From Implicit to Explicit: Token-Efficient Logical Supervision for Mathematical Reasoning in LLMs

Shaojie Wang[1] **Liang Zhang**[1]*

{shaojiewang, liangzhang}@hkustgz.edu.cn

[1] Hong Kong University of Science and Technology (Guangzhou)

Abstract

Recent studies reveal that large language models (LLMs) exhibit limited logical reasoning abilities in mathematical problem-solving, instead often relying on pattern-matching and memorization. We systematically analyze this limitation, focusing on logical relationship understanding, which is a core capability underlying genuine logical reasoning, and reveal that errors related to this capability account for over 90% of incorrect predictions, with Chain-of-Thought Supervised Fine-Tuning (CoT-SFT) failing to substantially reduce these errors. To address this bottleneck, we propose **F**irst-**S**tep **L**ogical **R**easoning (**FSLR**), a lightweight training framework targeting logical relationship understanding. Our key insight is that the first planning step-identifying which variables to use and which operation to apply without performing any calculation-encourages the model to derive logical relationships directly from the problem statement. By training models on this isolated step, FSLR provides explicit supervision for logical relationship understanding, unlike CoT-SFT which implicitly embeds such relationships within complete solution trajectories. Extensive experiments across multiple models and datasets demonstrate that FSLR consistently outperforms CoT-SFT under both in-distribution and out-of-distribution settings, with average improvements of 3.2% and 4.6%, respectively. Moreover, FSLR achieves 4-6× faster training and reduces training token consumption by over 80%.

1 Introduction

Mathematical reasoning, as a crucial cognitive skill that supports problem-solving in numerous scientific and practical applications, has attracted particular attention in LLM research (Ahn et al., 2024; Wang et al., 2025; Forootani, 2025). Although LLMs have achieved near-human accuracy on mathematical benchmarks such as GSM8K (Cobbe et al., 2021), recent studies (Mirzadeh et al., 2024; Huang et al.; Li et al., 2024) reveal that LLMs exhibit limited logical reasoning abilities—the capacity to understand "novel" problems and derive appropriate solution steps, rather than relying on pattern-matching and memorization.

As a fundamental capability for mathematical problem-solving, genuine logical reasoning enables models to understand the underlying rationale of problems and transfer knowledge to new situations (Serna M. and Serna, 2015). Recently, Supervised Fine-Tuning (SFT) with chain-of-thought (CoT) supervision has received increasing attention and has been widely adopted to improve reasoning abilities of LLMs by training models on complete solution trajectories (Sun et al., 2025). However, it remains unclear whether CoT-SFT can elicit genuine logical reasoning.

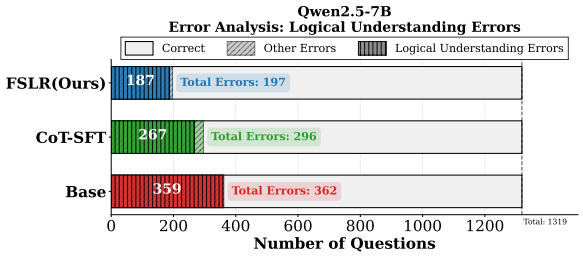

Figure 1: Error analysis on Qwen2.5-7B comparing Base, CoT-SFT, and FSLR(Ours) models. Each bar shows the breakdown of correct predictions, logical relationship understanding errors, and other errors.

To investigate this question, we conduct a systematic analysis comparing pretrained LLMs with their fine-tuned counterparts on mathematical reasoning tasks. Following the above discussion, genuine logical reasoning requires understanding a given problem before deriving solution steps. For mathematical problems, such understanding involves grasping how variables in the problem relate to each other and which operations connect them—what

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 28486–28500

we term logical relationships. We refer to the ability to grasp these relationships as logical relationship understanding, and adopt it as our diagnostic metric for genuine logical reasoning. Specifically, we use GPT-4o (Hurst et al., 2024) to analyze all incorrect predictions and identify whether failures arise from misunderstanding these logical relationships(Appendix A). As shown in Figure 1, using Qwen2.5-7B as an example, we find that logical relationship understanding errors account for over 90% of incorrect predictions—a pattern that holds consistently across multiple LLMs (Appendix B). Since CoT-SFT fails to substantially reduce these errors, they become the primary bottleneck limiting further improvements in overall accuracy.

These findings raise a natural research question: how can we more directly improve LLMs' understanding of logical relationships, thereby enhancing their overall performance? To answer this question, we examine why CoT-SFT falls short. CoT-SFT trains models to imitate complete solution trajectories, while logical relationships are implicitly embedded in these trajectories, never directly targeted by the training objective. We hypothesize that this implicit supervision leads to insufficient learning of logical relationships and consequently results in the phenomenon illustrated in Figure 1.

To address this limitation, we propose **F**irst-**S**tep **L**ogical **R**easoning (**FSLR**), a lightweight training framework that provides explicit supervision for enhancing logical relationship understanding. Specifically, we design a first-planning-step prompt schema that asks models to identify only what needs to be calculated first—which variables to use and which operation to apply, without solving the full problem. Then, FSLR isolates this first planning step as a standalone training task, thereby providing explicit supervision for logical relationship understanding.

This design offers several key advantages: (1) **More focused supervision**: FSLR provides a more direct training signal for logical relationship understanding by isolating the first planning step explicitly, alleviating the core limitation we identified earlier; (2) **Reduced training cost**: predicting only the initial planning step requires significantly fewer tokens than complete trajectories; (3) **Improved generalization**: since logical relationship understanding forms the foundation of the entire reasoning chain, strengthening this ability is expected to benefit the entire problem-solving process. As shown in Figure 1, FSLR achieves a substantial reduction in both logical relationship understanding errors and overall errors compared to CoT-SFT. Our main contributions are as follows:

- We systematically analyze the reasoning capabilities of LLMs and identify logical relationship understanding as a critical bottleneck: such errors account for over 90% of failures, and CoT-SFT fails to substantially mitigate this issue.
- We propose FSLR, a lightweight training framework that provides a more focused training task for logical relationship understanding by training models to identify the initial planning step obtained under a specially designed prompting scheme.
- Extensive experiments across multiple models and datasets demonstrate that FSLR substantially outperforms CoT-SFT in both in-distribution and out-of-distribution settings, while requiring significantly fewer training tokens.

2 Related Work

2.1 Mathematical Reasoning in LLMs

Mathematical reasoning has emerged as a critical testbed for evaluating whether large language models possess genuine reasoning abilities. Driven by this goal, the community has introduced diverse benchmarks (Cobbe et al., 2021; Patel et al., 2021; Miao et al., 2020; Koncel-Kedziorski et al., 2016; Lu et al., 2022; Gao et al., 2022), on which recent LLMs have achieved remarkable success. However, recent studies question whether these successes reflect genuine logical reasoning. GSM-Symbolic (Mirzadeh et al., 2024) and GSM-Plus (Li et al., 2024) reveal that LLMs exhibit noticeable performance variance across problem instantiations and are far from robust. Other work shows that LLMs are easily distracted by irrelevant information (Shi et al., 2023) and blindly apply learned skills without assessing their applicability to modified contexts (Huang et al.). While these works clearly diagnose that current LLMs often rely on pattern matching rather than true reasoning, they do not propose concrete training strategies to improve this capability.

2.2 Chain-of-Thought Fine-Tuning

Chain-of-Thought Supervised Fine-Tuning (CoT-SFT) has emerged as a widely adopted approach

for enhancing LLM reasoning by exposing models to step-by-step solution trajectories (Hwang et al., 2024; Lee et al., 2023; Yin et al., 2025; Lee et al., 2025). However, recent studies reveal several limitations: CoT-SFT decreases the faithfulness of reasoning (Lobo et al., 2025), leads models to memorize task-specific templates rather than acquiring transferable abilities (Chu et al., 2025), and introduces spurious features that cause hallucinations (Bao et al., 2025). While these studies reveal important limitations of CoT-SFT, none directly targets the logical relationship understanding bottleneck we identified.

3 Methodology

3.1 Task Formulation

We formalize CoT-SFT and FSLR to clarify how they differ in supervising logical relationship understanding.

CoT-SFT. Given a mathematical problem p and its complete solution trajectory $s = (s_1, s_2, ..., s_n)$, CoT-SFT trains models to generate the full reasoning chain by optimizing:

$$\mathcal{L}_{\text{CoT}} = -\sum_{i=1}^{n} \log P(s_i|p, s_1, ..., s_{i-1}; \theta)$$

While logical relationships are embedded within these steps, they are never explicitly targeted by the training objective. The supervision signal is distributed across the entire trajectory, diluting the focus on logical relationship understanding, we refer to this paradigm as *implicit supervision*.

FSLR. Given the same problem p, FSLR trains models to generate only the first planning step f_1 by optimizing:

$$\mathcal{L}_{\text{FSLR}} = -\log P(f_1|p; \theta)$$

where f_1 identifies which variables to use and which operation to apply, corresponding to the first planning step detailed in the next section. By isolating logical relationship understanding as an explicit training objective, our formulation directly targets this capability. We refer to this paradigm as *explicit supervision*.

3.2 FSLR Framework

Based on the above analysis, we now present the FSLR framework in detail. Specifically, FSLR trains on the first planning step—identifying relevant variables and operations without performing calculations—rather than the first calculation step. Corsidering logical relationship understanding forms the foundation of the entire reasoning chain, once the model correctly identifies which variables to use and which operation to apply, subsequent arithmetic steps follow naturally. As illustrated in Figure 2, FSLR consists of two stages: **(1)** constructing a training dataset $\mathcal{D}_{\text{FSLR}}$ that isolates logical relationship understanding, and **(2)** fine-tuning target models to generate f_1, providing explicit supervision for this core capability.

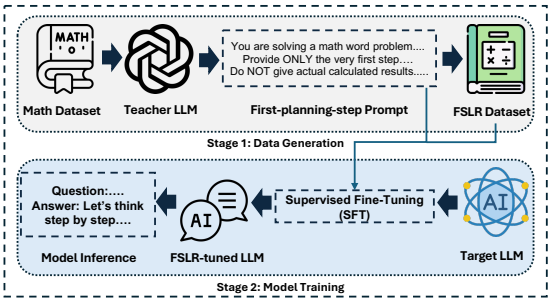

Figure 2: Overview of the FSLR framework, consisting of two stages: data generation and model training. The framework leverages a teacher LLM to generate first planning step guidance, which is then used to fine-tune the target LLM via supervised fine-tuning.

3.2.1 Training Dataset Construction

We construct training dataset $\mathcal{D}_{\text{FSLR}}$ by prompting a teacher model to generate first planning step for mathematical problems.

First-planning-step Prompt Design. Given a problem p, we use the following prompt template:

```
You are solving a math word problem step by step. Your task is to provide ONLY the very first step - stop immediately after identifying what to calculate first. Rules:

  • Provide ONLY the first calculation or identification
  • Do NOT solve the entire problem
  • Do NOT provide multiple steps
  • Do NOT give the actual calculated result, just identify what needs to be calculated

Problem: [problem text] First Step Only:
```

Design Rationale. The prompt instructs the model to identify only which variables to use and which operation to apply, ensuring that f_1 captures purely the logical reasoning decision. This aligns with our goal of providing explicit supervision for logical relationship understanding.

Data Generation. For each problem p_i, we query the teacher model to generate $f_1^{(i)}$ using the de-

signed prompt and construct the training dataset $\mathcal{D}_{\text{FSLR}} = \{(p_i, f_1^{(i)})\}_{i=1}^{N}$. We provide an example illustrating the training data format in Appendix C.

3.2.2 Model Training

Given $\mathcal{D}_{\text{FSLR}}$, the target LLM is fine-tuned to maximize the likelihood of $f_1^{(i)}$ conditioned on the input prompt(p_i) as follows:

$$\mathcal{L}_{\text{FSLR}} = -\frac{1}{N}\sum_{i=1}^{N} \log P(f_1^{(i)} | \text{prompt}(p_i); \theta)$$

where prompt(p_i) denotes p_i formatted with our prompt template, and $f_1^{(i)}$ is the first planning step for problem p_i. Through this training process, the model learns to explicitly identify logical relationships between variables. Notably, since FSLR trains only on the first planning step f_i, it requires significantly fewer tokens than CoT-SFT, substantially reducing training costs.

3.2.3 Model Inference

Since FSLR strengthens logical relationship understanding: the foundation of the entire reasoning chain, the improved capability is expected to benefit the full problem-solving process. At inference time, the FSLR-trained model generates complete solutions through standard autoregressive decoding, with no specialized prompting or additional modules required.

4 Experiments

In this section, we present comprehensive experiments to evaluate the effectiveness of FSLR training. We first describe the experimental setup (Section 4.1), then present our experiments results (Section 4.2).

4.1 Experimental Setup

Datasets: We evaluate FSLR on multiple mathematical reasoning benchmarks to assess both in-distribution and out-of-distribution performance. For **in-distribution evaluation**, we use GSM8K (Cobbe et al., 2021) and SVAMP (Patel et al., 2021). For **out-of-distribution evaluation**, we test on four additional datasets: ASDiv (Miao et al., 2020), MAWPS (Koncel-Kedziorski et al., 2016) , TabMWP (Lu et al., 2022), and GSM-Hard (Gao et al., 2022).

Models. We conduct experiments across multiple model families to ensure generalizability. For **training data generation**, we use three teacher models: LLaMA-3.1-70B-Instruct (Grattafiori et al., 2024), Qwen2.5-72B-Instruct (Team et al., 2024), and the finetuned target model itself (self-generated). For **target models**, we fine-tune three instruction-tuned checkpoints: LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-7B-Instruct (Team et al., 2024), and Qwen3-4B-Instruct (Yang et al., 2025). **Baselines.** We compare FSLR against the following baselines:

- **Base Model**: The original instruction-tuned model without additional mathematical reasoning training.
- **Math-Specialized Model**: Representative models fine-tuned specifically for mathematical reasoning at comparable scale, including DeepSeek-Math-7B-Instruct (Shao et al., 2024) and Qwen2.5-Math-7B (Yang et al., 2024).
- **CoT-SFT Model**: Models fine-tuned on complete chain-of-thought solution trajectories, representing the current mainstream approach.
- **Zero-LP**: A zero-shot prompting strategy that instructs the model to first identify relevant variables and operations before solving, without fine-tuning.
- **Few-LP**: A few-shot prompting strategy that provides demonstrations of explicit logical relationship identification before solving, without fine-tuning.

Training Details. For both CoT-SFT and FSLR, we use the complete training sets of GSM8K and SVAMP as our base datasets. For CoT-SFT data generation, we use two sources: (1) **Teacher LLM**: a larger model is prompted to generate complete step-by-step solution trajectories for each training problem; (2) **Self-generated**: the target model itself generates its own CoT solutions using the same prompt template. In both cases, we use greedy decoding (temperature=0) via vLLM for efficient batch inference, and retain only solutions yielding the correct final answer. For FSLR, we generate first planning steps using the prompt template described in Section 3.2.1.

Evaluation Protocol. We evaluate all models using greedy decoding with the standard "Let's think step by step" prompting strategy and report accuracy as the primary metric.

In-Distribution Results		*LLaMA3.1-8B*		*Qwen2.5-7B*		*Qwen3-4B*		Average
Data Source	**Method**	**GSM8K**	**SVAMP**	**GSM8K**	**SVAMP**	**GSM8K**	**SVAMP**	
DeepSeek-Math	Zero-shot	78.70	82.20	78.70	82.20	78.70	82.20	80.45
Qwen2.5-Math	Zero-shot	79.20	85.50	79.20	85.50	79.20	85.50	82.35
Base LLM	Zero-shot	62.90	67.60	72.60	83.00	84.70	85.00	75.97
	Few-shot	77.50	84.00	90.10	92.20	84.80	91.60	86.70
	Zero-LP	70.00	72.30	84.50	88.50	86.80	92.60	82.45
	Few-LP	67.70	71.80	86.20	87.30	91.30	91.70	82.67
LLaMA	CoT-SFT	77.90	79.30	77.60	88.10	85.70	85.10	82.28
	FSLR	**83.10**	**84.80**	**85.10**	**91.30**	**87.10**	**91.10**	**87.08**
Qwen	CoT-SFT	**85.70**	84.00	82.60	89.30	91.30	**93.90**	87.80
	FSLR	85.30	**85.90**	**86.40**	**92.70**	**91.80**	91.30	**88.90**
Self	CoT-SFT	74.10	78.80	84.50	90.80	89.80	90.10	84.68
	FSLR	**80.40**	**82.50**	**88.90**	**93.00**	**92.10**	**94.00**	**88.48**

Table 1: In-distribution evaluation on GSM8K and SVAMP. Models are trained on GSM8K and SVAMP using data generated by different teachers (LLaMA-3.1-70B, Qwen2.5-72B, or self-generated). All trained models are evaluated under zero-shot setting. Zero-LP and Few-LP denote zero-shot and few-shot logical planning prompting strategies that explicitly instruct models to identify logical relationships before solving, without fine-tuning. Math-specialized models (DeepSeek-Math-7B and Qwen2.5-Math-7B) are evaluated zero-shot as reference baselines. Green cells indicate FSLR outperforms CoT-SFT. Red cells indicate FSLR underperforms CoT-SFT.

4.2 Experiments and Analysis

In this section, we evaluate FSLR from eight perspectives: in-distribution performance (Section 4.2.1), out-of-distribution generalization (Section 4.2.2), performance across problem complexity levels (Section 4.2.3), training efficiency (Section 4.2.4), reliability of error attribution (Section 4.2.5), consistency analysis of teacher-generated first planning steps (Section 4.2.6), robustness to problem variations (Section 4.2.7), and case study (Section 4.2.8).

4.2.1 In-Distribution Performance

Table 1 presents results on in-distribution benchmarks (GSM8K and SVAMP), comparing FSLR against CoT-SFT using three teacher models for data generation.

FSLR consistently outperforms CoT-SFT across all settings. Across all three target models and teacher configurations, FSLR achieves superior or comparable performance to CoT-SFT, with particularly substantial improvements when using LLaMA-3.1-70B as the teacher: +5.2% on LLaMA-3.1-8B (GSM8K), +5.5% on LLaMA-3.1-8B (SVAMP), +7.5% on Qwen2.5-7B (GSM8K), averaging +4.8% improvement. When using Qwen2.5-72B and Self as data sources, FSLR achieves average improvements of +1.1% and +3.8% respectively. To further verify that FSLR's gains stem from the first-step design rather than general planning supervision, we compare against a Plan-and-Solve fine-tuning baseline (Parmar et al., 2025) in Appendix D.

FSLR enables general-purpose models to surpass math-specialized models. As shown in Table 1, math-specialized models such as DeepSeek-Math-7B and Qwen2.5-Math-7B achieve 78.70% and 79.20% on GSM8K, and 82.20% and 85.50% on SVAMP respectively. In contrast, FSLR-trained general-purpose models substantially outperform these specialized baselines. For example, Qwen2.5-7B with self-generated FSLR data achieves 88.90% on GSM8K and 93.00% on SVAMP, surpassing Qwen2.5-Math-7B by +9.7% and +7.5% respectively. These gains suggest that models benefit more from focused supervision that isolates the understanding of logical relationships.

FSLR demonstrates superior robustness to teacher model quality. While CoT-SFT performance varies substantially across different teachers (82.28% with LLaMA vs 87.80% with Qwen, a 5.52% gap), FSLR shows more consistent performance (87.08% vs 88.90%, only 1.82% gap). Notably, even with self-generated data, FSLR achieves 88.48% average accuracy, comparable to using larger teacher models. This robustness stems from the simplicity of first planning step supervision: generating a single logical planning decision is inherently easier and more reliable than generating complete solution trajectories, making FSLR less sensitive to teacher quality compared to CoT-SFT.

Out-of-Distribution Results		Models Trained on GSM8K					Average
Data Source	**Method**	**AsDIv**	**SVAMP**	**MAWPS**	**TabMWP**	**GSM-Hard**	
Math-Specialized Models							
DeepSeek-Math	Zero-shot	85.00	82.20	92.50	69.90	56.10	77.14
Qwen2.5-Math	Zero-shot	82.50	85.50	92.30	53.60	55.40	73.86
LLaMA3.1-8B							
Base LLM	Zero-shot	63.60	67.60	73.40	39.50	31.70	55.16
	Few-shot	85.80	84.00	97.00	55.00	38.70	72.10
LLaMA	CoT-SFT	72.30	76.80	80.30	49.20	35.60	62.84
	FSLR	**86.70**	**82.40**	**92.70**	**52.00**	**40.80**	**70.92**
Qwen	CoT-SFT	84.40	83.60	90.10	**70.00**	43.40	74.30
	FSLR	**87.80**	**83.90**	**93.80**	67.00	**43.60**	**75.22**
Self	CoT-SFT	74.40	80.20	79.70	46.20	33.00	62.70
	FSLR	**84.20**	**80.90**	**95.00**	**53.70**	**38.30**	**70.42**
Qwen2.5-7B							
Base LLM	Zero-shot	84.20	83.00	90.80	61.20	53.40	74.52
	Few-shot	90.90	92.20	97.60	70.40	62.90	82.80
LLaMA	CoT-SFT	79.70	85.10	83.40	**49.00**	50.70	69.58
	FSLR	**88.60**	**88.30**	**94.10**	47.20	**59.00**	**75.46**
Qwen	CoT-SFT	81.40	86.70	87.10	49.30	60.00	73.30
	FSLR	**89.30**	**87.40**	**94.60**	**52.10**	**60.70**	**76.82**
Self	CoT-SFT	84.70	90.10	87.20	**47.70**	61.50	74.24
	FSLR	**85.60**	**93.70**	**90.10**	45.80	**63.50**	**75.74**
Qwen3-4B							
Base LLM	Zero-shot	78.10	85.00	88.10	66.30	62.40	75.98
	Few-shot	88.40	91.60	95.90	70.30	56.00	80.44
LLaMA	CoT-SFT	78.10	83.40	85.40	55.00	53.10	71.00
	FSLR	**81.20**	**84.20**	**86.30**	**67.20**	**64.90**	**76.76**
Qwen	CoT-SFT	90.80	91.20	96.60	64.80	61.60	81.00
	FSLR	**90.80**	**93.80**	**96.70**	**69.10**	**66.90**	**83.34**
Self	CoT-SFT	86.20	89.90	92.00	66.40	53.50	77.60
	FSLR	**90.70**	**93.50**	**96.80**	**69.60**	**64.40**	**83.00**

Table 2: Out-of-distribution evaluation on five diverse benchmarks. Models are trained exclusively on GSM8K and evaluated on AsDIv, SVAMP, MAWPS, TabMWP, and GSM-Hard under zero-shot setting. Math-specialized models (DeepSeek-Math-7B and Qwen2.5-Math-7B) are evaluated zero-shot as reference baselines. Green cells indicate FSLR outperforms CoT-SFT. Red cells indicate FSLR underperforms CoT-SFT.

Comparison with prompting-based logical planning. To evaluate whether prompting alone without fine-tuning can achieve similar benefits by instructing models to identify logical relationships before solving, we test two strategies: Zero-shot Logical Planning (Zero-LP) and Few-shot Logical Planning (Few-LP), which provides demonstrations of explicit logical relationship identification. As shown in Table 1, both Zero-LP and Few-LP underperform FSLR by a considerable margin (82.45% and 82.67% vs. 87.08%–88.90%). This confirms that explicit logical relationship understanding requires internalization through fine-tuning rather than surface-level instruction following. Furthermore, Pass@k evaluation (Appendix E) confirms that FSLR expands model capability boundaries.

4.2.2 Out-of-Distribution Performance

Table 2 presents out-of-distribution results where models are trained exclusively on GSM8K then evaluated on five diverse benchmarks: AsDIv, SVAMP, MAWPS, TabMWP, and GSM-Hard. This setup tests whether the improvements from FSLR training generalize beyond the training distribution.

FSLR demonstrates superior generalization across diverse problem types. FSLR consistently outperforms CoT-SFT across all three models and most evaluation datasets. On LLaMA-3.1-8B, FSLR achieves substantial improvements over CoT-SFT: +14.4% on AsDIv (86.7% vs 72.3% with LLaMA teacher), +12.4% on MAWPS, and +5.2% on GSM-Hard, averaging +8.08% improvement across all OOD datasets. Similarly, Qwen2.5-7B

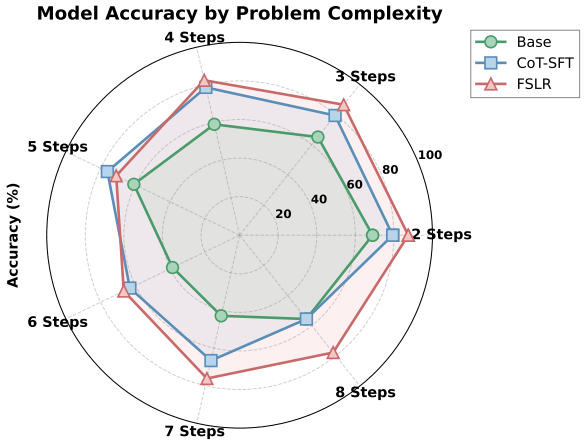

Figure 3: Radar chart showing model accuracy stratified by problem complexity (number of reasoning steps). Results shown are for LLaMA-3.1-8B with LLaMA-3.1-70B as the teacher model. FSLR consistently outperforms both Base and CoT-SFT methods, with particularly notable advantages on more complex problems requiring 6-8 reasoning steps.

and Qwen3-4B achieve average gains of +3.52% and +5.76% respectively across teacher configurations. Consistent with in-distribution results, **FSLR-trained general-purpose models also surpass math-specialized models on OOD benchmarks**. This demonstrates that a more focused supervising logical relationship understanding leads to more transferable reasoning capabilities. We provide complementary Out–of-distribution experiments with models trained on SVAMP in Appendix F, which show consistent results.

4.2.3 Performance Across Problem Complexity Levels

To understand how FSLR's benefits vary with problem difficulty, we analyze model performance across problems requiring different numbers of reasoning steps (2-8 steps) on the GSM8K test set. Figure 3 presents a radar chart comparing the base LLaMA-3.1-8B model, CoT-SFT, and FSLR across complexity levels. **FSLR demonstrates consistently superior performance across most complexity levels, with particularly strong advantages on challenging problems.** On simpler 2-4 step problems, FSLR achieves 82-87% accuracy compared to 78-79% for CoT-SFT and 59-69% for the base model, representing +4-8% improvements over CoT-SFT. Notably, on more complex 6-8 step problems where reasoning chains are much longer, FSLR shows dramatic gains compared to CoT-SFT: 67% vs 63% (6-step), 76% vs 67% (7-step), and 78% vs 56% (8-step), with the gap widening to +22% on 8-step problems. This pattern reveals that FSLR's focused training on understanding logical relationships provides compounding benefits in multi-step reasoning.

4.2.4 Training Efficiency

Beyond improved accuracy, **FSLR offers substantial computational advantages over CoT-SFT.** Table 3 shows that FSLR training data contains significantly fewer tokens: on GSM8K, FSLR produces sequences averaging 27–52 tokens, compared to 238–310 tokens for CoT-SFT (an 84-87% reduction), while on SVAMP, the reduction is similarly dramatic (34-46 tokens vs. 192-245 tokens, 81-86% reduction). This compression stems from FSLR's focus on first planning step rather than complete solution trajectories, eliminating the extensive computational execution steps that dominate CoT sequences.

Figure 4 demonstrates that these token savings directly translate to training acceleration. In GSM8K, the FSLR training is completed in 23-43 minutes compared to 144-175 minutes for CoT-SFT, achieving 4-6× speedup across all models and teacher configurations. Similar gains are observed on SVAMP (4–6× speedup), demonstrating that FSLR's training is not only more effective but also substantially more efficient.

4.2.5 Reliability of Error Attribution

To validate the robustness of error attribution, we conduct additional analysis using LLaMA-3.1-70B-Instruct and Qwen2.5-72B-Instruct as independent judges with the same classification prompt (Appendix G). As shown in Table 6, logical relationship understanding errors consistently account for over 90% of failures across all three judges, models, and methods. The convergence across judges with different architectures and training backgrounds substantially strengthens the validity of our error attribution. Furthermore, manual inspection of 100 randomly sampled cases confirms that judge classifications align with human judgment at over 95% accuracy.

4.2.6 Consistency Analysis of Teacher-Generated First Planning Steps

To investigate whether potential multiple first steps for a given problem introduce noise or limit the student model, we conduct a two-part consistency analysis on 500 randomly sampled problems from

Token length	Data Source	*LLaMA3.1-8B*		*Qwen2.5-7B*		*Qwen3-4B*	
		CoT-SFT	FSLR	CoT-SFT	FSLR	CoT-SFT	FSLR
GSM8K	**LLaMA**	237.64	**37.82**	254.98	**38.12**	254.98	**38.12**
	Qwen	293.76	**38.95**	310.38	**39.24**	310.38	**39.24**
	Self	256.41	**27.02**	301.22	**51.92**	262.11	**42.37**
SVAMP	**LLaMA**	195.51	**40.90**	207.43	**41.42**	207.43	**41.42**
	Qwen	206.19	**37.11**	215.12	**37.66**	215.12	**37.66**
	Self	244.67	**36.56**	217.36	**33.55**	192.36	**45.98**

Table 3: Average token length of training sequences for CoT-SFT and FSLR across different models and datasets. Green cells indicate FSLR uses fewer tokens than CoT-SFT (lower is better for efficiency).

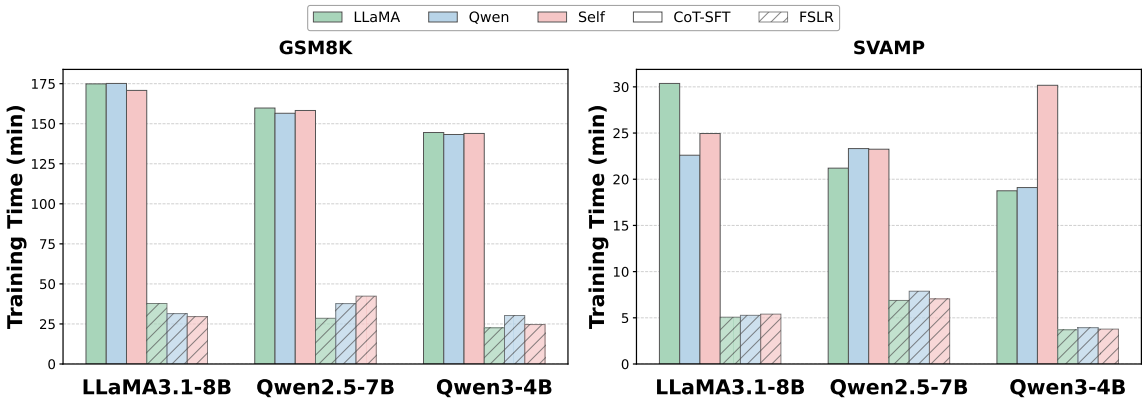

Figure 4: Training time (in minutes) for CoT-SFT and FSLR across different models and data sources. FSLR achieves substantial speedup over CoT-SFT on both GSM8K (left) and SVAMP (right), reducing training time by approximately 4-6× while maintaining competitive performance.

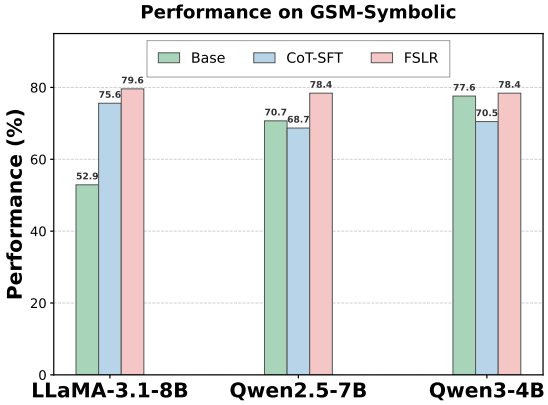

Figure 5: Zero-shot performance on GSM-Symbolic benchmark. All models are trained using LLaMA-3.1-70B as the teacher model. FSLR achieves the best performance across all three models, demonstrating superior generalization.

the GSM8K training set, using GPT-4o to judge semantic consistency of generated first planning steps.

Within-teacher consistency. We generate first planning steps using LLaMA-3.1-70B-Instruct at three temperatures (0, 0.5, 1.0) and evaluate whether outputs are semantically consistent across temperature settings. Results show that 84.7% of problems yield consistent first planning steps across all three temperatures, indicating that the first planning step is generally well-defined and robust to generation variance.

Across-teacher consistency. We further analyze whether different teacher models produce consistent first planning steps by comparing outputs from LLaMA-3.1-70B-Instruct, Qwen2.5-72B-Instruct, and the self-generated condition. We find that 75.4% of problems show consistent first planning steps across all three teachers. The 9.3% gap between within- and across-teacher consistency is expected given differences in model capabilities and training backgrounds, yet the majority of problems still yield consistent logical understanding across diverse teacher sources.

Together, the high consistency rates both within-teacher (84.7%) and across-teacher (75.4%) demonstrate that the first planning step is generally well-defined for mathematical problems.

4.2.7 Performance on GSM-Symbolic

To further validate FSLR's robustness beyond standard benchmarks, we evaluate on GSM-Symbolic (Mirzadeh et al., 2024), a benchmark specifically designed to test genuine reasoning. Figure 5 presents performance across all three models on GSM-Symbolic. **FSLR demonstrates superior robustness to problem variations.** While all models show performance drops compared to standard GSM8K, FSLR maintains substantially stronger performance than both base models and CoT-SFT. On LLaMA-3.1-8B, FSLR achieves 79.6% accuracy compared to 70.7% for CoT-SFT and 52.9% for the base model, achieving +8.9% improvement over CoT-SFT. Similarly, FSLR achieves improvements of +9.7% on Qwen2.5-7B (78.4% vs 68.7%) and +7.9% on Qwen3-4B (78.4% vs 70.5%) over CoT-SFT. This consistent advantage across different models suggests that FSLR's focused training on understanding logical relationships develops more robust reasoning capabilities, indicating that models trained with FSLR rely less on memorized patterns and more on genuine understanding of operational dependencies, which is the capability GSM-Symbolic is designed to test.

4.2.8 Case Study

To illustrate how FSLR improves logical relationship understanding, we present an example from GSM8K where the base model and CoT-SFT both fail, while FSLR succeeds. The following example uses LLaMA-3.1-8B as the target model with LLaMA-3.1-70B as the teacher model. Complete reasoning content is provided in Appendix H.

Problem: *Shiela bought 5 cell phones for $150 each for a 3-month installment. A 2% interest will be charged for each unit. How much will Shiela pay each month for 3 months?*

Ground Truth: *A $150 × 2/100 = $3 interest will be charged for each unit. So, each cell phone costs $150 + $3 = $153. The total amount to be paid for 5 cell phones is $153 × 5 = $765. Thus, Shiela will pay $765/3 = $255 each month.* **Answer: $255**

Base LLM: *...Interest = 2% of $150 = $3. Total interest = 3 × $3 = $9. Total amount = $750 + $9 = $759. Monthly payment = $759/3 = $253.* **Answer: $253**

CoT-SFT: *...The interest rate is 2% per unit, and there are 5 units. So, the total interest charged is: 5 × 2% = 10%. Total = $750 + $750 × 10% = $825. Monthly = $825/3 = $275.* **Answer: $275**

FSLR: *...Total cost = 5 × $150 = $750. Total interest: 2% of $750 = 0.02 × $750 = $15. Total amount = $750 + $15 = $765. Monthly payment = $765/3 = $255.* **Answer: $255**

Both the base LLM and CoT-SFT fail because they misidentify which quantity the interest rate should be applied to. The base model incorrectly interprets "interest for each unit" as applying to the payment period, computing "3 months × $3 = $9" total interest. CoT-SFT makes a different error: it aggregates the interest rate itself ("5 × 2% = 10%"), then applies this to the total cost, completely missing that "per unit" means the 2% must be calculated on the *individual unit price* of $150. In contrast, FSLR correctly identifies the operational structure, recognizing that the problem requires calculating "2% of $750", which properly captures the relationship between the interest rate, unit price, and quantity. Notably, all three models demonstrate sound arithmetic execution: the computational steps are performed correctly given their operational decisions.

5 Conclusion

In this work, we identified that logical relationship understanding errors constitute the primary bottleneck in mathematical reasoning, accounting for over 90% of incorrect predictions, and Chain-of-Thought Supervised Fine-Tuning(CoT-SFT) fails to substantially address this limitation. To bridge this gap, we proposed **F**irst-**S**tep **L**ogical **R**easoning (**FSLR**), a lightweight framework that provides a more focused training signal for logical relationship understanding by training models to identify the first planning step. Extensive experiments demonstrate that FSLR consistently outperforms CoT-SFT across multiple models and benchmarks, achieving stronger generalization on out-of-distribution tasks while requiring 81-87% fewer training tokens.

6 Limitations

Our work has several limitations. First, FSLR is evaluated on mathematical problems, and its effectiveness on other reasoning domains remains unexplored. Second, our framework relies on supervised fine-tuning with teacher-generated annotations, which may limit the model's ability. Exploring reinforcement learning approaches that reward correct logical relationship identification could potentially yield further improvements in logical relationship understanding and is left for future work.

References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*.

Guangsheng Bao, Hongbo Zhang, Cunxiang Wang, Linyi Yang, and Yue Zhang. 2025. How likely do llms with cot mimic human reasoning? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7831–7850.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ali Forootani. 2025. A survey on mathematical reasoning and optimization with large language models. *arXiv preprint arXiv:2503.17726*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, and 1 others. Math-perturb: Benchmarking llms' math reasoning abilities against hard perturbations, 2025. *URL https://arxiv.org/abs/2502.06453*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Hyeonbin Hwang, Doyoung Kim, Seungone Kim, Seonghyeon Ye, and Minjoon Seo. 2024. Self-explore: Enhancing mathematical reasoning in language models with fine-grained rewards. *arXiv preprint arXiv:2404.10346*.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. Mawps: A math word problem repository. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 1152–1157.

Jaehyeok Lee, Keisuke Sakaguchi, and JinYeong Bak. 2025. Self-training meets consistency: Improving llms' reasoning with consistency-driven rationale evaluation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10519–10539.

Nayoung Lee, Kartik Sreenivasan, Jason D Lee, Kangwook Lee, and Dimitris Papailiopoulos. 2023. Teaching arithmetic to small transformers. *arXiv preprint arXiv:2307.03381*.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. *arXiv preprint arXiv:2402.19255*.

Elita Lobo, Chirag Agarwal, and Himabindu Lakkaraju. 2025. On the impact of fine-tuning on chain-of-thought reasoning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11679–11698.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pages 975–984.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

Mihir Parmar, Palash Goyal, Xin Liu, Yiwen Song, Mingyang Ling, Chitta Baral, Hamid Palangi, and Tomas Pfister. 2025. Plan-tuning: Post-training language models to learn step-by-step planning for complex problem solving. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21430–21444.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.

Edgar Serna M. and Alexei Serna. 2015. Knowledge in engineering: A view from the logical reasoning. *International Journal of Computer Theory and Engineering*, 7:325–331.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan

Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, and 1 others. 2025. A survey of reasoning with foundation models: Concepts, methodologies, and outlook. *ACM Computing Surveys*, 57(11):1–43.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).

Peng-Yuan Wang, Tian-Shuo Liu, Chenyang Wang, Yi-Di Wang, Shu Yan, Cheng-Xing Jia, Xu-Hui Liu, Xin-Wei Chen, Jia-Cheng Xu, Ziniu Li, and 1 others. 2025. A survey on large language models for mathematical reasoning. *arXiv preprint arXiv:2506.08446*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Fangcong Yin, Zeyu Leo Liu, Liu Leqi, Xi Ye, and Greg Durrett. 2025. Learning composable chains-of-thought. *arXiv preprint arXiv:2505.22635*.

A Prompt for Error Analysis

Prompt Design. To categorize reasoning errors, we use GPT-4o (Hurst et al., 2024) with the following prompt template:

```
You are analyzing mathematical reasoning errors to identify failures in understanding logical relationships between variables.

Definition: Genuine logical reasoning requires understanding the logical relationships between variables in a problem, including:

1. Variable dependency: How variables depend on each other
2. Condition-solution mapping: How given conditions constrain the solution approach
3. Relevant information filtering: Which information is relevant vs. irrelevant to the solution
4. Logical step dependency: Each reasoning step logically follows from previous steps
5. Operation-relationship alignment: Choosing operations based on variable relationships, not surface-level keywords

Problem: [problem text]

Ground Truth Answer: [ground truth]

Model's Predicted Answer: [prediction]

Model's Reasoning Process: [reasoning]

Task: Categorize this error into ONE of the following categories:

1. STRUCTURAL_FAILURE: The error stems from misunderstanding logical relationships between variables

2. COMPUTATIONAL: The logical relationships are understood correctly, but arithmetic/calculation is wrong

3. COMPREHENSION: Misreading the problem statement itself

Response Format:
Category: [STRUCTURAL_FAILURE/COMPUTATIONAL/COMPREHENSION]
Explanation: [One sentence explaining why this category was chosen]
```

B Error Analysis

We present additional error analysis visualizations for LLaMA-3.1-8B and Qwen3-4B models to complement the Qwen2.5-7B analysis in Figure 1.

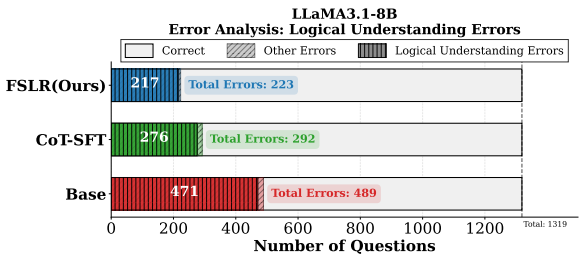

Figure 6: Error analysis on LLaMA3.1-8B comparing Base, CoT-SFT, and FSLR(Ours) models. Each bar shows the breakdown of correct predictions, logical relationship understanding errors, and other errors.

We observe consistent patterns across different models. On LLaMA-3.1-8B, FSLR reduces logical relationship understanding errors by 53.9% compared to the base model and 21.4% compared to CoT-SFT. On the stronger Qwen3-4B, logical relationship understanding errors still dominate (>90% of failures), and FSLR continues to outperform CoT-SFT. These results confirm that regardless of base model strength, FSLR effectively targets the primary bottleneck in mathematical reasoning.

In-Distribution Results		*LLaMA3.1-8B*		*Qwen2.5-7B*		*Qwen3-4B*		Average
Data Source	**Method**	**GSM8K**	**SVAMP**	**GSM8K**	**SVAMP**	**GSM8K**	**SVAMP**	
LLaMA	CoT-SFT	77.90	79.30	77.60	88.10	85.70	85.10	82.28
	Plan-and-Solve	81.70	83.30	83.70	89.00	**87.50**	90.10	85.88
	FSLR	**83.10**	**84.80**	**85.10**	**91.30**	87.10	**91.10**	**87.08**

Table 4: Comparison with Plan-and-Solve fine-tuning on in-distribution benchmarks GSM8K and SVAMP. All models use LLaMA-3.1-70B-Instruct as the teacher model. Green cells indicate the method outperforms CoT-SFT.

Out-of-Distribution Results		Models Trained on SVAMP					Average
Data Source	**Method**	**AsDIv**	**GSM8K**	**MAWPS**	**TabMWP**	**GSM-Hard**	
			Math-Specialized Models				
DeepSeek-Math	Zero-shot	85.00	82.20	92.50	69.90	56.10	77.14
Qwen2.5-Math	Zero-shot	82.50	85.50	92.30	53.60	55.40	73.86
			LLaMA3.1-8B				
Base LLM	Zero-shot	63.60	62.90	73.40	39.50	31.70	54.22
	Few-shot	85.80	77.50	97.00	55.00	38.70	70.80
LLaMA	CoT-SFT	75.70	77.40	82.10	47.10	34.00	63.26
	FSLR	**85.90**	**83.20**	**94.40**	**55.70**	**41.70**	**72.18**
Qwen	CoT-SFT	77.40	74.70	84.30	54.30	36.80	65.50
	FSLR	**79.90**	**79.60**	**94.20**	**59.00**	**39.20**	**70.38**
Self	CoT-SFT	73.60	75.40	78.10	40.40	35.90	60.68
	FSLR	**83.30**	**82.00**	**94.20**	**53.80**	**40.10**	**70.68**
			Qwen2.5-7B				
Base LLM	Zero-shot	84.20	72.60	90.80	61.20	53.40	72.44
	Few-shot	90.90	90.10	97.60	70.40	62.90	82.38
LLaMA	CoT-SFT	86.90	86.70	93.00	61.60	57.00	77.04
	FSLR	**90.50**	**91.10**	**94.80**	**67.20**	**57.20**	**80.16**
Qwen	CoT-SFT	81.40	80.40	85.80	48.60	**58.80**	71.00
	FSLR	**86.50**	**80.70**	**94.00**	**57.80**	57.50	**75.30**
Self	CoT-SFT	85.50	84.60	88.50	48.90	61.70	73.84
	FSLR	**86.00**	**90.80**	**92.60**	**54.70**	**66.20**	**78.06**
			Qwen3-4B				
Base LLM	Zero-shot	78.10	84.70	88.10	66.30	62.40	75.92
	Few-shot	88.40	84.80	95.90	70.30	56.00	79.08
LLaMA	CoT-SFT	81.10	85.70	89.30	62.10	58.50	75.34
	FSLR	**89.90**	**92.20**	**96.00**	**66.70**	**66.00**	**82.16**
Qwen	CoT-SFT	89.90	87.70	94.30	**66.60**	62.20	80.14
	FSLR	**90.20**	**90.00**	**96.20**	65.00	**63.70**	**81.02**
Self	CoT-SFT	89.10	90.10	96.20	**71.50**	55.00	80.38
	FSLR	**90.40**	**92.10**	**96.80**	70.20	**64.50**	**82.80**

Table 5: Out-of-distribution evaluation on five diverse benchmarks. Models are trained exclusively on SVAMP and evaluated on AsDIv, GSM8K, MAWPS, TabMWP, and GSM-Hard under zero-shot setting. Math-specialized models (DeepSeek-Math-7B and Qwen2.5-Math-7B) are evaluated zero-shot as reference baselines. Best results per setting are in **bold**. Green cells indicate FSLR outperforms CoT-SFT. Red cells indicate FSLR underperforms CoT-SFT.

C Training Data Example

To illustrate what the first-planning-step supervision looks like in practice, we provide a concrete example from our training data below.

Problem: *Betty picked 16 strawberries. Matthew picked 20 more strawberries than Betty and twice*

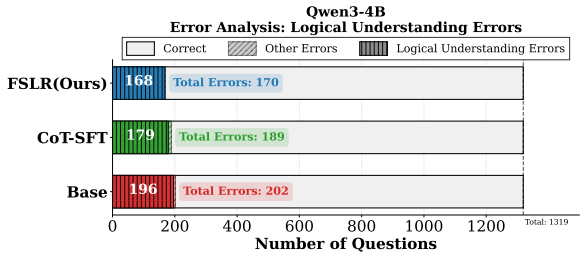

Figure 7: Error analysis on Qwen3-4B comparing Base, CoT-SFT, and FSLR(Ours) models. Each bar shows the breakdown of correct predictions, logical relationship understanding errors, and other errors.

as many as Natalie. They used their strawberries to make jam. One jar of jam used 7 strawberries and they sold each jar at $4. How much money were they able to make from the strawberries they picked?

First-Planning-Step (f_1): *First, we need to calculate the number of strawberries Matthew picked, which is 16 (Betty's strawberries) + 20 (more than Betty) = ?*

This example demonstrates how f_1 captures logical relationship understanding: the model identifies the relevant variables (Betty's 16 strawberries and the difference of 20), selects the appropriate operation (addition), and recognizes that "20 more than" implies an additive relationship. This isolation ensures the model is explicitly supervised on identifying the correct logical relationship from the problem statement.

D Comparison with Plan-and-Solve Fine-Tuning

To investigate whether FSLR's performance gains stem specifically from the first-step design or from general planning supervision, we compare FSLR against a Plan-and-Solve fine-tuning baseline (Parmar et al., 2025). Plan-and-Solve trains models to generate a complete plan before solving the problem, representing a broader form of planning supervision than FSLR's focused first-step approach.

Setup. Using the planning generation prompt template from the original Plan-and-Solve work, we generate training data with LLaMA-3.1-70B-Instruct as the teacher model, maintaining the same experimental setup as FSLR. Results are reported on in-distribution benchmarks GSM8K and SVAMP.

Results. As shown in Table 4, FSLR outperforms Plan-and-Solve by +1.20% on average across all models and datasets. While Plan-and-Solve already improves over CoT-SFT (+3.60% on average), FSLR achieves further gains, confirming that the first-step design contributes beyond general planning supervision. We attribute this to FSLR's more focused training signal: by isolating only the initial logical reasoning decision rather than generating a full plan, FSLR provides more direct supervision for logical relationship understanding without introducing the additional complexity of multi-step plan generation.

E Pass@k Evaluation on GSM8K

To investigate whether FSLR's improvements genuinely expanding model capability boundarie, we evaluate using Pass@k metrics on GSM8K, which measure whether the correct answer appears in k attempts and thus reflect the upper bound of model capability. As shown in Table 7, FSLR consistently outperforms both Base and CoT-SFT across all k values. The gains persist even as k increases (+1.74% at Pass@4, +1.16% at Pass@8, +0.68% at Pass@16), demonstrating that FSLR expands the model's capability boundary.

F Out-of-Distribution Performance (SVAMP Training)

Table 5 presents out-of-distribution results where models are trained exclusively on SVAMP and evaluated on five diverse benchmarks: AsDIv, GSM8K, MAWPS, TabMWP, and GSM-Hard. This complementary experiment validates whether the benefits of FSLR training generalize when using a different, smaller training dataset.

FSLR maintains strong generalization even with limited training data. Despite SVAMP being a smaller dataset than GSM8K, FSLR consistently outperforms CoT-SFT across nearly all configurations. On LLaMA-3.1-8B, FSLR achieves substantial improvements: +10.2% on AsDIv (85.9% vs 75.7% with LLaMA teacher), +16.1% on MAWPS (94.4% vs 78.1% with Self teacher), and +8.6% on TabMWP, averaging +8.92% improvement across all OOD datasets with the LLaMA teacher. Qwen2.5-7B and Qwen3-4B show similar trends, achieving average gains of +4.22% and +2.42%, respectively. These results demonstrate that FSLR's effectiveness is not dependent on large-scale training data. Consistent with GSM8K training results,

Judge	Model	Method	Total Errors	Logical Errors	Others
LLaMA-3.1-70B	LLaMA3.1-8B	Base	489	447	42
		CoT-SFT	292	282	10
		FSLR	223	215	8
	Qwen2.5-7B	Base	362	351	11
		CoT-SFT	296	276	20
		FSLR	197	185	12
	Qwen3-4B	Base	202	187	15
		CoT-SFT	189	172	17
		FSLR	170	162	8
Qwen2.5-72B	LLaMA3.1-8B	Base	489	443	46
		CoT-SFT	292	243	49
		FSLR	223	206	17
	Qwen2.5-7B	Base	362	330	32
		CoT-SFT	296	226	70
		FSLR	197	168	29
	Qwen3-4B	Base	202	164	38
		CoT-SFT	189	162	27
		FSLR	170	144	26

Table 6: Error attribution results using LLaMA-3.1-70B-Instruct and Qwen2.5-72B-Instruct as independent judges. Logical relationship understanding errors consistently account for over 90% of failures across all judges, models, and methods, consistent with the GPT-4o-based analysis in the main paper.

Metric	Base	CoT-SFT	FSLR
Pass@4	93.10	94.30	**96.04**
Pass@8	96.21	96.63	**97.79**
Pass@16	97.88	97.80	**98.56**

Table 7: Pass@k evaluation on GSM8K for LLaMA3.1-8B with LLaMA-3.1-70B-Instruct as teacher. FSLR consistently improves over Base and CoT-SFT across all k values.

FSLR-trained models surpass math-specialized models. The results suggest that teaching models *how to initiate reasoning* through logical relationship understanding is more effective for generalization than exposing them to large volumes of mathematical content. This reinforces our core hypothesis that the bottleneck in mathematical reasoning lies in the ability to identify and apply appropriate reasoning strategies from the problem's logical structure.

G Reliability of Error Attribution: Multi-Judge Analysis

Table 6 presents error attribution results using LLaMA-3.1-70B-Instruct and Qwen2.5-72B-Instruct as independent judges, complementing the GPT-4o-based analysis in the main paper. Logical relationship understanding errors consistently account for over 90% of failures across all judges, models, and methods, confirming the robustness of our error attribution.

H Complete Case Study

To illustrate how FSLR improves reasoning through better understanding of logical relationships, we present one example from GSM8K where LLaMA-3.1-8B-Instruct model and CoT-SFT both fail, while FSLR succeeds.

Problem: *Shiela bought 5 cell phones for $150 each for a 3-month installment. A 2% interest will be charged for each unit. How much will Shiela pay each month for 3 months?*

Ground Truth: *A $150 × 2/100 = $3 interest will be charged for each unit. So, each cell phone costs $150 + $3 = $153. The total amount to be paid for 5 cell phones is $153 × 5 = $765. Thus, Shiela will pay $765/3 = $255 each month.* ***Answer: $255***

LLaMA-3.1-8B-Instruct Response: *Step 1: Calculate the total cost of the cell phones. Total cost = 5 × $150 = $750. Step 2: Calculate the interest charged for each unit. Interest = 2% of $150 = 0.02 × $150 = $3. Step 3: Calculate the total interest charged for 3 months. Total interest = 3 × $3 = $9. Step 4: Calculate the total amount Shiela will pay each month for 3 months. Total amount = Total cost + Total interest = $750 + $9 = $759.*

Step 5: Calculate the monthly payment. Monthly payment = Total amount / 3 = $759 / 3 = $253. ***Answer: $253***

CoT-SFT Response: *First, we need to calculate the total cost of the cell phones. Shiela bought 5 cell phones for $150 each, so the total cost is: 5 × $150 = $750. Next, we need to calculate the interest charged for each unit. The interest rate is 2% per unit, and there are 5 units (cell phones). So, the total interest charged is: 5 × 2% = 10%. Now, we need to calculate the total amount Shiela needs to pay, including the interest. To do this, we'll add the total cost of the cell phones to the total interest charged: $750 (total cost) + $750 × 10% (interest) = $750 + $75 = $825. Since Shiela is paying in installments over 3 months, we need to divide the total amount by 3 to find out how much she'll pay each month: $825 ÷ 3 = $275.* ***Answer: $275***

FSLR Response: *First, calculate the total cost of the cell phones: 5 cell phones × $150 per cell phone = $750. Then, calculate the total interest: 2% of $750 = 0.02 × $750 = $15. Now, calculate the total amount Shiela will pay: $750 (cost of cell phones) + $15 (interest) = $765. Finally, calculate the monthly payment: $765 / 3 months = $255 per month.* ***Answer: $255***