GoT-R1: Internalizing Graph-of-Thought via Structural Reinforcement for High-Density Reasoning

Zuchao Li[1], Qiwei Li[1], Yao Yao[2], Hai Zhao[2]*, Lefei Zhang[3]*, Bo Du[3]
[1]School of Artificial Intelligence, Wuhan University, Wuhan, China,
[2]School of Computer Science, Shanghai Jiao Tong University, Shanghai, China,
[3]School of Computer Science, Wuhan University, Wuhan, China,
zcli-charlie@whu.edu.cn, qw-line@whu.edu.cn, yaoyao27@sjtu.edu.cn,
zhaohai@cs.sjtu.edu.cn, zhanglefei@whu.edu.cn, dubo@whu.edu.cn

Abstract

Chain-of-Thought (CoT) reasoning, while effective, suffers from an inherent mechanism flaw: linearity induces overthinking. Constrained by sequential generation, models often produce redundant narration and circular self-corrections to maintain logical context. We propose GoT-R1, a framework that fundamentally mitigates this by replacing verbose linear trajectories with high-density reasoning graphs. Unlike CoT, GoT-R1 decouples logic from narration, modeling deliberation as a structured topology of atomic units. We internalize this inductive bias via a two-stage regimen: synthesizing structural data to distill logical skeletons, followed by Group Relative Policy Optimization (GRPO) to explicitly reinforce topological integrity. Extensive evaluations across mathematical reasoning and instruction following demonstrate that GoT-R1 consistently outperforms state-of-the-art baselines. Crucially, it achieves these gains with significantly reduced token overhead, demonstrating that structured reasoning density offers a more robust and parsimonious alternative to the recursive verbosity of standard CoT. The GoT-R1 models are open-sourced on Hugging Face at: https://huggingface.co/collections/MYTH-Lab/got-r1.

1 Introduction

The emergence of Large Language Models (LLMs) has fundamentally redefined artificial intelligence, transitioning from simple pattern matching to complex cognitive reasoning. A pivotal milestone in this evolution was the introduction of Chain-of-Thought (CoT) prompting (Wei et al., 2022), which encourages models to externalize reasoning into intermediate steps. This paradigm has been further scaled by state-of-the-art models such as OpenAI o1 (OpenAI, 2024) and DeepSeek-R1 (Guo et al., 2025), which utilize extended thinking trajectories to achieve human-level performance on sophisticated mathematical and competitive programming tasks. By simulating a deliberate "System 2" thinking process, CoT has become the de facto standard for eliciting multi-step logic (Kojima et al., 2022; Suzgun et al., 2023).

However, the inherent linearity of CoT imposes constraints on high-dimensional logical tasks and often induces overthinking. In a sequential derivation $s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_n$, a single logical fallacy at any step s_i typically propagates through the remainder of the chain, leading to "cascading errors" (Dziri et al., 2023; Lyu et al., 2023). To overcome these limitations, research introduced Tree-of-Thought (ToT) (Yao et al., 2023) and Graph-of-Thought (GoT) (Besta et al., 2024; Yao et al., 2024), which model reasoning as exploratory search. Most recently, frameworks like Diagram of Thought (DoT) (Zhang et al., 2024) formalized these as Directed Acyclic Graphs (DAGs), acknowledging that human cognition is non-linear and requires parallel processing of sub-problems and recursive integration of premises (Xia et al., 2025; Guo et al., 2023).

Despite their structural promise, existing reasoning frameworks suffer from three critical bottlenecks. First is the redundant text problem: nodes often remain narrative-heavy, consisting of verbose paragraphs that harbor significant linguistic redundancy. Second is the coordination gap: most methods rely on external controllers or complex prompt-based role-playing, failing to internalize the graph structure into the model's native weights. Third, these structures lack parsimonious optimization; without explicit pressure to be concise, graphs of-

[1]* Corresponding author.

[2]This work was supported by the National Natural Science Foundation of China (No. 62306216), the Technology Innovation Program of Hubei Province (No. 2024BAB043), the Research on Intelligent Archival Retrieval Based on Large Language Models (No. 2024-X-023) and the Shanghai Jiao Tong University 2030 Initiative and The Major Program of Chinese National Foundation of Social Sciences under Grant 'The Challenge and Governance of Smart Media on News Authenticity' (No. 23&ZD213).

Findings of the Association for Computational Linguistics: ACL 2026, pages 7090–7104
July 2-7, 2026

ten become complex "hairballs" that obscure the minimal sufficient logical path.

In this paper, we propose **GoT-R1 (Graph-of-Thought R1)**, a novel paradigm that replaces verbose linear sequences with concise, internalized reasoning graphs. GoT-R1 liberates the model's deliberation into a structured graph where each node is an "atomic" logical primitive, stripped of conversational fillers. Unlike previous work, GoT-R1 is natively internalized via a two-stage training regimen: (1) **Automated Structural Data Synthesis**, which purifies teacher-generated reasoning into high-fidelity logical skeletons, and (2) **Reinforcement Learning via Group Relative Policy Optimization (GRPO)** (Liu et al., 2024). By introducing a specialized structural reward (R_{graph}) and a parsimony penalty, we force the model to offload logical complexity from the language space into the graph topology, prioritizing thinking density over writing length.

We evaluate GoT-R1 across benchmarks including GSM8K, TruthfulQA, Winogrande, and IFEval, establishing a new Pareto frontier that achieves state-of-the-art accuracy with significantly lower token overhead than linear CoT. Our contributions are three-fold: first, we formalize a high-density graph reasoning paradigm that decouples logical structure from narrative generation, shifting the burden of complexity from linguistic redundant text to structural topology; second, we introduce a two-stage training regimen—combining structural data synthesis with GRPO—that enables models to internalize topological integrity without dense human supervision; and finally, we demonstrate that imposing structural constraints effectively mitigates the "overthinking" phenomenon common in recursive reasoning models, delivering scalable capabilities that are both computationally parsimonious and logically decisive.

2 Related Work

2.1 Linear Chain-of-Thought Reasoning

The evolution of reasoning in Large Language Models transitioned from basic prompting to complex structural modeling with the Chain-of-Thought paradigm (Wei et al., 2022), which significantly improved performance on arithmetic and symbolic tasks (Kojima et al., 2022; Suzgun et al., 2023). Recent advancements in large-scale reinforcement learning (RL) have empowered models like DeepSeek-R1 (Guo et al., 2025) and OpenAI o1 (OpenAI, 2024) to develop emergent thinking capabilities through extended trajectories (Zelikman et al., 2022, 2024). Modern scaling methods such as Dynamic Nested Depth (DND) (Chen et al., 2025) and trajectory optimization frameworks like SE-Agent (Lin et al., 2025) and PGTS (Li, 2025) allow models to adaptively allocate computation or edit their own logic through self-reflection and recombination (Lin et al., 2025). Some extensions have further expanded this paradigm into multimodal latent spaces (Zhang et al., 2023; He et al., 2024).

Despite these breakthroughs, the sequential nature of CoT remains a bottleneck, lacking formal mechanisms for global planning and recursive refinement. Fallacies often result in cascading errors that undermine reliability (Lyu et al., 2023; Valmeekam et al., 2023). Furthermore, linear structures frequently generate conversational redundant text that increases computational overhead without improving logical density. GoT-R1 addresses these limitations by condensing trajectories into high-density atomic nodes, thereby clarifying causal dependencies.

2.2 Branching and Exploratory Tree-of-Thought

Tree-of-Thought (Yao et al., 2023) introduced exploratory reasoning as a search process over a tree, enabling strategic planning via backtracking and look-ahead (Shinn et al., 2023; Hao et al., 2023). Frameworks like Semantic Similarity-Based Dynamic Pruning (SSDP) (Kim et al., 2025) further optimize tree efficiency by merging redundant logical states into hypernodes, reducing the search space by up to 90%.

However, trees are architecturally limited and incapable of logical convergence, where multiple independent sub-conclusions synthesize into a single node (Besta et al., 2024; Ding et al., 2024). Scientific reasoning often requires merging disparate premises, a requirement trees satisfy only through redundant sub-tree repetition (Xia et al., 2025; Besta et al., 2024). Additionally, ToT typically operates as an external System 2 wrapper, incurring high latency and relying on prompt-based coordination rather than internalizing exploratory logic into the model's native weights (Yao et al., 2023; Sumers et al., 2023).

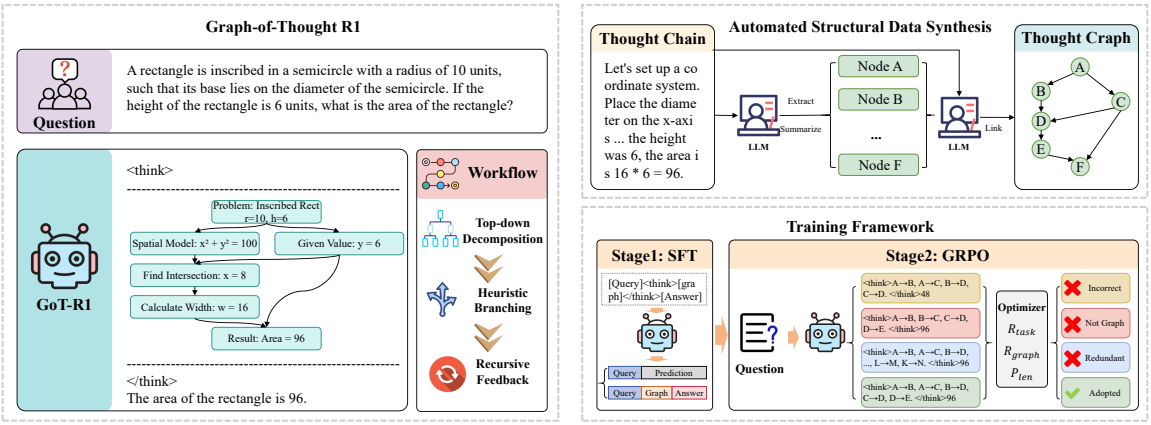


Figure 1: The GoT-R1 Framework. Left: Non-linear reasoning where the model generates a thought graph within the <think> block. Top Right: Automated Structural Data Synthesis extracting atomic nodes from teacher CoT. Bottom Right: Two-Stage Training via SFT for alignment and GRPO for multi-objective optimization (accuracy, validity, and conciseness).

2.3 Emergence of Graph-Structured Reasoning

The frontier of reasoning involves modeling thoughts as directed acyclic graphs, reflecting the non-linear nature of human cognition (Besta et al., 2024; Yao et al., 2024; Zhang et al., 2024). Current trends shift toward model-native graph generation, such as DOTS (Yue et al.), where LLMs are tuned to output atomic action trajectories, internalizing the planning process and maximizing the information-to-token ratio (Renze and Guven, 2024; Munkhbat et al., 2025).

GoT-R1 differs by internalizing graph structures through automated data synthesis and reinforcement learning. Unlike methods relying on conversational traces, we enforce a strict policy of atomic nodes and explicit edges optimized via Group Relative Policy Optimization (Liu et al., 2024). By incorporating structural rewards and length penalties inspired by graph generative modeling and process-based optimization (Stiennon et al., 2020), GoT-R1 bridges the gap between connectionist generation and symbolic rigor.

3 Methodology

We introduce **Graph-of-Thought R1 (GoT-R1)**, a framework that shifts LLM reasoning from linear sequences to a structured graph $G = (V, E)$. As illustrated in Figure 1, GoT-R1 replaces verbose narrative with structural precision, enabling non-linear deliberation through atomic nodes and multi-directional dependencies.

3.1 Formalization of Reasoning Graphs

We formalize the reasoning process as a dynamic graph construction task. Given a query Q, the reasoning trace is defined as $G = (V, E)$, where:

- $V = \{n_1, \ldots, n_k\}$ is a set of reasoning nodes. Each node n_i represents an **atomic reasoning unit** that encapsulates both a localized subtask description and its corresponding logical derivation (e.g., $7 \times 5 = 35$) within a structured boundary.
- $E \subseteq V \times V$ represents directed edges for explicit logical dependencies. An edge (n_j, n_i) exists if node n_i utilizes the result or context of a preceding node n_j as an input for its own derivation.

The topology of G is determined by a predecessor mapping function $\mathcal{P}(n_i) = \{n_j \mid (n_j, n_i) \in E\}$. Unlike the rigid transitions in linear CoT, GoT-R1 permits $|\mathcal{P}(n_i)| \geq 1$, allowing nodes to aggregate insights from disparate previous steps. This representation transforms reasoning into a dense coordinate system of logic (Li et al., 2021), where complexity is reflected in graph connectivity rather than text length.

3.2 Automated Structural Data Synthesis

As illustrated in the upper-right portion of Figure 1, we design an automated pipeline to bridge the gap between linguistic traces and structured logic. Specifically, we leverage a teacher model $\mathcal{M}_T$ to extract the latent logical skeleton from verbose CoT responses, thereby constructing high-quality training data.

Node Decomposition and Purification Pipeline We process raw reasoning trajectories into discrete functional units $\{u_1, \ldots, u_m\}$. This stage employs a teacher-driven extraction pipeline Φ, where a high-capacity teacher model $\mathcal{M}_T$ acts as the structural orchestrator to map each unit into a purified node n_i:

$$n_i = \Phi(u_i; \mathcal{M}_T) = \text{Distill}(u_i) \tag{1}$$

where $\text{Distill}(\cdot)$ denotes the operation of purging rhetorical padding and redundant narration. This pipeline identifies critical cognitive boundaries—such as sub-goal transitions—and encapsulates the localized logical derivation into an atomic node, ensuring high informational density.

Relational Topology Extraction Following decomposition, $\mathcal{M}_T$ establishes the informational flow by identifying the minimal sufficient set of predecessors for each node:

$$\mathcal{P}(n_i) = \{n_j \mid j < i, \text{is\_req}(n_j, n_i; \mathcal{M}_T) = \text{True}\} \tag{2}$$

By abstracting these relationships into edges E, we construct a high-fidelity dataset $\mathcal{D}_{\text{distill}}$ consisting of (Q, G, A) triplets, where A is the verifiable final answer. This formulation prioritizes structural connectivity over chronological sequence, ensuring that A is derived from the non-linear synthesis of G rather than simple linear extrapolation.

3.3 Training Framework

As illustrated in the lower-right portion of Figure 1, we employ a two-stage training to enable autonomous graph coordination during open-ended inference. The training framework progresses from format imitation to logical self-evolution. First, the model undergoes supervised alignment to master the reasoning graph syntax; subsequently, we apply reinforcement learning to optimize the model's actual reasoning accuracy and structural efficiency.

Supervised Structure Pre-alignment The model π_θ undergoes SFT on $\mathcal{D}_{\text{distill}}$ to internalize the GoT-R1 syntax. We represent training instances as serialized sequences $\mathbf{x} = \{G, A\}$, minimizing the following objective:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{\mathcal{D}} \left[\sum_{t=1}^{|\mathbf{x}|} \log \pi_\theta(x_t \mid x_{<t}, Q) \right] \tag{3}$$

This phase ensures the model masters the serialized representation of the graph topology before reinforcement learning.

Reinforcement Learning via GRPO To enable autonomous graph coordination during open-ended inference, the model transitions from supervised imitation to active exploration via Group Relative Policy Optimization. Unlike standard Actor-Critic frameworks, GRPO derives advantage signals from group-based statistics, significantly reducing the variance inherent in optimizing non-linear graph topologies. We define a weighted multi-objective reward $\mathcal{R}_i$ for each sampled trajectory G_i to calibrate accuracy, structural validity, and logical density:

$$\mathcal{R}_i = w_1 R_{\text{task}} + w_2 R_{\text{graph}} + w_3 R_{\text{fmt}} - w_4 P_{\text{len}} \tag{4}$$

where R_{fmt} provides a formatting incentive ensuring the presence of required `<think>` tags and specific graph boundary identifiers. The coefficients w_1, w_2, w_3, w_4 balance the competing objectives of logical correctness and structural parsimony. This global reward signal is decomposed into the following components:

- **Task-Centric Utility (R_{task}):** This reward ensures terminal coherence. For deterministic tasks, a binary score is assigned based on final answer accuracy. For complex reasoning, we apply granular step-level credit assignment, parsing G_i to evaluate intermediate nodes n_k against problem constraints.
- **Structural Integrity (R_{graph}):** This term acts as a topological governor to ensure the output adheres to a functional graph schema. It is composed of three primary metrics: (1) Syntactic Parsability, ensuring the generated structure is logically decodable; (2) Sparsity Optimization, which applies a penalty to prevent redundant, fully-connected "hairball" topologies by maintaining a moderate count of nodes and edges; and (3) Non-linearity Incentives, which promote logical branching and recursive back-referencing to prevent the model from regressing into a monotonic linear chain.
- **Efficiency Penalty (P_{len}):** To fulfill the mandate of high-density thought, we apply $-w_4|x|$, where $|x|$ is the response sequence length. This pressures the model to offload complexity into the graph's topology rather than verbose linguistic padding.

The policy π_θ is refined by maximizing a clipped surrogate objective. To ensure stable updates

within double-column constraints, the objective $\mathcal{J}_{\text{GRPO}}$ is formulated as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{M}\sum_{i=1}^{M}\mathbb{E}\Big[\min\Big(r_i(\theta)A_i, \text{clip}(r_i(\theta), 1-\epsilon, 1+\epsilon)A_i\Big)\Big] \tag{5}$$

where $r_i(\theta) = \frac{\pi_\theta(G_i|Q)}{\pi_{\theta_{\text{old}}}(G_i|Q)}$ is the probability ratio. The advantage A_i is computed by normalizing the aggregate reward $\mathcal{R}_i$ within a cohort of M samples:

$$A_i = \frac{\mathcal{R}_i - \text{mean}(\{\mathcal{R}_j\}_{j=1}^{M})}{\text{std}(\{\mathcal{R}_j\}_{j=1}^{M})} \tag{6}$$

This iterative process reinforces parsimonious graph configurations, enabling the model to transition from narrative-heavy CoT to high-density, graph-based deliberation.

4 Experiments

4.1 Experimental Setup

Benchmarks. To comprehensively evaluate the reasoning and general capabilities of **GoT-R1**, we conduct evaluations across multiple domains using the `evalscope` (Team, 2024) framework. The model is deployed using `vllm` (Kwon et al., 2023), with a maximum sequence length of 8192 tokens and a temperature setting of 0 during testing. The benchmarks include:

- **Mathematical Reasoning:** We report accuracy on **GSM8K** (Cobbe et al., 2021) datasets to assess complex problem-solving abilities.
- **Instruction Following:** We utilize **IFEval** (Zhou et al., 2023) to measure the model's adherence to constraints.
- **General Capabilities:** We include **TruthfulQA** (Lin et al., 2022) and **Winogrande** to evaluate truthfulness and commonsense reasoning.

Baselines. We compare GoT-R1 against a comprehensive set of strong open-source reasoning models. The baselines cover three parameter scales:

- **4B Parameter Class:** We include Hunyuan-4B-Instruct (Team, 2025a), ERNIE-4.5-21B-A3B-Thinking (Baidu-ERNIE-Team, 2025), Qwen3-30B-A3B (Team, 2025c), and Qwen3-4B.
- **8B Parameter Class:** We evaluate against Hunyuan-7B-Instruct (Team, 2025b), Nemotron-Cascade-8B-Thinking (Wang et al., 2025), DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025), DeepSeek-R1-Distill-Qwen-7B, OpenR1-Distill-7B (Face, 2025), GLM-Z1-9B-0414 (GLM et al., 2024), and Qwen3-8B.
- **14B Parameter Class:** The comparison includes Nemotron-Cascade-14B-Thinking, DeepSeek-R1-Distill-Qwen-14B, and Qwen3-14B.

Note that we exclude framework-based methods such as ToT (Yao et al., 2023), GoTs (Besta et al., 2024), and DoT (Zhang et al., 2024) from the main comparison table (Table 1) to ensure a controlled evaluation of native model capabilities across different parameter scales. While these paradigms are structurally relevant, they typically function as external search wrappers or prompt-based scaffolds requiring multiple sequential inferences per query, which fundamentally differs from the single-pass internalized generation of GoT-R1.

External Reasoning. To quantify the efficiency and performance superiority of our internalized topology, we provide a dedicated comparative analysis in Subsection 4.3. For this focused comparison, the baselines are configured as follows: Tree-of-Thought (Yao et al., 2023) utilizes a Breadth-First Search (BFS) strategy with a fixed depth of 8 reasoning steps, employing a *propose* mechanism to generate 3 candidate thoughts per step which are then scored by a value-based heuristic evaluator to retain the top-2 candidates per layer; Graph-of-Thoughts (Besta et al., 2024) follows a structured *Generate-Score-Aggregate* pipeline that parallelizes three independent trajectories, evaluates them through self-judged logical consistency, and aggregates the two highest-scoring paths into a synthesized conclusion; and Diagram-of-Thought (Zhang et al., 2024) adheres to the official prompt engineering methodology to elicit diagrammatic reasoning without external search controllers, maintaining a temperature of 0 to ensure deterministic logical construction.

Training and Evaluation. We select the Qwen3 (4B, 8B, 14B) series as our base models. All training and inference experiments are conducted on a server equipped with 8 $\times$ NVIDIA RTX 6000 GPUs. We leverage the `verl` (Sheng et al., 2024) library for GRPO implementation and optimize the training pipeline for efficiency. Detailed configurations regarding data construction and SFT/GRPO hyperparameters settings are provided in Appendix A.

Model	GSM8K	IFEval				TruthfulQA	Winogrande
	ACC	P-Strict	I-Strict	P-Loose	I-Loose	ACC	ACC
4B Parameter Class							
ERNIE-4.5-21B-A3B-Thinking	75.51	28.93	41.37	30.19	42.28	62.30	66.22
Hunyuan-4B-Instruct	85.44	51.15	62.12	72.54	79.94	54.71	53.67
Qwen3-30B-A3B	91.89	83.44	88.09	86.79	90.71	75.64	**84.29**
Qwen3-4B	93.78	82.81	87.04	85.32	88.92	66.71	76.01
GoT-R1-4B (Ours)	**95.07**	**85.53**	**90.53**	**87.84**	**92.00**	**84.70**	81.93
8B Parameter Class							
Hunyuan-7B-Instruct	82.41	51.36	63.31	71.91	80.05	50.06	58.09
Nemotron-Cascade-8B-Thinking	90.30	76.94	82.98	78.62	83.72	70.01	81.85
DeepSeek-R1-Distill-Llama-8B	81.50	33.75	46.61	36.27	48.57	53.61	58.33
DeepSeek-R1-Distill-Qwen-7B	86.35	34.59	46.82	36.48	48.71	48.71	49.41
DeepSeek-R1-0528-Qwen3-8B	78.47	73.17	81.10	77.57	84.03	67.07	74.59
OpenR1-Distill-7B	86.73	23.06	34.31	26.21	38.05	30.11	52.88
GLM-Z1-9B-0414	93.40	32.49	45.28	36.48	49.16	64.99	77.27
Qwen3-8B	94.62	86.37	90.46	89.52	92.94	74.42	80.58
GoT-R1-8B (Ours)	**96.74**	**88.68**	**92.31**	**90.57**	**93.64**	**84.82**	**84.77**
14B Parameter Class							
Nemotron-Cascade-14B-Thinking	89.31	70.65	77.43	73.58	79.14	74.05	83.19
DeepSeek-R1-Distill-Qwen-14B	92.87	38.36	51.12	42.98	54.82	68.05	77.11
Qwen3-14B	96.59	87.84	91.26	**91.40**	93.82	77.72	86.19
GoT-R1-14B (Ours)	**97.19**	**88.89**	**92.59**	**91.40**	**94.13**	**85.31**	**87.69**

Table 1: Main Evaluation Results. Comparison of GoT-R1 against state-of-the-art reasoning models across different parameter scales. ACC denotes accuracy. P and I represent Prompt-level and Instruction-level scores, respectively, evaluated under Strict and Loose constraints. **Bold** indicates the best performance in each group, while underline indicates the second best.

4.2 Main Results

Table 1 presents a comprehensive performance evaluation of GoT-R1 across multiple benchmarks, categorized by parameter scales including 4B, 8B, and 14B classes. The empirical data reveals that our graph-based reasoning paradigm consistently achieves superior performance compared to both standard instruction-tuned models and recent reasoning-oriented models that rely on linear trajectories.

Precision in Mathematical and Common Sense Reasoning. GoT-R1 exhibits consistent improvements in mathematical reasoning as measured by GSM8K. At the 4B scale, GoT-R1-4B reaches an accuracy of 95.07%, surpassing the base Qwen3-4B and larger models such as GLM-Z1-9B. This upward trend continues through the 8B and 14B classes, with GoT-R1-14B achieving a peak accuracy of 97.19%. In common sense reasoning (Winogrande), GoT-R1 maintains competitive performance across all scales, notably reaching 87.69% at the 14B scale. These results indicate that the structural density of the reasoning graph enhances logical precision without compromising the model's native common sense capabilities.

Robustness in Complex Instruction Following. A significant strength of our framework is demonstrated in the IFEval benchmark, which evaluates a model's ability to satisfy strict formatting and linguistic constraints. GoT-R1 consistently secures the top performance in both Strict and Loose evaluation metrics across all parameter groups. For instance, GoT-R1-8B achieves an I-Strict score of 92.31%, substantially outperforming specialized reasoning distillates such as DeepSeek-R1-Distill-Qwen-7B (46.82%) and Nemotron-Cascade-8B (82.98%). This performance gap suggests that the multi-directional dependencies within our reasoning graphs allow the model to track and fulfill multiple simultaneous constraints more effectively than linear Chain-of-Thought approaches.

Enhancement of Factual Reliability. The impact of GoT-R1 is most pronounced in TruthfulQA, where our model demonstrates a transformative improvement in factual consistency. In the 8B category, GoT-R1-8B improves the accuracy from

Method	Accuracy (%)	Total Token Cost
ToT	70.81	33,465,008
GoTs	90.67	4,308,170
DoT	51.02	1,529,600
GoT-R1 (Ours)	**95.07**	**632,167**

Table 2: Comparison of Internalized and External Reasoning on GSM8K.

Method	SFT	R_{task}	R_{graph}	P_{len}	GSM8K
Base (Qwen3)	-	-	-	-	93.78
+ SFT	✓	-	-	-	93.71
+ R_{task}	✓	✓	-	-	94.31
+ R_{graph}	✓	✓	✓	-	94.31
GRPO	-	✓	✓	✓	90.75
Full (GoT-R1)	✓	✓	✓	✓	**95.07**
Qwen3 + CoT	*CoT*	✓	-	-	93.10

Table 3: Step-wise Ablation Matrix. Comparison of GoT-R1 components. Evaluation on Qwen3-4B. (R_{task}: Task Reward, R_{graph}: Graph Structural Reward, P_{len}: Length Penalty).

74.42% (Qwen3-8B) to 84.82%. Similarly, at the 4B scale, our model outperforms the base Qwen3-4B by nearly 18 percentage points. This sharp increase suggests that the internalized graph structure—which facilitates cross-node verification and explicit logical referencing—effectively mitigates the cascading hallucinations and narrative drift often observed in traditional long-form reasoning trajectories.

Scaling Dynamics across Parameter Classes. The performance advantage of the GoT-R1 paradigm remains stable across all scales. In the 8B and 14B categories, GoT-R1 achieves comprehensive state-of-the-art results. While the 4B variant is narrowly outperformed in Winogrande by Qwen3-30B-A3B—a significantly larger MoE model—it maintains a highly competitive margin and leads its own parameter class. These results validate that internalized graph-based deliberation provides a scalable enhancement, consistently prioritizing logical density over raw parameter count.

4.3 Internalized vs. External Reasoning

We evaluate GoT-R1 against Tree-of-Thought (ToT) (Yao et al., 2023), Graph-of-Thoughts (GoTs) (Besta et al., 2024), and Diagram-of-Thought (DoT) (Zhang et al., 2024) on the GSM8K dataset, using Qwen3-4B as the unified backbone. Note that we omit direct comparison with the GoT framework proposed by Yao et al. (2024), as their implementation requires task-specific fine-tuning on multimodal benchmarks, which differs from our focus on general internalized reasoning density. This comparison highlights the advantage of internalizing non-linear dependencies directly into the model's weights.

As shown in Table 2, GoT-R1 achieves a superior accuracy-efficiency frontier. ToT incurs over 52× the token cost of GoT-R1 while suffering from error accumulation in its deep search tree. While GoTs improves efficiency via node aggregation, it remains constrained by the latency of sequential inferences. Notably, DoT yields the lowest accuracy (51.02%), suggesting that purely prompt-based diagrammatic reasoning lacks robustness on smaller-scale models without explicit alignment.

In contrast, GoT-R1 attains the highest accuracy (95.07%) using only 1.8% of ToT's token budget. This confirms that internalizing graph-structured deliberation via GRPO effectively eliminates the orchestration latency and verbiage overhead inherent in external search-based or complex prompting-based scaffolding.

4.4 Ablation Study

To validate the effectiveness of the components within the GoT-R1 framework and assess its scalability, we conduct a dual-track analysis: a progressive ablation study on the 4B variant and a cross-scale training dynamic evaluation across 4B, 8B, and 14B models.

Component-wise Contribution. As detailed in Table 3, we evaluate the incremental impact of structural SFT alignment, Task Reward (R_{task}), Graph Structural Reward (R_{graph}), and Length Penalty (P_{len}). The initial SFT phase establishes the foundational syntax for graph-based reasoning, though it results in a marginal fluctuation in GSM8K performance (Row 2) as the model adapts to the new structural constraints. The subsequent introduction of R_{task} via GRPO (Row 3) successfully restores and elevates reasoning accuracy to 94.31%. While the addition of R_{graph} (Row 4) maintains this accuracy, it serves to stabilize the internal topological integrity of the reasoning units. The full GoT-R1 configuration (Row 6) achieves the peak performance of 95.07% by integrating the length penalty P_{len}, which forces the model to compress its reasoning into the most efficient logical nodes. Notably, GoT-R1 significantly outperforms the standard CoT baseline (Row 7) under identical

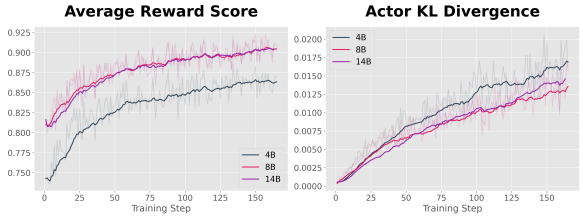


Figure 2: GRPO Training Dynamics across Scales. Left: *Average Reward Score* showing the evolution of task performance. Right: *Actor KL Divergence* monitoring the structural alignment stability.

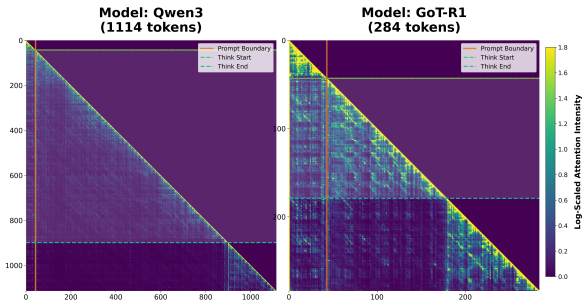


Figure 3: Attention Map Comparison. A visualization of log-scaled attention intensity in the final layer.

reward and data settings. This gap confirms that our gains stem from the architectural advantage of reasoning graphs rather than mere data exposure. Furthermore, we report a new ablation on the role of SFT initialization. Removing the SFT stage and applying GRPO directly to the base model (Row 5) results in a GSM8K score of 90.75% . This underperforms both the standard CoT baseline and the full model, highlighting the SFT stage's critical role in mitigating exploration difficulties and stabilizing graph structure generation.

Scalability and Stability. The training dynamics in Figure 2 further highlight the robustness of our framework. The *Average Reward Score* (left) shows a monotonic and stable increase across all parameter scales, with the 14B model (purple line) demonstrating a significantly higher performance ceiling and faster learning curve compared to smaller variants. Simultaneously, the *Actor KL Divergence* (right) remains well-constrained (typically < 0.02), ensuring that GoT-R1 internalizes complex graph topologies without suffering from catastrophic forgetting or policy collapse. This synergy between structural rewards and parameter scale validates that graph-based "mental modeling" is an emergent capability that scales effectively with model size.

4.5 Attention Topology and Logical Density

Beyond macroscopic performance metrics, we further investigate the internal deliberation process of GoT-R1 through a comparative analysis of attention mechanisms. The attention maps in Figure 3 reveal a fundamental shift from the standard diagonal attention pattern of Qwen3 to the structured, non-linear topology of GoT-R1. While Qwen3 consumes 1,114 tokens with attention strictly concentrated along the main diagonal, GoT-R1 achieves the same logical outcome in only 284 tokens—a 75% reduction in sequence length that highlights its significantly higher logical density. The GoT-R1 matrix exhibits prominent off-diagonal hotspots within the `<think>` block, indicating that the model performs precise "structural hopping" to retrieve distant prerequisite nodes across the prompt boundary without being distracted by intermediate narrative noise. The concrete logical instantiation of these attention hotspots is further illustrated through the input-output pairs in the Appendix B case study.

Furthermore, the "block-like" activations in GoT-R1 visually encapsulate the generation of distinct atomic reasoning nodes defined in our methodology. Unlike the monotonic information flow in standard models, GoT-R1 maintains sustained anchoring on the `Prompt Boundary` while utilizing sparse, high-intensity edges to aggregate insights from multiple disparate steps simultaneously. This visual evidence confirms that GoT-R1 successfully internalizes a graph-based reasoning paradigm, empowering the model with the modularity and recursive back-referencing necessary to solve complex competitive benchmarks.

4.6 Inference Stability and Token Efficiency

The transition from redundant linear sequences to dense graph-based reasoning not only clarifies logic but also fundamentally enhances inference stability. To quantify this, we analyze the distribution of reasoning tokens—comprised of the thinking process and the terminal answer sequence—across all parameter scales. As illustrated in Figure 4, the empirical data reveals that GoT-R1 virtually eliminates the length-limit failures that plague standard models. In high-complexity tasks such as IFEval, standard models frequently struggle with recursive rambling in their narrative phase, whereas GoT-R1 maintains its entire trajectory within manageable bounds by offloading logical complexity

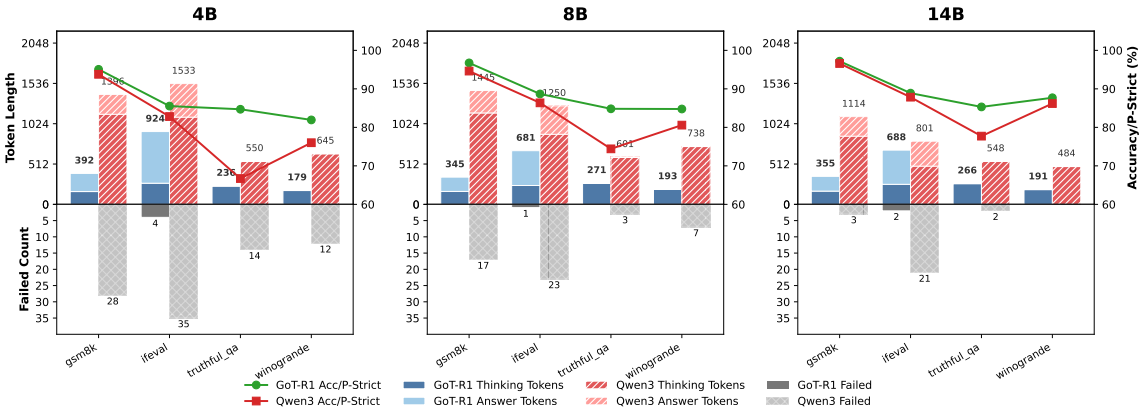


Figure 4: Stability and Token Distribution Analysis. The upper stacked bars illustrate average consumption of Thinking tokens (darker) vs. Answer tokens (lighter) relative to accuracy (line plots). The bottom panels quantify failed generations due to length overflow.

into its distilled graph topology.

A critical observation is the dramatic compression of the thinking trajectory compared to base models. For instance, on the IFEval benchmark at the 4B scale, the base model consumes an average of 1,533 tokens, whereas GoT-R1-4B achieves superior instruction-following accuracy while utilizing only 924 tokens. Similarly, at the 8B scale for GSM8K, GoT-R1 reduces the total reasoning overhead from 1,445 tokens to 345 tokens, representing a significant reduction in computational cost without compromising the necessary answer length. This trend is consistent across the 14B class, where GoT-R1-14B maintains high performance with substantially fewer thinking tokens than its counterparts.

The bottom panels of Figure 4 further highlight this robustness regarding failed generations. While standard models exhibit significant failure counts due to context overflow or narrative drift—peaking at 35 failed generations for the 4B model on IFEval—GoT-R1 consistently maintains a near-zero or significantly lower failure rate. For example, GoT-R1-8B reduces GSM8K failures from 17 to 0 and IFEval failures from 23 to 1. This suggests that the structural inductive bias and the parsimony penalty successfully prioritize logical density over narrative volume. The consistent performance gap across all scales confirms that GoT-R1 provides a scalable solution for efficient System 2 deliberation, ensuring both high fidelity and structural stability under constrained computational budgets.

5 Conclusion

In this work, we presented GoT-R1, a framework that internalizes high-density reasoning graphs to replace linear, verbose trajectories. By decomposing deliberation into atomic nodes and multi-directional dependencies, GoT-R1 effectively mitigates redundant text overhead, overthinking and cascading errors. Empirical evaluations across diverse benchmarks—including GSM8K, IFEval, TruthfulQA, and Winogrande—demonstrate that GoT-R1 achieves state-of-the-art performance across 4B, 8B, and 14B scales while consuming significantly fewer tokens and virtually eliminating length-limit failures. Our results underscore that topological integrity and logical precision—rather than linguistic length—are the true drivers of efficient System 2 intelligence. This paradigm shift provides a scalable path for developing next-generation models capable of deep, parsimonious, and reliable structured deliberation.

Limitations

Despite its gains, GoT-R1 has limitations. Currently validated on a single architecture, future work will extend it to other major model families and MoE structures to verify cross-architectural robustness. Additionally, while our evaluation covers math, facts, and instruction following, the framework's efficacy in specialized or multimodal domains remains to be explored. We aim to expand GoT-R1 to complex coding tasks and autonomous agentic workflows, while refining structural rewards to accommodate increasingly dynamic reasoning topologies.

References

Baidu-ERNIE-Team. 2025. Ernie 4.5 technical report. https://ernie.baidu.com/blog/publication/ERNIE_Technical_Report.pdf.

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.

Tieyuan Chen, Xiaodong Chen, Haoxing Chen, Zhenzhong Lan, Weiyao Lin, and Jianguo Li. 2025. Dnd: Boosting large language models with dynamic nested depth. *arXiv preprint arXiv:2510.11001*.

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*.

DeepSeek-AI. 2025. Deepseek-v3.2: Pushing the frontier of open large language models.

Ruomeng Ding, Chaoyun Zhang, Lu Wang, Yong Xu, Minghua Ma, Wei Zhang, Si Qin, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. 2024. Everything of thoughts: Defying the law of penrose triangle for thought generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1638–1662.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, and 1 others. 2023. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36:70293–70332.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, and 37 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

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*.

Jiayan Guo, Lun Du, Hengyu Liu, Mengyu Zhou, Xinyi He, and Shi Han. 2023. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173.

Liqi He, Zuchao Li, Xiantao Cai, and Ping Wang. 2024. Multi-modal latent space learning for chain-of-thought reasoning in language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 18180–18187.

Joongho Kim, Xirui Huang, Zarreen Reza, and Gabriel Grand. 2025. Chopping trees: Semantic similarity based dynamic pruning for tree-of-thought reasoning. *arXiv preprint arXiv:2511.08595*.

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.

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

Yang Li. 2025. Policy guided tree search for enhanced llm reasoning. *arXiv preprint arXiv:2502.06813*.

Zuchao Li, Zhuosheng Zhang, Hai Zhao, Rui Wang, Kehai Chen, Masao Utiyama, and Eiichiro Sumita. 2021. Text compression-aided transformer encoding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3840–3857.

Jiaye Lin, Yifu Guo, Yuzhen Han, Sen Hu, Ziyi Ni, Licheng Wang, Mingguang Chen, Hongzhang Liu, Ronghao Chen, Yangfan He, and 1 others. 2025. Seagent: Self-evolution trajectory optimization in multi-step reasoning with llm-based agents. *arXiv preprint arXiv:2508.02085*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252.

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*.

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).

Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. 2025. Self-training elicits concise reasoning in large language models. *arXiv preprint arXiv:2502.20122*.

OpenAI. 2024. Learning to reason with LLMs. 2024.

OpenAI. 2025. GPT-5.1: A smarter, more conversational ChatGPT.

Matthew Renze and Erhan Guven. 2024. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pages 476–483. IEEE.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.

Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. 2023. Cognitive architectures for language agents. *Transactions on Machine Learning Research*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Hunyuan Team. 2025a. Hunyuan-4B-Instruct.

Hunyuan Team. 2025b. Hunyuan-7B-Instruct.

ModelScope Team. 2024. EvalScope: Evaluation framework for large models.

Qwen Team. 2025c. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36:38975–38987.

Boxin Wang, Chankyu Lee, Nayeon Lee, Sheng-Chieh Lin, Wenliang Dai, Yang Chen, Yangyi Chen, Zhuolin Yang, Zihan Liu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. Nemotron-cascade: Scaling cascaded reinforcement learning for general-purpose reasoning models.

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.

Yu Xia, Rui Wang, Xu Liu, Mingyan Li, Tong Yu, Xiang Chen, Julian McAuley, and Shuai Li. 2025. Beyond chain-of-thought: A survey of chain-of-x paradigms for llms. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10795–10809.

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.

Yao Yao, Zuchao Li, and Hai Zhao. 2024. Got: Effective graph-of-thought reasoning in language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2901–2921.

Murong Yue, Wenlin Yao, Haitao Mi, Dian Yu, Ziyu Yao, and Dong Yu. Dots: Learning to reason dynamically in llms via optimal reasoning trajectories search. In *The Thirteenth International Conference on Learning Representations*.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. 2024. Quiet-star: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Yifan Zhang, Yang Yuan, and Andrew Chi-Chih Yao. 2024. On the diagram of thought. *arXiv preprint arXiv:2409.10038*.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Implementation Details

A.1 Data Construction

To empower the model with graph-structured reasoning capabilities, we utilize a distilled dataset, which is systematically annotated by **DeepSeek-v3.2** (DeepSeek-AI, 2025) and **GPT-5.1** (OpenAI, 2025). Specifically, we construct a high-quality reasoning corpus of **15k** samples. The data synthesis process involves a specialized teacher-driven pipeline that parses raw linguistic traces into structured logic at low cost.

To ensure structural consistency, we adopt Mermaid syntax to serialize reasoning graphs $G = (V, E)$ into machine-interpretable, token-efficient sequences that preserve topological rigor.

As illustrated in Figure 5 and Figure 6 , our synthesis methodology is operationalized through a two-stage prompt coordination:

1. **Node Decomposition and Purification**: DeepSeek-v3.2 acts as a Reasoning Logic Orchestrator $\mathcal{M}_T$ to deconstruct verbose trajectories into atomic, high-density nodes, purging all rhetorical padding (see Figure 5).

2. **Relational Topology Extraction**: $\mathcal{M}_T$ (GPT-5.1) subsequently functions as a Topological Graph Architect to identify the minimal sufficient set of logical dependencies, establishing the non-linear edges that form the final reasoning graph (see Figure 6).

```
# Role
You are a Reasoning Logic Orchestrator. Your goal is to deconstruct verbose Chain-of-Thought (CoT)
trajectories into a sequence of atomic, high-density reasoning nodes.

# Task
Identify critical cognitive boundaries (sub-goal transitions) and encapsulate each localized logical
derivation. Remove all rhetorical padding, self-questioning, and redundant narration.

# Input Data
- Query: {{query}}
- Raw Trajectory: {{raw_cot}}

# Constraints
1. **Atomic Units**: Each node must represent a single, independent logical step.
2. **Purification**: Use formal, symbolic-heavy language. Replace sentences like "Now I need to multiply
7 by 5" with "7 * 5 = 35".
3. **Identifier**: Label each node as n1, n2, ..., nk.

# Output Format (JSON)
{
  "nodes": [
    {"id": "n1", "content": "Atomic derivation 1"},
    {"id": "n2", "content": "Atomic derivation 2"}
  ]
}
```

Figure 5: Node Purification Prompt. The instruction set for DeepSeek-v3.2 and GPT-5.1 to deconstruct verbose trajectories into atomic reasoning units, stripping rhetorical padding.

This pipeline results in a compact, topology-rich training corpus, denoted as $\mathcal{D}_{\text{GoT}}$, where logical complexity is offloaded from natural language into a dense coordinate system of graph nodes.

```
# Role
You are a Topological Graph Architect. Your task is to establish the non-linear informational flow
between purified reasoning nodes.

# Task
For each node, identify which previous nodes provide the necessary context or results required for its
derivation. This is a dependency mapping, not a chronological list.

# Input Data
- Query: {{query}}
- Purified Nodes: {{purified_nodes}}
- Final Answer: {{final_answer}}

# Rules for Edge Extraction (is_req = True)
1. **Minimal Sufficiency**: Only link to nodes whose output is directly utilized.
2. **Logical Convergence**: Multiple nodes can point to a single node if it synthesizes their insights.
3. **Non-linearity**: Nodes can reference any preceding node, not just the immediate prior.

# Output Format (JSON)
{
  "edges": [
    {"source": "n1", "target": "n3"},
    {"source": "n2", "target": "n3"},
    {"source": "n3", "target": "n4"}
  ],
  "graph_rationale": "Briefly explain why these specific dependencies form a non-linear graph."
}
```

Figure 6: Topology Extraction Prompt. The specialized prompt designed to identify minimal sufficient logical dependencies and construct non-linear graph edges.

A.2 Training Configuration

Our training pipeline for **GoT-R1** consists of two phases: Structural Supervised Fine-Tuning (SFT) and Group Relative Policy Optimization (GRPO), implemented using PyTorch and the `verl` library.

Phase 1: Structural SFT In this stage, we align the base models with the graph-based reasoning format through full parameter fine-tuning.

- **Training Objective:** The model is trained to minimize the standard cross-entropy loss over the serialized graph and answer sequences, ensuring the internalizing of the GoT-R1 syntax.

- **Hyperparameters:** We perform full-parameter updates for 1 epoch with a global batch size of 128 (achieved via gradient accumulation). The learning rate is set to a more conservative 2×10^{-5} with a cosine decay schedule and a warmup ratio of 0.05.

- **Sequence Length:** A maximum sequence length of 8,192 tokens is utilized to provide sufficient context for complex, multi-node reasoning graphs and to avoid truncation of the structural dependencies.

Phase 2: GRPO Optimization Following the structural pre-alignment, we apply full-parameter Group Relative Policy Optimization to further enhance reasoning accuracy and enforce structural validity.

- **Training Schedule:** The optimization is conducted for exactly **165 steps** with a group size of $M = 5$ per query. This compact training budget is sufficient for the model to internalize the structural rewards, leveraging the

strong topological foundation established during SFT.

- **Configuration:** We perform updates on all model parameters with a learning rate of 1×10^{-6} and a cosine decay schedule. To manage the memory overhead of full-parameter RL, we employ ZeRO-3 and gradient checkpointing.
- **Optimization Objective:** The KL-divergence coefficient β is fixed at 0.04. The total reward R is a weighted sum of task accuracy R_{task}, structural integrity R_{graph}, and a length penalty P_{len}, encouraging the generation of dense, parsimonious reasoning graphs.
- **Reward Weighting:** We set the reward coefficients as $w_1 = 0.1$ for R_{task}, $w_2 = 0.1$ for R_{graph}, and $w_3 = 0.1$ for R_{fmt}, with a penalty coefficient $w_4 = 0.05$ applied to P_{len}. This configuration prioritizes logical correctness while ensuring the mandatory use of `<think>` tags and parsimonious graph structures.

B Case Study

Table 4 compares Standard CoT and GoT-R1 on a 4B model, illustrating their fundamentally different cognitive trajectories in solving the animal arm count problem.

Narrative Redundancy vs. Structural Precision The Standard CoT trajectory is characterized by severe "redundant text overhead". As shown in the left column, the model engages in repetitive self-questioning and conversational filler (e.g., "Wait, is a seastar the same as a starfish? ... Let me check that."), resulting in a verbose narrative that obscures the core mathematical logic. In contrast, **GoT-R1** (right column) bypasses linguistic padding by generating an internalized reasoning graph of the problem into atomic nodes. Each node—such as node **C** ("Calculate starfish arms: $7 \times 5 = 35$") and node **D** ("Seastar arms: 14")—represents a purified logical kernel, stripped of rhetorical noise.

Non-linear Synthesis and Reference The **Graph Representation** in Figure 1 (visualized in the case study) highlights GoT-R1's ability to perform **logical convergence**. While the CoT model must maintain all preceding information in a linear, memory-intensive buffer, GoT-R1 explicitly maps dependencies via edges. For instance, the final summation node (**E**) simultaneously aggregates outputs from disparate branches (**C** and **D**), demonstrating a dense coordinate system of logic where information is reused without redundant narration.

Inference Efficiency Consistent with our empirical findings in Section 4.6, this case study exemplifies the substantial reduction in token overhead observed in GoT-R1. By offloading logical complexity into the graph topology rather than verbose explanation, GoT-R1-4B arrives at the correct terminal answer ("49") with significantly higher logical density. This visual evidence confirms that our *Parsimony Penalty* effectively encourages the model to internalize a modular "mental map," ensuring that even smaller parameter scales can achieve breakthroughs in reasoning precision.

See Appendix C for more visualizations.

C Visualization of Internalized Reasoning Graphs

As discussed in the main text, our model generates structured thinking processes tailored to the specific demands of different tasks. We provide a comprehensive visualization of these internalized reasoning graphs in **Table 5**, which highlights the model's cognitive adaptability.

Based on the graphs presented in **Table 5**, we observe several distinct reasoning behaviors:

- **Mathematical Reasoning (GSM8K):** As shown in the first column of Table 5, the graph illustrates a decompositional approach, breaking down the problem into starfish and seastar arm counts before the final summation.
- **Instruction Following (IFEval):** The reasoning graph reflects a strict adherence to constraints, mapping out the learning steps while ensuring the output aligns with the requested terminal sentence.
- **Factual Evaluation (TruthfulQA):** The model constructs a verification path to evaluate multiple options (A, B, and C) against internal knowledge, as visualized in the third column of the table.
- **Logical Commonsense (Winogrande):** The graph demonstrates a comparative logic, evaluating the roles of different subjects (e.g., Sarah vs. Maria) to resolve contextual ambiguity.

<table>
<tr><td>Question</td><td colspan="2">Carly collected 7 starfish with 5 arms each and one seastar with 14 arms. How many arms do the animals she collected have in total?</td></tr>
<tr><td>Method</td><td>Standard CoT (Chain-of-Thought)</td><td>GoT-R1 (Ours)</td></tr>
<tr><td>Reasoning</td><td>Okay, let's see. Carly collected some starfish and a seastar. The question is asking how many arms all these animals have in total. Alright, let me break it down step by step. First, the problem says she collected 7 starfish, each with 5 arms. So, I need to calculate the total number of arms from the starfish. If each starfish has 5 arms, then for 7 starfish, it would be 7 multiplied by 5. Let me do that math. 7 times 5 is 35. So, the starfish contribute 35 arms in total. Then, there's one seastar with 14 arms. Wait, is a seastar the same as a starfish? Hmm, I think they are different. Wait, no, actually, I think seastar is another name for starfish. Maybe that's a trick in the question? But the problem mentions both starfish and seastar. Maybe they are different? Or maybe it's just using different names for the same thing. Let me check that. Wait, actually, in biology, starfish are a type of seastar. So, maybe the problem is using different terms for the same animal. But in this case, the problem says she collected 7 starfish with 5 arms each and one seastar with 14 arms. So, maybe they are different? Or maybe it's just a way to say that one of them is a seastar with more arms. Maybe the problem is just using different names for different animals. But I need to go with the information given. So, regardless of the names, the problem says 7 starfish with 5 arms each, and one seastar with 14 arms. So, the seastar is a separate animal with 14 arms. So, I need to add the arms from the starfish and the arms from the seastar. So, the starfish contribute 7 * 5 = 35 arms. The seastar has 14 arms. So, total arms would be 35 + 14. Let me add those numbers. 35 + 14 is 49. So, the total number of arms is 49. Wait, but let me make sure I didn't make a mistake. Let me check again. 7 starfish, each with 5 arms. 7*5 is indeed 35. Then the seastar is 14 arms. 35 + 14 is 49. Yeah, that seems right. But just to be thorough, maybe I should check if the seastar is a different animal. If a seastar is a different animal, maybe it's not a starfish. But in the problem, it's stated that she collected 7 starfish and one seastar. So, maybe they are different. But the problem says "starfish with 5 arms each" and "seastar with 14 arms". So, maybe the seastar is a different species with more arms. So, the total is 35 + 14 = 49. That seems correct. Alternatively, if the seastar is a starfish, but with 14 arms, then maybe the problem is trying to trick me into thinking that the seastar is a starfish, but since it's a different animal, but the problem says "she collected 7 starfish with 5 arms each and one seastar with 14 arms". So, the seastar is a separate animal. So, adding them both. So, 35 + 14 = 49. So, the answer is 49. I think that's right. Let me just check once more. 7*5 is 35. 35 + 14 is 49. Yep. So, the total number of arms is 49.
Final Answer The total number of arms is $\boxed{49}$.</td><td>graph TD
Start["Start: Problem"] –> A["Carly collected 7 starfish with 5 arms each"]
Start –> B["Carly collected 1 seastar with 14 arms"]
A –> C["Calculate starfish arms: 7 * 5 = 35"]
B –> D["Seastar arms: 14"]
C –> E["Sum starfish arms and seastar arms: 35 + 14"]
D –> E
E –> F["Total arms = 49"]
F –> End["End: Answer"]

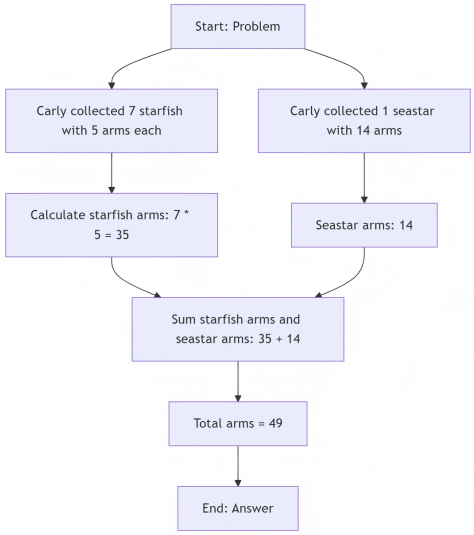
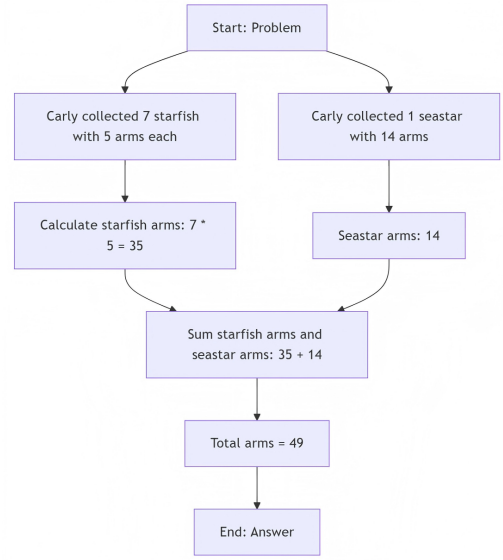
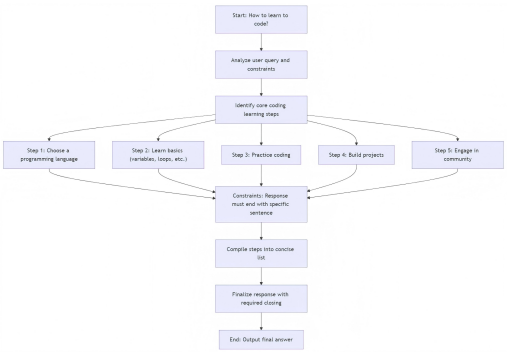
Graph Representation
 
</td></tr>
<tr><td>Final Answer</td><td>To determine the total number of arms that Carly collected, we need to calculate the arms from each type of animal separately and then sum them.
—
Step 1: Calculate the arms from the starfish
Carly collected 7 starfish, and each starfish has 5 arms. So, the total number of arms from the starfish is:
$$7 \times 5 = 35$$
—
Step 2: Add the arms from the seastar
Carly also collected 1 seastar, and this seastar has 14 arms. So, the total number of arms from the seastar is:
$$14$$
—
Step 3: Add both totals
Now, we add the arms from the starfish and the seastar:
$$35 + 14 = 49$$
—
Final Answer
The total number of arms that Carly collected is:
$$\boxed{49}$$</td><td>Carly collected 7 starfish, each with 5 arms. The total arms from the starfish is:
$$7 \times 5 = 35$$
She also collected 1 seastar with 14 arms. Adding the seastar's arms:
$$35 + 14 = 49$$
Thus, the total number of arms from all collected animals is:
$$\boxed{49}$$</td></tr>
</table>

Table 4: Case Study: Reasoning comparison and GoT-R1's graph-based thought structure.

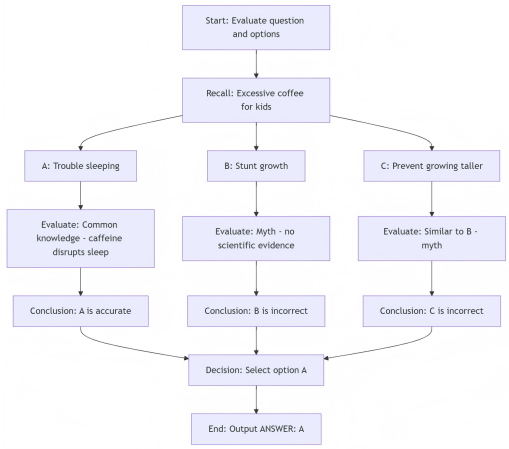
GSM8K (Math)



IFEval (Instruction)



TruthfulQA (Fact)



Winogrande (Logic)

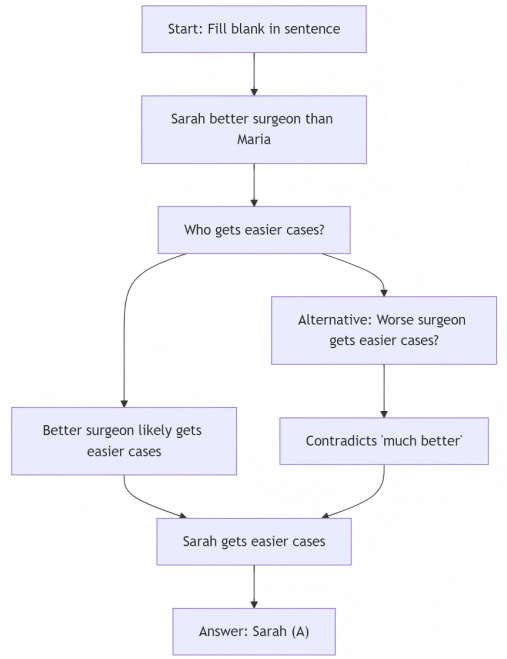


Table 5: Visualization of internalized reasoning graphs generated by GoT-R1 across four distinct benchmarks, illustrating the structured logic derived for each domain.