Diagnosing and Mitigating Structural Pathologies in Mathematical Reasoning of Large Language Models

Anonymous ACL submission

Abstract

Despite recent progress, large language models (LLMs) for mathematical reasoning often exhibit fragile behaviors, where correct answers are produced despite invalid or incoherent intermediate reasoning. We diagnose two recurring structural pathologies in Chain-of-Thought (CoT) reasoning: disconnected steps, where intermediate results are not reused, and weak logical flow, where steps are loosely or incorrectly linked yet may yield correct answers. These failures are difficult to address under outcome-only supervision. To mitigate these issues, we propose the Graph-structured Stepwise Reasoning Framework (GSRF), which reformulates implicit CoT into a Graph-structured Stepwise CoT (GS-CoT) that makes inter-step dependencies explicit. Building on this structure, we introduce Graph-guided Group Relative Policy Optimization (G-GRPO), incorporating process-level rewards that encourage step reuse and alignment with the final answer. Extensive experiments on both textual and multimodal mathematical reasoning benchmarks demonstrate that GSRF achieves competitive performance while producing more faithful, coherent, and structurally grounded reasoning traces.

1 Introduction

LLMs have achieved impressive progress on mathematical and geometric reasoning (Yan et al., 2025; Chen et al., 2025). Yet their reliability remains fragile, i.e., final-answer correctness is often decoupled from the validity of the underlying reasoning process. A trustworthy solution should satisfy a simple contract, i.e., each intermediate step should produce a result that is subsequently reused to support the final conclusion. In practice, standard CoT (Wei et al., 2022; Wang et al., 2022; Kojima et al., 2022) frequently violates this contract, i.e., the models may reach the correct answer while producing traces that are redundant, partially hallucinated, or logically disconnected, resulting in a convincing appearance of reasoning without verifiable structural support.

We argue that a major source of this unreliability is the implicit, free-form nature of standard CoT. Most current paradigms treat reasoning as a flat sequence of text, which obscures the dependency structure that makes a derivation checkable. As shown in Fig 1, this implicit CoT often exhibits two recurring structural pathologies: (i) *Disconnected Steps*, where intermediate results are introduced but never used (Turpin et al., 2023; Lanham et al., 2023); and (ii) *Weak Logical Flow*, where the intermediate steps exhibit unclear or even incorrect logical transitions, but may lead to the correct answers (Lyu et al., 2023; Yee et al., 2024). Importantly, these pathologies are difficult to penalize under outcome-only supervision: as long as the final answer is correct, optimization signals provide little guidance on which steps are essential and which are merely spurious or redundant (Lightman et al., 2023).

To make reasoning verifiable and optimizable, we propose to capture reasoning faithfulness through explicit graph structure. We transform a solution into a Directed Dependency Graph (DDG), where nodes represent reasoning steps and edges encode variable-level reuse between steps. This view yields two efficiently graph-computable objectives that directly target the above failures: (i) *inter-step dependency*, encouraging steps to actually build on prior results; and (ii) *global step-to-answer support*, requiring that every step contributes (directly or indirectly) to the derivation of the final answer.

Based on these insights, we introduce a Graph-structured Stepwise Reasoning Framework (GSRF). GSRF replaces implicit CoT with an explicit and dependency-aware CoT called Graph-structured Stepwise CoT (GS-CoT), where step i is represented as a tuple (I_i, O_i) consisting of its inputs I_i (from problem conditions or previous

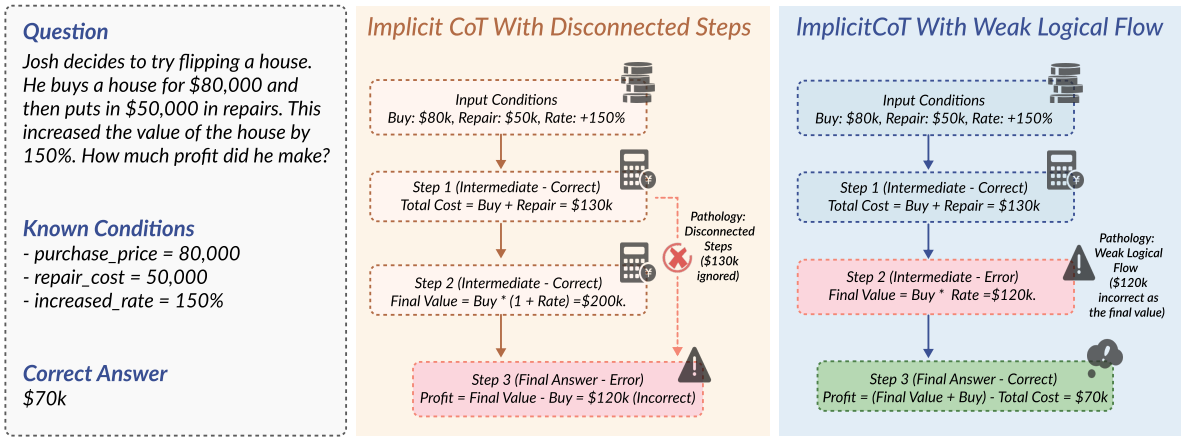

Figure 1: An example for solutions of implicit CoT with respectively disconnected steps and weak logical flow

outputs) and newly derived output O_i. To complement GS-CoT, we introduce Graph-guided Group Relative Policy Optimization (G-GRPO) to optimize the reasoning structure that moves beyond conventional outcome-level rewards (Zhang et al., 2025; Wang et al., 2025) and introduces two novel process-level reward signals: (i) *inter-step dependency reward* and (ii) *step-to-answer alignment reward* to assign process-level rewards for coherent reuse and global support. This directly discourages unused steps and promotes tight, reusable derivations. Experiments on both textual and multimodal benchmarks show that GSRF improves not only predictive accuracy but also the faithfulness and structural integrity of reasoning traces, suggesting a general principle for process optimization via explicit dependency structure. Our GSRF consists of two phases, and Fig. 2 shows an overview of our GSRF.

The **main contributions** of our work are fourfold. First, through systematic analysis of failure cases, we diagnose two fundamental structural pathologies in implicit Chain-of-Thought for mathematical reasoning: (i) *disconnected steps* and (ii) *weak logical flow*. Second, we propose **GSRF** that replaces implicit CoT with GS-CoT, enabling the explicit construction of a DDG to model inter-step dependencies. Third, to further optimize reasoning structure, we introduce G-GRPO within GSRF, which deemphasizes outcome-level rewards and instead leverages two process-level rewards to guide dependency-aware and globally aligned reasoning. Fourth, GSRF establishes a novel training paradigm for mathematical reasoning in LLMs that shifts the optimization focus from learning step-level correctness to modeling inter-step dependency structures, demonstrating that structural coherence can serve as an effective inductive bias for improving reasoning capabilities.

2 Related Works

Mathematical Reasoning with Large Language Models Recent studies have shown that LLMs exhibit impressive yet fragile reasoning abilities in mathematical domains. Early works such as GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) benchmarked models' symbolic reasoning capacities, revealing their tendency to produce correct answers through flawed reasoning chains. To mitigate this, recent approaches such as DeepSeekMath-RL (Shao et al., 2024) and Math-Shepherd (Wang et al., 2024) leverage large-scale mathematical data and process-aware supervision to improve numerical precision and reasoning stability. However, these methods primarily focus on outcome-level or local step supervision, without explicitly modeling global reasoning dependencies, which limits interpretability and structural faithfulness.

Chain-of-Thought and Structured Reasoning Chain-of-Thought supervision has become a cornerstone for improving LLM reasoning by exposing intermediate steps. Subsequent works such as Tree-of-Thought (Yao et al., 2023) and Graph-of-Thought (Besta et al., 2024) attempted to model reasoning as structured search or graph traversal, encouraging exploration of diverse reasoning paths. Yet, most of these methods rely on heuristic sampling and lack an explicit optimization objective to align reasoning structure with answer correctness, often leading to redundant or inconsistent reasoning steps.

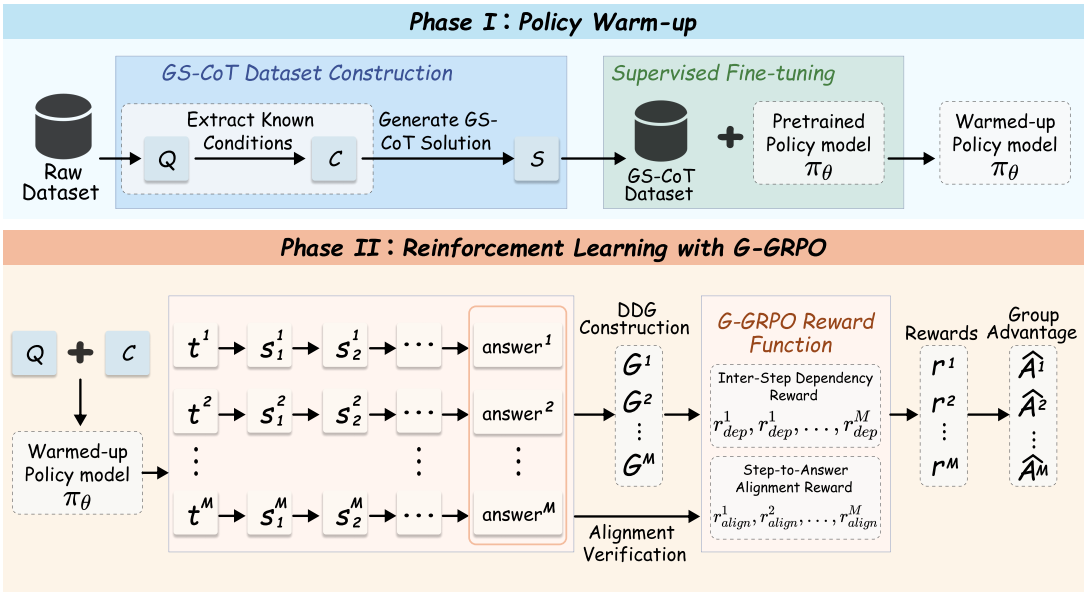

Figure 2: An overview of our GSRF consisting of two phases, i.e., Policy Warm-up and RL with G-GRPO.

Reinforcement Learning for Reasoning Optimization. Reinforcement learning (RL) has been increasingly used to optimize reasoning behavior in LLMs. Methods like GRPO (Shao et al., 2024) and R1 (Guo et al., 2025) directly optimize reward signals derived from reasoning outcomes and format, effectively improving factual accuracy and reasoning stability. However, these trajectory-level or outcome-level objectives ignore inter-step dependencies, which are particularly critical in mathematical reasoning where symbolic relations between steps dictate final correctness.

Multimodal and Structured Reasoning Extensions Recent efforts extend reasoning optimization into multimodal contexts, such as MathVista (Lu et al., 2023) and MM-ReAct (Yang et al., 2023). More recent work explores enhanced multimodal geometric reasoning, including unified vision–language pre-training methods for geometry such as GeoX (Xia et al., 2024), formal-verified multimodal problem generation via TrustGeoGen (Fu et al., 2025), and neuro-symbolic geometric data frameworks like NeSyGeo (Wu et al., 2025). Despite their progress, these approaches still ignore to explicitly model inter-step dependency and global structural consistency.

3 Methodology

3.1 Phase I: Policy Warm-up

GS-CoT Dataset Preparation. Before the policy warm-up phase, we construct a GS-CoT solution as the annotation for each mathematical problem. Specifically, given a mathematical problem Q, we first explicitly extract m known conditions, including numerical values, symbolic relations, and textual constraints, employing the DeepSeek-V3 model (Liu et al., 2024). These extracted conditions are denoted as $C = \{C_1, C_2, \ldots, C_m\}$. Next, given Q and its known conditions C, we employ DeepSeek-V3 again to generate a structured n-step solution $S = (s_1, s_2, \ldots, s_n)$, in which s_i $(i \leq n)$ strictly conforms to our predefined reasoning template with explicit inputs I_i and an output O_i where inputs I_i are subject to a following constraint:

$$I_i \subseteq \{C_1, C_2, \ldots, C_m\} \cup \{O_1, O_2, \ldots, O_{i-1}\}.$$

This constraint ensures that s_i depends only on the known conditions or the outputs of earlier steps, preserving dependency consistency in the reasoning process. Finally, we construct a high-quality GS-CoT dataset $\mathcal{D} = \{P_n | P_n = (Q_n, C_n, S_n)\}_{n=1}^{N}$ through a dual-review process combining the LLMs and human verification, which is then used for subsequent supervised fine-tuning. Note that, all GS-CoT annotations strictly satisfy the input–output dependency constraints by construction.

Supervised Fine-Tuning. After constructing the GS-CoT dataset, we train the policy model via supervised fine-tuning. This stage enables the model to learn explicit inter-step dependencies and to generate reasoning traces that strictly conform to the GS-CoT template. Formally, the supervised fine-tuning objective is defined as:

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{T} \log \pi_\theta(y_t \mid y_{<t}, Q, C) \quad (1)$$

where y_t denotes the t-th token in the reasoning sequence and Q and C are the input problem and known conditions, respectively. This stage establishes a strong initialization for subsequent policy refinement.

3.2 Phase II: RL with G-GRPO

To further refine the structural quality of the dependency graph and enhance alignment between reasoning steps and final answers, we introduce G-GRPO, which augments GRPO with two novel process-level rewards: Inter-Step Dependency Reward and Step-to-Answer Alignment Reward. Both rewards are computed from the Directed Dependency Graph (DDG), as illustrated in Fig. 3.

DDG Construction. Given a mathematical problem Q and its known conditions C , the warmed-up policy model generates its n-step GS-CoT solution $S = (s_1, s_2, \ldots, s_n)$. For the GS-CoT solution, we can define a DDG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = (v_0, v_1, \ldots, v_n)$. The initial node v_0 represents the set of known conditions C, and other nodes, e.g., $v_i (1 \leq i \leq n)$ represents the reasoning step s_i with its inputs I_i and output O_i. For an edge $e_{ij} \in \mathcal{E}$, $e_{ij} = 1$ if the input of v_j contains the output variable of v_i (i<j), that is $O_i \in I_j$; otherwise, $e_{ij} = 0$.

Inter-Step Dependency Reward. To encourage the model to produce reasoning chains that are both logically coherent and concise, we introduce a *dependency reward* r_{dep} based on the given DDG $\mathcal{G}$. Before calculating r_{dep}, we construct the adjacency matrix $A \in \{0,1\}^{N \times N} (N \geq 2)$ of DDG $\mathcal{G}$ with N nodes to describe inter-step dependencies and calculate the number of edges $|E(\mathcal{G})| = \sum_{i<j} A_{ij}$. We then calculate the inter-step dependency reward based on a fully connected structure:

$$r_{\text{dep}} = \begin{cases} \lambda + 2(1-\lambda)P, & \text{if } |E(\mathcal{G})| \geq N-1, \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where λ is the normalization factor, usually taken as 0.6 and $P = \dfrac{|E(\mathcal{G})| - (N-1)}{(N-1)(N-2)}$. In addition, $r_{\text{dep}} = \lambda$ When $N = 2$ to avoid division by zero and prevent score leakage, as a 2-node graph trivially satisfies connectivity without reflecting meaningful structural complexity.

This reward measures the proportion of valid dependency relations among all possible step pairs. A higher value indicates that reasoning steps are more tightly connected, reflecting a stronger structural dependency consistency. The discussion on dependency structure design is presented in Appendix A.

Step-to-Answer Alignment Reward. While the dependency reward encourages dense and coherent inter-step connections, it does not explicitly guarantee that all reasoning steps contribute to the final answer. To explicitly enforce global step-to-answer consistency, we introduce a graph-theoretic *Step-to-Answer Alignment Reward* based on reachability in the reasoning graph.

Given a DDG $\mathcal{G}$ with $n+1$ reasoning nodes $(v_0, v_1, v_2, \ldots, v_n)$, where v_n corresponds to the final answer node, we define the reasoning steps are *aligned* with the final answer if and only if every preceding node v_i $(0 \leq i < n)$ can directly or indirectly influence v_n. Formally, this requires that for the node v_i, there exists at least one directed path from v_i to v_n in the graph $\mathcal{G}$. From a graph-theoretic perspective, this is equivalent to checking whether v_n is reachable from all other nodes in the graph $\mathcal{G}$. Let I_{reach} denote an indicator function defined as:

$$I_{\text{reach}} = \begin{cases} 1, & \text{if } \forall 0 \leq i < n,\ \exists \text{ a path } v_i \rightsquigarrow v_n, \\ 0, & \text{otherwise.} \end{cases}$$

Given the standard answer accuracy reward r_{acc}, the step-to-answer alignment reward is defined as:

$$r_{\text{align}} = \begin{cases} 1, & \text{if } I_{\text{reach}} = 1 \text{ and } r_{acc} = 1, \\ \gamma, & \text{if } I_{\text{reach}} = 1 \text{ and } r_{acc} \neq 1, \\ 0, & \text{if } I_{\text{reach}} = 0 \end{cases} \quad (3)$$

where $\gamma \in (0,1)$ is a constant, empirically taken as 0.4, which assigns partial credit to reasoning graphs whose steps are structurally aligned with the final answer but yield an incorrect prediction.

This reward enforces that all preceding steps must support the final answer, either directly or indirectly, thereby discouraging redundant or disconnected steps and ensuring global coherence.

Policy Optimization. Given the inter-step dependency reward r_{dep} and the step-to-answer alignment

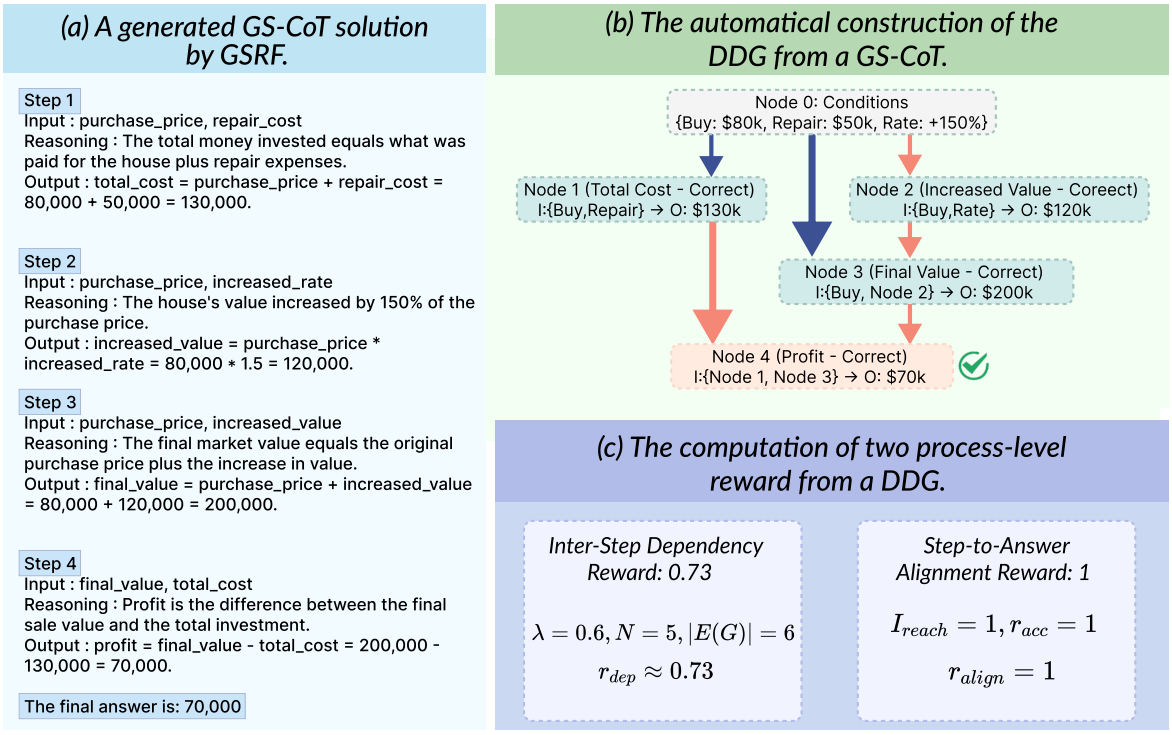

Figure 3: An overview on how to automatically construct a DDG from a GS-CoT data and calculate r_{dep} and r_{align} (we use the same case in Fig. 1 and the two paths highlighted by orange arrows illustrate that every intermediate node in the DDG has a directed path to the final node).

reward r_{align} computed from the DDG $\mathcal{G}$, we define the overall reward for the i-th reasoning path as $r^i = r^i_{\text{dep}} + r^i_{\text{align}}$. By sampling M reasoning paths, we obtain a reward set $\{r^1, r^2, \ldots, r^M\}$, based on which we compute the relative advantage for each reasoning path as follows:

$$\hat{A}^i = \frac{r^i - mean(\{r^1, r^2, \ldots, r^M\})}{std(\{r^1, r^2, \ldots, r^M\})} \tag{4}$$

The mean group reward serves as a baseline, with $\hat{A}_i$ measures the deviation r_i deviates from the group average. We then optimize the policy model using the following loss function:

$$\mathcal{L}_{\text{G-GRPO}} = -\mathbb{E}_{Q \in D}\left[\frac{1}{M}\sum_{i=1}^{M}\left(\frac{\pi_\theta(c^i|Q)}{\pi_{\theta_{\text{old}}}(c^i|Q)}\right)\hat{A}^i - \beta D_{\text{KL}}\left(\pi_\theta \| \pi_{\text{ref}}\right)\right] \tag{5}$$

where a KL-divergence regularization term is applied to constrain the deviation of the policy model from a reference model. The reference model is initialized identically to the policy model and remains frozen throughout reinforcement learning. The KL divergence is estimated as follows:

$$D_{KL}(\pi_\theta \| \pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(c^i|Q)}{\pi_\theta(c^i|Q)} - \log\frac{\pi_{\text{ref}}(c^i|Q)}{\pi_\theta(c^i|Q)} - 1 \tag{6}$$

The overall training pipeline is summarized in Appendix B.

4 Experiments

4.1 Experimental Setup

We conduct experiments on both textual and multimodal mathematical reasoning benchmarks to comprehensively evaluate the effectiveness of GSRF.

Textual Mathematical Reasoning. For pure textual reasoning, we evaluate our method on two widely used mathematical reasoning benchmarks, i.e., **GSM8K** and **MATH**, via Qwen2.5-3B and Qwen2.5-7B (Base Models) (Qwen et al., 2025). GSM8K focuses on grade-school arithmetic, while MATH contains competition-level problems requiring complex symbolic reasoning. These two datasets together provide a comprehensive testbed for assessing structured reasoning ability under varying difficulty levels.

Multimodal Mathematical Reasoning. For multimodal reasoning, we adopt the GeoQA (Chen

et al., 2021) benchmark and the two baseline models, i.e., Qwen2.5-VL-3B and Qwen2.5-VL-7B (Bai et al., 2025). GeoQA consists of geometry problems that require joint reasoning over visual diagrams and textual descriptions, posing significant challenges for cross-modal alignment and multi-step deduction.

Evaluation Metrics. We adopt **exact-match accuracy** as the primary metric. For GSM8K, we extract the final numerical result; for MATH, we use symbolic equivalence after normalization. For GeoQA, we evaluate under two protocols: *open-ended* (same as GSM8K) and *multiple-choice*. In the multiple-choice setting, we additionally provide the full set of candidate answer options for model and require it to select the the most consistent one. Fig. 4 presents a quantitative comparison between Qwen2.5-VL-7B and GSRF-VL-7B under the multiple-choice setting on GeoQA. Meanwhile, To quantify structural quality of GSRF, we use APC (Average Predecessor Count): APC $= \frac{1}{n}\sum_{i=1}^{n} \text{pred}(v_i)$, where $\text{pred}(v_i)$ is the number of incoming edges to node v_i. Higher APC indicates stronger inter-step dependencies, with a theoretical maximum of $(n-1)/2$ for an n-node graph.

Implementation details can be found in Appendix D.

4.2 Main Results and Analyses

Results on Textual Mathematical Reasoning. Table 1 reports the performance of GSRF on GSM8K and MATH compared with closed-source and open-source models. GSRF consistently brings substantial improvements over Qwen2.5 baselines across both benchmarks and model scales. On GSM8K, GSRF-3B and GSRF-7B achieve 88.4% (+9.3%) and 93.6% (+8.2%), respectively. On MATH, the gains are even more pronounced: GSRF-3B reaches 64.0% (+21.4%) and GSRF-7B reaches 74.3% (+24.5%).

In addition, Table 2 reports the average APC and node count of DDGs on GSM8K and MATH. GSRF-3B and GSRF-7B both produce more structurally sound reasoning chains, with APC values approaching the expected theoretical maximum $(n-1)/2$ (n is the average node count) and fewer redundant nodes. These structural improvements mitigate the two pathologies in Fig. 1. Expected APC indicates frequent reuse of intermediate results, mitigating *disconnected steps*. Meanwhile, fewer redundant nodes combined with GSRF's alignment reward reduce *weak logical flow* by penalizing steps that do not contribute to the final answer. Detailed pathology statistics are provided in Appendix C.

Table 1: Performance of GSRF on GSM8K and MATH.

Model	GSM8K	MATH
Closed-Source Models		
GPT4	92.0	52.9
Claude 3 Opus	95.0	60.1
Open-Source Models		
LLama3-70B-instruct	93.0	51.0
Qwen2.5-Math-72B	90.8	66.8
DeepSeekMath-RL-7B	88.2	51.7
Qwen2.5-3B (Baseline)	79.1	42.6
GSRF-3B (Ours)	**88.4**	**64.0**
Qwen2.5-7B (Baseline)	85.4	49.8
GSRF-7B (Ours)	**93.6**	**74.3**

Table 2: Structural analysis of GSRF on GSM8K and MATH.

Model	GSM8K		MATH	
	Nodes	APC	Nodes	APC
Qwen2.5-3B+Warm-up	4.265	0.772	3.419	0.591
GSRF-3B	3.193	1.021	3.114	0.983
Qwen2.5-7B+Warm-up	4.615	1.348	5.621	1.889
GSRF-7B	2.481	0.723	4.243	1.591

Results on Multimodal Mathematical Reasoning. Table 3 summarizes the results on GeoQA under both multiple-choice and open-ended protocols. Under the multi-choice setting, GSRF-VL-3B improves accuracy to 67.9% (+29.3%), while GSRF-VL-7B achieves 73.5% (+14.2%). Notably, GSRF-VL-7B surpasses or matches strong open-source models such as MAVIS-7B (Zhang et al., 2024), G-LLaVA-13B (Gao et al., 2023), and Chimera-Reasoner-8B (Peng et al., 2025). Similar trends are observed in the open-ended setting: GSRF-VL-3B improves from 37.0% to 60.0%, and GSRF-VL-7B from 46.2% to 68.4%, demonstrating effective generalization to free-form solution generation.

Table 4 presents the structural analysis on GeoQA, showing similar trends as observed on GSM8K and MATH, indicating that GSRF can also enhance the structural quality of multimodal reasoning, not just textual reasoning.

4.3 Ablation Studies

To further validate the effectiveness of GSRF, we conducted a series of ablation experiments. All ablation experiments follow the same training and

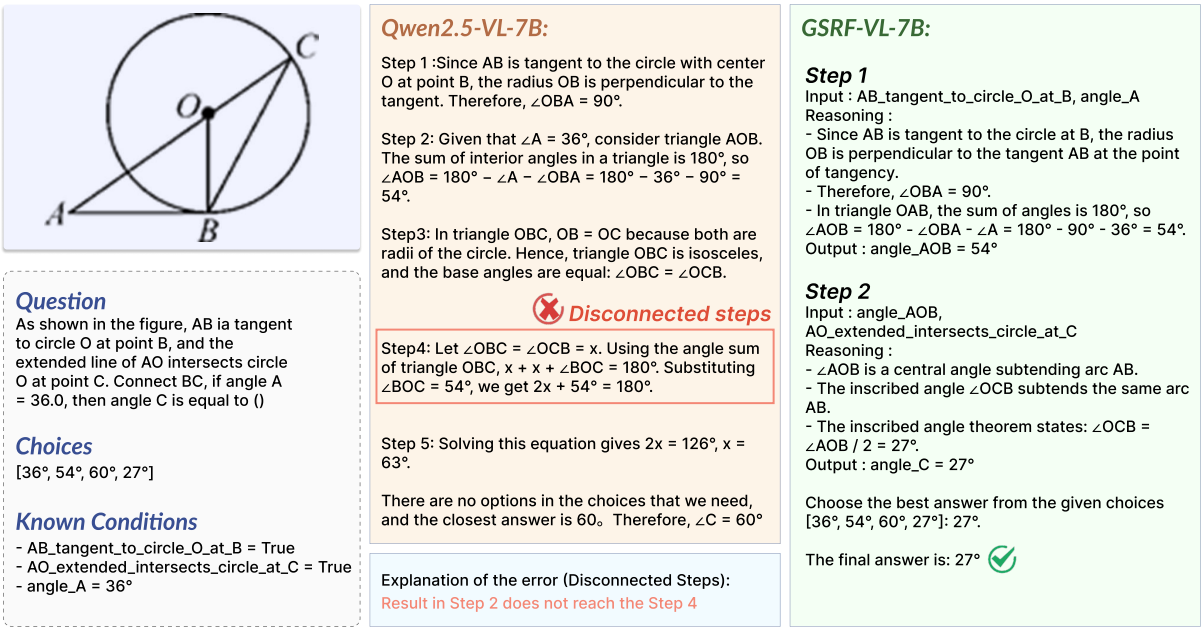

Figure 4: Quantitative comparison between Qwen2.5-VL-7B and GSRF-VL-7B (multi-choice) on GeoQA.

Table 3: Performance of GSRF on GeoQA.

Model	Accuracy
Multi-choice	
Closed-Source Models	
GPT-4o	58.9
Claude-3.5-Sonnet	65.1
Open-Source Models	
MAVIS-7B	68.3
G-LLaVA-13B	67.0
Chimera-Reasoner-8B	69.6
Qwen2.5-VL-3B (Baseline)	38.6
GSRF-VL-3B (Ours)	**67.9**
Qwen2.5-VL-7B (Baseline)	59.3
GSRF-VL-7B (Ours)	**73.5**
Open-ended	
Qwen2.5-VL-3B (Baseline)	37.0
GSRF-VL-3B (Ours)	**60.0**
Qwen2.5-VL-7B (Baseline)	46.2
GSRF-VL-7B (Ours)	**68.4**

Table 4: Structural analysis of GSRF on GeoQA in open-ended setting.

Model	Nodes	APC
Qwen2.5-VL-3B + Warm-up	4.468	0.945
GSRF-VL-3B	2.878	0.920
Qwen2.5-VL-7B + Warm-up	5.271	1.327
GSRF-VL-7B	3.414	1.189

Table 5: Comparison of different RL strategies.

Model	GSM8K	MATH
Qwen2.5-3B	79.1	42.6
+Warm-up	80.6	57.4
+Warm-up+GRPO	85.4	65.9
+Warm-up+GMPO	84.7	65.6
+Warm-up+G-GRPO	**88.4**	**64.0**
Qwen2.5-7B	85.4	49.8
+Warm-up	89.5	66.8
+Warm-up+GRPO	90.0	71.4
+Warm-up+GMPO	91.4	72.9
+Warm-up+G-GRPO	**93.6**	**74.3**

evaluation protocols as the main experiments unless otherwise specified.

Comparison of RL Strategies. Table 5 compares different reinforcement learning strategies under the same backbone and training data. Policy warm-up (GS-CoT) consistently improves performance on both GSM8K and MATH, indicating that graph-structured supervision provides a strong initialization for RL. On GSM8K, **G-GRPO** achieves the best results for both model scales. For the 3B model on MATH, G-GRPO (64.0%) is slightly below GRPO (65.9%) and GMPO (Zhao et al., 2025) (65.6%), likely due to the limited capacity of smaller models when enforcing stricter dependency constraints. For the 7B model, G-GRPO consistently outperforms GRPO and GMPO on both benchmarks, reaching 93.6% on GSM8K and 74.3% on MATH.

Effect of Graph Structure Beyond Distillation. To disentangle whether the gains of GSRF stem solely from distillation from a stronger teacher (DeepSeek-V3) or also from the proposed graph-

Table 6: Comparison between Linear CoT and GS-CoT using the same teacher model.

Model	GSM8K	MATH
Qwen2.5-7B	85.4	49.8
+ Warm-up (Linear CoT)	89.2	59.1
+ Warm-up (Linear CoT) + GRPO	92.0	71.7
+ Warm-up (GS-CoT)	89.5	66.8
+ Warm-up (GS-CoT) + G-GRPO	**93.6**	**74.3**

Table 7: Ablation of reward components.

Model	GSM8K	MATH
Qwen2.5-7B + Warm-up	89.5	66.8
+ GRPO (Standard outcome-only)	90.0	71.4
+ G-GRPO (without r_{dep})	91.1	70.9
+ G-GRPO (without r_{align})	85.6	63.8
+ Full G-GRPO	**93.6**	**74.3**

structured reasoning itself, we construct a strict control baseline where Qwen2.5-7B is fine-tuned on standard **Linear CoT** traces generated by DeepSeek-V3 and further optimized with GRPO. This setup ensures both approaches use the same teacher supervision and RL pipeline, differing only in reasoning representation: *linear CoT* vs. *Graph-Structured CoT*. Results are shown in Table 6. While Linear CoT distillation significantly improves the base model, GS-CoT consistently performs better. This demonstrates that the improvements cannot be solely attributed to distillation; the proposed graph-structured formulation provides additional benefits beyond standard CoT supervision.

Reward Component Analysis. We further analyze the contributions of the two reward components in G-GRPO: r_{dep} and r_{align}. Table 7 reports results when removing each component separately. Removing r_{dep} leads to slightly worse performance compared with the full model (comparable to standard GRPO), indicating that dependency optimization provides useful structural supervision. In contrast, removing r_{align} causes a significant performance drop, suggesting that alignment regularization is crucial for stabilizing RL training. Overall, the best performance is achieved when both rewards are jointly applied, demonstrating their complementary roles in optimizing structured reasoning.

Parameter Analysis. We conduct the parameter analysis on the number of generations M. As shown in Fig. 5, increasing M from 2 to 4 leads to consistent performance gains, while further increasing M yields marginal or no improvements. The results indicate that G-GRPO is not highly sensitive to the M, and a moderate value of M already provides competitive and stable performance.

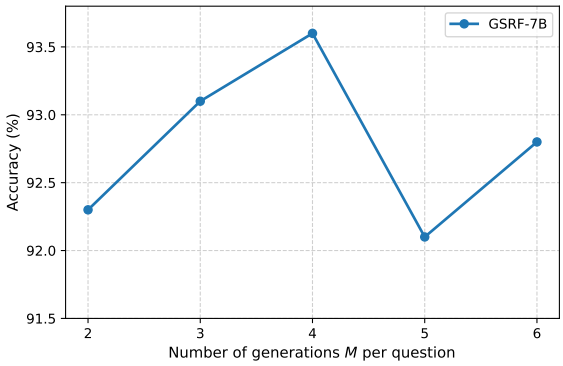

Figure 5: Parameter analysis of M on GSM8K.

5 Conclusion

We identify two structural pathologies in implicit Chain-of-Thought reasoning for mathematics: *disconnected steps* and *weak logical flow*. To address these issues, we propose GSRF, a graph-structured reasoning framework that makes step dependencies explicit and optimizable through Graph-structured Stepwise CoT (GS-CoT) and Directed Dependency Graphs (DDGs) and introduce Graph-guided Group Relative Policy Optimization (G-GRPO) in GSRF, which incorporates process-level rewards to penalize these pathological traces. Experiments on both textual and multimodal benchmarks demonstrate that GSRF achieves competitive performance while producing more faithful and structurally coherent reasoning traces.

Limitations

Our approach has several limitations. First, due to the structured nature of GS-CoT solutions, they cannot be reliably generated through prompt engineering alone. Instead, a GS-CoT dataset must be constructed in advance to warm up the model, which incurs additional annotation and training cost. Second, the two proposed rewards are complementary and need to be applied jointly. Third, the step-level reward design introduces the risk of reward hacking. Further analysis and discussion are provided in Appendix E.

References

The claude 3 model family: Opus, sonnet, haiku.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17682–17690.

Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric Xing, and Liang Lin. 2021. Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 513–523.

Zui Chen, Tianqiao Liu, Mi Tian, Qing Tong, Weiqi Luo, and Zitao Liu. 2025. Advancing mathematical reasoning in language models: The impact of problem-solving data, data synthesis methods, and training stages. *arXiv preprint arXiv:2501.14002*.

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*.

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

Daocheng Fu, Zijun Chen, Renqiu Xia, Qi Liu, Yuan Feng, Hongbin Zhou, Renrui Zhang, Shiyang Feng, Peng Gao, Junchi Yan, and 1 others. 2025. Trustgeogen: Scalable and formal-verified data engine for trustworthy multi-modal geometric problem solving. *arXiv preprint arXiv:2504.15780*.

Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, and 1 others. 2023. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

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*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023)*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Tianshuo Peng, Mingsheng Li, Jiakang Yuan, Hongbin Zhou, Renqiu Xia, Renrui Zhang, Lei Bai, Song Mao, Bin Wang, Aojun Zhou, and 1 others. 2025. Chimera: Improving generalist model with domain-specific experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3011–3022.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan

Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Teng Wang, Zhangyi Jiang, Zhenqi He, Shenyang Tong, Wenhan Yang, Yanan Zheng, Zeyu Li, Zifan He, Hailei Gong, Zewen Ye, and 1 others. 2025. Towards hierarchical multi-step reward models for enhanced reasoning in large language models. *arXiv preprint arXiv:2503.13551*.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Weiming Wu, Jin Ye, Zi-kang Wang, Zhi Zhou, Yu-Feng Li, and Lan-Zhe Guo. 2025. Nesygeo: A neuro-symbolic framework for multimodal geometric reasoning data generation. *arXiv preprint arXiv:2505.17121*.

Renqiu Xia, Mingsheng Li, Hancheng Ye, Wenjie Wu, Hongbin Zhou, Jiakang Yuan, Tianshuo Peng, Xinyu Cai, Xiangchao Yan, Bin Wang, and 1 others. 2024. Geox: Geometric problem solving through unified formalized vision-language pre-training. *arXiv preprint arXiv:2412.11863*.

Yibo Yan, Jiamin Su, Jianxiang He, Fangteng Fu, Xu Zheng, Yuanhuiyi Lyu, Kun Wang, Shen Wang, Qingsong Wen, and Xuming Hu. 2025. A survey of mathematical reasoning in the era of multimodal large language model: Benchmark, method & challenges. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11798–11827.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. 2023. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Evelyn Yee, Alice Li, Chenyu Tang, Yeon Ho Jung, Ramamohan Paturi, and Leon Bergen. 2024. Dissociation of faithful and unfaithful reasoning in llms. *arXiv preprint arXiv:2405.15092*.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. 2025. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *arXiv preprint arXiv:2503.12937*.

Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, and 1 others. 2024. Mavis: Mathematical visual instruction tuning with an automatic data engine. *arXiv preprint arXiv:2407.08739*.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, Fang Wan, and Furu Wei. 2025. Geometric-mean policy optimization. *Preprint*, arXiv:2507.20673.

A Discussion on Dependency Structure Design

In this section, we provide an in-depth analysis of why the fully-connected dependency structure outperforms the chain structure in our Graph-structured Stepwise Reasoning Framework (GSRF). Our experiments demonstrate consistent performance advantages of the fully-connected structure across different model scales, as shown in Table 8.

Table 8: Performance comparison between chain and fully-connected dependency structures.

Model	Structure	GSM8K	MATH
GSRF-3B	Chain	86.9	62.8
GSRF-3B	Fully-Connected	**88.2**	**64.1**
GSRF-7B	Chain	89.1	68.3
GSRF-7B	Fully-Connected	**89.5**	**69.7**

The fundamental difference between these two structures lies in information flow and accessibility. In a chain dependency structure, information flows linearly: Step 1 $\rightarrow$ Step 2 $\rightarrow$ Step 3 $\rightarrow$ ... $\rightarrow$ Step N. Each step can only access outputs from its immediate predecessor, creating several critical limitations. First, if Step i produces valuable intermediate results, Step $i+2$ or later steps cannot directly access this information without it being explicitly passed through Step $i+1$, creating an information bottleneck. Second, a single error in Step i affects all subsequent steps, as they depend on the corrupted information flowing through the chain. Third, each step operates with a restricted view of the problem-solving process, seeing only the immediately preceding step's output. In contrast, the fully-connected structure allows each step to access outputs from *all* previous steps, providing rich contextual awareness where each step can leverage the complete history of intermediate computations, enabling more informed decision-making.

This architectural difference directly mitigate the two pathological structures: disconnected steps and weak logical flow. In a chain structure, if Step i's output is not needed by Step $i+1$, it becomes disconnected from the reasoning path. The fully-connected structure provides multiple opportunities for each step's output to be utilized by any subsequent step, significantly reducing the likelihood of disconnection. Similarly, when a step contains a logical error in a chain structure, subsequent steps have no alternative but to work with this flawed information. The fully-connected structure enables later steps to cross-reference multiple previous steps, potentially identifying and correcting inconsistencies.

The quality of the dependency reward signal also differs substantially between the two structures. The Inter-Step Dependency Reward (r_{dep}) measures the connectivity of the reasoning graph. In a fully-connected structure, the model receives richer gradient signals about which dependencies are truly useful versus spurious. The optimization process can learn to selectively attend to relevant prior steps while ignoring irrelevant ones, and the reward function can better distinguish between meaningful logical connections and superficial dependencies. In a chain structure, the dependency reward is constrained to only evaluate sequential connections, providing less informative feedback for learning optimal reasoning patterns. Our analysis of the trained models reveals that models with fully-connected structures learn to selectively utilize relevant prior steps rather than uniformly attending to all connections, suggesting the model develops sophisticated dependency selection strategies.

An important advantage of our approach is that the fully-connected structure does not introduce additional computational overhead during training or inference. This is because the Directed Dependency Graph (DDG) is constructed through our GS-CoT (Graph-structured Chain-of-Thought) process, which explicitly identifies the actual dependencies between steps based on their semantic relationships.

In summary, the fully-connected dependency structure's superiority stems from its ability to provide richer contextual information to each reasoning step, reduce pathological reasoning patterns, enable more robust error handling through multiple information pathways, and generate higher-quality dependency reward signals for training.

B GSRF Training Pipeline

The overall training pipeline of GSRF is summarized in Algorithm 1.

C Pathological diagnostic method

To systematically identify the two structural pathologies in reasoning traces, we employ a automated diagnostic approach combining large language model analysis with human verification.

Algorithm 1 GSR Training Pipeline

Input: Pretrained policy model π_θ; GS-CoT dataset $\mathcal{D} = \{P_n\}_{n=1}^{N}$.
Output: Trained policy model π_θ

Policy Warm-up:
for iter $= 1$ to I **do**
  Sample P from $\mathcal{D}$
  Update policy model π_θ by Eq. (1)
end for

RL with G-GRPO:
for iter $= 1$ to I **do**
  Sample problem P from $\mathcal{D}$
  Generate M reasoning trajectories $\{t^i\}_{i=1}^{M}$ and Parse steps and construct dependency graph $\{\mathcal{G}^i\}_{i=1}^{M}$
  Compute inter-step dependency reward $\{r^i_{\text{dep}}\}_{i=1}^{M}$ by Eq. (2) and step-to-answer alignment reward $\{r^i_{\text{align}}\}_{i=1}^{M}$ by Eq. (3)
  Compute the relative advantage $\hat{A}^i$ by Eq. (4)
  Optimize Policy model π_θ by Eq. (5)–(6)
end for
  return policy model π_θ

Specifically, we use DeepSeek-V3 with a carefully designed prompt to analyze each reasoning solution and detect the presence of *disconnected steps* and *weak logical flow*. The prompt instructs the model to: (1) decompose the solution into structured reasoning steps with explicit input-output dependencies, (2) construct a dependency graph to trace variable reuse, and (3) identify pathologies based on graph-theoretic criteria such as reachability and logical coherence. Part of model-generated diagnoses are subsequently reviewed by human annotators to ensure accuracy. Table 9 presents the diagnostic statistics on GSM8K and MATH. The results show that GSRF significantly reduces the occurrence of both pathologies compared to the warm-up baselines across both model scales. These reductions confirm that the proposed graph-structured supervision and process-level rewards effectively mitigate structural pathologies in mathematical reasoning.

D Implementation Details

For all datasets, we first employ **DeepSeek V3** to automatically extract the *known conditions* for each mathematical problem, including numerical constants, symbolic relations, and textual constraints. Based on these extracted conditions, DeepSeek V3 is further prompted to generate structured solutions that strictly follow our graph-structured reasoning template. This procedure is applied uniformly to all training data in both text-only and multimodal settings.

In policy warm-up phase, we fine-tune all backbone models using supervised learning on the generated graph-structured reasoning data. To ensure parameter-efficient adaptation across different model scales, we adopt **LoRA** for fine-tuning. Specifically, we apply LoRA to all eligible modules with rank 8 and scaling factor 16. The per-device training batch size is set to 8, and the learning rate is fixed at 5×10^{-5}.

In RL with G-GRPO phase, both the policy model and the reference model are initialized from the warmed-up checkpoint, but the reference model remains frozen throughout this phase. we further optimize the policy model using the proposed Graph-guided Group Relative Policy Optimization (G-GRPO). For each question, we sample 4 reasoning trajectories to form a comparison group, with the maximum response length set to 1024 tokens. The policy model is optimized with a learning rate of 1×10^{-6}, and the per-device training batch size is 32. The KL regularization coefficient between the policy and reference models is set to 0.001 to stabilize training while allowing sufficient exploration. All experiments are conducted on 8 RTX L40 48GB GPUs.

E Additional Analysis and Discussion

Effect of Model Backbone. Table 10 shows the performance of GSRF when applied to LLaMA3.1-8B as the backbone model on GSM8K. Compared to LLaMA3.1-8B-instruct, GSRF improves accuracy from 84.5% to 90.5%, yielding a gain of +6.0%. This result demonstrates that GSRF is not restricted to the Qwen family and can be effectively transferred to different backbone architectures, consistently enhancing mathematical reasoning performance.

Comparison with Tree-of-Thought (ToT). A natural point of comparison is Tree-of-Thought (ToT), which also introduces non-linear structure into the reasoning process. However, GSRF and ToT differ fundamentally in both their design philosophy and operational paradigm.

ToT is an *inference-time search heuristic*: it organizes candidate reasoning steps as a tree, uses heuristic value functions to evaluate intermediate nodes, and employs backtracking to navigate toward a correct final answer. This design is inherently exploratory — it assumes the base model may produce flawed reasoning and compensates by searching over multiple alternative branches at test

Table 9: Diagnostic statistics of two pathology.

Model	Disconnected Steps	Weak Logical Flow	Both
GSM8K			
Qwen2.5-3B + Warm-up	88 (6.7%)	316 (24.0%)	83 (6.3%)
GSRF-3B	49 (3.7%)	191 (14.5%)	33 (2.5%)
Qwen2.5-7B + Warm-up	70 (5.3%)	247 (18.7%)	59 (4.5%)
GSRF-7B	44 (3.3%)	148 (11.2%)	40 (3.0%)
MATH			
Qwen2.5-3B + Warm-up	491 (9.8%)	2523 (50.5%)	451 (9.0%)
GSRF-3B	427 (8.5%)	2064 (41.3%)	379 (7.6%)
Qwen2.5-7B + Warm-up	468 (9.4%)	1921 (38.4%)	411 (8.2%)
GSRF-7B	354 (7.1%)	1273 (25.5%)	291 (5.8%)

Table 10: Performance on GSM8K with LLaMA3.1-8B as the backbone.

Model	Accuracy (%)
LLaMA3.1-8B-Instruct	84.5
GSRF-8B	**90.5**

time. As a result, ToT incurs significant inference overhead, requiring multiple forward passes and external evaluation calls per query, without modifying the model's underlying reasoning behavior. GSRF, by contrast, is a *training-time structural optimization* framework. Rather than searching over alternative paths at inference, GSRF embeds structural constraints directly into the model's weights through G-GRPO. By representing each solution as a Directed Dependency Graph (DDG) and optimizing two process-level rewards — inter-step dependency reward (r_{dep}) and step-to-answer alignment reward (r_{align}) — GSRF trains the model to *intrinsically* produce structurally faithful reasoning traces. Consequently, at inference time, GSRF requires only standard single-pass decoding with no external search, tree expansion, or multi-branch evaluation.

The topological distinction is equally important. ToT constructs a *multi-branch tree* to explore alternative reasoning paths, where each branch represents a different candidate continuation. GSRF instead constructs a *single-path dependency graph* within one solution trajectory, where edges encode variable-level reuse between steps. This means GSRF does not aim to find the best path among many; it trains the model to generate a single, internally consistent, and dependency-grounded path from the outset.

In summary, ToT addresses reasoning failures by searching around them at test time, while GSRF addresses them by eliminating them during training. The two approaches are therefore complementary: GSRF could in principle be combined with inference-time search methods for settings where both structural faithfulness and exploratory robustness are required.

Limitations. Despite its effectiveness, GSRF has several limitations. First, the framework relies on GS-CoT annotations to initialize training. Unlike standard CoT, GS-CoT cannot be reliably induced through prompting alone, and constructing such data—especially with human verification—introduces additional cost. This limits the immediate applicability of GSRF to domains where structured reasoning annotations are available or affordable. Second, the two step-level rewards used in G-GRPO capture different but complementary aspects of reasoning structure. The inter-step dependency reward encourages coherent dependency chains, while the step-to-answer alignment reward promotes global relevance. Empirically, removing either reward leads to performance degradation, indicating that each alone is insufficient. This coupling makes reward design less modular and may complicate extension to more complex or heterogeneous reasoning tasks. Third, as with other reinforcement learning approaches, GSRF is susceptible to reward exploitation. Because the rewards are defined over graph structure rather than semantic validity, models may learn to satisfy structural criteria in superficial ways. Although this risk is mitigated through joint supervision and regularization, it remains an inherent challenge of process-level optimization. Finally, enforcing explicit structural constraints may restrict exploration, particularly for smaller models or tasks requiring long and flexible

reasoning chains. Our results suggest that this effect diminishes with increased model capacity, but it highlights a trade-off between structural guidance and exploratory freedom.

Overall, GSRF represents a step toward more structurally faithful and interpretable reasoning, while inheriting some limitations common to structured supervision and reinforcement learning. We view it as a complementary component that can be combined with semantic verification or external reasoning tools in future work.