Reasoning Graph-Structured Question Answering: Datasets and Insights from LLM Benchmarking

Khin Saw Yone[1†], Devasha Trivedi[1†], Anish Pahilajani[1*], Jincen Shuai[1*], Samyak Rajesh Jain[1*], Ryan A. Rossi[2], Nesreen K. Ahmed[3], Franck Dernoncourt[2], Yu Wang[4], Namyong Park

[1]University of California Santa Cruz, [2]Adobe Research,
[3]Cisco Outshift, [4]University of Oregon
{kyone, detrived, apahilaj, jshuai, srajeshj}@ucsc.edu
ryrossi@adobe.com, nesahmed@cisco.com, dernonco@adobe.com,
yuwang@uoregon.edu, namyongp@cs.cmu.edu

Abstract

Large Language Models (LLMs) have shown remarkable success in multi-hop question-answering (M-QA) due to their advanced reasoning capabilities. However, the influence of reasoning structures on their performance remains underexplored, primarily due to the lack of M-QA datasets that explicitly encode the reasoning pathways underlying each question-answer pair. While existing benchmarks such as HotpotQA, 2WikiMultiHopQA, MuSiQue, and GSM8K evaluate multi-step reasoning, they do not provide a unified, explicit representation of reasoning structure that enables controlled structural analysis. In this work, we introduce GRS-QA, a reasoning graph-structured question answering dataset that augments multi-hop QA instances from exisiting datasets, with directed reasoning graphs representing intermediate inference steps. By unifying and extending graph-style annotations across multiple textual and mathematical benchmarks, GRS-QA enables fine-grained evaluation of LLM performance across varying context structures, prompting styles, and data domains. We conduct a systematic prompting study comparing reasoning graphs as contextual grounding versus as in-context exemplars. Our experiments reveal that example-based structural prompting consistently outperforms graph-as-context conditioning, suggesting that LLMs benefit more from explicit reasoning guidance than supplying contextual information alone. These findings highlight the importance of unified structural annotations for understanding and improving multi-hop reasoning in LLMs.

Keywords: multi-hop question answering, multi-step reasoning, large language models, reasoning graphs

1. Introduction

Reasoning in natural language is the fundamental aspect of intelligence (Huang and Chang, 2023), and QA tasks provide a quantifiable way to test the reasoning capabilities of intelligent systems (Yang et al., 2018). The emergence of LLMs has demonstrated an unprecedented reasoning capacity in answering questions (Wei et al., 2022; Wang et al., 2024). However, real-world applications often demand more complex reasoning capability, such as multi-hop reasoning (Atif et al., 2023), where systems integrate information from multiple sources and perform multiple steps of thinking in a certain order to arrive at the final answer and conclusion.

To evaluate the multi-hop reasoning capabilities of LLMs, researchers have developed several multi-hop question-answering (M-QA) datasets, including HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020) (Wiki), and MuSiQue (Trivedi et al., 2022). Multi-step mathematical reasoning datasets and dynamically extendable benchmarks such as GSM8K (Cobbe et al., 2021), and DARG (Zhang et al., 2024) have also been developed.

HotpotQA is a large-scale and crowd-sourced dataset comprising 113,000 Wikipedia-based QA pairs, offering sentence-level supporting facts for explainable predictions. MuSiQue constructs genuine multi-step QA pairs by composing connected single-hop questions through a bottom-up approach and mitigating existing common shortcuts. 2WikiMultiHopQA integrates structured and unstructured data to provide comprehensive and evidence based reasoning paths, which ensures authentic multi-hop reasoning. GSM8K consists of high quality, linguistically diverse grade school math word problems, and DARG introduces dynamically extended versions of current benchmarks, including GSM8K.

Despite their contributions to benchmarking LLMs' multi-hop reasoning capabilities, the aforementioned widely used M-QA datasets lack explicit reasoning structures for each QA pair, preventing LLMs from leveraging predefined reasoning pathways and forcing them to rely solely on their internal knowledge. Furthermore, while some prior works have introduced reasoning graphs and explanation

[†]These authors contributed equally to this work.
[*]These authors contributed equally.

Code and dataset used in this paper are available at https://github.com/kyone138/grs-qa.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 5301–5316
11–16 May 2026.

structures, these efforts remain domain-specific. For example, WorldTree (Jansen et al., 2018) constructs explanation graphs for elementary science questions; QA-GNN (Yasunaga et al., 2021) builds joint reasoning graphs over QA contexts and knowledge graphs for commonsense reasoning; and Reason2Drive (Nie et al., 2023) introduces reasoning chains for autonomous driving perception tasks. However, none of these works provides a unified, multi-domain benchmark that systematically annotates reasoning graphs across existing multi-hop QA datasets of varying complexity.

To address the lack of a unified benchmark for multi-hop QA and enable structure-aware analysis of multi-hop reasoning, we introduce GRS-QA, a novel question answering dataset augmented with explicit directed reasoning graphs. Each question-answer pair is associated with an inference graph in which nodes represent contextual evidence and edges encode logical dependencies. This unified representation enables structure-aware and systematic evaluation of how LLMs leverage structured reasoning signals across domains and prompting strategies. Our contributions can be summarized as follows:

- **A Graph-Structured Multi-Hop QA Dataset:** We introduce GRS-QA, a large-scale QA dataset that explicitly pairs each question-answer instance with a directed reasoning graph representing the logical steps required to derive the answer. GRS-QA covers both textual and mathematical domains and reasoning complexities.
- **Unified Structural Annotation Across Benchmarks.** By standardizing reasoning graphs across datasets with varying characteristics and reasoning complexities, GRS-QA enables fine-grained analysis of how reasoning structure influences LLM performance. This unified representation allows comparison across domains and reasoning types under a common structural formalism.
- **Analysis of Structure Prompting Effects:** We conduct controlled experiments comparing different prompting styles:
  - Reasoning Graphs as Context (Factual Groundings): Using reasoning graphs as input context alongside the question.
  - Reasoning Graphs as Examples (Prompting Reasoning Paths): Using reasoning graphs as examples in the prompt that demonstrate reasoning steps and how to approach a similar type of task.

  Incorporating *Reasoning Graph Examples* consistently outperforms *Reasoning Graph Context*, demonstrating that example-based prompting provides effective guidance for LLM reasoning even without access to ground-truth reasoning structures.
- **Impact of Domain Complexity and Model Capacity:** Smaller LLMs struggle to interpret reasoning graph structures in the multi-hop textual QA domains, while larger models exhibit better adaptability.
- **Robustness and Sensitivity Analysis:** LLMs demonstrate higher sensitivity to variations and perturbations in reasoning graph examples within textual QA tasks, but remain more stable in the mathematical reasoning domain. Negatively perturbed graphs lead to notable performance degradation.

2. Dataset Creation

GRS-QA constructs reasoning graphs that trace logical paths from questions to answers using QA pairs from HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), 2WikiMultiHopQA (Ho et al., 2020), and DARG-processed GSM8K (Zhang et al., 2024). GRS-QA builds these graphs by leveraging structured fields—question, answer, and supporting facts—where sentences serve as nodes and edges represent logical relationships derived from annotations.

2.1. Positive Graphs

Positive reasoning graphs represent the ground truth reasoning steps from a question to its answer. In these graphs, nodes correspond to sentences or mathematical equations of the golden context that address portions of the question. Edges between nodes define the logical flow that the LLM should follow to arrive at the correct answer. The logical flow is determined by the context or evidence provided in each of the datasets and how each instance points to the others.

In the HotpotQA dataset, questions are categorized as either "bridge" or "comparison." Each question is paired with two sentences from the golden context, which serve as nodes in a reasoning graph. For "comparison" questions, no edges are established between nodes. For "bridge" questions, we use the "keyword" field to identify the second sentence as the tail node, designate the other sentence as the head node, and create an edge from head to tail. Figure 1 shows an example of a HotpotQA data instance.

In the MuSiQue dataset, we utilize the provided question category (e.g., "4hop2") to determine the graph structure. The supplied sentence IDs are used to set up the nodes accordingly.

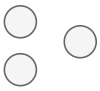
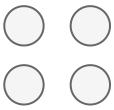
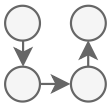
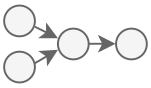
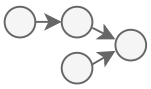
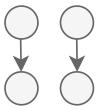
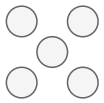
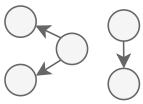
Graph	Type	Question	Decomposition
	Comparison_2_1 (C-2-1)	Between Athlete and Fun, which band has more members? **Athlete**	**1.** How many members are in Athlete? **Four members** **2.** How many members are in Fun? **Three members**
	Bridge_2_1 (B-2-1)	Who beat the player that won the 2017 Australian men's open tennis single title in the US open? **Novak Djokovic**	**1.** Who wins the 2017 australian men's open tennis single title? **Roger Federer** **2.** Who beat **Roger Federer** in the us open? **Novak Djokovic**
	Comparison_3_1 (C-3-1)	In which country is the administrative territorial entity for the city where Charlie Harper was born? **United Kingdom**	**1.** Where was Charlie Harper born? **Hackney** **2.** In which administrative territorial entity is **Hackney** located? **Middlesex** **3.** Which country is **Middlesex** located in? **United Kingdom**
	Bridge_3_1 (B-3-1)	In which country is the administrative territorial entity for the city where Charlie Harper was born? **United Kingdom**	**1.** Where was Charlie Harper born? **Hackney** **2.** In which administrative territorial entity is **Hackney** located? **Middlesex** **3.** Which country is **Middlesex** located in? **United Kingdom**
	Compositional_3_2 (CO-3-2)	In which country is Midway, in the same county as McRae in the same state as KAGH-FM? **U.S.**	**1.** What state is KAGH-FM located? **Arkansas** **2.** In which administrative territorial entity is McRae located? **White County** **3.** Which country is Midway (near Pleasant Plains), **White County**, **Arkansas** located in? **U.S.**
	Comparison_4_1 (C-4-1)	Did Albrecht Alt and Asli Hassan Abade have the same occupation? **no**	**1.** ["Asli Hassan Abade", "occupation", "pilot"] **2.** ["Asli Hassan Abade", "occupation", "military figure"], **3.** ["Asli Hassan Abade", "occupation", "civil activist"] **4.** ["Albrecht Alt", "occupation", "theologian"] **5.** ["Albrecht Alt", "occupation", "lecturer"] **6.** ["Albrecht Alt", "occupation", "professor"] "supporting_facts": [["Asli Hassan Abade", 0], ["Albrecht Alt", 0],["Albrecht Alt", 2], ["Albrecht Alt", 6]]
	Bridge_4_1 (B-4-1)	When did Ukraine gain independence from the first Allied nation to reach the German city where the director of The Man from Morocco was born? **1917**	**1.** Who is the director of The Man from Morocco? **Mutz Greenbaum** **2.** What is the place of birth of **Mutz Greenbaum**? **Berlin** **3.** What allied nation was the first to reach the german capitol of **Berlin**? **Soviet Union** **4.** When did Ukraine gain independence from **Soviet Union**? **1917**
	Compositional_4_2 (CO-4-2)	Where is the place of death of the man who became leader of the largest country in Europe in square miles after the collapse of the nation Germany agreed to sign a non-aggression pact with in 1939? **Moscow**	**1.** What is the largest country in europe by square miles? **Russia** **2.** In 1939 Germany agreed to sign a non-aggression pact with which country? **the Soviet Union** **3.** Who became leader of **Russia** after the collapse of **the Soviet Union**? **Boris Yeltsin** **4.** Where did **Boris Yeltsin** die? **Moscow**
	Compositional_4_3 (CO-4-3)	In what country is Tuolumne, which is within a county that borders the county containing Jamestown, and is located within the state where Some Like It Hot was filmed? **United States**	**1.** In which administrative territorial entity is Jamestown located? **Tuolumne County** **2.** Which entities share a border with **Tuolumne County**? **Stanislaus County** **3.** Where did they film some like it hot? **in California** **4.** Which country is **Tuolumne**, **Stanislaus County**, in California located in? **United States**
	Bridge_Comparison_4_1 (BC-4-1)	Are both directors of films The Blue Bird (1940 Film) and Bharya Biddalu from the same country? **no**	**1.** ['The Blue Bird (1940 film)', 'director', **'Walter Lang'**] **2.** ['Bharya Biddalu', 'director', **'Tatineni Rama Rao'**] **3.** [**'Walter Lang'**, 'country of citizenship', 'American'] **4.** [**'Tatineni Rama Rao'**, 'country of citizenship', 'India']
	Comparison_5_1 (CO-5-1)	Which film has more directors, Red Cow (Film) or Chillerama? **Chillerama**	**1.** ["Red Cow (film)", "director", "Tsivia Barkai Yacov"] **2.** ["Chillerama", "director", "Adam Rifkin"] **3.** ["Chillerama", "director", "Tim Sullivan"] **4.** ["Chillerama", "director", "Adam Green"] **5.** ["Chillerama", "director", "Joe Lynch"]
	Bridge_Comparison_5_1 (BC-5-1)	"Do both films The Falcon (Film) and Valentin The Good have the directors from the same country? **no**	**1.** ["The Falcon (film)", "director", **"Vatroslav Mimica"**] **2.** ["Valentin the Good", "director", **"Martin Frič10d"**] **3.** [**"Vatroslav Mimica"**, "country of citizenship", "Croatian"] **4.** [**"Vatroslav Mimica"**, "country of citizenship", "Yugoslavia"] **5.** [**"Martin Frič10d"**, "country of citizenship", "Czech"]

Table 1: This table shows the reasoning graphs of GRS-QA. The reasoning graphs demonstrate the decomposition of the larger question and the reasoning paths to approach the answer. Each of these is constructed using the context and relevant entities for each question. The decomposition is shown with varying formats in the right-most column of the graph, including more questions derived from the original question as well as triples that represent the relations between entities and, in turn, provide subsets of the context. This is consistent with the multiple datasets that each of the question types are extracted from.

For 2WikiMultiHopQA, the triplets in the "evidences" field provide sufficient information to establish the edges for our graph. We extract entities from the initial sentence to locate their corresponding sentences. Furthermore, the MuSiQue dataset includes an 'answerable' field that indicates whether the provided context contains the necessary sentences to address parts of the question.

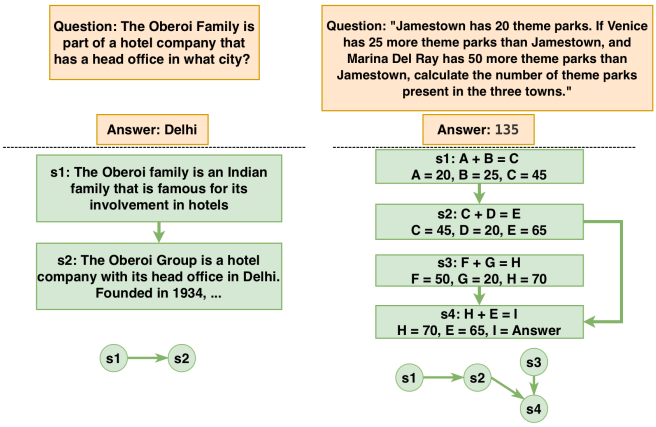


Figure 1: Example reasoning graphs constructed from HotpotQA (Yang et al., 2018) (left) and DARG-processed GSM8K (Zhang et al., 2024) (right), illustrating the logical steps required to derive the answer. For HotpotQA, each node corresponds to a supporting paragraph in the original dataset. For GSM8K, each supporting sub equation corresponds to a node. These reasoning graphs represent the *gold reasoning path* to answer the question.

If the "answerable" field is marked as False, we exclude that data point, as we cannot generate a reasoning graph with incomplete context.

Lastly, from the DARG processed GSM8K dataset we were able to extract a mid-step rendition of reasoning graphs, DARG reasoning graphs, and Width and Depth increased reasoning graphs which provide irrelevant extra node(s) for the graph that acts as a perturbation. However, unlike the rest of the datasets, GSM8K has too many variations in each hop type, therefore, the breakdown of GSM8K is limited to distribution of hops. Since GSM8K is a math domain dataset, we were able to use the intermediate equations that were given in the evidence field as the content of the nodes. The graph structure, edges, was created using the results of each of the intermediate steps. An edge is created from one node to the other if the result of a node is a part of the operands of another node. More information about graph construction and dataset processing can be found in A, B.

2.2. Negative Graphs

In addition to the positive reasoning graphs, GRS-QA introduces structurally perturbed variants, referred to as *negative reasoning graphs*, to assess model sensitivity to reasoning path integrity. Unlike traditional negative samples, which differ semantically, these perturbations alter only the graph structure, keeping sentence content largely unchanged. This design isolates the impact of structural coherence on model behavior. We implement two primary types of structural perturbation.

Edge Perturbation: Gold graph nodes are retained, but the edge connectivity is altered. This includes deleting edges, adding incorrect edges, or reversing edge directions, resulting in a misleading or incoherent reasoning flow.

Node Perturbation: Distraction sentences, sourced from a global context pool but irrelevant to the gold reasoning path, are inserted or swapped with gold nodes. These nodes are linked through new edges, disrupting the logical progression.

Graph Type	MuSiQue	Wiki	Hotpot	GSM8K
Bridge	1875	7652	7298	–
Compositional	121	–	–	–
Comparison	–	5200	1747	–
Bridge-Comparison	–	3448	–	–
Multi-hop (3–11 hops)	–	–	–	2796
Total	1996	16301	9045	2796

Table 2: Distribution of reasoning graph structures across datasets used for experiments. GSM8K has 3 to 11 hop questions with numerous edge variations that do not fit into the shown types. This distribution shows the breakdown of a subset of MuSiQue, Wiki, Hotpot for evaluation purposes. GSM8K is shown in its entirety. C provides more information about the statistical analysis

3. Experiments

To assess the challenges GRS-QA pose to state-of-the-art LLMs, we benchmark their performance from three core perspectives:

- **Effects of Structured Prompting on Performance:** How do differently structured prompting strategies, such as using reasoning graphs as examples or as context, affect the reasoning performance of language models?
- **Domain Complexity and Model Capacity Effects:** How do different language models perform across diverse domains, such as factual and mathematical question answering, and what does this reveal about their domain-dependent reasoning capabilities?
- **Robustness and Sensitivity:** How robust and consistent are language models when exposed to varied versus static few-shot demonstrations, in the presence of prompt noise, and to what extent are they sensitive to prompt variations?

3.1. Experimental Settings

Prompt Settings. The prompt used in this work consists of two components: *Context* and *Examples*. The context refers to the ground truth reasoning graph that can be used to derive the answer

Setting	Abbreviation	Description
• No Context + No Examples	NC + NE	The question is provided without any context (i.e., zero-shot)
• No Context + Reasoning Graph Examples	NC + Graph Ex	The question is provided with examples from other questions (i.e., few-shot), which consist of question-graph-answer triples
• Reasoning Graph Context + No Examples	Graph Context + NE	The question is provided with its graph-structured context, but without any examples from other questions
• Reasoning Graph Context + Reasoning Graph Examples	Graph Context + Graph Ex	The question is provided with its graph-structured context and examples from other questions (question-graph-answer triples)
• No Context + Randomized Reasoning Graph Examples	NC + Ran Graph Ex	The question is provided without context but with randomly selected examples, consisting of question-graph-answer triples

Table 3: Prompt settings and descriptions.

to a given question. The examples are reasoning graph instances included in the prompt to demonstrate how different questions can be approached. To evaluate the impact of different prompting structures, we experimented with various combinations of these settings, as listed in Table 3.

LLMs. The LLMs evaluated in this paper are: GPT4o-mini (OpenAI, 2024), Llama8b-instruct (Grattafiori et al., 2024) (8.03B), Qwen2 (Yang et al., 2024) (1.54B), and Phi3.5 (Abdin et al., 2024) (3.82B). These LLMs show how differences in model size and training methods can lead to variations in performance.

Evaluation Metrics. For evaluation, we use Exact Match (EM), F1, and an LLM-as-a-Judge metric (Zheng et al., 2023), where GPT-4o-mini serves as the judge. EM measures correctness by checking if the generated answer exactly matches the ground truth answer, while F1 offers a more lenient token-level comparison that rewards partial overlap between the predicted and ground truth answers.

The LLM Judge assesses the quality of semantics and reasoning capabilities beyond lexical similarity. We report primarily the F1 score in the main text, as it captures nuanced correctness and shows trends observed over all metrics. To reduce noise and ensure fair comparison, model outputs are normalized and truncated to include only the text after the keyword "Answer:", isolating the final predicted answer from any preceding reasoning steps.

3.2. Effects of Structured Prompting

Prompts can have varying impacts on reasoning performance depending on how they are structured. To examine how different prompt designs influence performance, we compare several prompting strategies, including reasoning graphs as examples, reasoning graphs used as context, and evidence-based examples. Table 3 summarizes the prompt settings used for evaluation in this section.

These prompt settings are applied across all the domains in the GRS-QA dataset, which includes Wiki, MuSiQue, Hotpot, and GSM8K. Table 2 shows the distribution of reasoning graph structures across subsets of data that were used to evaluate the models. Experimental results are reported separately for each dataset, leading to the following key observations.

LLMs struggle to fully utilize the provided graph context. We first compare two settings: NC + Graph Ex and Graph Context + NE. This comparison reveals how well LLMs perform when given relevant context for a question versus when they are provided reasoning graph-based guidelines for solving the question within the prompt. The first two points on the x-axis of Figure 2 correspond to these two settings.

We see that there is a common trend across Wiki, Hotpot, and GSM8K, where NC + Graph Ex considerably outperforms Graph Context + NE across LLama, GPT4o, and Phi3.5. Phi3.5 has some fluctuation on Hotpot, and this variation is further amplified in Qwen2's performance. Qwen2 performs better for Hotpot, Wiki, and MuSiQue, for the Graph Context + No Ex setting.

This shows that even when LLMs are given the relevant context to a question, they may not know how to approach or effectively use it to their advantage, and providing guidelines or demonstrations on how to approach a question can be more useful. Overall, Hotpot, Wiki, and GSM8K yield better results across all LLMs (except Hotpot with Phi3.5) in the NC + Graph Ex setting, whereas MuSiQue is an outlier, showing higher performance in the Graph Context + No Ex setting.

LLMs understand relevant context better when it is paired with examples of how they are used in the prompt. Based on the previous experiments, we present an additional setting: Graph Context + Graph Ex. This experiment shows that combining the two previous settings, NC + Graph Ex and Graph Context + NE, and giving relevant context and guidance to LLMs improves performance in most cases as shown in Figure 2. This shows that LLMs capable of effectively understanding Graph Examples independently tend to perform even better when the Graph Context and Graph Examples are combined into one setting, as shown in Figure 2d. The rest of Figure 2 shows the same trend for Llama and 4o-mini.

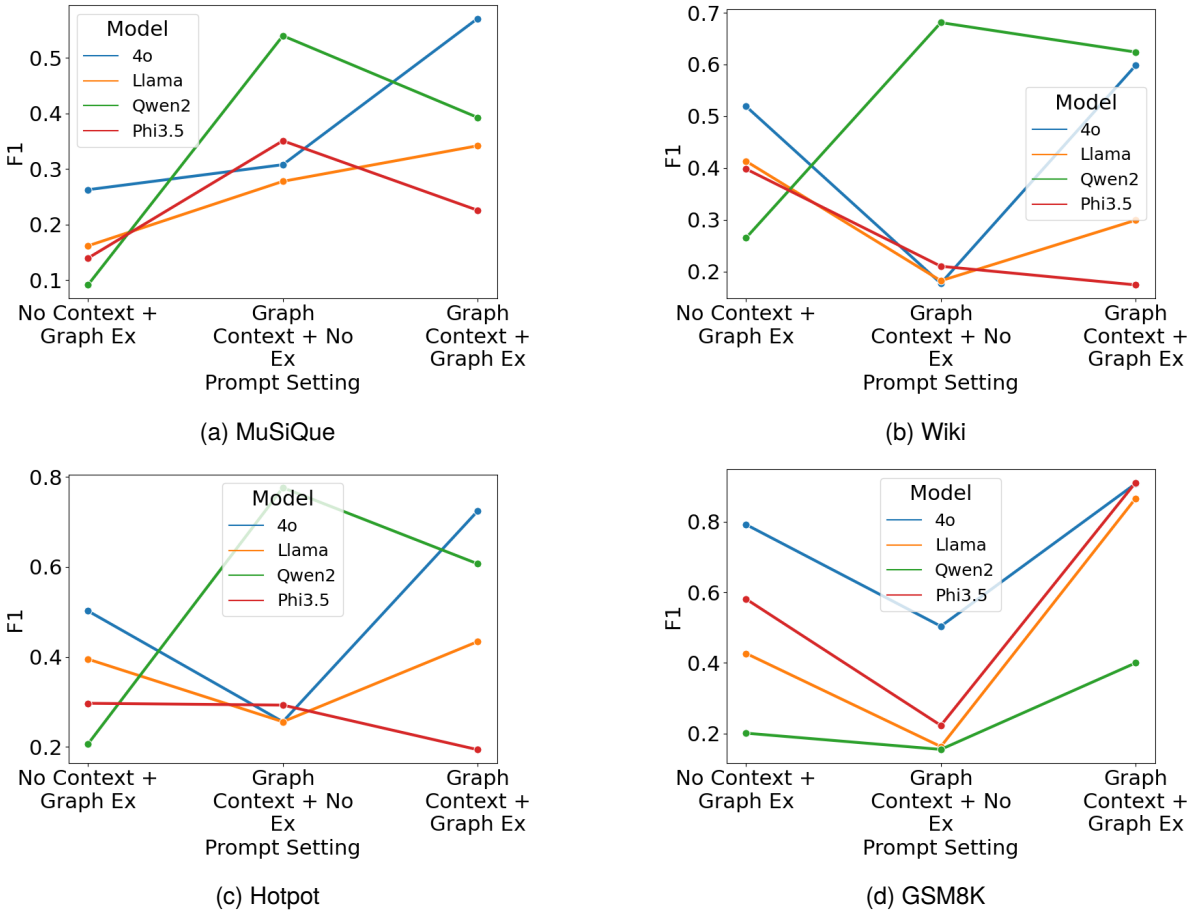


Figure 2: Performance of LLMs across data domains in GRS-QA under different prompt settings: (1) No Context + Graph Examples, (2) Graph Context + No Examples, and (3) Graph Context + Graph Examples.

Context Setting	GPT-4o-mini			LLaMA			Qwen2			Phi-3.5		
	EM	F1	LLM	EM	F1	LLM	EM	F1	LLM	EM	F1	LLM
Evidence Context	0.479	0.479	0.481	0.244	0.244	0.246	0.186	0.191	0.198	0.174	0.276	0.374
Reasoning Graph Context	0.504	0.504	0.505	0.161	0.163	0.166	0.151	0.155	0.162	0.179	0.223	0.259
No context + Evidence Examples in Prompt	0.791	0.791	0.791	0.449	0.449	0.449	0.243	0.243	0.244	0.591	0.591	0.593
No context + Reasoning Graph Examples	0.792	0.792	0.791	0.427	0.427	0.428	0.201	0.201	0.202	0.581	0.581	0.582
Evidence Context + Evidence Examples in Prompt	0.919	0.919	0.920	0.845	0.845	0.845	0.278	0.316	0.399	0.886	0.886	0.887
Reasoning Graph Context + Reasoning Graphs Examples	0.908	0.908	0.909	0.866	0.866	0.870	0.385	0.400	0.415	0.906	0.911	0.923

Table 4: Performance across different prompt settings for GSM8K for each model, evaluated using Exact Match (EM), F1, and LLM Judge (GPT-4o-mini).

However, we also see that the two smallest models, Phi3.5 and Qwen2, perform the worst with the Graph Context + Graph Ex setting on Wiki, Hotpot, and MuSiQue. This indicates that reasoning graph examples may be too complicated for the smaller models to process and understand.

LLMs struggle to understand reasoning graph structures. In addition to the settings shown in Table 3, we evaluate the same settings with evidence context. Evidence context has the same information that the reasoning graph context has except it is just unstructured plain text. The trends we observe is that the performances of reasoning graphs and evidence context are on par with each other in most cases, with some LLMs preferring evidence context over reasoning graph context. These trends are shown in Table 4. These results show that in many cases the reasoning graphs still help the LLMs perform better by providing a better breakdown of the information necessary to arrive to the correct answer. However, many LLMs remain unequipped to handle reasoning graphs without the aid of prompts or additional fine-tuning, highlighting the need to enhance their ability to learn from and

reason over graph-structured information.

3.3. Domain Complexity and Model Capacity Effects

To address the research question, "How do language models perform across diverse domains?", we evaluate language models on both textual and mathematical QA tasks to analyze domain-dependent reasoning capabilities, which leads to the following key observation.

Smaller LLMs struggle to understand the Reasoning Graph Structures provided for textual domain. Overall, we see in Figure 2d that GPT4o-mini outperforms all other LLMs in the mathematical domain across all prompting styles. Similarly, GPT4o-mini also outperforms LLama and Phi3.5 in the NC + Graph Examples setting for all data domains. However, Qwen2 and Phi3.5 outperforms the other LLMs in the Graph Context + No Examples setting as illustrated in Figures 2a to 2c.

Looking at the performance of each LLM, we can conclude that smaller LLMs like Phi3.5 and Qwen2 have a harder time understanding the reasoning graph structures given in the prompts for textual domains. For MuSiQue and Hotpot, Phi3.5 and Qwen2 have the same pattern of performing better for Graph Context + No Ex Prompt setting compared to the No Context + Graph Ex setting. This is also the case for Wiki with Qwen2. We can see that while ***Graph Context*** is helpful for the smaller LLMs, the ***Graph Examples*** are not as easily understood in textual reasoning settings. Therefore, we can assume smaller LLMs have a harder time replicating the reasoning graph structures for the given question when facing textual data domains like Wiki, Hotpot, and MuSiQue.

These results show that across all LLMs, GSM8K gains the most out of the reasoning graph structures. GSM8K is the dataset in which all models have consistent performance (Figure 2d). This indicates that GSM8K consists of the least complex reasoning paths compared to the textual dataset domains such as MuSiQue. The mathematical nature of GSM8K likely contributes to the consistent performance in all models, while the more linguistically complex content of textual domains demands more reasoning capabilities.

In addition, we observe that in the textual datasets, MuSiQue is an outlier, indicating that it is the most complex and difficult to comprehend for most LLMs. Lastly, Phi3.5 and Qwen2's performance across the textual domains deviates from the other LLMs, which suggests potential limitations of their modeling capabilities. Smaller Models, such as Phi3.5 and Qwen2, may benefit more from the ground truth relevant graph context and have a harder time understanding longer prompts that have too many examples.

3.4. Robustness and Sensitivity

To explore how consistent language models are when exposed to varied versus static few-shot demonstrations of reasoning graphs, we analyze the settings, NC + Ran Graph Ex and NC + Graph Ex, revealing how stable or sensitive each model is to prompt variation.

3.4.1. Effect of Graph Example Choice

LLMs exhibit higher sensitivity to variations in reasoning graph examples for multi-hop textual QA tasks, while showing greater stability in the mathematical domain. As shown in Figure 3d, we see that the mathematical tasks (GSM8K) perform slightly better with the randomized set of three examples in the prompt across all LLMs, while the performance of Wiki (Figure 3b), Hotpot (Figure 3c), and MuSiQue (Figure 3a) is greatly reduced when random examples are given in the prompt across all LLMs.

It is understandable that GSM8k's performance is not impacted by the randomized example set, which consists only of GSM8K examples in the prompt, since GSM8K is a mathematical and procedural reasoning task where both the necessary steps to perform and the information needed to derive an answer are part of the question itself. Therefore, even if the examples change in length and variety, it does not affect the performance as much as the steps to derive the reasoning path stays the same across all the questions in GSM8K.

However, the rest of GRS-QA mainly consists of textual question answer pairs. As a result, the randomized example set pool is made up of a combination of Wiki, MuSiQue, and Hotpot. Therefore, although the datasets might be similar at their core, there are still a variety of graph structures, question types, and entity relations that could end up confusing the model more if the randomization favors one dataset over the others. To summarize, reasoning graph diversity aids procedural reasoning tasks (e.g., math) where reasoning patterns are abstract and generalizable, but can hinder knowledge-based reasoning tasks where factual grounding and relational consistency are crucial.

3.4.2. Effect of Negatively Perturbed Graphs

We further test the robustness and sensitivity of a language model by incorporating negative graphs into the settings. Specifically, we introduce three new types of graphs: Darg Graph, Increased Width 1 Graph, Increased Depth 1 Graph. Darg Graph Context for GSM8K basically has the same information as the Reasoning Graph in GRS-QA, but

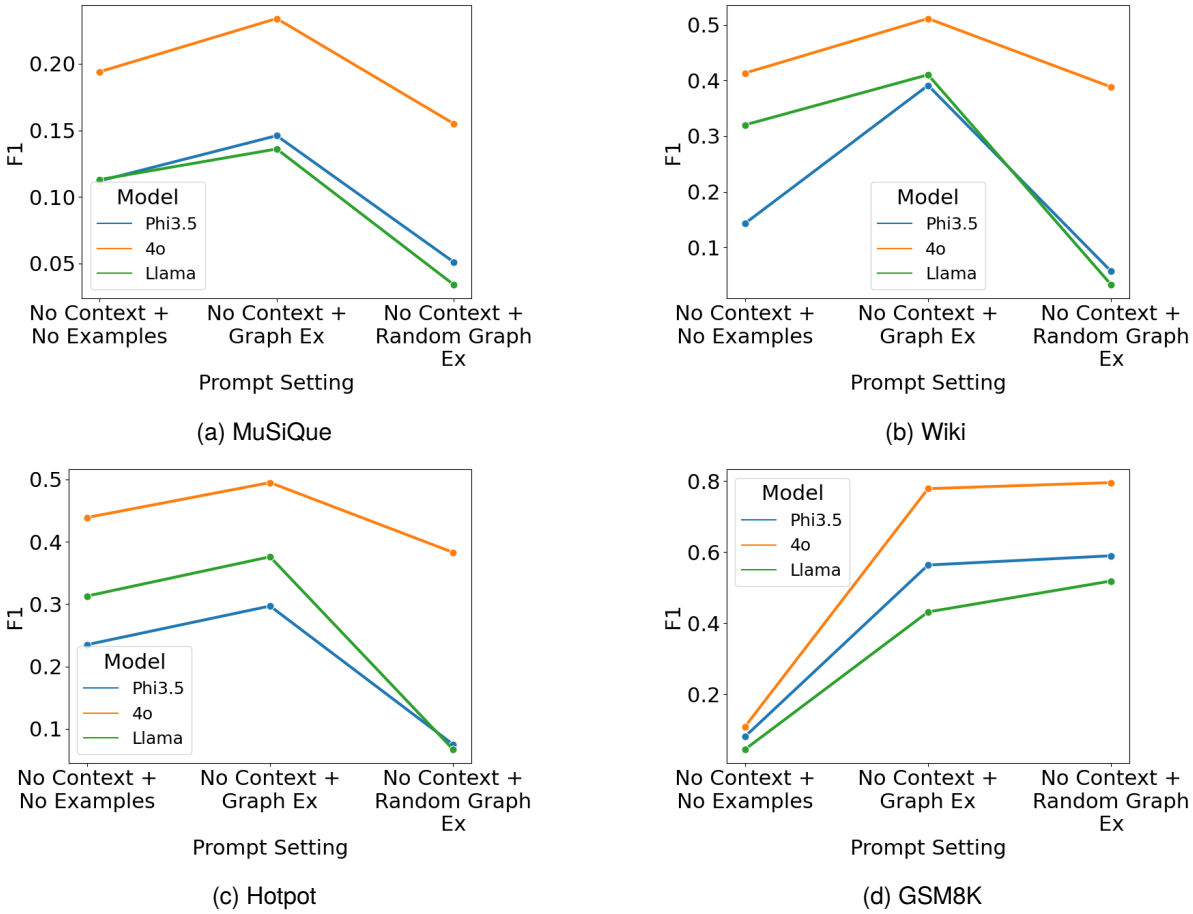


Figure 3: Performance of LLMs across data domains in GRS-QA under different prompt settings: (1) No Context + No Examples, (2) No Context + Same Set of Graph Examples, (3) No Context + Random Set of Graph Examples. This set of examples are evaluated on a smaller subset of data points from Table 2.

it is presented in a more expanded notation of nodes and edges compared to GRS-QA. Increased Width 1 and Depth 1 graphs are perturbations of the graphs and they are also referred to as negative graphs. In the case of Width 1 and Depth 1 increased graphs ((Zhang et al., 2024)), the perturbation adds extra nodes to the reasoning graphs that add an extra step to the derivation of the answer, which is not present in the question itself.

We observe that the GRS-QA reasoning graphs outperform DARG Graph Context in Table 5. In addition, we also see that negative graphs of GSM8K also impact the performance for the worse. Therefore, we can conclude that the correctness and relevance of nodes and edges in reasoning graphs are crucial for guiding multi-hop reasoning. Furthermore, LLMs are sensitive to extraneous steps, which can mislead the reasoning process even if the additional nodes are logically coherent but irrelevant to the question.

4. Related Work

GRS-QA draws on foundational insights from prominent multi-hop QA datasets, such as HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), 2WikiMultiHopQA (Ho et al., 2020), and GSM8K (Zhang et al., 2024; Cobbe et al., 2021). Several prior works have introduced structured reasoning representations as well. WorldTree (Jansen et al., 2018) constructs explanation graphs for elementary science questions. QA-GNN (Yasunaga et al., 2021) integrates joint reasoning graphs over textual contexts and knowledge graphs for commonsense reasoning. Reason2Drive (Nie et al., 2023) provides structured reasoning chains tailored to autonomous driving perception tasks. These efforts demonstrate the value of graph-based or structured explanations for interpretability and reasoning control. However, they are largely domain-specific and do not systematically annotates reasoning structures across multiple established multi-hop QA benchmarks of varying complexity.

Advancements in retrieval and reasoning also inform GRS-QA. SURE summarizes retrieved pas-

Context Setting	EM	F1
Reasoning Graph Context + No Ex	**0.504**	**0.504**
DARG Graph Context + No Ex	0.244	0.254
Increased Width 1 Reasoning Graph Context + No Ex	0.425	0.425
Increased Depth 1 Reasoning Graph Context + No Ex	0.422	0.422
Increased Width 1 DARG Graph Context + No Ex	0.190	0.195
Increased Depth 1 DARG Graph Context + No Ex	0.180	0.184
No context + Reasoning Graph Examples + No Ex	**0.792**	**0.792**
No context + DARG Reasoning Graph Examples	0.685	0.685

Table 5: Performance of GPT-4o-mini across variations in graph context and reasoning graph examples, evaluated with Exact Match (EM) and F1.

sages (Kim et al., 2024), dense retrieval with dual encoders is explored in (Karpukhin et al., 2020), and ORQA retrieves evidence via question-answer pairs (Lee et al., 2019), while generative models aid passage retrieval (Izacard and Grave, 2021). In KGQA, (Wu et al., 2023) rewrites KG knowledge as text, and MRPQA enhances answer prediction with minimal labeled data (Wang et al., 2022).

While these approaches improve question answering through retrieval augmentation, structural guidance, or model tuning, *GRS-QA focuses on evaluating reasoning robustness by embedding structured reasoning pathways directly into the dataset*. In contrast to prior structured reasoning efforts that focus on specific domains or model architectures, GRS-QA unifies and extends graph-style reasoning annotations across multiple existing multi-hop QA benchmarks. By embedding structured reasoning pathways directly into the dataset, it enables controlled and fine-grained evaluation of reasoning robustness across domains and reasoning complexities. This unified, cross-benchmark design distinguishes GRS-QA within the broader structured-reasoning QA literature.

5. Conclusion

In this paper, we introduce GRS-QA, which captures different types of reasoning pathways necessary for multi-hop and multi-step question answering. In GRS-QA, we map out logical steps to reach the answer, and construct the reasoning graph for each question. This structured representation provides a new lens for analyzing how LLMs handle complex multi-hop reasoning and offers a fine-grained evaluation of the reasoning steps. Through our analysis of the reasoning structures in GRS-QA, we found that while current QA models perform well on many general tasks, they often struggle with questions that exhibit complex reasoning structures, especially those requiring multi-step inference. This highlights the need for improvements in models' ability to process and reason through intricate logical pathways.

Limitations. GRS-QA's primary limitation is the imbalance in graph types, with simpler structures like bridge_2_1 (60.54%) dominating, while complex types like bridge_comparison_5_1 are less frequent. This may bias models toward easier patterns, affecting their ability to generalize. The dataset's broad domain range, though enhancing generalization, poses challenges in domain-specific reasoning, as questions span diverse topics without clear boundaries. Additionally, the complexity of multi-hop reasoning graphs, while a strength, presents challenges for current LLMs in handling intricate, multi-step reasoning.

Future Work. To address the imbalance in graph types, future work could focus on generating synthetic data to better represent complex structures. Additionally, segmenting the dataset by domain (e.g., historical, scientific) could lead to domain-adapted models, improving QA performance in specialized contexts. Expanding negative graph varieties could also offer deeper insights into how different graph structures impact model performance. Also, testing diverse model architectures, such as graph neural networks (GNNs) and retrieval-augmented models, could reveal the most effective approaches for handling graph structures.

An important open question raised by our findings is why LLMs benefit from reasoning graph examples yet still struggle to understand the underlying structure of the reasoning graphs. Future work should investigate whether models are genuinely learning to follow graph connectivity — traversing logical dependencies between nodes — or simply imitating surface-level patterns in the examples. Probing studies and attention analyses could help disentangle these two behaviors, shedding light on the degree to which current LLMs are capable of true structure-aware reasoning versus shallow pattern matching. Understanding this distinction is critical for designing prompting strategies and architectures that more faithfully leverage explicit reasoning structures.

6. Ethics Statement

The GRS-QA dataset promotes interpretability in language models by explicitly providing reasoning pathways in a structured and concise form. These reasoning graphs can serve as training material

to help models develop more structured, step-by-step reasoning capabilities, making their thought process more understandable and transparent to users.

7. Acknowledgements

We thank Prof. Ian Lane (University of California, Santa Cruz) for his leadership as Program Director of the Natural Language Processing Program at UCSC (2023-2025). We also thank Neng Wan for insightful discussions during the early stages of this work. Khin Saw Yone was supported in part by the QUAD Fellowship.

8. Bibliographical References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone.

Farah Atif, Ola El Khatib, and Djellel Difallah. 2023. Beamqa: Multi-hop knowledge graph question answering with sequence-to-sequence prediction and beam search.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti,

Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby,

Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Summarizing retrievals using answer candidates for open-domain qa of llms.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering.

OpenAI. 2024. Gpt-4o-mini: Advancing cost-efficient intelligence.

Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2024. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19206–19214.

Yu Wang, V.srinivasan@samsung.com V.srinivasan@samsung.com, and Hongxia Jin. 2022. A new concept of knowledge based question answering (KBQA) system for multi-hop reasoning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4007–4017, Seattle, United States. Association for Computational Linguistics.

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

Yike Wu, Nan Hu, Sheng Bi, Guilin Qi, Jie Ren, Anhuan Xie, and Wei Song. 2023. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering.

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

Zhehao Zhang, Jiaao Chen, and Diyi Yang. 2024. Darg: Dynamic evaluation of large language models via adaptive reasoning graph.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

9. Language Resource References

Karl Cobbe and Vineet Kosaraju and Mohammad Bavarian and Mark Chen and Heewoo Jun and Lukasz Kaiser and Matthias Plappert and Jerry Tworek and Jacob Hilton and Reiichiro Nakano and Christopher Hesse and John Schulman. 2021. *Training Verifiers to Solve Math Word Problems*.

Ho, Xanh and Duong Nguyen, Anh-Khoa and Sugawara, Saku and Aizawa, Akiko. 2020. *Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps*. International Committee on Computational Linguistics.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018. WorldTree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. 2023. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving.

Trivedi, Harsh and Balasubramanian, Niranjan and Khot, Tushar and Sabharwal, Ashish. 2022. *MuSiQue: Multihop Questions via Single-hop Question Composition*. MIT Press One Broadway, 12th Floor, Cambridge, Massachusetts 02142, USA ....

Yang, Zhilin and Qi, Peng and Zhang, Saizheng and Bengio, Yoshua and Cohen, William W and Salakhutdinov, Ruslan and Manning, Christopher D. 2018. *HotpotQA: A dataset for diverse, explainable multi-hop question answering*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Zhehao Zhang and Jiaao Chen and Diyi Yang. 2024. *DARG: Dynamic Evaluation of Large Language Models via Adaptive Reasoning Graph*.

10. Appendix

A. Illustrative Graph Construction Example

To concretely demonstrate how decompositions correspond to reasoning graphs, we present a representative example from the `bridge_3_1` category. Consider the following question:

```
\textit{"In which country is the
administrative territorial entity for
the city where Charlie Harper was born?"}
```

Its stepwise decomposition is:

1. Where was Charlie Harper born? → Hackney
2. In which administrative territorial entity is Hackney located? → Middlesex
3. Which country is Middlesex located in? → United Kingdom

Each sentence relevant to these steps is treated as a node in the reasoning graph. Edges are added between nodes based on logical progression: Step 1 → Step 2 → Step 3, forming a linear directed path. This sequence reflects the stepwise reasoning needed to arrive at the final answer.

Another example that has a different decomposition is as below:

```
"Are Captain Robert S. Craig
Cottage and Wiichen Men'S Meetinghouse
located in the same country?"
```

Its stepwise decomposition is:

1. `['Captain Robert S. Craig Cottage', 0]` → `['Captain Robert S. Craig Cottage', 'country', 'United States']`
2. `["Wiichen Men's Meetinghouse", 3]` → `["Wiichen Men's Meetinghouse", 'country', 'United States']`
3. `["Wiichen Men's Meetinghouse", 0]` → `["Wiichen Men's Meetinghouse", 'country', 'Federated States of Micronesia']`

Each entity has an accompanying relation that contains the context needed to answer the question. Each of the relations are treated as nodes. However, unlike Bridge-3-1, there is no linear path since these are not directly related to each other, rather they are compared to derived the final answer. Therefore, there are no edges for this graph.

B. GRS-QA Dataset Construction

B.1. HotPotQA Processing

For each data item, nodes and edges are constructed, with nodes generated from positive paragraphs and characterized by attributes such as evidence and type. In cases where the data item is classified as a "bridge" type, specific handling is applied to determine the presence of a "bridge" in the positive paragraphs, influencing edge connections. This results in a primary graph structure containing nodes and edges. The process also generates 5 sets of negative examples, where each set consists of two graphs: one with an additional node and another with altered edge connections, creating challenging negative samples. Additionally, another

five sets of negative examples are created by randomly selecting nodes from negative paragraphs and constructing corresponding edges. The final output comprises a list of processed data items, including the original question, concatenated positive evidence, correct answers, the primary positive graph, and sets of negative graphs.

B.2. MuSiQue Processing

For each data item, nodes and edges are constructed, with nodes generated from the answers of the question decomposition and edges created by using attributes such as the type. The process also generates five sets of negative examples, where each set consists of two graphs: one with an additional node and another with altered edge connections, creating challenging negative samples. Additionally, another five sets of negative examples are created by randomly selecting nodes from negative paragraphs and constructing corresponding edges. The final output comprises a list of processed data items, including the original question, concatenated positive evidence, correct answers, the primary positive graph, and sets of hard and easy negative graphs.

B.3. 2WikiMultiHopQA Processing

To construct 2WikiMultiHopQA reasoning graphs, we extracted four primary graph types—Inference, Comparison, Bridge Comparison, and Compositional—from the raw dataset, varying node counts to generate additional structures. Gold graphs were built using supporting facts and evidences, with nodes representing supporting facts and edges created based on relations between entities shown in the evidences. Five sets of negative graphs were created by using nodes from the context to add an extra node. Another five sets of negative graphs were created by randomly selecting and constructing the corresponding edges. The final output comprises a list of processed data items, including the original question, concatenated positive evidence, correct answers, the primary positive graph, and sets of negative graphs.

Due to entity string inconsistencies in the data for 2WikiMultiHopQA train set, there were 4507 instances of the data that were discarded since they were unable to be used to create positive graphs.

B.4. GSM8K Data Processing

Lastly, from the DARG-processed GSM8K dataset, we construct several variants of reasoning graphs, including (i) positive reasoning graphs and (ii) perturbed graphs with increased width and depth. The latter introduces additional, irrelevant node(s) that act as controlled perturbations to the original reasoning structure based on DARG. The positive graphs are derived directly from the structured intermediate reasoning steps provided in GSM8K. Each problem instance contains a question equation, a sequence of intermediate equations (with answers), and a final answer. We treat each intermediate equation as a node in the reasoning graph. The value computed at each step serves as the semantic output of the node and determines the graph connectivity. Specifically, we create a directed edge between two nodes when the numerical result of one intermediate equation is used as an operand in a subsequent equation. In this way, the graph encodes the functional dependencies between intermediate computations, yielding a structured representation of the solution trajectory. Using the DARG framework, we then generate perturbations of these positive graphs to evaluate reasoning robustness. Width-based perturbations introduce additional nodes at a given reasoning depth that are syntactically valid but semantically irrelevant to the final answer. Depth-based perturbations extend the graph with extraneous reasoning steps that do not alter the correct solution path but increase the structural complexity of the graph. These perturbations preserve the original question and correct answer while modifying the reasoning steps it takes to arrive to the answer.

B.5. Licensing

Our use of the following datasets is consistent with their licenses, specifically:

- 2WikiMultiHopQA: Apache-2.0 License https://github.com/Alab-NII/2wikimultihop?tab=Apache-2.0-1-ov-file
- HotpotQA: CC BY-SA 4.0 License https://hotpotqa.github.io/
- MuSiQue: CC BY 4.0 License https://github.com/stonybrooknlp/musique
- GSM8K: MIT License https://github.com/openai/grade-school-math/blob/master/LICENSE
- DARG: ODC Attribution License (ODC-By) https://salt-nlp.github.io/DARG_website/

Our dataset is released under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

C. Statistical Analysis – Breakdown of each dataset

Dataset Distribution The constructed reasoning graphs in GRS-QA span four primary reasoning categories: comparison, bridge, compositional, and

Question Type	Train	Val	Test
B-2-1	58384	7298	7298
C-2-1	13964	1745	1747
total	**72348**	**9043**	**9045**

Table 6: Breakdown of Question Types and Unique Question Count for HotpotQA

Question Type	Train	Val	Test
B-2-1	61209	7651	7652
C-2-1	41324	5165	5167
C-3-1	234	29	30
C-4-1	10	1	2
C-5-1	-	-	1
C-3-2	3	-	1
BC-4-1	27266	3408	3409
BC-5-1	308	38	39
total	**130354**	**16292**	**16301**

Table 7: Breakdown of Question Types and Unique Question Count for 2WikiMultiHopQA

bridge-comparison, yielding a total of 12 distinct graph structures. (Graph Types for GSM8k is not included in this since there are too many distinct types.)

Table 1 summarizes the characteristics of each graph type, including representative example questions, corresponding decompositions, and structural properties.

Figure 4 presents key dataset statistics: (a) distribution of question types across splits, (b) average number of nodes and edges per graph, and (c) average token count per graph structure.

We observe that bridge and comparison graphs constitute the majority of the dataset, reflecting both their dominance in the source datasets and their relevance to common reasoning patterns.

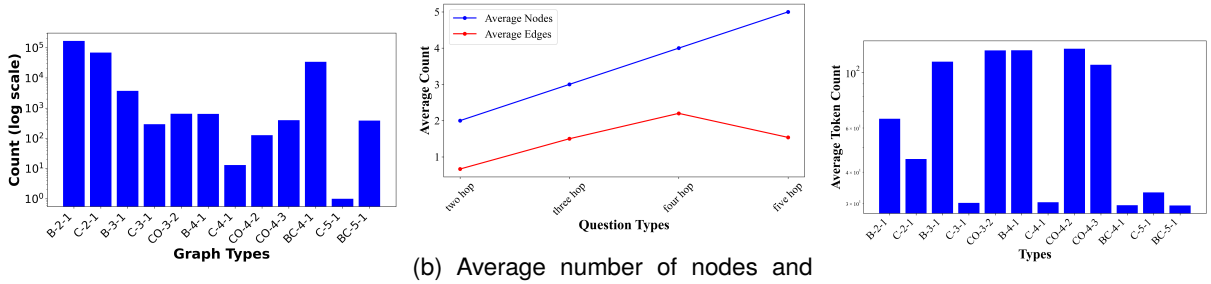
Interestingly, question types with simpler graph structures tend to include fewer but longer sentences, leading to higher average token counts, as seen in Figure 4. This indicates a trade-off between graph complexity and sentence granularity, with implications for model performance and reasoning analysis.

Question Type	Train	Val	Test
B-2-1	11478	1434	1436
B-3-1	2987	373	374
CO-3-2	519	64	66
B-4-1	516	64	65
CO-4-2	101	12	14
CO-4-3	319	39	41
total	**15920**	**1986**	**1996**

Table 8: Breakdown of Question Types and Unique Question Count for MuSiQue

Graph Type	Total Instances	Unique Edges
3-hop	848	6
4-hop	882	34
5-hop	565	96
6-hop	263	155
7-hop	150	117
8-hop	59	47
9-hop	20	20
10-hop	7	7
11-hop	2	2
Total	**2796**	**484**

Table 9: Hop-based distribution of graph types in the GSM8K dataset.



(a) Number of Questions by Graph types in all dataset splits

(b) Average number of nodes and edges in each question type Positive Graphs

(c) Average number of tokens in each question type's Positive Graphs

Figure 4: Statistical Analysis of the Distribution of GRS-QA. (Wiki, Hotpot, Musique)