From Chains to Graphs: Self-Structured Reasoning for General-Domain LLMs

Anonymous ACL submission

Abstract

Large Language Models (LLMs) show strong reasoning ability in open-domain question answering, yet their reasoning processes are typically linear and often logically inconsistent. In contrast, real-world reasoning requires integrating multiple premises and solving subproblems in parallel. Existing methods, such as Chain-of-Thought (CoT), express reasoning in a linear textual form, which may appear coherent but frequently leads to inconsistent conclusions. Recent approaches rely on externally provided graphs and do not explore how LLMs can construct and use their own graph-structured reasoning, particularly in open-domain QA. To fill this gap, we novelly explore graph-structured reasoning of LLMs in general-domain question answering. We propose Self-Graph Reasoning (SGR), a framework that enables LLMs to explicitly represent their reasoning process as a structured graph before producing the final answer. We further construct a graph-structured reasoning dataset that merges multiple candidate reasoning graphs into refined graph structures for model training. Experiments on five QA benchmarks across both general and specialized domains show that SGR consistently improves reasoning consistency and yields a 17.74% gain over the base model. The LLaMA-3.3-70B model fine-tuned with SGR performs comparably to GPT-4o and surpasses Claude-3.5-Haiku, demonstrating the effectiveness of graph-structured reasoning.[1]

1 Introduction

Large Language Models (LLMs) (Hurst et al., 2024; Dubey et al., 2024) have exhibited impressive performance on a wide range of open-domain natural language understanding and question-solving tasks (Zhao et al., 2023; Bang et al., 2023; Yang et al., 2024b). In recent years, research on reasoning-oriented LLMs has progressed rapidly, showing that explicit intermediate reasoning can significantly enhance complex inference and provide more interpretable explanations of model decisions (Ke et al., 2025; Patil and Jadon, 2025). Representative examples include Chain-of-Thought (CoT) (Wei et al., 2022) and dedicated reasoning models (Jaech et al., 2024; Guo et al., 2025).

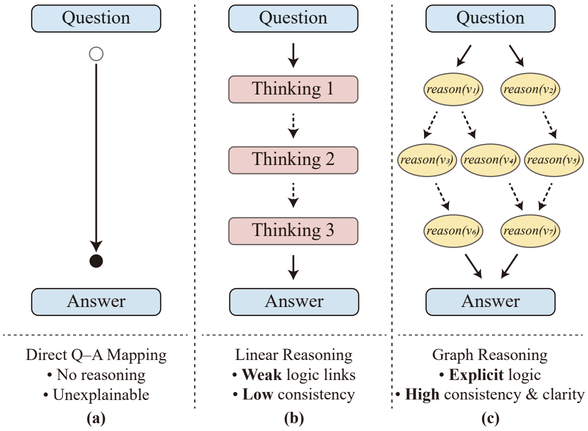


Figure 1: Comparison of reasoning and answering paradigms in general-domain question answering. (a) **Direct Answering** without explicit reasoning. (b) **Linear reasoning** with weak logical alignment between intermediate reasoning and the final answer. (c) Our **Graph-structured reasoning** with explicit logical connections, yielding higher reasoning consistency.

However, current approaches often exhibit inconsistencies between the generated reasoning process and the final answer (Wang et al., 2025; Arcuschin et al., 2025), especially in general-domain question answering (QA), where the inference lacks a clear and explicit logical structure (Xu et al., 2024; Lee and Lee, 2025). This makes it difficult to ensure that intermediate steps form a coherent reasoning path toward the final conclusion. Such challenges are less pronounced in specialized domains, such as mathematics and medicine, where reasoning typically follows more formalized and constrained structures. Early paradigms

---

[1] Our anonymous code is available at `https://anonymous.4open.science/r/SGR-F43C`.

in general-domain question answering focused on direct answer prediction without explicit reasoning (Figure 1, a). Recent advances, such as CoT, introduced intermediate reasoning steps to improve interpretability and performance. However, because the reasoning process in LLMs is represented as a linear textual sequence (Figure 1, b) optimized for next-token prediction rather than for structured logical inference, the reasoning chain may appear superficially coherent yet lead to incorrect answers, or, conversely, flawed reasoning may occasionally yield correct results (Turpin et al., 2023). Furthermore, unlike explicitly structured reasoning such as knowledge graphs (Chen et al., 2020; Tan et al., 2025), such linear reasoning makes it difficult for LLMs to maintain consistent dependencies between intermediate reasoning steps and the final answer (Patil and Jadon, 2025). These limitations are particularly evident in smaller-scale LLMs.

To overcome these limitations, we posit that reasoning should extend beyond simple linear sequences. Real-world reasoning often involves parallel sub-problems and the integration of multiple premises, which cannot be adequately represented by a single linear chain. As illustrated in Figure 1 (c), graph-structured reasoning enables many-to-one dependencies, allowing multiple independent inferences to be explicitly integrated into a unified conclusion (Yao et al., 2024). By enforcing explicit parent–child dependencies between reasoning steps, such a structure tightly couples the reasoning process with the final answer, ensuring that each conclusion is grounded in its supporting premises. However, existing graph-based reasoning methods primarily incorporate externally constructed graph structures to guide inference (Jin et al., 2024; Luo et al., 2024; Han et al., 2025). As a result, whether LLMs can perform explicit self-graph reasoning, particularly in the general-domain, remains largely unexplored.

Building on this insight, we propose *self-graph reasoning* (SGR), a paradigm that compels LLMs to externalize their latent reasoning into an explicitly structured graph before producing a final answer, and we use that as intermediate hints for the model. In SGR, nodes represent reasoning units, while edges encode explicit logical dependencies, forming a structured bridge between the input question and the final prediction. Unlike prior approaches that either rely on linear text-based reasoning or external graph structures, SGR enables LLMs to internalize graph construction as part of the reasoning process itself. Furthermore, we construct a dataset to train models for self-graph reasoning. Given a question, the LLM generates multiple candidate reasoning graphs, which are then aggregated and refined into an optimal reasoning graph, which serves as supervision for training self-graph reasoning. We conduct experiments across five QA benchmarks, covering both general-domain and specialized-domain tasks, demonstrating the effectiveness of our proposed framework. Moreover, we publicly release our constructed graph-reasoning dataset to support future research on structured and interpretable reasoning in LLMs.

In summary, our contributions are: (1) **Self-Graph Reasoning Method.** We introduce a novel reasoning paradigm that enables LLMs to perform structured graph reasoning within the inference process itself, enhancing the transparency and consistency between intermediate reasoning and final answer. (2) **Graph-Reasoning Dataset.** We construct a general-purpose graph-reasoning dataset of 10K instances that provides explicit structured supervision, enhancing LLMs' capability in graph-based reasoning. (3) **Empirical Effectiveness.** We demonstrate the effectiveness of SGR, with a 17.74% improvement over its base model across five benchmarks while showing consistent effectiveness in specialized domains, including mathematics and medicine.

2 Related Works

Chain-of-Thought (CoT) Reasoning. Early language models mainly performed direct answer prediction, where reasoning was implicit (Brown et al., 2020; Chowdhery et al., 2023; Li et al., 2025). To address the limited capability in reasoning over complex problems and the lack of interpretability, the Chain-of-Thought (CoT) paradigm was proposed (Wei et al., 2022; Kojima et al., 2022). CoT encourages models to explicitly generate an intermediate reasoning process before producing the final answer, thereby improving performance on tasks requiring complex logical reasoning (Zhou et al., 2022; Zhang et al., 2022). However, despite its effectiveness, CoT reasoning remains essentially linear, modeling reasoning as a single textual sequence optimized for next-token prediction, which can lead to plausible but incorrect reasoning chains (Turpin et al., 2023; Lanham et al., 2023). In contrast, our self-graph reasoning represents internal reasoning as a structured topology, where each claim is grounded in its ancestral arguments, resulting in clearer and more coherent reasoning.

Reasoning LLMs. Recent studies have shifted toward reasoning LLMs (Patil and Jadon, 2025), which primarily enhance multi-step reasoning through reasoning-oriented training, encouraging models to generate or internally perform reasoning before producing final answers, often via specialized training objectives. Representative models include OpenAI o1 (Jaech et al., 2024), DeepSeek-R1 (Guo et al., 2025), and Qwen3-thinking (Yang et al., 2025). Despite their remarkable performance, these models still rely on a fundamentally linear reasoning process, similar to CoT. As a result, their reasoning remains limited in capturing complex logical dependencies and maintaining consistency between reasoning and final answers (Turpin et al., 2023), particularly in general-domain QA tasks with intricate logical structures.

Graph-based Reasoning. To address the limitations of linear reasoning, recent studies have explored incorporating graph structures into the reasoning process (Luo et al., 2023; Sun et al., 2023; Jin et al., 2024; Tian et al., 2024; Cao, 2024). Specifically, Reasoning on Graphs (Luo et al., 2023) generates a reasoning graph to capture logical relations and use it to generate the final answer. MindMap (Wen et al., 2024) retrieves external evidence graphs and performs reasoning based on these graphs. However, these approaches typically use pre-extracted logical graphs or retrieved knowledge graphs, requiring external structures to support reasoning. In contrast, our work is the first to explore self-graph reasoning, where a reasoning LLM autonomously externalizes its internal reasoning process into a structured topological graph before producing the final answer, particularly in general-domain settings.

3 Methods

In this section, we introduce our proposed Self-Graph Reasoning (SGR), a framework designed to enable LLMs to externalize their latent reasoning process into a structured graph before generating the final answer. By explicitly organizing intermediate reasoning into a graph structure, SGR provides clearer logical constraints and reduces the drift and inconsistency often seen in linear Chain-of-Thought reasoning. The implementation of SGR consists of two main components, as illustrated in Figure 2: (1) Reasoning Graph Construction, which constructs a dataset of explicit reasoning graphs capturing the logical process between questions and answers, serving as supervision for model training; and (2) Self-Graph Reasoning Framework, where the model is fine-tuned on the constructed graph data to internalize the ability to perform self-graph reasoning before generating the final answer.

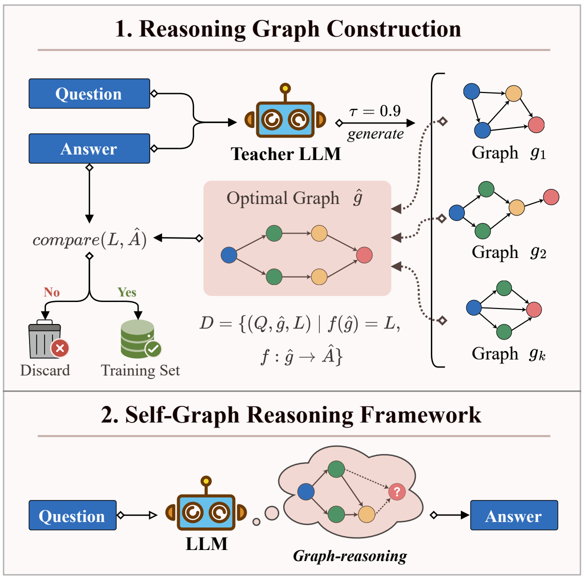
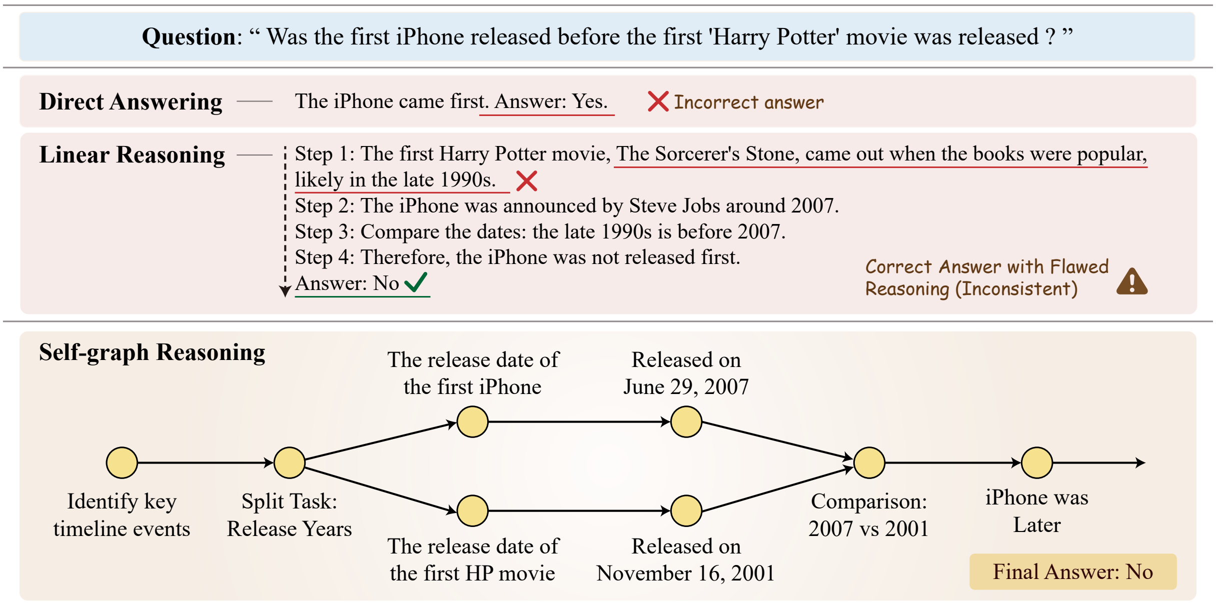


Figure 2: Overview of the proposed Self-Graph Reasoning (SGR).

3.1 Reasoning Graph Construction

To provide the model with high-quality graph reasoning supervision, we transform raw question-answer pairs into structured reasoning graphs.

Diverse Trajectory Exploration. To capture a diverse set of reasoning trajectories specific to the general-domain question Q, we prompt a teacher LLM (GPT-4o) to explore multiple reasoning paths. General-domain questions often admit multiple reasoning paths, as they can be solved from different starting points or perspectives, each leading to a distinct sequence of logical steps and a valid answer. By setting a higher temperature ($\tau = 0.9$), we encourage stochasticity and diversity in the reasoning process. Formally, for each Q, we sample k independent reasoning trajectories, which are represented as a set of candidate graphs $\mathcal{S} = g_1, g_2, \ldots, g_k$. Each $g_i = (V_i, E_i)$ represents a potential logical reasoning path leading to an answer, where nodes V denote atomic reasoning steps and edges E represent logical dependencies. These candidate graphs are then aggregated and refined to construct high-quality training data for supervising the LLM in learning structured self-graph


Question: " Was the first iPhone released before the first 'Harry Potter' movie was released ? "

Direct Answering — The iPhone came first. Answer: Yes. ✗ Incorrect answer

Linear Reasoning — Step 1: The first Harry Potter movie, The Sorcerer's Stone, came out when the books were popular, likely in the late 1990s. ✗
Step 2: The iPhone was announced by Steve Jobs around 2007.
Step 3: Compare the dates: the late 1990s is before 2007.
Step 4: Therefore, the iPhone was not released first.
Answer: No ✓

Correct Answer with Flawed Reasoning (Inconsistent) ⚠

Self-graph Reasoning

Figure 3: An illustrative example of the Self-Graph Reasoning framework. Our method constructs a structured reasoning graph where each node is explicitly grounded in its predecessors to ensure a logically consistent and clear path to the final answer. For comparison, we also illustrate other paradigms: Direct Answering lacks an explicit thinking process and is prone to errors, while Linear Reasoning often exhibits process-answer inconsistency (e.g., reaching a correct answer through flawed factual steps).

reasoning.

Graph Integration and Data Cleaning. Individual reasoning paths often contain fragmented or suboptimal logic reasoning path. To construct an optimal reasoning trajectory, we integrate all candidate graphs $\mathcal{S}$ into a unified, optimal reasoning graph $\hat{g}$, which we define as the integrated graph that best preserves logical consistency and convergence toward the correct answer. The resulting graph is linearized into a structured template as follows:

`<reasoning>`
`<step>` $v_i \to v_j$ `</step>`
...
`</reasoning>`
`<answer>` Final Answer `</answer>`

Furthermore, to ensure the integrity of the training data, we evaluate the final answer $\hat{A}$ derived from the synthesized graph $\hat{g}$ against the known correct answer L for question Q. The graph $\hat{g}$ is retained in our final training set $\mathcal{D}$ if and only if the reasoning process leads to the correct answer:

$$\mathcal{D} = \left\{ (Q, \hat{g}, L) \mid f(\hat{g}) = L,\ f : \hat{g} \to \hat{A} \right\}. \quad (1)$$

By discarding instances where the synthesized reasoning fails to reach the correct conclusion, we ensure that SGR is trained exclusively on logically consistent and factually grounded reasoning-answer pairs.

3.2 Self-Graph Reasoning Framework

The core of SGR is to transform the LLM's latent cognitive process into an explicit, verifiable graph topology before reaching a conclusion.

Supervised Graph Learning. To empower the LLM with structured self-reasoning capability, we perform Supervised Fine-Tuning (SFT) with LoRA (Hu et al., 2022) on the training data $\mathcal{D}$ constructed in Section 3.1. Each training instance is a triplet $(Q, \hat{g}, L)$, where the graph $\hat{g}$ serves as the structural intermediate reasoning between the question Q and the ground truth L. Formally, we optimize the model parameters θ by minimizing the standard cross-entropy loss over the reasoning graph and the final answer:

$$\mathcal{L}(\theta) = - \sum_{(Q,\hat{g},L) \in \mathcal{D}} \log P_\theta(\hat{g}, L \mid Q). \quad (2)$$

By generating $\hat{g}$ token-by-token, the model effectively constructs a structural reasoning graph that constrains the final answer L to be a direct consequence of verified antecedent steps. Compared to *Direct Answering*, which collapses the reasoning space into a single mapping, and *Linear Reasoning* (e.g., CoT), which is restricted to a single-path dependency, SGR ensures logical consistency between the reasoning process and the final answer. Each node v_j must be explicitly justified by its parent nodes $Pa(v_j)$, thereby eliminating the "logical drift" often observed in linear reasoning (Lanham et al., 2023). By externalizing internal thoughts into V and E, SGR provides a transparent and structured reasoning process.

Inference Stage. At inference time, given a question Q, the model generates a structured reasoning graph $\hat{g}$ followed by the final answer L, according to the learned conditional distribution $P_\theta(\hat{g}, L \mid Q)$. An illustrative example is shown in Figure 3. Unlike Direct Answering, which directly outputs an answer without any intermediate reasoning, or Linear Reasoning, which may produce a correct answer but with inconsistent or hallucinatory reasoning (e.g., wrongly stating that *The Sorcerer's Stone was released in the 1990s*), our Self-Graph Reasoning (SGR) generates a structured reasoning graph that provides a consistent and clear reasoning process. As shown, two reasoning branches respectively consider *"the release date of the first iPhone"* and *"the release date of the first Harry Potter movie,"* which eventually converge to the correct conclusion. Each step is explicit and verifiable, allowing easy detection of possible hallucinations or reasoning errors.

4 Experimental Setup

4.1 Datasets

Training Dataset. To enable the LLM to perform self-graph reasoning before answering questions, we construct a dataset of about 10K samples based on the training subset of the LogiQA dataset (Liu et al., 2020). LogiQA is a general-domain QA benchmark that involves complex logical reasoning, making it particularly suitable for supervising structured graph-based reasoning. The final 10K samples are obtained after data cleaning and filtering. For each question–answer pair, we use GPT-4o (Hurst et al., 2024) to generate an explicit reasoning graph represented as {reasoning step$_i$→reasoning step$_j$}, which captures the logical dependencies leading to the correct answer. The resulting dataset is organized in the format of {*Question*, *Graph Reasoning*, *Label*}. More information is shown in Appendix B. We randomly split the dataset into training and validation subsets with a ratio of 9:1.

Evaluation Benchmarks. Our work primarily targets general-domain question answering. To evaluate the model's performance in this setting, we adopt several widely used benchmarks, including the LogiQA test set (Liu et al., 2020), AIW, AIW+ (Nezhurina et al., 2024), and AR-LSAT (Wang et al., 2022). In addition, to evaluate the cross-domain generalization of our method, which is trained on general-domain data, we further evaluate it on MedQA (Jin et al., 2020) and MathQA (Amini et al., 2019). Details of the benchmarks are provided in Appendix A.

4.2 Baselines

We compare our proposed framework with three categories of models, including proprietary LLMs, open-source LLMs, and specialized methods.

For proprietary LLMs, we include GPT-4o (Hurst et al., 2024), GPT-5.1 (OpenAI), Claude-3.5-Haiku (Anthropic), and Gemini-2.5-Pro (Google, 2025), which serve as strong closed-source baselines with advanced reasoning capabilities. For open-source LLMs, we evaluate a series of models with varying scales, including LLaMA-3.2-3B, LLaMA-3.1-8B, LLaMA-3.1-8B, LLaMA-3.3-70B (Dubey et al., 2024), Qwen2.5-7B, Qwen2.5-72B (Yang et al., 2024a). We also compare our method with the specialized method Reasoning with Graphs (RwG) (Han et al., 2025), which relies on externally pre-extracted graphs to enhance model reasoning.

4.3 Implementation Details

We perform supervised fine-tuning of LLaMA-3.3-70B [2] using LoRA (Hu et al., 2022), keeping the base model frozen. Models are trained for 1-3 epochs with early stopping (patience=1), using a batch size of 4 and learning rate 6×10^{-5}. Gradient accumulation is employed to effectively increase the batch size. Maximum generation length is set to 1024. The checkpoint with the best validation performance is selected for evaluation. Experiments are conducted on 8 NVIDIA A100 40GB GPUs. Full hyperparameter details are provided in the Appendix C. We use accuracy (Acc) as the evaluation metric for our experiments, following previous methods (Cao, 2024; Han et al., 2025).

---

[2] `https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct`

	General Domain			Specialized Domain		
Method	LogiQA	AIW	AR-LSAT	MedQA	MathQA	Overall Avg. (%)
Proprietary LLMs						
GPT-4o (Hurst et al., 2024)	74.01	32.50	31.75	88.29	81.05	61.52
GPT-5.1 (OpenAI)	76.34	57.00	33.33	89.55	39.09	59.06
Claude-3.5-Haiku (Anthropic)	65.97	2.50	29.41	76.36	79.36	50.72
Gemini-2.5-Pro* (Google, 2025)	85.75	76.00	96.22	-	73.10	-
Open-source LLMs						
LLaMA-3.2-3B (Dubey et al., 2024)	41.28	1.70	20.00	49.57	29.21	28.35
LLaMA-3.1-8B (Dubey et al., 2024)	49.17	5.00	26.96	55.22	29.39	33.15
LLaMA-3.3-70B (Dubey et al., 2024)	64.01	19.50	31.30	63.55	38.09	43.29
Qwen2.5-7B (Yang et al., 2024a)	34.10	5.00	17.39	59.54	38.29	30.96
Qwen2.5-72B (Yang et al., 2024a)	**76.91**	5.00	**34.78**	74.42	52.60	49.00
Specialized Graph-based Methods						
RwG-LLaMA3.1-70B† (Han et al., 2025)	59.13	12.00	31.73	-	-	-
RwG-Claude-3-sonnet† (Han et al., 2025)	45.16	2.60	30.86	-	-	-
Ours						
SGR-Llama3.3-70B	69.91	**57.50**	31.74	**78.81**	**67.17**	**61.03**

Table 1: Accuracy of models across all benchmarks, covering both general-domain and specialized-domain question answering. Methods are categorized into proprietary LLMs, open-source LLMs, and specialized reasoning approaches. The best results among non-proprietary LLMs are highlighted in **bold**, and the second-best are underlined. The red and blue indicate the average performance of our SGR method and its base model, respectively. *Gemini results are omitted for medical-domain questions due to policy restrictions.†RwG results are taken from the original publication due to reproduction issues; therefore, MedQA and MathQA are excluded.

5 Results and Analysis

5.1 Main Results

Table 1 reports the accuracy of various models, including proprietary LLMs, open-source LLMs, and specialized graph-based methods, across general-domain and specialized-domain question answering benchmarks. In general, our SGR-Llama3.3-70B achieves an average accuracy of 61.03%, demonstrating competitive performance across both general and specialized domains. Through efficient graph reasoning fine-tuning, our method performs on par with the powerful proprietary LLM GPT-4o, while substantially outperforming all comparable open-source models of similar scale.

Specifically, in the general-domain setting, including LogiQA, AIW, and AR-LSAT, our SGR-Llama3.3 70B achieves an average accuracy of 53.05%, surpassing GPT-4o (46.08 %) and significantly outperforming all existing open-source LLMs. Compared to its base model LLaMA-3.3-70B, our approach brings a 17.74% improvement. Particularly on the AIW dataset that relies on logical reasoning, our method achieves 57.50%, outperforming GPT-4o by 25 points and LLaMA-3.3-70B by 38 points on average across the three benchmarks. Compared with graph-based baselines, our method achieves an 18.76% higher average accuracy than RwG-LLaMA3.1-70B (52.73%), which depends on pre-extracted external graphs for reasoning. These results highlight the effectiveness of self-graph reasoning, especially under complex reasoning scenarios in general-domain tasks.

To further evaluate the generalization capability of our method, we conduct experiments on specialized-domain tasks, including MedQA and MathQA. Notably, the training data are from general-domain sources, containing no domain-specific knowledge of medicine or mathematics. SGR-Llama3.3 70B achieves 78.81% on MedQA and 67.17% on MathQA, reaching an average accuracy of 72.99%, a 22.17% improvement over the base model. These results indicate that self-graph reasoning enables LLMs to develop a structured, human-like reasoning process that generalizes beyond specific domains and remains effective even without task-specific fine-tuning.

5.2 Comparison with Existing Paradigms.

To further demonstrate the effectiveness of our self-graph reasoning (SGR) compared with existing reasoning paradigms, we conduct comparative

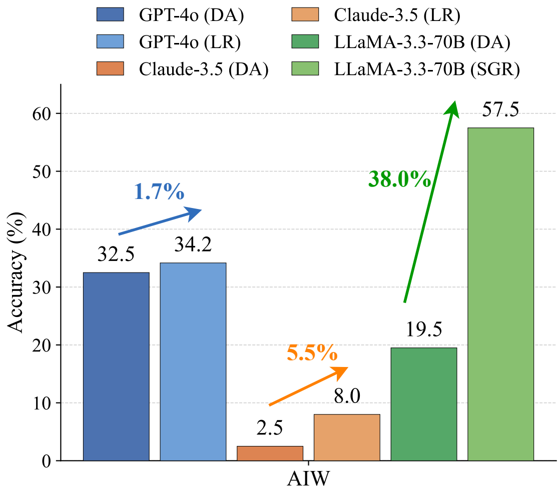


Figure 4: Comparison of accuracy under different reasoning paradigms on the AIW dataset. We evaluate three paradigms: Direct Answering (DA), Linear Reasoning (LR), and our Self-Graph Reasoning (SGR) across GPT-4o, Claude-3.5-haiku, and LLaMA-3.3-70B.

experiments on the AIW dataset, which requires strong reasoning ability. Specifically, we evaluate three paradigms: direct answering, linear reasoning (CoT), and our SGR approach. For CoT, we apply standard CoT prompting to both GPT-4o and Claude-3.5-Haiku. For comparison, we train LLaMA-3.3-70B with our SGR framework as the representative implementation of our method. The results are presented in Figure 4. The experimental results show that, compared to direct answering, GPT-4o achieves a 1.7% gain and Claude-3.5-Haiku achieves a 5.5% gain with CoT reasoning. In contrast, our SGR-LLaMA-3.3-70B achieves a remarkable 38% improvement, surpassing both proprietary models. These results demonstrate the effectiveness of graph-structured reasoning, which enables clearer and logically consistent reasoning processes.

5.3 Ablation Studies

To further assess the effectiveness of our proposed SGR, we conduct ablation studies across LLMs of different scales, comparing base models with and without self-graph reasoning. We evaluate LLaMA-3.3-8B and LLaMA-3.3-70B, with results summarized in Figure 5. The results show that the 8B model exhibits only marginal gains on MathQA and MedQA, and even degradation on more complex general-domain QA datasets, suggesting that limited base capabilities constrain it to effectively perform self-graph reasoning. In contrast, the 70B model demonstrates substantial and consistent improvements across all benchmarks.

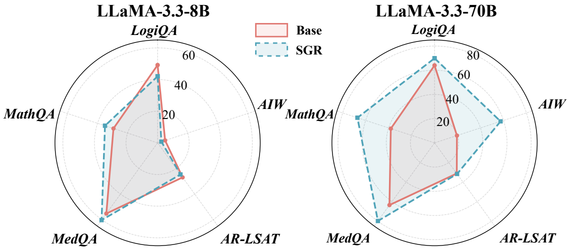


Figure 5: Comparison of accuracy between the base models and our SGR across all five benchmarks for LLaMA-3.3-8B and LLaMA-3.3-70B.

Overall, these findings indicate that the benefits of SGR correlate with the model's underlying reasoning capacity, with stronger LLMs better able to reason in a graph-structured manner, and the impact of SGR becomes more pronounced as model size increases. We expect that applying SGR to even larger or more capable LLMs will yield further improvements. Further experimental analyses are presented in the Appendix.

5.4 Case studies

We conduct a case study on general-domain QA to illustrate how our method enables LLMs to perform self-graph reasoning. We also visualize the reasoning graph generated by our model, as shown in Figure 6. Unlike linear reasoning paradigms such as CoT, SGR explicitly constructs a structured reasoning graph before producing the final answer. This graph representation clearly demonstrates how the model decomposes the problem into interpretable reasoning units.

For the given question, the model generates a reasoning graph consisting of 9 nodes and 8 directed edges, where each node corresponds to a reasoning step, and each edge encodes an explicit dependency between reasoning units. This structured organization makes the reasoning process clearer and reduces the risk of potential context inconsistency and error propagation that may arise in linear reasoning paradigms. Specifically, the branching node *"Identify siblings (brothers & sisters)"* separates the interpretation of the two sibling types, creating two independent reasoning branches. This prevents the mixing of different reasoning units, a common issue in linear CoT, and allows the model to focus separately on the brother-related and sister-related subproblems. Notably, in the "brothers" branch, SGR correctly halts further expansion once it recognizes that the question concerns the sister count per brother, and subsequently redirects reasoning

Question Alice has 4 brothers and she also has 3 sisters. How many sisters does one of Alice's brother have?

Actual Output

<reasoning>
<step>Alice exists -> leads to -> Consider family members related to her</step>
<step>Consider family members related to her -> leads to -> Identify siblings (brothers & sisters)</step>
<step>**Identify siblings (brothers & sisters)** -> leads to -> Note number of brothers = 4</step>
<step>**Identify siblings (brothers & sisters)** -> leads to -> Note number of sisters = 3 + Alice herself for sibling count from brother perspective</step>
<step>Note number of brothers = 4 -> leads to -> Each brother is part of this group but question focuses on sister count per brother</step>
<step>Note number of sisters = 3 + Alice herself for sibling count from brother perspective -> leads to -> Total sisters seen by each brother includes Alice now counted among them</step>
<step>Total sisters seen by each brother includes Alice now counted among them -> leads to -> **Calculate total sisters including Alice from brother's view = 3 actual sisters + 1 (Alice)**</step>
<step>Calculate total sisters including Alice from brother's view = 3 actual sisters + 1 (Alice) -> leads to -> **Conclusion that each brother sees 4 sisters**</step>
</reasoning>
<answer>4</answer>

Reasoning Graph Visualization

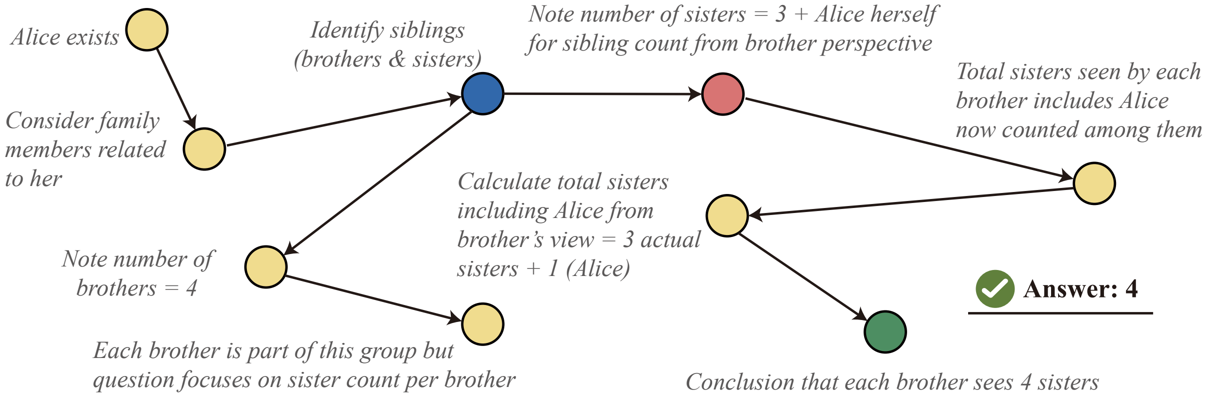


Figure 6: A case study in general-domain QA, illustrating the model's actual output and its reasoning graph visualization. The blue nodes denote branch nodes, the red nodes denote key nodes, and the green node denotes the final decision node.

to the sister-related branch.

Further, a crucial advantage of SGR is evident in the node *"Note number of sisters = 3 + Alice herself for sibling count from brother perspective"*, which explicitly encodes the perspective shift required to answer the question. The structured reasoning graph enforces the creation of an independent node to handle this logical transition from Alice's view (3 sisters) to a brother's view, where Alice must now be counted as a sister. This step is typically implicit or mishandled in linear reasoning paradigms. By explicitly constructing this node, SGR ensures that this key reasoning step is neither overlooked nor weakened.

Finally, SGR aggregates information from multiple upstream nodes through the intermediate computation node *"Calculate total sisters including Alice from brother's view = 3 actual sisters + 1 (Alice)"*, and converges on the decision node *"Conclusion that each brother sees 4 sisters."* The graph structure ensures that intermediate reasoning aligns with the final output, leading the model to the correct answer: 4.

6 Conclusion

In this work, we presented Self-Graph Reasoning (SGR), a framework that enables LLMs to perform graph-structured reasoning on their own for open-domain question answering. SGR overcomes the limitations of linear reasoning and externally provided graphs by allowing models to express their reasoning explicitly in a structured form. We also construct a graph-structured reasoning dataset for training LLMs to perform graph-based reasoning effectively. Experiments across five QA benchmarks show that SGR substantially improves reasoning consistency, yielding a 17.74% average gain over its base model across all benchmarks.

Limitations

Training Data. The scale of the training data for self-graph reasoning is relatively limited. The constructed self-graph reasoning dataset contains approximately 10K training instances, which may constrain the full potential of the proposed SGR framework. We expect that expanding the training data to a larger scale would further improve both the performance and generalization of SGR.

Base Model Scale. As discussed in the paper, the effectiveness of SGR is inherently tied to the capability of the base models. Due to computational constraints, our experiments are conducted on a 70B model. Applying SGR to larger-scale models may further improve reasoning consistency and overall performance.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pages 2357–2367.

Anthropic. Claude 3.5 haiku. `https://www.anthropic.com/news/3-5-models-and-computer-use`. Retrieved December 12, 2025.

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. 2025. Chain-of-thought reasoning in the wild is not always faithful. *arXiv preprint arXiv:2503.08679*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, and 1 others. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Lang Cao. 2024. Graphreason: Enhancing reasoning capabilities of large language models through a graph-based verification approach. In *Proceedings of the 2nd Workshop on Natural Language Reasoning and Structured Explanations (@ ACL 2024)*, pages 1–12.

Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. A review: Knowledge reasoning over knowledge graph. *Expert systems with applications*, 141:112948.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

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*.

Google. 2025. Gemini-2.5-pro. `https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-pro`. Retrieved December 12, 2025.

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*.

Haoyu Han, Yaochen Xie, Hui Liu, Xianfeng Tang, Sreyashi Nag, William Headden, Yang Li, Chen Luo, Shuiwang Ji, Qi He, and 1 others. 2025. Reasoning with graphs: Structuring implicit knowledge to enhance llms reasoning. *arXiv preprint arXiv:2501.07845*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

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*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and 1 others. 2024. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. *arXiv preprint arXiv:2404.07103*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*.

Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, Peifeng Wang, Silvio Savarese, and 1 others. 2025. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems. *arXiv preprint arXiv:2504.09037*.

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*.

Jihyung Lee and Gary Geunbae Lee. 2025. Structured reasoning and answer verification: Enhancing question answering system accuracy and explainability. *Knowledge-Based Systems*, 311:113091.

Jindong Li, Yali Fu, Li Fan, Jiahong Liu, Yao Shu, Chengwei Qin, Menglin Yang, Irwin King, and Rex Ying. 2025. Implicit reasoning in large language models: A comprehensive survey. *arXiv preprint arXiv:2509.02350*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061*.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. 2024. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. *arXiv preprint arXiv:2410.13080*.

Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. 2024. Alice in wonderland: Simple tasks showing complete reasoning breakdown in state-of-the-art large language models. *arXiv preprint arXiv:2406.02061*.

OpenAI. Gpt5.1. `https://openai.com/ja-JP/index/introducing-gpt-5/`. Retrieved December 12, 2025.

Avinash Patil and Aryan Jadon. 2025. Advancing reasoning in large language models: Promising methods and approaches. *arXiv preprint arXiv:2502.03671*.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M Ni, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *arXiv preprint arXiv:2307.07697*.

Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, and Wenjie Zhang. 2025. Paths-over-graph: Knowledge graph empowered large language model reasoning. In *Proceedings of the ACM on Web Conference 2025*, pages 3505–3522.

Yijun Tian, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, Nitesh V Chawla, and Panpan Xu. 2024. Graph neural prompting with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19080–19088.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Preprint*, arXiv:2305.04388.

Siyuan Wang, Zhongkun Liu, Wanjun Zhong, Ming Zhou, Zhongyu Wei, Zhumin Chen, and Nan Duan. 2022. From lsat: The progress and challenges of complex reasoning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Zezhong Wang, Xingshan Zeng, Weiwen Liu, Yufei Wang, Liangyou Li, Yasheng Wang, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. 2025. Chain-of-probe: Examining the necessity and accuracy of CoT step-by-step. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2586–2606, Albuquerque, New Mexico. Association for Computational Linguistics.

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.

Yilin Wen, Zifeng Wang, and Jimeng Sun. 2024. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10370–10388.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024. Faithful logical reasoning via symbolic chain-of-thought. *arXiv preprint arXiv:2405.18357*.

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024a. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Rui Yang, Qingcheng Zeng, Keen You, Yujie Qiao, Lucas Huang, Chia-Chun Hsieh, Benjamin Rosand, Jeremy Goldwasser, Amisha Dave, Tiarnan Keenan, and 1 others. 2024b. Ascle—a python natural language processing toolkit for medical text generation: development and evaluation study. *Journal of Medical Internet Research*, 26:e60601.

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.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *Preprint*, arXiv:2210.03493.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and 1 others. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Benchmark Details

A.1 General Domain Benchmarks

LogiQA. The LogiQA dataset (Liu et al., 2020) is a multiple-choice reading comprehension benchmark designed to evaluate logical reasoning in natural language understanding. Each instance consists of a context, a question, and four candidate answers, only one of which is logically correct. The questions are sourced from real-world examination materials, such as national civil service and postgraduate entrance tests, ensuring a high level of reasoning complexity. We use the training set of LogiQA2.0 to construct our graph reasoning dataset and the test set as a benchmark for evaluation.

AIW. The AIW dataset (Nezhurina et al., 2024) is a benchmark of reasoning problems. Each problem typically presents a simple natural language scenario, such as familial relationships with variables, and requires the model to derive a logically correct answer from basic commonsense logic. Unlike traditional multiple-choice datasets, AIW focuses on minimal text examples that expose reasoning breakdowns in state-of-the-art models, making it a useful diagnostic for evaluating core inferential competence.

AR-LSAT. The AR-LSAT dataset (Wang et al., 2022) is a large-scale benchmark constructed from the analytical reasoning (logic games) section of the official LSAT (Law School Admission Test). It is designed to evaluate the formal reasoning and deductive inference abilities of language models. Each problem describes a context and requires the model to select the correct answer. We use the test set as a benchmark for evaluation.

Dataset	Size	Question Length
General Domain		
LogiQA	1572	930.57
AIW	200	98.79
AR-LSAT	230	892.36
Specialized Domain		
MedQA	1273	877.57
MathQA	2985	216.76

Table 2: Statistics of Benchmark Datasets. We report the size of each benchmark and the average text length of questions.

Hyperparameter	Value
seed	42
batch_size	8
num_epochs	1–3
learning_rate	5×10^{-6}
weight_decay	0.01
grad_steps	4
warmup	0.05
early_stop_patience	1
lora_r	4
lora_alpha	16
lora_dropout	0.1
lora_target_modules	q_proj, v_proj
max_txt_len	1024
max_new_tokens	1024

Table 3: Hyperparameters.

A.2 Specialized Domain Benchmarks

MedQA. The MedQA dataset (Jin et al., 2020) is a large-scale medical question answering benchmark designed to evaluate the clinical knowledge and reasoning abilities of language models. Each question is a multiple-choice item sourced from real-world medical licensing examinations. The dataset covers a wide range of medical domains, requiring models to integrate factual recall with domain-specific reasoning.

MathQA. The MathQA dataset (Amini et al., 2019) is a large-scale benchmark for evaluating the mathematical reasoning and problem-solving abilities of language models. It is gathered by using a new representation language to annotate over the AQuA-RAT dataset, covering a wide range of mathematical domains. Each question requires the model to translate a natural language description into a formal reasoning process and compute the correct numerical answer.

A.3 Preprocessing for Benchmarks

To facilitate a unified evaluation across different benchmarks, we standardize all datasets into a consistent {question, label} format. For datasets originally containing a separate *context* field, we concatenate them with the question text to form a single *question* input. The details are summarized in Table 2.

Method	General Domain			Specialized Domain		Overall Avg. (%)
	LogiQA	**AIW**	**AR-LSAT**	**MedQA**	**MathQA**	
LLaMA-3.1-8B	49.17	5.0	26.96	55.22	29.39	33.15
SGR-LLaMA-3.1-8B	42.05	2.50	24.68	60.36	35.03	32.92
SGR-LLaMA-3.1-8B w/ GRPO	56.68	5.5	24.35	59.47	36.25	36.45

Table 4: Accuracy of the base LLaMA-3.1-8B model, our SRG version, and the GRPO-fine-tuned version across all benchmarks.

B Our Graph Reasoning Dataset

We construct our graph-based reasoning dataset based on the training set of LogiQA 2.0. The original LogiQA 2.0 training corpus contains a total of 12,547 samples. After reasoning graph construction and data cleaning, we retain 9,869 high-quality samples for our graph reasoning benchmark. Examples of the constructed graph reasoning data is shown in Figure 7.

C Detail of Hyperparameter.

We list all the parameters used for SGR-Llama3.3 70B, as shown in table 3. This includes configuration details such as batch size, learning rate, LoRA, and optimizer settings.

D Analysis of Computational Cost

We compared the computational cost of our proposed SRG-LLaMA-3-70B model with several LLMs on the LogiQA benchmark. For our locally deployed model, the cost was estimated at $0.8 per GPU hour, requiring approximately 42 GPU hours in total. As shown in Table 5, our method incurs a total cost of about $33.6, which is substantially lower than that of GPT-4o (CoT) (approximately $80), while achieving comparable performance and offering a more interpretable, graph-structured reasoning process. Given that our model can be deployed locally with a moderate computational budget, these results highlight the efficiency and scalability of the SRG framework for reasoning tasks.

Model	Cost ($)
GPT-4o	~26
Claude-3.5-haiku	~10
Gemini-2.5-Pro	~24
GPT-4o CoT	~80
Claude-3.5-haiku CoT	~32
SGR-Llama3.3-70B(Ours)	~33.6

Table 5: Comparison of the cost of our method with other LLMs on the LogiQA test set.

E GRPO Fine-Tuning

To mitigate the potential dilution of supervision on the final answer caused by intermediate reasoning steps, we perform an additional round of GRPO fine-tuning following standard SFT. This stage strengthens both the structured reasoning format and the accuracy of the final answer through two complementary reward functions. The first reward evaluates whether the model's output strictly adheres to the predefined structured reasoning–answer template, while the second reward assesses whether the content within the <*answer*> tag matches the label y_i

We conduct GRPO fine-tuning based on the LLaMA-3.3-8B, using a batch size of 8, a maximum of 6000 steps, and a learning rate of 5×10^{-6}. This post-SFT training phase encourages the model to preserve structured reasoning while maximizing the correctness of the final answer. Results in Table 4 indicate that the additional GRPO fine-tuning substantially enhances the model's answer accuracy, particularly on the in-domain LogiQA dataset, where it achieves a 7.51% improvement over the base model. This demonstrates that GRPO

Method	LogiQA	AIW	MedQA
Qwen3-8B	73.20	43.00	66.80
Qwen3-8B-thinking	80.40	80.00	75.80
GPT-5.1	76.34	57.00	89.55
SGR-LLaMA-3.3-70B	69.91	57.50	79.81

Table 6: Comparison of accuracy for our Self-Graph Reasoning (SGR) framework against recent reasoning LLMs, including Qwen-3-8B-thinking and other LLMs, on the LogiQA, AIW, and MedQA datasets.

effectively mitigates the performance degradation associated with the limited capacity of small-scale LLMs. However, we also observe that the reasoning process of the GRPO-tuned model becomes overly concise, occasionally omitting intermediate reasoning steps. These findings suggest an inherent trade-off between answer accuracy and reasoning completeness in small-scale language models.

F Comparison with Reasoning LLMs

To assess the effectiveness of our Self-Graph Reasoning (SGR) framework, we compare its performance against the recent reasoning LLM Qwen-3-8B-thinking (Yang et al., 2025), evaluated on the LogiQA, AIW, and MedQA datasets, as shown in Table 6. Qwen-3B-thinking achieves strong results across all three benchmarks, even outperforming the proprietary GPT-5.1 model. This may be partly due to the inclusion of these public benchmarks in its pretraining or instruction-tuning data, which likely provides prior exposure to the evaluation distributions. In contrast, our SGR-LLaMA-3.3-70B model is fine-tuned with only a small amount of graph data and without any benchmark-specific priors, yet it still achieves competitive accuracy.

G Prompts

We provide the prompts used in our experiments. To generate candidate reasoning graphs, the model is prompted as illustrated in Figure 8. For integrating candidate graphs into an optimal reasoning graph, the model is prompted as shown in Figure 9. A standard question-answering prompt is constructed for the baseline LLMs, as depicted in Figure 10. Within our Self-Graph Reasoning (SGR) framework, the model is prompted as illustrated in Figure 11.

Question	Reasoning Path	Label
In a magic show, from the seven magicians - G.H.K.L.N.P and Q, select 6 to perform, perform into two teams: 1 team and 2 team. Each team by the front, middle and after three positions, playing the magician just each occupies a position, the selection and position arrangement of the magician must meet the following conditions :(1) if the arrangement of G or H play, they must be in the front. (2) If K is to play, he must be in the middle. (3) If L is to play, he must be in team 1. (4) Neither P nor K can be on the same team as N. (5) P cannot be in the same team as Q. (6) If H is in team 2, Q is in the middle position of team 1. If H is in team 2, which of the followings is listed as an acceptable arrangement for team 1? Options: A. Front: L Middle: Q Rear: N B. Front: G Middle: K Rear: N C. Front: L Middle: Q Rear: G D. Front: Q Middle: K Rear: L	["H is in team 2", "leads to", "Q is in the middle of team 1"] ["Q is in the middle of team 1", "leads to", "Evaluate possible positions for L"] ["Evaluate possible positions for L", "leads to", "L must be in team 1"] ["L must be in team 1", "leads to", "L can be in the front or rear of team 1"] ["L can be in the front or rear of team 1", "leads to", "Consider L in the front"] ["L can be in the front or rear of team 1", "leads to", "Consider L in the rear"] ["Consider L in the front", "leads to", "Evaluate rear position for team 1"] ["Consider L in the rear", "leads to", "Evaluate front position for team 1"] ["Evaluate rear position for team 1", "leads to", "N can be at the rear"] ["Evaluate front position for team 1", "leads to", "N can be in the front"] ["N can be at the rear", "leads to", "Front: L, Middle: Q, Rear: N"] ["N can be in the front", "leads to", "Front: N, Middle: Q, Rear: L"] ["Front: L, Middle: Q, Rear: N", "leads to", "Option A is valid"] ["Front: N, Middle: Q, Rear: L", "leads to", "Check validity based on other rules"] ["Check validity based on other rules", "leads to", "Option A satisfies all rules"] ["Option A is valid", "leads to", "Conclusion: Option A is an acceptable arrangement for team 1"]	A
Among people who have a history of chronic trouble falling asleep, some rely only on sleeping pills to help them fall asleep, and others practice behavior modification techniques and do not take sleeping pills. Those who rely only on behavior modification fall asleep more quickly than do those who rely only on sleeping pills, so behavior modification is more effective than are sleeping pills in helping people to fall asleep. Which one of the following, if true, most weakens the argument? Options: A. The people who are the most likely to take sleeping pills rather than practice behavior modification techniques are those who have previously had the most trouble falling asleep. B. People who do not take sleeping pills spend at least as many total hours asleep each night as do the people who take sleeping pills. C. Most people who have trouble falling asleep and who use behavior modification techniques fall asleep more slowly than do most people who have no trouble falling asleep. D. The people who are the most likely to practice behavior modification techniques rather than take sleeping pills are those who prefer not to use drugs if other treatments are available.	["Some people have trouble falling asleep", "leads to", "Some people use sleeping pills"] ["Some people have trouble falling asleep", "leads to", "Some people use behavior modification"] ["Some people use sleeping pills", "leads to", "People using pills fall asleep slower than those using behavior modification"] ["Some people use behavior modification", "leads to", "People using behavior modification fall asleep faster"] ["People using pills fall asleep slower than those using behavior modification", "leads to", "Behavior modification is more effective than pills"] ["Behavior modification is more effective than pills", "leads to", "Behavior modification helps people fall asleep faster than pills"] ["Some people use sleeping pills", "leads to", "People using pills may have had more trouble falling asleep initially"] ["People using pills may have had more trouble falling asleep initially", "leads to", "Initial severity of insomnia affects falling asleep speed"] ["Initial severity of insomnia affects falling asleep speed", "leads to", "Behavior modification may not inherently be more effective"] ["Behavior modification may not inherently be more effective", "leads to", "Option A weakens the argument"] ["People using pills may have had more trouble falling asleep initially", "leads to", "People with severe insomnia prefer sleeping pills"] ["People with severe insomnia prefer sleeping pills", "leads to", "Severity affects effectiveness comparison"] ["Severity affects effectiveness comparison", "leads to", "Ignoring severity difference weakens argument"] ["Ignoring severity difference weakens argument", "leads to", "Option A is correct"]	A
The so-called memory representation refers to the image of things perceived by people in the human brain. Which of the following activities is designed for memory content that does not belong to memory representation? Options: A. Tom still remembers her eating Quanjude roast duck in China. B. Lao Li often recalls the hard life of his childhood. C. The beautiful sound of the piano echoed in her ears. D. Xiao Hong can recite the multiplication formula skillfully.	["Memory representation is the image of things perceived", "leads to", "Memory representation involves sensory or emotional experiences"] ["Memory representation involves sensory or emotional experiences", "leads to", "Memory representation is not abstract knowledge"] ["Memory representation involves sensory or emotional experiences", "leads to", "Tom remembers eating roast duck is a sensory experience"] ["Memory representation involves sensory or emotional experiences", "leads to", "Lao Li recalls hard life is a sensory experience"] ["Memory representation involves sensory or emotional experiences", "leads to", "The sound of the piano is a sensory experience"] ["Memory representation is not abstract knowledge", "leads to", "Reciting the multiplication formula is abstract knowledge"] ["Tom remembers eating roast duck is a sensory experience", "leads to", "Option A relates to memory representation"] ["Lao Li recalls hard life is a sensory experience", "leads to", "Option B relates to memory representation"] ["The sound of the piano is a sensory experience", "leads to", "Option C relates to memory representation"] ["Reciting the multiplication formula is abstract knowledge", "leads to", "Option D does not relate to memory representation"] ["Option A relates to memory representation", "leads to", "Option A is not the answer"] ["Option B relates to memory representation", "leads to", "Option B is not the answer"] ["Option C relates to memory representation", "leads to", "Option C is not the answer"] ["Option D does not relate to memory representation", "leads to", "Option D is the answer"]	D

Figure 7: Samples of our graph reasoning data. Each sample consists of a question, its corresponding reasoning graph, and the correct answer (label).

Prompt for candidate reasoning graph generation

Question: **{question}**
Answer: **{answer}**

Construct a **concise reasoning graph** that explores the key reasoning paths leading to the correct answer.
The graph should capture multiple hypotheses or branches, but keep it **short and high-quality**: only 3-5 key steps plus the final conclusion.

Granularity rules:
- Each node must be an **atomic proposition**: one subject + one predicate (+ optional object/value).
- Split sentences by meaning: if a claim can be divided into smaller independent claims, make multiple nodes.
- No pronouns; always use the same canonical entity names.
- Each node should be concise and only include essential information.
- Avoid redundancy: do not repeat the same fact multiple times.

Graph rules:
- Start from the initial facts given in the question.
- Allow **branching** into multiple reasoning directions only when alternative possibilities exist.
- Branches should converge to the final conclusion that supports the correct answer.
- Each edge represents a **small, justified inference step**, formatted as ["premise", "leads to", "next step"].
- Keep the reasoning graph **connected from start to conclusion**.

Output format (strictly):
<reasoning>
["node1", "leads to", "node2"]
["node1", "leads to", "node3"]
["node2", "leads to", "node4"]
...
["nodeN", "leads to", "conclusion"]
</reasoning>

<answer>
A/B/C/D
</answer>

Constraints:
- Output only the <reasoning> graph and the <answer> block.
- Ensure reasoning is **concise**, **atomic**, and **high-quality**.
- Represent multiple branches when necessary, but limit the graph to key steps (3-5 + final conclusion).

Figure 8: Prompt for candidate reasoning graph generation.

Prompt for candidate reasoning graph integration

Question: **{question}**
Answer: **{answer}**

You are given three reasoning paths that attempt to explain the answer to this question.
Your task is to integrate them into a **single optimal reasoning graph** that includes multiple possible branches but ultimately converges on the correct answer.

Reasoning 1: **{reasoning1}**
Reasoning 2: **{reasoning2}**
Reasoning 3: **{reasoning3}**

Integration rules:
1. If two entities are similar or co-refer, merge them into one, keeping the more meaningful/specific name.
2. Each node must be an **atomic proposition** (one subject + one predicate + optional object/value).
3. If a sentence expresses multiple independent claims, split it into multiple nodes.
4. Each node must represent exactly one indivisible fact/idea.
5. Each edge must represent a **small justified inference step**, not a leap.
6. When evaluating options, base judgments only on explicit prior nodes.
7. At least one node must branch into ≥2 alternative reasoning directions before convergence.
8. All branches must remain connected and converge toward the final conclusion.

Output format:
<reasoning>
["thinking1", "leads to", "thinking2"]
["thinking1", "leads to", "thinking3"]
["thinking2", "leads to", "thinking4"]
["thinking3", "leads to", "thinking5"]
...
["thinkingN", "leads to", "conclusion"]
</reasoning>

<answer>
A/B/C/D
</answer>

Constraints:
- Output only the <reasoning> graph and the <answer> block, nothing else.
- All nodes must be atomic, unique, and quoted strings.
- Ensure at least one explicit branching before convergence to the conclusion.

Figure 9: Prompt for candidate reasoning graph integration.

Prompt for the baseline LLM

Question: **{question}**
Please provide the correct answer in the format: <answer>...</answer>
For example, if choosing option B, the output is: <answer> B </answer>.

Figure 10: Prompt for the baseline LLMs.

Prompt used in our Self-Graph Reasoning (SGR) framework

Question: **{question}**
Instruction:
1. Carefully think step by step to determine the correct answer.
2. Show your graph-style step-by-step reasoning in a concise <reasoning> block.
3. After reasoning, always provide the final answer in <answer> tag.
4. Format strictly as:
<reasoning>
<step>node A -> leads to -> node B</step>
...
</reasoning>
<answer>...</answer>
5. Do NOT generate anything outside <reasoning> and <answer>.
6. Keep <reasoning> concise—just enough to justify your answer, do not overexpand.

Figure 11: Prompt used in our Self-Graph Reasoning (SGR) framework.