How Does Atomic Data Composition Shape LLM Mathematical Reasoning?

Anonymous authors
Paper under double-blind review

Abstract

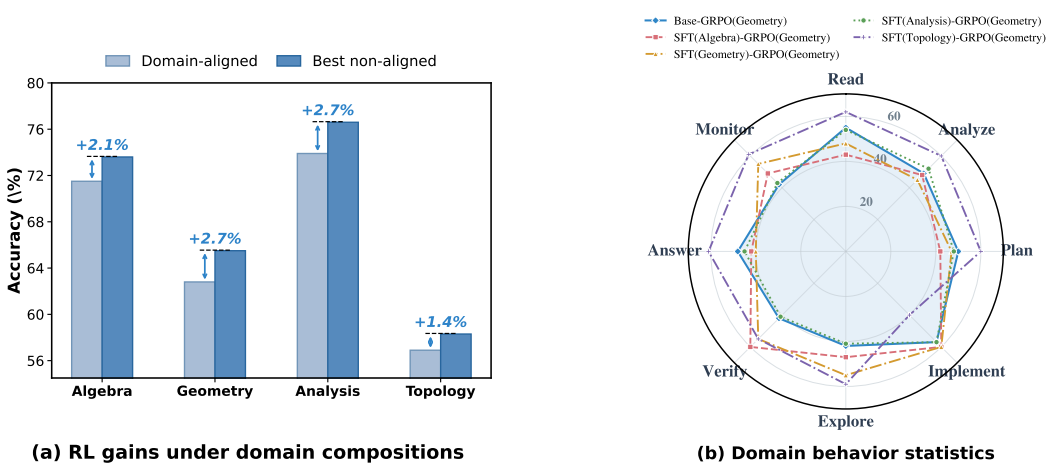
Reinforcement learning (RL) has substantially advanced the mathematical reasoning capabilities of Large Language Models (LLMs). However, how the domain composition of training data shapes target-capability amplification remains unclear. Motivated by the field-atomic perspective, we investigate how supervised fine-tuning (SFT) data from algebra, geometry, analysis, and topology affects subsequent target-domain GRPO. Beyond final performance, we annotate reasoning chains using ThinkARM's eight reasoning behaviors and analyze behavior distributions, transition dynamics, and their conditional associations with correctness. Our results show that domain-aligned SFT does not consistently provide the strongest initialization, largest GRPO gain, and best final performance, with the most effective source domain varying across target capabilities. Different target domains exhibit distinct behavioral demands, while models initialized with different field data retain different reasoning profiles after the same target-domain GRPO. Moreover, the strength and direction of transition–correctness associations vary across **domains capability** and **training compositions** and Path-reward experiments show that reinforcing high-value transitions can improve target-domain reasoning. These findings highlight the importance of behavior-aware data selection and post-training for more effective mathematical capability amplification.

1 Introduction

In recent years, large language models (LLMs) have made significant progress in language comprehension(Zheng et al., 2024; Wu et al., 2023; Li et al., 2024; Gong et al., 2025; Bezou-Vrakatseli et al., 2025; Du et al., 2024; Yamamoto et al., 2026), complex reasoning(Li et al., 2025; He et al., 2025a; Havrilla et al., 2024; Tan et al., 2025; Huang et al., 2026; Agarwal & Yamada, 2026; He et al., 2025b), and multimodal tasks(Dong et al., 2025; Man et al., 2025; Thawakar et al., 2025; Wang et al., 2025b; Ye et al., 2025; Tang et al., 2026; Grassucci et al., 2025; Ren et al., 2025; Deitke et al., 2025), gradually demonstrating the potential to approach general intelligence. Against this backdrop, mathematical reasoning has emerged as a crucial benchmark for assessing models' deep cognitive capabilities(Danxin et al., 2026; Huber & Niklaus, 2025; Ruan et al., 2026), garnering widespread attention. Recent studies, reinforcement learning (Zhu et al., 2025; Yue et al., 2026; Jia et al., 2026; Wu et al., 2026), have improved performance on challenging mathematical benchmarks, further fueling expectations for general-purpose reasoning models.

Current efforts to improve mathematical reasoning broadly follow two complementary directions: **(1) advancing training paradigms** and **(2) scaling reasoning data**. On the training side, reinforcement learning with verifiable rewards enables models to improve through outcome feedback, as demonstrated by DeepSeek-R1 (Guo et al., 2025a) and DAPO (Yu et al., 2026), while on-policy distillation provides dense teacher supervision over trajectories sampled from the student policy (Li et al., 2026b). On the data side, recent work scales mathematical supervision through problem synthesis, solution augmentation, filtering, and verification. Representative efforts such as OpenMathInstruct-2 (Toshniwal et al., 2025), MegaMath (Zhou et al., 2025), and OpenMathReasoning (Moshkov et al., 2025) have expanded mathematical training corpora to millions of reasoning examples or hundreds of billions of tokens. However, as improvements from increasing model and data scale gradually approach a bottleneck, these methods are facing challenges:

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.



(a) RL gains under domain compositions **(b) Domain behavior statistics**

Figure 1: **Motivation for moving beyond domain alignment to reasoning behavior.** (a) The best non-aligned initialization outperforms the domain-aligned initialization across all four target domains. (b) Five data compositions produce distinct eight-behavior profiles on Geometry.

- *Does scaling target-aligned data guarantee the strongest capability amplification, or can cross-domain interactions yield greater gains?*
- *Do observed capability gains arise from target-domain data, subsequent optimization, or changes in the underlying reasoning behavior?*

Recent work by Kuang et al. (Kuang et al., 2026) introduces a field-atomic perspective for understanding LLM mathematical reasoning, which has received increasing attention in the LLM community. Their framework reorganizes mathematical data into four field-atomic capabilities—**Algebra**, **Geometry**, **Analysis**, and **Topology**—and evaluates how selectively strengthening one capability affects performance in the others. Their empirical results reveal asymmetric cross-domain transfer, providing preliminary evidence that mathematical capabilities interaction. However, existing analyses remain largely focused on performance comparisons, leaving unclear how different field-specific SFT sources affect target-capability amplification during subsequent post-training and which changes in the underlying reasoning process account for the resulting performance differences.

To clarify how field-specific data compositions affect target-domain capability amplification, as is shown in Figure 1, we conduct a controlled study across the four field-atomic capabilities. For each target domain, we compare the base model and **four field-specific SFT initializations before** and **after the same target-domain GRPO**, allowing us to examine how the capability foundation established by SFT shapes subsequent reinforcement. We then use a validated LLM annotator and the eight reasoning behaviors defined by ThinkARM(Li et al., 2026a) to characterize token allocation and transition dynamics in the generated reasoning chains. Finally, we model the conditional associations between these behavioral features and answer correctness at both the **(1) domain-capability** and **(2) training-composition** levels, and further validate whether reinforcing positively associated reasoning paths can improve target-domain reasoning. Our analysis reveals four key insights:

- **Limited benefits of domain alignment:** Domain-aligned SFT does not consistently yield the strongest initialization, largest GRPO gain, or best final performance. The most effective source domain varies across target capabilities.
- **Behavioral Demands Are Domain-Specific:** Different mathematical domains exhibit distinct reasoning-behavior distributions, emphasizing different balances among analysis, planning, implementation, verification, and answer formulation.
- **Behavioral Alignment Does Not Follow Domain Alignment:** Models trained with different SFT sources retain distinct reasoning profiles after the same target-domain GRPO, while the domain-aligned composition does not consistently exhibit the behavioral pattern associated with the strongest target performance.

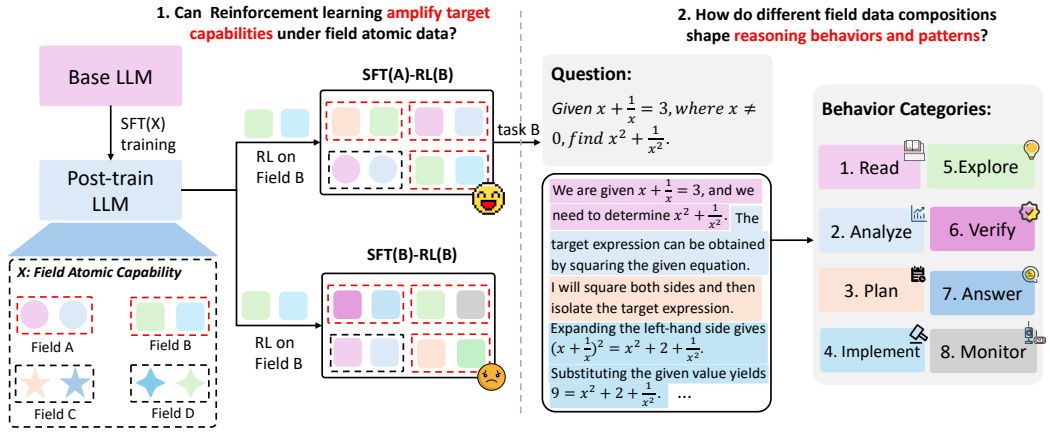


Figure 2: **This figure illustrates target-domain capability amplification and reasoning behavior analysis.** The left part shows models initialized with different field-specific atomic data and subsequently optimized with target-domain GRPO. The right part shows the sentence-level annotation of model reasoning chains using ThinkARM's eight behavior categories.

- **Transition Associations Are Composition-Sensitive:** Different target capabilities exhibit distinct positive and negative transition–correctness associations. Although reasoning traces share a broad progression from iterative analysis and execution toward answer formation, the strength and direction of individual transition associations vary across SFT compositions.
- **Path Rewards Improve Reasoning Beyond Data Composition:** Jointly reinforcing high-value reasoning transitions with correctness rewards can improve target-domain reasoning beyond composition-based training.

2 Related Work

Reinforcement Learning and Reasoning Process Modeling in LLMs Recent studies have shown that reinforcement learning (RL) is an effective paradigm for enhancing the mathematical reasoning ability of large language models (Zhang et al., 2025; Wong et al., 2025; Wang et al., 2025a; Tian et al., 2026; Yadav & Verma, 2026). Representative work such as **DeepseekMath** (Shao et al., 2024) demonstrates that RL can directly incentivize reasoning capability, while **OREAL** (Lyu et al., 2025) shows that strong performance can be achieved even under binary feedback. At the same time, studies like the **Reasoning Boundary Paradox** (Nguyen et al., 2025) reveal potential trade-offs between accuracy and reasoning diversity under RL. Beyond optimizing final outcomes, a growing body of work has focused on modeling intermediate reasoning processes (Lin et al., 2025; Zhang et al., 2024), including process supervision, step-level verification, and trajectory-level analysis. For example, **Let's Verify Step by Step** (Lightman et al., 2023) introduces step-level supervision to improve reasoning correctness by explicitly validating intermediate steps. These efforts highlight the importance of reasoning dynamics beyond final answer accuracy.

Benchmarks for Mathematical Ability Evaluation A growing number of benchmarks have been proposed to evaluate mathematical reasoning in LLMs (Liu et al., 2024), (Lu et al., 2024), (Yang & Deng, 2019). **MATH** (Hendrycks et al., 2021) and **GSM8K** (Cobbe et al., 2021) provide foundational datasets for competition-level and arithmetic reasoning, respectively. More recent benchmarks focus on increasing difficulty and diversity: **MathOdyssey** (Fang et al., 2025) spans high-school to university-level problems, while **FrontierMath** (Glazer et al., 2024) targets expert-level mathematical reasoning. **UGMathBench** (Xu et al., 2025) focuses on undergraduate-level reasoning with diversified subjects. **RealMath** (Zhang et al., 2026) considers research-level mathematical problems, and **RBench** (Guo et al., 2025b) evaluates multidisciplinary, graduate-level reasoning ability.

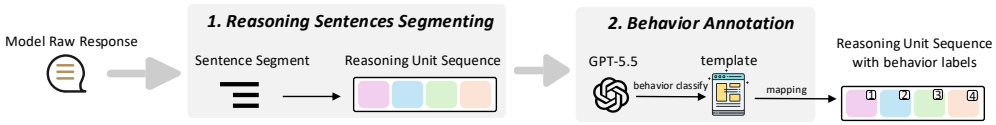


Figure 3: The illustration of **reasoning sentences segmenting** and **behavior annotation process.**

3 Problem Formulation

Existing studies on LLM mathematical reasoning primarily rely on final answer accuracy, which makes it difficult to explain why different domain data compositions lead to heterogeneous capability gains. To enable a more fine-grained investigation, we formalize the three research questions and define how they are operationalized in our analysis.

3.1 RL-based Capability Amplification under Data Compositions

Reinforcement learning is widely used to enhance the reasoning ability of large language models, yet it remains unclear whether its amplification effect is consistent when models are initialized with different field-specific atomic data compositions. For each target domain, we first use domain-specific SFT data to establish different capability initializations and then apply the same target-domain RL training to all initialized models. By comparing target-domain performance before and after RL, we reveal ***how data composition influences the extent of RL-based capability amplification?***

3.2 Fine-Grained Analysis of Reasoning Behaviors

To better explain the performance differences observed in Section 3.1, we further analyze the internal structure of model-generated reasoning processes. We follow the taxonomy introduced by ThinkARM (Li et al., 2026a), as it provides a theory-grounded and sentence-level representation of mathematical reasoning behaviors based on Schoenfeld's Episode Theory (Schoenfeld, 1985). The taxonomy contains eight behaviors: **Read**, **Analyze**, **Plan**, **Implement**, **Explore**, **Verify**, **Answer**, and **Monitor**. Given a reasoning trace r_i, we segment it into sentences and assign each sentence a behavioral label:

$$r_i \to (s_{i,1}, \ldots, s_{i,L_i}), \qquad s_{i,t} \to e_{i,t}, \quad e_{i,t} \in \mathcal{E}, \tag{1}$$

where $\mathcal{E}$ denotes the eight behavior categories.

Let $\tau_{i,t}$ denote the token count of sentence $s_{i,t}$. For model m, target domain T, and behavior c, its token-level proportion is defined as:

$$P_{m,d}(c) = \frac{\sum_{i \in \mathcal{D}_{m,d}} \sum_{t=1}^{L_i} \tau_{i,t} \mathbb{I}(e_{i,t} = c)}{\sum_{i \in \mathcal{D}_{m,d}} \sum_{t=1}^{L_i} \tau_{i,t}}. \tag{2}$$

We compare these proportions and average response lengths across the Base and four domain-specific SFT initializations after target-domain RL to examine how training-data composition reshapes reasoning behaviors. More details are provided in Appendix E.

3.3 Reasoning-Transition Dynamics and Correctness Association

We extend the aggregate behavior analysis in Section 3.2 by modeling directed transitions between adjacent behaviors, including self-transitions. For each response, the feature vector includes log-transformed length, eight behavior-token proportions, and counts of all 8×8 transitions, with features standardized within each target domain across training compositions.

Composition-Specific Modeling. To examine how data composition changes the reasoning patterns associated with the same target capability, we fit a class-balanced L2-regularized logistic regression separately for each target domain d and training composition m:

$$\Pr(y_i = 1 \mid \mathbf{x}_i, d, m) = \sigma\left(\alpha_{d,m} + \widetilde{\mathbf{x}}_i^{\top} \boldsymbol{\beta}_{d,m}\right), \tag{3}$$

where y_i denotes answer correctness and $\boldsymbol{\beta}_{d,m}$ represents the conditional associations between reasoning features and correctness for a specific training composition.

Domain-Capability Modeling. To identify the reasoning patterns associated with a target capability as a whole, we pool responses from all five training compositions and estimate a shared coefficient vector while controlling for composition-specific baseline differences:

$$\min_{\boldsymbol{\alpha}_d, \boldsymbol{\beta}_d} -\frac{1}{N_d} \sum_{i=1}^{N_d} [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] + \lambda \|\boldsymbol{\beta}_d\|_2^2, \qquad \hat{y}_i = \sigma\left(\alpha_{d,m_i} + \widetilde{\mathbf{x}}_i^\top \boldsymbol{\beta}_d\right). \quad (4)$$

Here, N_d is the number of responses in target domain d, α_{d,m_i} controls for the training composition that produces response i, and the shared coefficient vector $\boldsymbol{\beta}_d$ characterizes the reasoning patterns associated with correctness at the overall domain-capability level.

Target-Path-Driven Validation. To further investigate whether reasoning patterns that are positively associated with correctness can effectively improve target-domain reasoning ability, we conduct a target-path-driven validation experiment. We represent candidate reasoning patterns by a set of directed transitions:

$$\mathcal{P} = \{(p, q) \mid p, q \in \mathcal{E}\}, \quad (5)$$

where each element (p, q) denotes a transition from behavior p to behavior q. For a response with behavior sequence: $\mathbf{e}_i = (e_{i,1}, \ldots, e_{i,L_i})$, its path-reward score is defined as:

$$r_{\text{path}}(\mathbf{e}_i; \mathcal{P}) = \frac{1}{L_i - 1} \sum_{t=1}^{L_i - 1} \mathbb{I}\left[(e_{i,t}, e_{i,t+1}) \in \mathcal{P}\right]. \quad (6)$$

Within the GRPO, this path-level reward is jointly combined with the answer-correctness reward during training, encouraging the model to generate reasoning paths that are both structurally valuable and associated with correct answers. More details are provided in Appendix D.

4 Experiments Setup

4.1 Field Atomic Datasets

We adopt the field atomic datasets proposed by Kuang et al. (Kuang et al., 2026), which are constructed from diverse mathematical benchmarks, including MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), Gaokao-Bench (Zhang et al., 2023), OlympiadBench (He et al., 2024), and MMLU (Wang et al., 2024). The original problems are reclassified into four mathematical domains—algebra, geometry, analysis, and topology—through template-based rules and LLM-assisted annotation, with balanced train/test splits. Dataset statistics are reported in Appendix A.

4.2 Training and Evaluation Protocol

Training Paradigm We conduct experiments on two representative mathematical reasoning backbones, **Qwen2.5-Math-Instruct-7B** (Yang et al., 2024) and **MetaMath-Mistral-7B** (Yu et al., 2024), to evaluate the robustness of our findings across models. We adopt a two-stage training paradigm of supervised fine-tuning (SFT) followed by reinforcement learning (RL) on 2×A100 80GB GPUs. In the second stage, we apply GRPO (Shao et al., 2024) with rewards based on answer correctness. Detailed settings are provided in Appendix B.

Evaluation Protocol We follow standard Chain-of-Thought (CoT) prompting, requiring the model to generate step-by-step reasoning and enclose the final answer within `\boxed{}` for reliable extraction. Final performance is evaluated using exact-match accuracy against reference answers.

5 Analysis and Discussion

5.1 Analysis of Capability Amplification under Domain Data Compositions

Limited Capability Amplification of In-Field Alignment. Table 1 shows that a stronger in-domain initialization does not consistently translate into better final capability. This pattern appears in both algebra and geometry, where weaker Base or cross-domain initializations can achieve larger gains

Table 1: **Performance of Qwen2.5-Math and models initialized by SFT on different data and after GRPO on each target domain T.** Change (Δ) reports the performance change before and after RL. † denotes the domain-aligned initialization; bold values indicate the best result in each column within a target-domain block.

SFT Source	Before GRPO(T)			After GRPO(T)			Change (Δ)		
	Low	High	Avg.	Low	High	Avg.	Low	High	Avg.
Algebra Capability									
Base	86.0	68.3	77.2	87.0	**72.5**	79.8	+1.0	**+4.2**	**+2.6**
Algebra†	86.3	**69.4**	**77.9**	**88.7**	71.5	80.1	**+2.4**	+2.1	+2.3
Geometry	85.9	69.1	77.5	87.8	72.1	80.0	+1.9	+3.0	+2.5
Analysis	86.1	69.1	77.6	88.4	71.7	80.1	+2.3	+2.6	+2.5
Topology	**86.8**	68.6	77.7	88.5	71.8	**80.2**	+1.7	+3.2	+2.5
Geometry Capability									
Base	64.1	57.9	61.0	64.7	63.3	64.0	+0.6	+5.4	+3.0
Algebra	64.8	59.5	62.2	**66.3**	62.5	64.4	+1.5	+3.0	+2.3
Geometry†	**65.0**	**59.9**	**62.5**	64.2	62.8	63.5	-0.8	+2.9	+1.1
Analysis	64.8	59.4	62.1	64.4	63.0	63.7	-0.4	+3.6	+1.6
Topology	63.4	58.2	60.8	65.2	**63.9**	**64.6**	**+1.8**	**+5.7**	**+3.8**
Analysis Capability									
Base	75.5	68.3	71.9	**77.9**	73.9	**75.9**	+2.4	**+5.6**	**+4.0**
Algebra	76.6	**71.0**	73.8	76.7	73.6	75.2	+0.1	+2.6	+1.4
Geometry	76.4	70.0	73.2	77.7	73.6	75.7	**+1.3**	+3.6	+2.5
Analysis†	76.7	**71.0**	73.9	77.2	73.9	75.6	+0.5	+2.9	+1.7
Topology	**77.3**	70.6	**74.0**	77.4	**74.1**	75.8	+0.1	+3.5	+1.8
Topology Capability									
Base	52.2	56.2	54.2	52.1	57.4	54.8	-0.1	+1.2	+0.6
Algebra	53.5	**56.3**	**54.9**	**56.2**	**58.7**	**57.5**	**+2.7**	**+2.4**	**+2.6**
Geometry	53.6	55.4	54.5	52.7	56.9	54.8	-0.9	+1.5	+0.3
Analysis	53.2	55.7	54.5	55.6	57.1	56.4	+2.4	+1.4	+1.9
Topology†	**53.9**	54.7	54.3	55.2	56.9	56.1	+1.3	+2.2	+1.8

and stronger final performance. In particular, the advantage of SFT(Topology) on geometry and the degradation of SFT(Geometry) on easier problems reveal that different training domains can introduce both positive and negative interaction effects. The same tendency is observed in Appendix C.1. Thus, adaptation depends not only on domain alignment, but also on whether the source data provides complementary reasoning structures for the target capability.

Effective Transfer Source Varies across Target Domains. The most effective initialization differs substantially across target domains, revealing no universal ranking among source-domain data. The Base initialization is most competitive after algebra GRPO, topology SFT produces the strongest geometry model, and algebra SFT yields the best topology performance. Even before RL on analysis, SFT(Topology) slightly surpasses the domain-aligned initialization. These results indicate different SFT datasets may establish distinct procedural priors and reasoning behaviors, which are reinforced differently by target-domain GRPO. The final capability therefore depends on the compatibility between the reasoning structure induced during SFT and the optimization signal provided during RL, motivating the fine-grained behavioral analysis in the following section.

5.2 Analysis of Reasoning Behaviors under Different Data Compositions

Target Capabilities Induce Distinct Reasoning Profiles. Table 2 shows that different target domains induce distinct allocations of reasoning behaviors. Algebra reasoning is more execution-oriented: approximately 54% of its tokens are assigned to *Implement*, while *Analyze* occupies a smaller proportion than in geometry. Algebra also allocates more tokens to *Plan*, consistent with the need to formulate and execute symbolic solution procedures. In contrast, geometry assigns more tokens to *Analyze* and fewer to *Plan*, reflecting greater emphasis on constructing and interpreting structural relationships before execution. These differences suggest that target capabilities are distinguished not only by performance, but also by how reasoning effort is distributed across functional behaviors.

Table 2: **Token-level distributions (%) of the eight behaviors after target-domain GRPO.** Avg. Tok. denotes the average number of tokens per response.

SFT Source	Percentage (%)								Avg. Tok.
	Read	Ana	Plan	Imp	Exp	Ver	Ans	Mon	
Algebra Capability									
Base	5.59	21.84	8.92	54.19	0.30	2.73	6.36	0.08	533
Algebra†	4.99	18.80	8.46	54.41	0.40	2.80	5.70	4.44	594
Geometry	5.73	20.91	8.98	53.90	0.38	3.07	6.46	0.57	546
Analysis	5.01	19.01	8.74	54.53	0.62	3.45	5.75	2.89	596
Topology	4.94	19.70	8.41	54.47	0.54	3.25	5.61	3.07	616
Geometry Capability									
Base	7.79	24.41	7.47	52.25	0.21	1.43	6.38	0.04	537
Algebra	6.96	24.30	7.04	53.52	0.25	1.86	5.93	0.14	567
Geometry†	7.36	23.70	7.32	53.75	0.31	1.77	5.58	0.21	570
Analysis	7.69	25.26	7.38	51.83	0.20	1.40	6.19	0.05	531
Topology	8.32	27.43	8.00	46.85	0.34	1.76	6.94	0.35	508
Analysis Capability									
Base	4.97	19.05	9.72	58.41	0.18	1.79	5.81	0.07	570
Algebra	4.59	18.04	9.37	60.16	0.09	1.81	5.61	0.32	583
Geometry	4.55	17.57	9.52	59.18	0.31	2.00	5.71	1.15	583
Analysis†	4.72	18.00	9.63	59.68	0.18	1.73	5.74	0.32	576
Topology	4.61	17.78	9.59	60.54	0.17	1.73	5.41	0.18	601
Topology Capability									
Base	7.37	26.25	10.42	45.33	0.38	3.29	6.48	0.50	579
Algebra	6.80	24.19	10.02	45.54	0.37	4.12	6.87	2.10	652
Geometry	7.54	26.27	10.42	44.61	0.40	3.40	6.41	0.95	586
Analysis	7.15	25.91	10.08	46.13	0.38	3.61	6.50	0.23	619
Topology†	6.94	24.37	9.92	46.38	0.36	3.61	6.67	1.76	637

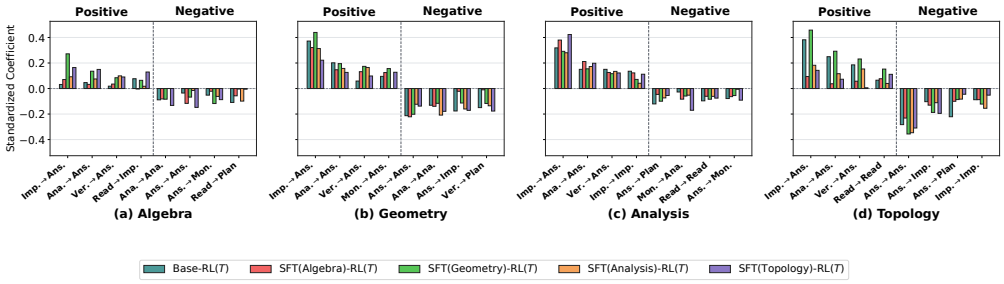


Figure 4: **Associations between representative reasoning transitions and answer correctness.** For each domain, the four strongest positive associations and the four strongest negative associations by absolute coefficient magnitude are shown on the left and right, respectively.

Behavioral Alignment Does Not Follow Domain Alignment. However, behavioral profiles also vary across training compositions within the same target domain. For geometry, the Geometry- and Algebra-initialized models after GRPO(Geometry) remain strongly implementation-oriented, allocating over 53% of their tokens to *Implement*. In contrast, the Topology-initialized model reduces this proportion to 46.85% while increasing *Analyze*, *Plan*, and *Monitor*; it also produces the shortest average trace among the five models. Notably, this model achieves the strongest final geometry performance in Table 1, indicating that better transfer is not necessarily associated with longer reasoning or greater execution intensity. This motivates the correctness-oriented analysis of behavior transitions in the next section.

5.3 Analysis of Capability Amplification Depends on Specific Reasoning Patterns

Shared and Domain-Specific Transfer Patterns. We first examine whether target-domain capabilities rely on a common reasoning structure or domain-specific strategies. As shown in Table 3,

Table 3: **Top ten domain-level reasoning features most associated with answer correctness in each target domain.**

Algebra Capability					
Positive Association			**Negative Association**		
#	**Feature**	β	**#**	**Feature**	β
1	Trans (Imp → Ans)	+0.37	1	Length (Total)	-0.85
2	Trans (Ana → Ans)	+0.34	2	Ratio (Ans)	-0.24
3	Ratio (Imp)	+0.25	3	Ratio (Mon)	-0.22
4	Trans (Ver → Ans)	+0.23	4	Ratio (Ana)	-0.14
5	Trans (Read → Imp)	+0.08	5	Ratio (Read)	-0.09
6	Trans (Plan → Ana)	+0.07	6	Ratio (Exp)	-0.08
7	Trans (Plan → Imp)	+0.06	7	Trans (Read → Plan)	-0.06
8	Trans (Plan → Plan)	+0.04	8	Trans (Ans → Ans)	-0.05
9	Ratio (Ver)	+0.03	9	Trans (Ana → Plan)	-0.05
10	Trans (Imp → Ver)	+0.03	10	Trans (Ana → Ana)	-0.03

Geometry Capability					
Positive Association			**Negative Association**		
#	**Feature**	β	**#**	**Feature**	β
1	Trans (Ana → Ans)	+0.61	1	Length (Total)	-0.59
2	Trans (Imp → Ans)	+0.59	2	Trans (Imp → Ana)	-0.24
3	Trans (Ver → Ans)	+0.32	3	Ratio (Ana)	-0.23
4	Ratio (Imp)	+0.30	4	Ratio (Read)	-0.19
5	Trans (Ana → Imp)	+0.27	5	Trans (Ans → Ans)	-0.19
6	Trans (Ana → Read)	+0.10	6	Trans (Ana → Ana)	-0.12
7	Trans (Plan → Imp)	+0.09	7	Ratio (Mon)	-0.11
8	Trans (Read → Imp)	+0.05	8	Trans (Imp → Plan)	-0.11
9	Trans (Plan → Read)	+0.04	9	Trans (Plan → Ana)	-0.10
10	Trans (Read → Read)	+0.03	10	Ratio (Ver)	-0.10

Analysis Capability					
Positive Association			**Negative Association**		
#	**Feature**	β	**#**	**Feature**	β
1	Trans (Imp → Ans)	+0.66	1	Length (Total)	-0.82
2	Trans (Ana → Ans)	+0.52	2	Ratio (Ans)	-0.19
3	Trans (Ver → Ans)	+0.35	3	Ratio (Read)	-0.11
4	Trans (Ana → Imp)	+0.17	4	Trans (Read → Read)	-0.09
5	Trans (Imp → Imp)	+0.13	5	Ratio (Ana)	-0.05
6	Ratio (Imp)	+0.12	6	Ratio (Exp)	-0.05
7	Trans (Plan → Ana)	+0.08	7	Trans (Imp → Ana)	-0.02
8	Trans (Read → Imp)	+0.08	8	Trans (Read → Plan)	-0.01
9	Trans (Ana → Ana)	+0.07	9	–	–
10	Trans (Ans → Ans)	+0.07	10	–	–

Topology Capability					
Positive Association			**Negative Association**		
#	**Feature**	β	**#**	**Feature**	β
1	Trans (Imp → Ans)	+0.74	1	Length (Total)	-0.45
2	Trans (Ana → Ans)	+0.71	2	Trans (Ans → Ans)	-0.36
3	Trans (Ver → Ans)	+0.52	3	Ratio (Read)	-0.26
4	Ratio (Imp)	+0.27	4	Trans (Imp → Ana)	-0.24
5	Trans (Ana → Imp)	+0.24	5	Trans (Plan → Ana)	-0.17
6	Trans (Read → Read)	+0.14	6	Ratio (Mon)	-0.14
7	Trans (Mon → Imp)	+0.13	7	Ratio (Ana)	-0.13
8	Ratio (Ans)	+0.07	8	Trans (Imp → Imp)	-0.13
9	Trans (Read → Imp)	+0.07	9	Ratio (Exp)	-0.12
10	Trans (Read → Ana)	+0.07	10	Trans (Imp → Ver)	-0.11

Table 4: **Validation of path-reward reinforcement for target-domain reasoning.** S denotes the best-performing non-domain-aligned source configuration under different domain capability.

Model	**Algebra**		**Geometry**		**Analysis**		**Topology**	
	Low	**High**	**Low**	**High**	**Low**	**High**	**Low**	**High**
Qwen2.5-Math	86.0	68.3	64.1	57.9	75.5	68.3	52.2	56.2
Mutilple Domain Composition								
SFT(T)-RL(T)	**88.7**	71.5	64.2	62.8	77.2	73.9	55.2	56.9
SFT(S)-RL(T)	88.5	71.8	65.2	**63.9**	77.4	74.1	**56.2**	**58.7**
Target Domain Reinforcement								
Base-RL(T)	87.0	72.5	64.7	63.3	77.9	73.9	52.1	57.4
Base-RL$_{\text{PATHREWARD}}$(T)	88.4	**73.0**	**65.5**	63.5	**78.4**	**74.3**	55.8	57.2

both domains favor *Analysis → Answer*, suggesting that transferable reasoning depends on consolidating intermediate deductions into a clear conclusion. Beyond this shared structure, geometry is positively associated with *Plan → Implementation*, indicating the importance of translating a geometric construction or proof plan into concrete derivation steps. Topology instead favors *Monitor → Implementation*, suggesting that successful reasoning more often requires checking definitions or structural conditions before applying them. The complete transition-frequency matrices for all target domains are provided in Appendix C.3.

Data composition reconfigures the predictive role of transitions. Figure 4 shows that transition–correctness associations vary substantially across SFT sources within each target domain. Although *Implement→Answer* is positively associated with correctness in all domains, its strength depends on the source composition; this variation is particularly evident in topology. Negative associations are also composition-dependent: *Answer→Answer* is consistently negative for algebra, geometry, and topology, whereas analysis is more strongly associated with *Answer→Plan* and *Monitor→Analyze*.

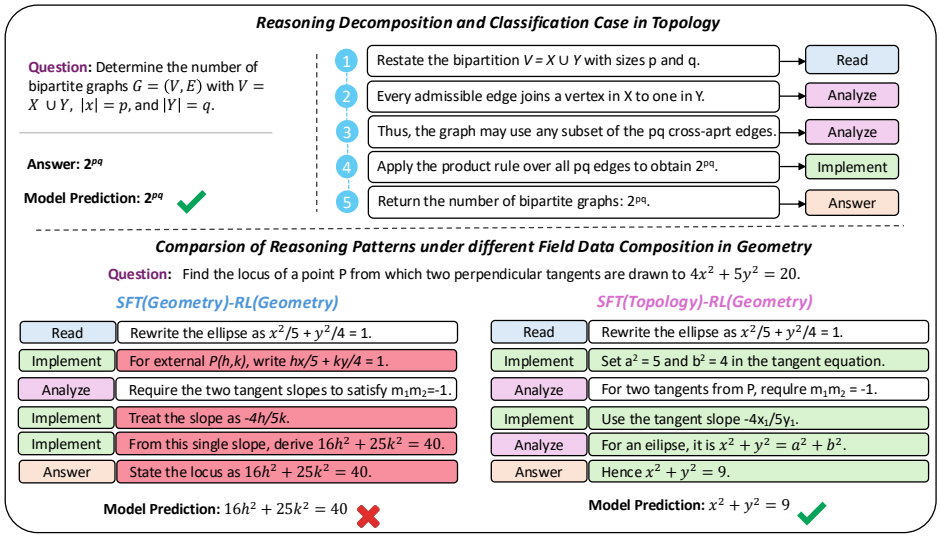


Figure 5: **Case studies of behavior-annotated reasoning in topology and geometry under different domain data composition.**

These results indicate that SFT composition reweights the predictive role of individual transitions. Full coefficient matrices are provided in Appendix C.4.

Path-Reward Validation of High-Value Reasoning Patterns. Building on the preceding analyses, we instantiate the path reward with ***Analyze*** $\rightarrow$ ***Answer***, ***Implement*** $\rightarrow$ ***Answer***, and ***Verify*** $\rightarrow$ ***Answer***, which were identified as relatively high-value transitions in the transition–correctness analysis. As shown in Table 4, augmenting Base-RL(T) with path rewards improves performance. The largest gain occurs for Topology-low, which increases from 52.1 to 55.8, followed by Algebra-low (87.0$\rightarrow$88.4), and Analysis-low (77.9$\rightarrow$78.4). The path-reward model achieves the best results on Algebra-high, and both Analysis settings, while remaining competitive with the strongest non-path-reward models in most other settings. Overall, high-value reasoning paths are an important mechanism underlying domain competence, while their effectiveness depends on the target domain and the difficulty level.

5.4 Case Study.

To better understand the decomposed reasoning processes of models trained with different field-data compositions, Figure 5 first presents a topology example in which a complete solution is represented as an ordered sequence of behaviors. This decomposition makes both the function of each reasoning step and the transitions between successive behaviors explicit. The geometry example then compares two models on the same problem: although they begin from similar information, different field-data compositions lead to distinct organizations of analysis and implementation behaviors.

6 Conclusion

In our work, we focus on how different data compositions shape the amplification of target-domain mathematical capabilities through RL. We construct models with SFT initializations from different fields, apply target-domain GRPO and compare their capabilities before and after RL. To explain the resulting performance differences beyond final accuracy, we decompose reasoning chains into eight interpretable behaviors, analyze their token-level proportions and sequential transitions, and employ regularized logistic regression to examine their associations with answer correctness at both the model and domain levels. Our experiments indicate that capability amplification depends not only on domain alignment but also on the reasoning patterns induced by the initialization data. We further find that jointly reinforcing high-value reasoning transitions with correctness rewards can improve target-domain reasoning. Future work will extend the evaluation scope and incorporate beneficial reasoning patterns into RL objectives to achieve more effective capability amplification.

AI USE STATEMENT

We use LLM-based supervised fine-tuning and reinforcement learning methods in our mathematical reasoning experiments. We additionally use LLMs to annotate model-generated reasoning chains by segmenting them into sentences and assigning the eight predefined reasoning-behavior categories for the subsequent behavior and transition analyses. LLMs also assisted with manuscript language polishing and preparation. All LLM-assisted annotations, analyses, code, and text were reviewed and verified by the authors, who take full responsibility for the final content of this paper.

ETHICS STATEMENT

This study uses publicly available mathematical problems and model-generated reasoning traces and does not involve sensitive personal data or high-risk human-subject research. A small human-annotated gold set was prepared by mathematics students solely to evaluate the consistency of candidate LLM annotators; the annotation task involved no personally identifiable information. The authors declare no conflicts of interest or external sponsorship that could influence the study, and take responsibility for the integrity and ethical use of the released analyses and materials.

REPRODUCIBILITY STATEMENT

We provide our data and code in the supplementary materials for review. We will further disclose the relevant data and code in GitHub and Huggingface to facilitate subsequent researchers to reproduce our results.

REFERENCES

Aakash Kumar Agarwal and Moyuru Yamada. Hypothesis-driven reasoning for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1686–1693, 2026.

Elfia Bezou-Vrakatseli, Oana Cocarascu, and Sanjay Modgil. Can large language models understand argument schemes? In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 13666–13681, 2025.

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

Cui Danxin, Sihang Jiang, Keyi Wang, Zhiyi Duan, Yanghua Xiao, Bi Yude, Jiaqing Liang, Minggui He, Shimin Tao, and Yilun Liu. Measuring the unmeasurable: Unveiling latent cognitive capabilities of llm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30485–30493, 2026.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 91–104, 2025.

Yuhao Dong, Zuyan Liu, Hai-Long Sun, Jingkang Yang, Winston Hu, Yongming Rao, and Ziwei Liu. Insight-v: Exploring long-chain visual reasoning with multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9062–9072, 2025.

Jiangshu Du, Yibo Wang, Wenting Zhao, Zhongfen Deng, Shuaiqi Liu, Renze Lou, Henry Peng Zou, Pranav Narayanan Venkit, Nan Zhang, Mukund Srinath, et al. Llms assist nlp researchers: Critique paper (meta-) reviewing. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 5081–5099, 2024.

Meng Fang, Xiangpeng Wan, Fei Lu, Fei Xing, and Kai Zou. Mathodyssey: Benchmarking mathematical problem-solving skills in large language models using odyssey math data. *Scientific data*, 12(1):1392, 2025.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, et al. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*, 2024.

Jie Gong, Biaoshuai Zheng, and Qiwang Hu. Eventrelbench: A comprehensive benchmark for evaluating event relation understanding in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 9084–9099, 2025.

Eleonora Grassucci, Giordano Cicchetti, and Danilo Comminiello. Closing the modality gap enables novel multimodal learning applications. In *Second Workshop on Representational Alignment at ICLR 2025*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025a. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Meng-Hao Guo, Jiajun Xu, Yi Zhang, Jiaxi Song, Haoyang Peng, Yi-Xuan Deng, Xinzhi Dong, Kiyohiro Nakayama, Zhengyang Geng, Chen Wang, et al. R-bench: Graduate-level multi-disciplinary benchmarks for llm & mllm complex reasoning evaluation. *arXiv preprint arXiv:2505.02018*, 2025b.

Alexander Havrilla, Sharath Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravynski, Eric Hambro, and Roberta Raileanu. Glore: When, where, and how to improve llm reasoning via global and local refinements. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 17719–17733, 2024.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Tao He, Hao Li, Jingchang Chen, Runxuan Liu, Yixin Cao, Lizi Liao, Zihao Zheng, Zheng Chu, Jiafeng Liang, Ming Liu, et al. Breaking the reasoning barrier a survey on llm complex reasoning

through the lens of self-evolution. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 7377–7417, 2025a.

Yancheng He, Shilong Li, Jiaheng Liu, Weixun Wang, Xingyuan Bu, Ge Zhang, Zy Peng, Zhaoxiang Zhang, Zhicheng Zheng, Wenbo Su, et al. Can large language models detect errors in long chain-of-thought reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18468–18489, 2025b.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Jiameng Huang, Baijiong Lin, Guhao Feng, Jierun Chen, Di He, and Lu Hou. Efficient reasoning for large reasoning language models via certainty-guided reflection suppression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31176–31184, 2026.

Thomas Huber and Christina Niklaus. Llms meet bloom's taxonomy: A cognitive view on large language model evaluations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5211–5246, 2025.

Weitao Jia, Jinghui Lu, Haiyang Yu, Siqi Wang, Guozhi Tang, An-Lan Wang, Weijie Yin, Dingkang Yang, Yuxiang Nie, Bin Shan, et al. Meml-grpo: Heterogeneous multi-expert mutual learning for rlvr advancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31283–31291, 2026.

Jiayi Kuang, Haojing Huang, Yinghui Li, Xinnian Liang, Zhikun Xu, Yangning Li, Xiaoyu Tan, Chao Qu, Meishan Zhang, Ying Shen, et al. Atomic thinking of llms: Decoupling and exploring mathematical reasoning abilities. *Advances in Neural Information Processing Systems*, 38:166142–166167, 2026.

Ming Li, Chenrui Fan, Yize Cheng, Soheil Feizi, and Tianyi Zhou. Schoenfeld's anatomy of mathematical reasoning by language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32773–32802, 2026a.

Yangning Li, Weizhi Zhang, Yuyao Yang, Wei-Chieh Huang, Yaozu Wu, Junyu Luo, Yuanchen Bei, Henry Peng Zou, Xiao Luo, Yusheng Zhao, et al. Towards agentic rag with deep reasoning: A survey of rag-reasoning systems in llms. *arXiv preprint arXiv:2507.09477*, 2, 2025.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe, 2026b. URL https://arxiv.org/abs/2604.13016.

Yinghui Li, Qingyu Zhou, Yuanzhen Luo, Shirong Ma, Yangning Li, Hai-Tao Zheng, Xuming Hu, and Philip S Yu. When llms meet cunning texts: A fallacy understanding benchmark for large language models. *Advances in Neural Information Processing Systems*, 37:112433–112458, 2024.

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*, 2023.

Yen-Ting Lin, Di Jin, Tengyu Xu, Tianhao Wu, Sainbayar Sukhbaatar, Chen Zhu, Yun He, Yun-Nung Chen, Jason E Weston, Yuandong Tian, et al. Step-kto: Optimizing mathematical reasoning through stepwise binary feedback. In *Proceedings of The 3rd Workshop on Mathematical Natural Language Processing (MathNLP 2025)*, pp. 15–33, 2025.

Hongwei Liu, Zilong Zheng, Yuxuan Qiao, Haodong Duan, Zhiwei Fei, Fengzhe Zhou, Wenwei Zhang, Songyang Zhang, Dahua Lin, and Kai Chen. Mathbench: Evaluating the theory and application proficiency of llms with a hierarchical mathematics benchmark. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6884–6915, 2024.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024.

Chengqi Lyu, Songyang Gao, Yuzhe Gu, Wenwei Zhang, Jianfei Gao, Kuikun Liu, Ziyi Wang, Shuaibin Li, Qian Zhao, Haian Huang, et al. Exploring the limit of outcome reward for learning mathematical reasoning. *arXiv preprint arXiv:2502.06781*, 2025.

Yunze Man, De-An Huang, Guilin Liu, Shiwei Sheng, Shilong Liu, Liang-Yan Gui, Jan Kautz, Yu-Xiong Wang, and Zhiding Yu. Argus: Vision-centric reasoning with grounded chain-of-thought. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14268–14280, 2025.

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *arXiv preprint arXiv:2504.16891*, 2025.

Phuc Minh Nguyen, Chinh D La, Duy MH Nguyen, Nitesh V Chawla, Binh T Nguyen, and Khoa D Doan. The reasoning boundary paradox: How reinforcement learning constrains language models. *arXiv preprint arXiv:2510.02230*, 2025.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15781–15791, 2025.

Kai Ruan, Xuan Wang, Jixiang Hong, Peng Wang, Yang Liu, and Hao Sun. Evaluating llms' divergent thinking capabilities for scientific idea generation with minimal context. *Nature Communications*, 2026.

Alan H Schoenfeld. *Mathematical problem solving*. New York, 1985.

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.

Xiaoyu Tan, Haoyu Wang, Xihe Qiu, Leijun Cheng, Yuan Cheng, Wei Chu, Yinghui Xu, and Yuan Qi. Struct-x: Enhancing the reasoning capabilities of large language models in structured data scenarios. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2584–2595, 2025.

Zichen Tang, E Haihong, Rongjin Li, Jiacheng Liu, Linwei Jia, Zhuodi Hao, Zhongjun Yang, Yuanze Li, Haolin Tian, Xinyi Hu, et al. Finmmdocr: Benchmarking financial multimodal reasoning with scenario awareness, document understanding, and multi-step computation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25858–25866, 2026.

Omkar Thawakar, Dinura Dissanayake, Ketan Pravin More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Ilmuz Zaman Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, et al. Llamav-o1: Rethinking step-by-step visual reasoning in llms. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24290–24315, 2025.

Changyuan Tian, Zhicong Lu, Shuang Qian, Nayu Liu, Peiguang Li, Li Jin, Leiyi Hu, Zhizhao Zeng, Sirui Wang, Ke Zeng, et al. Rectify evaluation preference: Improving llms' critique on math reasoning via perplexity-aware reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33241–33249, 2026.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 19243–19275, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/302ce0673c00aee2cf84bb43d0117553-Paper-Conference.pdf.

Huaijie Wang, Shibo Hao, Hanze Dong, Shenao Zhang, Yilin Bao, Ziran Yang, and Yi Wu. Offline reinforcement learning for llm multi-step reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8881–8893, 2025a.

Peijie Wang, Zhong-Zhi Li, Fei Yin, Dekang Ran, and Cheng-Lin Liu. Mv-math: Evaluating multimodal math reasoning in multi-visual contexts. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19541–19551, 2025b.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024.

Zhen Hao Wong, Jingwen Deng, Runming He, Zirong Chen, Qijie You, Hejun Dong, Hao Liang, Chengyu Shen, Bin Cui, and Wentao Zhang. Logicpuzzlerl: cultivating robust mathematical reasoning in llms via reinforcement learning. *arXiv preprint arXiv:2506.04821*, 2025.

Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Huijie Lv, Ming Zhang, et al. Reasoning or memorization? unreliable results of reinforcement learning due to data contamination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33944–33952, 2026.

Xiaoqian Wu, Yong-Lu Li, Jianhua Sun, and Cewu Lu. Symbol-llm: leverage language models for symbolic system in visual human activity reasoning. *Advances in neural information processing systems*, 36:29680–29691, 2023.

Xin Xu, Jiaxin Zhang, Tianhao Chen, Zitong Chao, Jishan Hu, and Can Yang. Ugmathbench: A diverse and dynamic benchmark for undergraduate-level mathematical reasoning with large language models. In *International Conference on Learning Representations*, volume 2025, pp. 78770–78813, 2025.

Sourabh Yadav and Nishchal K Verma. Teaching small llms to reason using reinforcement learning. In *2026 IEEE 16th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1347–1353. IEEE, 2026.

Taisei Yamamoto, Ryoma Kumon, Danushka Bollegala, and Hitomi Yanaka. Neuron-level analysis of cultural understanding in large language models. In *International Conference on Learning Representations*, volume 2026, pp. 131074–131115, 2026.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Kaiyu Yang and Jia Deng. Learning to prove theorems via interacting with proof assistants. In *International Conference on Machine Learning*, pp. 6984–6994. PMLR, 2019.

Xinwu Ye, Chengfan Li, Siming Chen, Wei Wei, and Robert Tang. Mmscibench: Benchmarking language models on chinese multimodal scientific problems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 14621–14663, 2025.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 45040–45061, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Chuhuai Yue, Chengqi Dong, Yinan Gao, Hang He, Jiajun Chai, Wei Lin, and Guojun Yin. Promoting efficient reasoning with verifiable stepwise reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34530–34538, 2026.

Jie Zhang, Cezara Petrui, Kristina Nikolić, and Florian Tramèr. Realmath: A continuous benchmark for evaluating language models on research-level mathematics. *Advances in Neural Information Processing Systems*, 38, 2026.

Kepu Zhang, Guofu Xie, Weijie Yu, Mingyue Xu, Xu Tang, Yaxin Li, and Jun Xu. Legal mathematical reasoning with llms: Procedural alignment through two-stage reinforcement learning. In *EMNLP (Findings)*, pp. 1586–1598, 2025.

Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li, Yueting Zhuang, and Weiming Lu. Agent-pro: Learning to evolve via policy-level reflection and optimization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5348–5375, 2024.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric Xing, et al. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. In *International Conference on Learning Representations*, volume 2024, pp. 22225–22257, 2024.

Fan Zhou, Zengzhi Wang, Nikhil Ranjan, Zhoujun Cheng, Liping Tang, Guowei He, Zhengzhong Liu, and Eric P Xing. Megamath: Pushing the limits of open math corpora. *arXiv preprint arXiv:2504.02807*, 2025.

Huaisheng Zhu, Siyuan Xu, Hangfan Zhang, Teng Xiao, Zhimeng Guo, Shijie Zhou, Shuyue Hu, and Vasant G Honavar. Reinforcement learning for large language models via group preference reward shaping. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21398–21411, 2025.

A Additional Details about Data

To ensure consistency with prior work (Kuang et al., 2026), we adopt the same field-level atomic training split for all four mathematical domains. In the original benchmark, both the training and test sets are organized into two coarse difficulty levels (**Level 1** and **Level 2**). Table 5 reports the train and test data statistics.

Table 5: Original Data statistics of **FIELD** atomic capabilities

Field	Algebra		Geometry		Analysis		Topology	
	Level 1	Level 2	Level 1	Level 2	Level 1	Level 2	Level 1	Level 2
Train/Test	3813/1277	4517/1505	3351/1117	3391/1331	3276/1092	4077/1358	3336/1112	3176/1058

B Training Details

We fine-tune all models using the GRPO algorithm implemented in the Open-Instruct framework. Unless otherwise specified, we use the same training configuration across all experiments. The main hyperparameters are summarized in Table 6.

Table 6: **Main hyperparameters for GRPO training.**

Parameter	Value	Parameter	Value
Learning rate	2×10^{-5}	Optimizer	AdamW
Batch size per device	8	Gradient accumulation steps	2
Number of epochs	1	Precision	bfloat16
Number of generations	4	Max model length	2048
Max completion length	1024	vLLM GPU utilization	0.4

Table 7: **Performance of candidate annotators on the human-annotated gold set across four mathematical domains.** GPT-5.5 shows the highest agreement with human annotations overall and is used for further large-scale annotation. The best result in each column is highlighted in bold.

Model	Algebra		Geometry		Analysis		Topology		Overall	
	Acc.	κ	Acc.	κ	Acc.	κ	Acc.	κ	Acc.	κ
GPT-5.5	**90.73**	**85.68**	**91.28**	**87.48**	88.87	83.30	**87.31**	**82.42**	**89.55**	**84.72**
Claude-Opus-4.8	89.90	84.64	90.69	86.72	**90.09**	**85.41**	84.73	79.07	88.85	83.96
Gemini-3.1-Pro	87.08	80.13	85.51	79.34	86.66	80.07	82.22	75.28	85.37	78.70

C Additional Experiments and Analysis

C.1 Domain Alignment Shows Consistent Limitations across Backbones.

Table 8 further examines whether our main finding generalizes to MetaMath-Mistral-7B. The results are consistent with those of the primary backbone: domain-aligned SFT does not reliably produce the strongest final capability or the largest target-domain GRPO gain. For geometry, Geometry SFT provides the strongest initialization, whereas Topology SFT achieves the best post-GRPO performance on both difficulty levels and exhibits larger gains. For topology, Algebra SFT similarly outperforms the domain-aligned Topology initialization after GRPO. This cross-backbone consistency indicates that the limited advantage of domain alignment is not specific to a particular model family; instead, target-domain amplification depends on how the capability foundation induced by the source-domain data interacts with subsequent RL optimization.

C.2 Effective Reasoning Favors Forward Progression.

To further examine how transition patterns unfold throughout reasoning, we analyze their frequency distributions over normalized response progress. As shown in Figure 6, *Analyze→Analyze* is concentrated in the early and middle stages, reflecting continued refinement of the problem representation. *Analyze→Implement* remains active during the main derivation stage, where analysis is translated into concrete solution steps, whereas *Implement→Answer* rises sharply near completion as intermediate results are consolidated into the final response. These temporal profiles show that the role of a transition depends not only on its overall frequency but also on when it occurs.

C.3 Domain-Level Behavior Transition Matrices

To provide a complete view of the sequential behavior structure, Figure 7 reports the full 8×8 sentence-level directed transition matrix for each target domain. Rows denote the current behavior and columns the next behavior. The matrices show a common concentration on Analyze-to-Analyze, Analyze-to-Implement, Implement-to-Implement, and Answer-to-Answer transitions, while the relative mass assigned to planning, verification, and cross-stage transitions varies across domains. This appendix view complements the coefficient-based RQ3 analysis by describing the overall transition structure without selecting only the most correctness-associated paths.

Table 8: **Performance of MetaMath-Mistral-7B before and after target-domain GRPO on Geometry and Topology.** Each capability block reports Low and High accuracy before and after GRPO(T).

SFT Source	*Geometry Capability*						*Topology Capability*					
	Before GRPO(T)		After GRPO(T)		Change (Δ)		Before GRPO(T)		After GRPO(T)		Change (Δ)	
	Low	High	Low	High	Low	High	Low	High	Low	High	Low	High
Base	29.6	16.0	44.8	19.6	+15.2	+3.6	31.5	16.7	51.8	22.6	**+20.3**	+5.9
Algebra	32.5	17.5	46.4	21.9	+15.9	+5.4	**35.6**	18.2	**52.9**	**24.8**	+17.3	**+6.6**
Geometry	**35.2**	**18.9**	45.6	21.4	+10.4	+2.5	33.7	18.5	51.2	23.3	+17.5	+4.8
Analysis	31.6	18.6	46.1	22.3	+14.5	+3.7	34.2	**19.8**	34.6	23.6	+0.4	+3.8
Topology	30.1	17.3	**46.8**	**23.0**	**+16.7**	**+5.7**	35.4	19.1	51.6	24.5	+16.2	+5.4

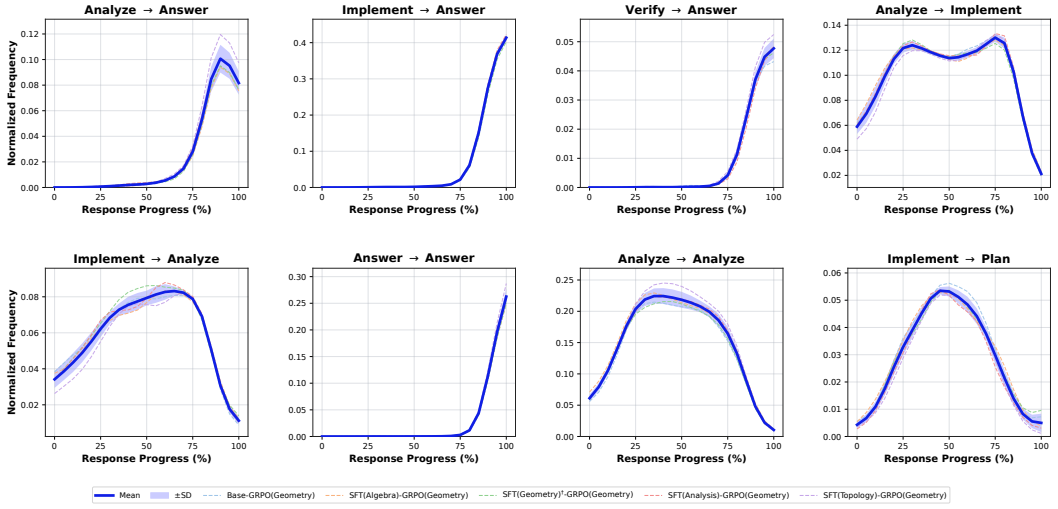


Figure 6: **Temporal distributions of eight representative reasoning-behavior transitions on the geometry target domain.** The horizontal axis denotes token-normalized response progress, and the vertical axis reports the normalized frequency of each transition.

C.4 Transition Associations with Answer Correctness

The transition probabilities above describe how often behaviors follow one another, whereas Figure 8 summarizes their correctness associations. Each cell is the coefficient $\beta_{d,p\to q}$ of a directed transition in the pooled, domain-capability L2-regularized logistic model. The model additionally controls for total response length, eight behavior token shares, and model initialization; thus, a positive or negative cell indicates a conditional association with correctness. A shared diverging color scale makes the magnitude and sign comparable across domains.

D Implementation Details of Path-Reward Training

The implementation of path-reward training is summarized in Figure 9. For each training prompt, the current policy generates a group of four complete rollouts. Each rollout is segmented into sentences and passed to the ThinkARM annotator, which assigns one of the eight reasoning-behavior labels to every sentence. The resulting ordered labels are used to construct the adjacent behavior transitions required by the path-level reward.

As is shown in figure 9, the correctness reward is computed from the final answer using the same answer normalization and verification procedure as the standard GRPO training. The path reward is computed online from the annotated behavior sequence. In the implementation, the path signal is represented by the proportion of adjacent transitions that match the predefined target set. The annotation and reward calculation are performed for every rollout before the GRPO update. In all experiments, the path-reward coefficient is set to $\lambda = 0.40$. The complete rollout reward is:

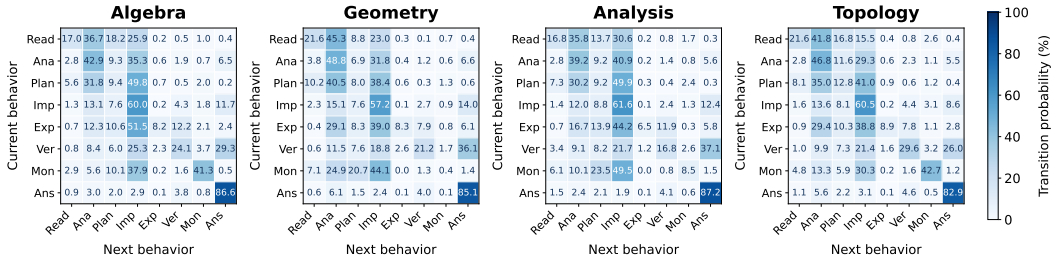


Figure 7: Complete **sentence-level transition matrices** of the eight ThinkARM behaviors for the four target domains; entries are transition probabilities (%).

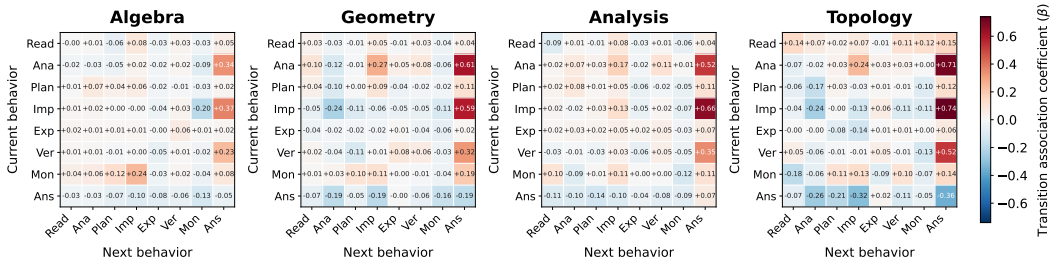


Figure 8: **Pooled domain-level transition-association coefficients** for the eight ThinkARM behaviors.

$$R = r_{\text{correct}} + \lambda r_{\text{path}}, \tag{7}$$

For each prompt, the four rollout rewards are normalized within the group to obtain the relative advantages used by GRPO.

E Annotation Protocol for Mathematical Reasoning

For analyzing LLM-generated reasoning on mathematical problems, we adopt the sentence-level annotation scheme and guidebook introduced by ThinkARM (Li et al., 2026a). ThinkARM likewise investigates mathematical reasoning in LLMs, and its adaptation of Schoenfeld's episode framework and eight-category taxonomy has received broad recognition in the LLM community. Each sentence is assigned one primary behavior based on both its content and the surrounding context.

E.1 Annotator Selection

For the annotation-agreement pilot, we sampled 100 responses from each of the four domains, including 50 low-level and 50 high-level responses per domain. These 400 responses were manually annotated by mathematics students to construct the human-annotated gold set. We then evaluated GPT-5.5, Opus-4.8, and Gemini-3.1-Pro against this gold set using sentence-level accuracy and Cohen's κ. As is shown in Table 7, GPT-5.5 achieved the highest overall agreement and was therefore selected for subsequent large-scale annotation.

E.2 Eight ThinkARM Behavioral Categories

Table 9 summarizes the eight behavior categories, their operational definitions, distinguishing boundaries, and potential keyword indicators used during annotation.

E.3 Prompt Template for Annotation

Every annotator receives the same sentence segmentation, guidebook, problem, preceding context, and current batch. Responses are segmented with the ThinkARM sentence splitter and processed in

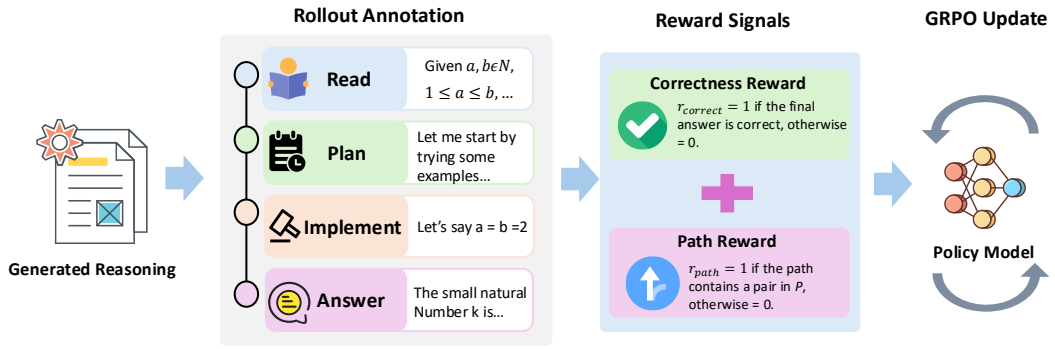


Figure 9: **Overview of the path-reward validation procedure.**

Table 9: **Definitions**, **distinguishing boundaries**, and **potential keyword indicators** for the eight ThinkARM reasoning behaviors.

Behavior	Definition and distinguishing boundary	Potential keywords/indicators
Read	Extracts or restates the given information, conditions, and goal of the problem. It is limited to understanding the input; analysis, evaluation, and solution operations are not Read.	"The question asks", "we are given", "the goal is", "the problem requires", "the choices are"
Analyze	Constructs or recalls relevant theories, introduces relationships, and makes certain logical inferences. It explains the mathematical basis without concrete substitution or calculation.	"According to", "we can define", "this implies", "therefore", "based on this", "we can infer", "let's recall"
Plan	Announces and commits to the next step or an overall solution strategy before execution. Definite intent distinguishes it from the uncertainty of Explore; "let's verify" is Verify.	"Next", "the next step is", "we need to", "let's proceed", "the plan is", "we should first", "I will now"
Implement	Carries out the mathematics, including setting notation, substitutions, calculations, diagram construction, case enumeration, and generation of intermediate results.	"Substituting", "expanding", "the matrix becomes", "let me set", "let's denote", displayed equations or concrete calculations
Explore	Generates a tentative idea, alternative approach, analogy, guess, or trial without commitment to a solution path. Any subsequent analysis, calculation, or check receives its own functional label.	"Maybe", "perhaps", "what if", "another possibility", "could this be", "alternatively", "let's try"
Verify	Judges the correctness, consistency, feasibility, or simplicity of a result or method, using checks, comparisons, alternative derivations, or bounds.	"Double-check", "consistent with", "plugging it back in", "let's confirm", "check again", "is there any mistake", "verify"
Monitor	Expresses a brief meta-level pause, hesitation, self-monitoring, or transition without substantive mathematical content.	"Hmm", "wait", "let me think", "okay", "let's see", "hold on"
Answer	Directly states the solution or conclusion for the given problem, either as a final or provisional answer; intermediate calculation results are not Answer.	"The answer is", "hence the result is", "the final answer is", "therefore"

batches of 20 sentences. Future sentences are not shown. The following box presents the prompt template; placeholders are populated for each response.

Prompt Template for ThinkARM Annotation

In this project, we aim to analyze the reasoning process of current large language models (LLMs) with advanced reasoning capabilities, i.e., Large Reasoning Models (LRMs), based on a modified version of Alan Schoenfeld's (1985) "Episode-Timeline" framework for problem-solving. Given the model response, annotate its sentence-level behavior with eight categories: Read, Analyze, Explore, Plan, Implement, Verify, Monitor, and Answer.
The [Guidebook]–[End of the Guidebook] section provides the detailed definitions and examples of each category. The [Math Problem]–[End of the Math Problem] section provides the problem. The [Overall Response]–[End of the Overall Response] section provides the complete model response. The [Previous Context]–[End of the Previous Context] section provides the preceding sentences and their annotations. The [Input]–[End of the Input] section provides the sentences to annotate, and the [Format]–[End of the Format] section specifies the output format.
[Guidebook]
(Guidebook Content)
[End of Guidebook]
[Math Problem]
(The Question)
[End of the Math Problem]
[Overall Response]
(The Complete Model Response)
[End of the Overall Response]
[Previous Context]
The previous sentences are: *(Previous batch sentences and labels)*
[End of Previous Context]
[Input]
The following sentences need to be classified:
([1] sentence 1; [2] sentence 2; . . . ; [batch_num] sentence n)
[End of Input]
[Format]
Return JSON containing an index, a short rationale, and one category for each sentence:
`{"sentences": [{"index": "sentence index", "reason": "short rationale", "category": "one of the eight categories"}]}`
[End of Format]
Annotate only the sentences in [Input], follow their indices exactly, and return one category and a short rationale for every input sentence.