Enhancing Logical Reasoning in Large Language Models through Graph-based Synthetic Data

Jiaming Zhou[1♠] **Abbas Ghaddar**[1] **Ge Zhang**[1] **Liheng Ma**[2] **Yaochen Hu**[1]
Soumyasundar Pal[1] **Mark Coates**[2] **Bin Wang**[3] **Yingxue Zhang**[1♠] **Jianye Hao**[3]
[1] Huawei Noah's Ark Lab, Montréal, Canada
[2] McGill University and Mila - Québec AI Institute
[3] Huawei Noah's Ark Lab, Beijing, China
jiaming.zhou@h-partners.com, yingxue.zhang@huawei.com

Abstract

Despite recent advances in training and prompting strategies for Large Language Models (LLMs), these models continue to face challenges with complex logical reasoning tasks that involve long reasoning chains. In this work, we explore the potential and limitations of using graph-based synthetic reasoning data as training signals to enhance LLMs' reasoning capabilities. Our extensive experiments, conducted on two established natural language reasoning tasks—inductive reasoning and spatial reasoning—demonstrate that supervised fine-tuning (SFT) with synthetic graph-based reasoning data effectively enhances LLMs' reasoning performance, without compromising their effectiveness on other standard evaluation benchmarks.

1 Introduction

The reasoning capabilities of Large Language Models (LLMs) Touvron et al. (2023); Jiang et al. (2023); Dubey et al. (2024) can be greatly enhanced by post-training techniques Ouyang et al. (2022); Zhang et al. (2023b) and prompting strategies Wei et al. (2022b); Yao et al. (2023); Madaan et al. (2023). However, even with the aforementioned techniques, the multi-hop reasoning tasks remain challenging Touvron et al. (2023); Jiang et al. (2023); Dubey et al. (2024): LLMs struggle to reason over steps Agrawal et al. (2024); Zhao & Zhang (2024), and are fragile to minor perturbations Ullman (2023); Chen et al. (2024) in the input prompt.

Recently, several works Xu et al. (2024); Abdin et al. (2024); Anil et al. (2023) have demonstrated the efficacy of boosting the LLMs' reasoning capacity via fine-tuning on synthetic data generated by stronger LLMs. However, how to make such synthetic data generation effective and controllable for specific applications remains an open question. Extensive prompt engineering and quality filtering are required to guide LLMs' generation, yet the quality of generated reasoning questions and their labels remains uncertain Gudibande et al. (2023); Wang et al. (2023); Tan et al. (2024).

Motivated by the fact that natural language reasoning tasks can be represented as structured data with finite nodes and edges Jin et al. (2024), and inspired by existing works on constructing reasoning benchmarks Fatemi et al. (2024); Kazemi et al. (2023b); Agrawal et al. (2024), we propose to leverage synthetic graph-based data for task-specific post-training adaptation to improve the correctness of the generated reasoning questions and labels.

In this paper, we carefully design a random walk sampling algorithm on graphs and introduce a new prompting strategy that first extracts a reasoning chain and then derives the answer. These processes complement each other to enable efficient, task-specific adaptation of LLMs for reasoning tasks. Extensive experiments on two well-established benchmarks for inductive and spatial reasoning—*CLUTRR* Sinha et al. (2019) and *StepGame* Shi et al. (2022)—demonstrate that our framework leads to significant performance gains compared to standard prompting and training methods. Our findings suggest that when carefully

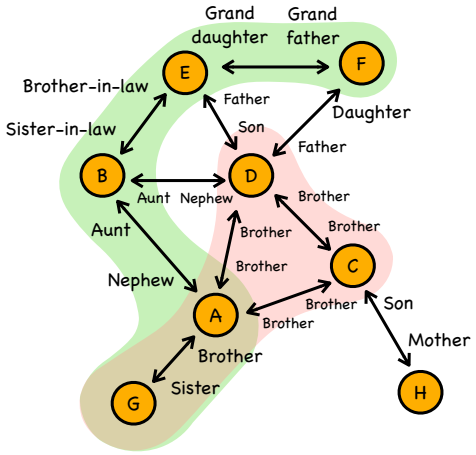


Figure 1: Illustration of a kinship graph highlighting a reasoning chain sampled by our algorithm (green) for LLM adaptation, and an ignored simpler chain (red).

curated, graph-based structured data can effectively enhance LLMs' reasoning abilities on the targeted task while maintaining its performance on standard evaluation benchmarks.

2 Methodology

Reasoning tasks in natural language often involve a structured representation of facts that can be abstracted into a graph following predefined rules Ji et al. (2022). For example, family relationships can be systematically represented in a logical graph, where nodes denote family members and edges define their relationships, all governed by logical rules. Let $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathcal{R})$ represent a relational graph with a set of nodes $\mathcal{V}$, a set of edges $\mathcal{E}$ between the nodes, and a set of relations $\mathcal{R}$ expressed in first-order logic. In this graph, vertices (e.g., family members) are denoted as $v_i \in \mathcal{V}$, and directed edges (e.g., familial relationships) are represented as $(v_i, r_{i,j}, v_j) \in \mathcal{E}$, with relations $r_{i,j} \in \mathcal{R}$. Our goal is to generate synthetic examples of graph-structured data from such relation graphs to adapt LLMs for targeted reasoning tasks. Existing graph-based data generation methods, such as Sinha et al. (2019), may produce large amounts of data, but this data is frequently redundant and lacks the necessary complexity (see Figure 1). Therefore, we propose a random-walk-based algorithm Lovász (1993) that produces a manageable yet diverse set of examples by sampling sub-graphs from $\mathcal{G}$. In the remainder of this section, we describe our algorithm for constructing $\mathcal{G}$ in § 2.1, the process of generating synthetic reasoning data in § 2.2, and their deployment to enhance LLM capabilities in natural language reasoning tasks in § 2.3.

2.1 Relational Graph Construction

The relational graph $\mathcal{G}$ is built iteratively by adding new nodes connected to existing nodes via basic relations. We start with an initial graph $\mathcal{G}_0 = (\{v_0\}, \emptyset, \mathcal{R})$, where v_0 is a randomly sampled root node. In each successive iteration l, we grow the graph by a) searching for absent relations between the nodes in $\mathcal{G}_{l-1}$, and b) adding new nodes with those relations, if such absent relations are found in step a). Specifically, for every node v in $\mathcal{G}_{l-1}$ and each relation $r \in \mathcal{R}$, we check if there is another node v' in $\mathcal{G}_{l-1}$ such that there is an edge between v and v' with relation r. If no such node exists in $\mathcal{G}_{l-1}$, we create a new node v_r and connect it to node v in $\mathcal{G}_{l-1}$ with relation r. Then, a deduction function f is used to evaluate the relation between v_r and each other node v' in $\mathcal{G}_{l-1}$, except v. Edges (v', r', v_r)-s are added to $\mathcal{G}_{l-1}$ if the deduction function f computes a relation $r' \in \mathcal{R}$ between v' and v_r. This process is repeated once for each node in $\mathcal{G}_{l-1}$ within the l-th iteration. When this is complete, we assign the expanded $\mathcal{G}_{l-1}$ to $\mathcal{G}_l$ and proceed to the next iteration. We terminate this procedure after L iterations and take $\mathcal{G}_L$ as the relational graph $\mathcal{G}$.

Algorithm 1 Reasoning Chain Sampling

Input: Graph $\mathcal{G}$, walk length l , transition probabilities π

Sample a starting vertex $v_0 \sim \text{Uniform}(\mathcal{V})$
Initialize $walk \leftarrow [v_0]$
for $i = 0$ to $l - 1$ **do**
  $j \leftarrow i$
  while $v_j \in walk$ **do**
    Sample $j \sim Categorical([\pi_{i,1}, \cdots \pi_{i,n}])$
    if $v_j \notin walk$ **then**
      $v_{i+1} \leftarrow v_j$
      Append v_{i+1} to $walk$
      break
    end if
  end while
end for
Initialize Reasoning chain $c \leftarrow [\,]$
for $i = 0$ to $l - 1$ **do**
  Retrieve and append $(v_i, r_{i,i+1}, v_{i+1}) \in \mathcal{E}$ to c
end for

Return: Reasoning chain c

2.2 Sub-Graphs Sampling

Given a relational graph $\mathcal{G}$ with $|\mathcal{V}|{=}n$ nodes, the desired walking length (e.g., number of hops) $l{\in}\mathbb{Z}^+$, and isotropic random walk probabilities $\pi{\in}\{0,1\}^{n\times n}$ ($\pi_{i,j}$ denotes the probability of transition to v_j from v_i in one step of the random walk), we construct a reasoning chain c by conducting a random walk of length l, which starts at a random node $v_0{\in}\mathcal{V}$ and avoids any repetition of visited nodes (to avoid circular reasoning in the data). The overall procedure is summarized in Algorithm 1.

For each item c in the generated set of reasoning chains $\mathcal{C}$, we apply one of the following augmentation techniques to further introduce diversity and additional complexity: **Permutation**, where we apply a permutation function σ to reorder the triples in c; **Edge Noise**, which involves introducing noise by adding edges that connect nodes not initially in the chain, specifically, for a vertex $v_i \in \mathcal{V}'$ from the chain, we add an edge to a vertex $v_j \in \mathcal{V} \setminus \mathcal{V}'$: $(v_i, r_{i,j}, v_j)$; and **Edge Direction Flip** where we randomly flip the direction of some edges in c, altering the flow of reasoning.

2.3 Graph Synthetic Data for LLM Tuning

The resultant set of reasoning chains $\mathcal{C}$ can be converted into LLM-supervised fine-tuning data as follows. First, input-output pairs are created based on the requirements of the targeted reasoning task. For instance, a corruption function f' operates by removing an edge $(v_i, r_{i,j}, v_j)$ from a chain c, treating this edge as the output y, while the remaining chain $c' = c \setminus \{(v_i, r_{i,j}, v_j)\}$ is used as the input x for tasks such as family or spatial relation predictions. Second, the input c' needs to be converted into natural language textual input by applying a verbalizer, which can be either rule-based templates or more advanced techniques, such as utilizing a powerful LLM. Finally, while a standard prompt directs the model to answer immediately, we propose a graph-based reasoning task-specific prompting technique that mimics human cognitive processes in solving these types of tasks Sinha et al. (2019). We propose a new prompting technique we call `ETA-P` (Extract then Answer) prompting, which instructs the model to extract the relational graph before generating the answer, as opposed to standard prompting `STD-P`, which instructs the LLM to directly generate the answer. Details of the prompt design are described in Appendix C.

3 Experimental Setup

3.1 Tasks and Datasets

To evaluate our framework, we benchmark it in two different logical reasoning tasks: **CLUTRR** Sinha et al. (2019) supports an inductive reasoning task that requires predicting the relationship between two family members (e.g. Alice is the sister of Bob) based on a story snippet that describes relevant familial relations; **StepGame** Shi et al. (2022) is a spatial reasoning task that involves determining the positional relationship between two entities (e.g., A is to the upper left of B) by navigating through a sequence of steps that describe relationships with neighboring entities. Both benchmarks feature logical reasoning problems with natural language story inputs followed by queries that require multi-hop reasoning. The accuracy of predicting the exact relationships is reported as the evaluation metric. Dataset statistics and synthetic data processing details are listed in Appendix B.1.

3.2 Baselines

We evaluate three system configurations, all using an instruction-tuned LLM as the backbone model. **FS**: The model is tested in a few-shot setting with no additional tuning Brown et al. (2020); **SFT-S**: The model is supervised fine-tuned on the official training set of natural language stories; and **SFT-S**+k: The fine-tuning data consists of training story and k systemic samples generated by our framework of § 2. For all main experiments[1], we use Mistral-2-7B-Instruct Jiang et al. (2023) as the backbone LLM. Additionally, we include the few-shot test results of the commercial closed-source GPT-4o OpenAI (2024a) model. Implementation details of prompt design, fine-tuning, and inference hyperparameters can be found in Appendices B.2 and B.3.

4 Results and Analysis

4.1 Main Results

Figure 2 shows the performance on CLUTRR (top) and StepGame (bottom) of the Mistral-7B LLM under few-shot (**FS**), supervised fine-tuning on stories (**SFT-S**), and supervised fine-tuning on both stories and synthetic data of various sizes[2] (**SFT-S**+k) settings, on both the CLUTRR (top) and StepGame (bottom) datasets. In addition, it includes the few-shot performances of the GPT4-o model.[3]

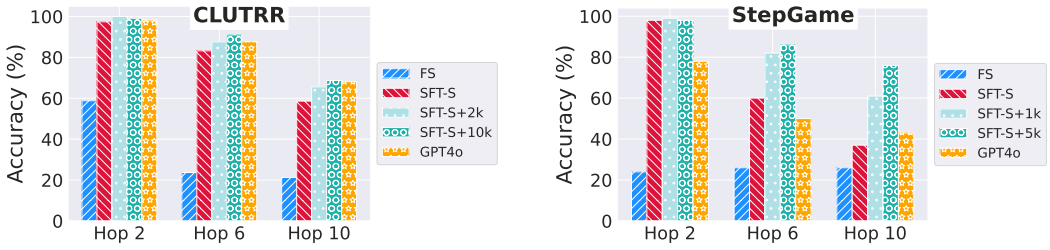
First, we observe that on both datasets, **FS** significantly underperforms all **SFT** models across various reasoning hops, indicating that supervised fine-tuning is essential for enhancing performance on reasoning tasks in moderate-size open-source LLMs. Second, we observe that tuning with our synthetic data (**SFT-S**+k models) consistently yields performance gains on the CLUTRR dataset, with these improvements becoming more pronounced at mid (6 hop) and high (10 hop) complexity levels of reasoning. Interestingly, we notice that extra tuning with synthetic data was necessary, as **SFT-S** underperformed compared to GPT-4o in most cases. This adaptation is crucial for open LLMs in domain-specific settings to achieve performance comparable to the GPT-4o model.

Nevertheless, we observe that on StepGame, a more challenging task with limited SFT data, training with our synthetic data leads to significant improvements compared to using only SFT on textual stories as well as when compared to the closed-source GPT-4o model across

[1] We also experiment with 2 other LLMs, see Appendix D.

[2] The plot only shows the impact of adding the minimum and maximum amounts of synthetic data for CLUTRR (+2k and +10k) and StepGame (+1k and +5k).

[3] We were unable to include the results for all 10 hops and synthetic data size variants in the plot due to visualization constraints. Instead, we selected hops 2, 6, and 10 as representatives of the observed trends for low, mid, and high complexity reasoning, respectively. However, neighboring hops mostly exhibit similar result patterns, and detailed performance data are presented in Table 3 in Appendix D.



(a) CLUTRR (b) StepGame

Figure 2: System performance on the CLUTRR and StepGame datasets for 2, 6, and 10 hop.

all hops. Furthermore, we notice that scaling up with our synthetic data results in more pronounced performance gaps between **SFT-S** and GPT-4o, compared to those observed on CLUTRR. These results suggest that synthetic data has a more significant impact on low-resource scenarios or challenging reasoning problems for task-specific adaptation of LLMs. Finally, we observe a systematic degradation in the performance of all models as we progress from low to mid to high hop reasoning complexity on both tasks, indicating that particularly complex reasoning cases continue to pose significant challenges for LLMs.

4.2 Prompt Strategy Ablation

We study the impact of our proposed prompting strategy in § 2.3 by comparing it to a model using standard prompting (see Table 5 in § D) in both few-shot (**FS**) and story-based supervised fine-tuning (**SFT-S**) settings; results are shown in Figure 3.

Surprisingly, we observed that our task-specific prompt leads to performance drops under the **FS** setting[4]. Our manual inspection revealed that models fail to correctly extract the graph relations from the story, resulting in error propagation into the predictions of the answer. In contrast, we notice that performing supervised fine-tuning (SFT) with our prompt leads to significant gains in most cases, compared to using the standard prompt during tuning. These results suggest that prompt engineering is complementary rather than a replacement for in-domain SFT for reasoning tasks.

4.3 LLM Benchmarks Evaluation

We validate whether LLMs can retain their open domain knowledge and problem-solving abilities, thereby avoiding catastrophic forgetting, after undergoing task-specific adaptation. We do so by evaluating models that have been tuned with the maximum amount of synthetic data, specifically **SFT-S**+10k on CLUTRR and **SFT-S**+5k on StepGame, on MMLU Hendrycks et al. (2021), GPQA Rein et al. (2023), and GSM8K Cobbe et al. (2021) benchmarks.

Benchmark	w.o. SFT	CLUTRR	StepGame
$\text{MMLU}_{\text{0-shot}}$	57.54%	58.46%	58.52%
$\text{GPQA}_{\text{0-shot}}$	29.46%	29.02%	31.70%
$\text{GSM8K}_{\text{8-shot}}$	38.82%	38.36%	38.74%

Table 1: Performance of the original Mistral-2-7B, **SFT-S**+10k on CLUTRR, and **SFT-S**+5k on StepGame models across three LLM evaluation benchmarks.

Results presented in Table 1 show minor variations in performance and, in some cases, improvements—as observed for **SFT-S**+10k on CLUTTR during MMLU—across most benchmarks between the original model without SFT and those tuned on the two reasoning

[4]It is important to note that **FS** with `STD-P` consistently underperforms all **SFT** models across all settings.

tasks. These observations suggest that task-specific adaptation of LLMs for reasoning tasks is feasible without sacrificing factual knowledge and generalization abilities, provided that the synthetic data for SFT is carefully curated.

5 Conclusion

In this work, we propose a synthetic data augmentation algorithm and prompting strategy that effectively complement each other, enabling efficient task-specific adaptation of LLMs for reasoning tasks. We plan to expand our work to include a broader range of graph-based reasoning tasks.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021.

Palaash Agrawal, Shavak Vasania, and Cheston Tan. Exploring the limitations of graph reasoning in large language models. *arXiv preprint arXiv:2402.01805*, 2024.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, 2020.

Xinyun Chen, Ryan A Chi, Xuezhi Wang, and Denny Zhou. Premise order matters in reasoning with large language models. In *Proc. Int. Conf. Mach. Learn.*, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv e-prints*, pp. arXiv–2110, 2021.

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

Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. Test of Time: A Benchmark for Evaluating LLMs on Temporal Reasoning. *arXiv preprint arXiv:2406.09170*, 2024.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *Proc. Int. Conf. Learn. Represent.*, 2021.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2), 2022. ISSN 2162-2388.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Suhang Wang, Yu Meng, and Jiawei Han. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. In *Proc. Annu. Meet. Assoc. Comput. Linguist.*, 2024.

Mehran Kazemi, Hamidreza Alvari, Ankit Anand, Jialin Wu, Xi Chen, and Radu Soricut. GeomVerse: A Systematic Evaluation of Large Models for Geometric Reasoning, 2023a.

Mehran Kazemi, Quan Yuan, Deepti Bhatia, Najoung Kim, Xin Xu, Vaiva Imbrasaite, and Deepak Ramachandran. Boardgameqa: A dataset for natural language reasoning with contradictory information. In *Adv. Neural Inf. Process. Syst.*, volume 36, 2023b.

Erran Li. Improving multi-hop reasoning in llms by learning from rich human feedback. In *ACM Conf. Intell. User Interfaces*, number 1, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

László Lovász. Random walks on graphs. *Combinatorics, Paul erdos is eighty*, 2(1-46):4, 1993.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *Adv. Neural Inf. Process. Syst.*, volume 36, 2023.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. SPARTQA: A Textual Question Answering Benchmark for Spatial Reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021.

OpenAI. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/, 2024a. Accessed: 2024-05-26.

OpenAI. Learning to reason with llms, 2024b. URL https://openai.com/index/learning-to-reason-with-llms/. Accessed: 2024-09-19.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Seyed Mehran Kazemi, Najoung Kim, and He He. Testing the General Deductive Reasoning Capacity of Large Language Models Using OOD Examples, 2023.

Zhengxiang Shi, Qiang Zhang, and Aldo Lipani. StepGame: A New Benchmark for Robust Multi-Hop Spatial Reasoning in Texts. In *Proc. AAAI Conf. Artif. Intell.*, pp. 11321–11329, 2022.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L Hamilton. Clutrr: A diagnostic benchmark for inductive reasoning from text. In *Proc. Conf. Empir. Methods Nat. Lang. Process. Int. Joint Conf. Nat. Lang. Process.*, pp. 4506–4515, 2019.

Saurabh Srivastava, Annarose M. B, Anto P V, Shashank Menon, Ajay Sukumar, Adwaith Samod T, Alan Philipose, Stevin Prince, and Sooraj Thomas. Functional Benchmarks for Robust Evaluation of Reasoning Performance, and the Reasoning Gap, 2024.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. ProofWriter: Generating Implications, Proofs, and Abductive Statements over Natural Language, 2021.

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In *Proc. Conf. North Am. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol.*, pp. 641–651, 2018.

Zhen Tan, Alimohammad Beigi, Song Wang, Ruocheng Guo, Amrita Bhattacharjee, Bohan Jiang, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. Large language models for data annotation: A survey. *arXiv preprint arXiv:2402.13446*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tomer Ullman. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv preprint arXiv:2302.08399*, 2023.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. In *Adv. Neural Inf. Process. Syst.*, volume 36, pp. 74764–74786, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In *Adv. Neural Inf. Process. Syst.*, 2022a.

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

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Trans. Assoc. Comput. Linguist.*, 6:287–302, 2018.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks, 2015.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. In *Int. Conf. Learn. Represent.*, 2024.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024a.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? *arXiv preprint arXiv:2402.16837*, 2024b.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proc. Conf. Empir. Methods Nat. Lang. Process.*, pp. 2369–2380, 2018.

Zhun Yang, Adam Ishay, and Joohyung Lee. Coupling large language models with logic programming for robust and general reasoning from text. In *Findings of the Association for Computational Linguistics: ACL*, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Adv. Neural Inf. Process. Syst.*, 2023.

Hanlin Zhang, Jiani Huang, Ziyang Li, Mayur Naik, and Eric Xing. Improved logical reasoning of language models via differentiable symbolic programming. In *Findings Assoc. Comput. Linguist.*, pp. 3062–3077, 2023a.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023b.

Jinman Zhao and Xueyan Zhang. Exploring the limitations of large language models in compositional relation reasoning. *arXiv preprint arXiv:2403.02615*, 2024.

A Related Works

Logical Reasoning: Logical reasoning has been highlighted in language modeling Zhang et al. (2023a); Yang et al. (2024b); Li (2024). Especially, Yang et al. (2024b) reveal that LLMs possess the capability of latent multi-hop reasoning, in which, however, the moderative evidence of second-hop reasoning does not become stronger with increasing model size.

Reasoning Datasets: Various question-and-answer (QA) datasets have been proposed to evaluate the logical reasoning ability of language models. Commonly used datasets include HotpotQA Yang et al. (2018), ComplexWebQuestions Talmor & Berant (2018), QAngaroo Welbl et al. (2018) and CLUTTER Sinha et al. (2019). These datasets emphasize the knowledge-based reasoning ability of language models. Besides, several datasets focus on spatial reasoning, e.g., bAbI Weston et al. (2015), SpartQA Mirzaee et al. (2021) and StepGame Shi et al. (2022).

Synthetic datasets: A new trend in probing various LLMs capabilities, especially in the case of reasoning, is through synthetic data that allows for a more systematic evaluation. Previous work has developed synthetic datasets for probing and improving various kinds of reasoning including logical reasoning Tafjord et al. (2021); Kazemi et al. (2023b); Saparov et al. (2023), mathematical reasoning Srivastava et al. (2024) Kazemi et al. (2023a) and temporal reasoning Fatemi et al. (2024). Several works add KG triplet information directly to the LM input Agarwal et al. (2021).

B Experimental Setup

B.1 Data Processing

Table 2 shows the number of natural language story[5] samples in the train and test splits of the CLUTRR Sinha et al. (2019) and StepGame Shi et al. (2022) benchmarks per hop, as well as the total number of graph-based synthetic data we generate for each benchmark. Note that the original StepGame dataset consists of structured data for training and test splits only. We deployed GPT-4o to convert these splits into natural language story snippets for StepGame, thereby ensuring consistency in the experimental settings with the train-test splits of its counterpart in the CLUTRR benchmark.

Hop	CLUTRR			StepGame		
	train	test	syn.	train	test	syn.
2	333	114	1162	333	100	555
3	352	229	1170	333	100	555
4	317	219	1129	333	100	555
5	–	308	1219	–	100	555
6	–	178	1224	–	100	555
7	–	246	1231	–	100	555
8	–	228	1120	–	100	555
9	–	172	945	–	100	555
10	–	163	795	–	100	555
Total	1002	1857	9995	999	900	4995

Table 2: Number of samples in the train-test splits of the CLUTRR and StepGame benchmarks per hop, as well as teh total number of synthetic data we populate for each benchmark.

We use the algorithm described in § 2 in order to generate $10k$ and $5k$ reasoning chains for CLUTRR and StepGame, respectively. The synthetic samples are evenly distributed across the 9 different hop categories (2-10). We opted to generate twice as much synthetic data to support CLUTRR because its set of label classes is larger than that of StepGame.

[5]In addition to the textual story, the structured data corresponding to each story is provided in each benchmark.

We used the ASP solver from Yang et al. (2023) as the deduction function f in § 2.1 for CLUTRR deductive reasoning family relationship predict task. Conversely, for the StepGame spatial reasoning task, the relationship is deduced based on relative coordinates. For both tasks, the corruption function f' described in § 2.1 involves removing the edge between the head and tail of the reasoning chain c and using it as the output label. For both datasets, we employ a simple syntactic rule-based heuristic system as a verbalizer to convert our synthetic input reasoning chain c' into a natural language story. We prefer this cost-free verbalizer over LLM API options to accurately quantify the contribution of the synthetic chains themselves, without the enhancements provided by LLMs. Additionally, while LLM-based verbalizers may generate richer text stories, they can introduce errors in story generation, especially in larger hop scenarios. For each task, we carefully design both `STD-P` and `ETA-P` prompts, resulting in a total of four prompts. The prompt design is described later in Appendix C

B.2 Baselines

For the Few-shot (**FS**) setting, we conduct experiments with Mistral-7B-Instruct and GPT-4o [6] using both the `STD-P` and `ETA-P` prompting strategies. In contrast, all settings involving fine-tuning (**SFT-S** and **SFT-S**+k) are conducted exclusively with Mistral-7B. For the **SFT-S** setting, we conduct experiments with both `STD-P` and `ETA-P` settings, while we experiment only with `ETA-P` for **SFT-S**+k. For CLUTRR and StepGame, we create three configurations for **SFT-S**+k where $k \in \{2000, 5000, 10000\}$ and $k \in \{500, 2000, 5000\}$ respectively. These configurations simulate fine-tuning with small, medium, and large amounts of synthetic data. For the small and medium configurations, synthetic data are sampled proportionally across the 9 different hop categories.

In addition to Mistral-7B-Instruct, we conduct parallel experiments with two additional open-source models: Qwen2.5-7B-Instruct Yang et al. (2024a) and Llama3-7B-Instruct Dubey et al. (2024). Similar results and observed trends, as seen with Mistral-7B in Tables 3, 5, and 1, are also observed with these two models in Tables 4, 6, and 7, respectively.

B.3 Implementation Details

For **FS** experiments with GPT-4o, we access the model through the official OpenAI API [7] using the default generation parameters. Inference and fine-tuning experiments were performed on a single GPU server that consists of 8 NVIDIA Tesla V100 cards with 32GB of memory. The pre-training code is based on the PyTorch Paszke et al. (2019) version of the Transformers library Wolf et al. (2020). In all fine-tuning experiments, we train Mistral-7B models for five epochs using a learning rate of 5e-7 with a batch size of 64. We always use AdamW Loshchilov & Hutter (2017) optimizer with a linear decay learning rate scheduler and a warm-up phase for the first 10% of the training. During this phase, the learning rate gradually decreases to reach 1% of its initial value at the end of the fine-tuning process. We found that five epochs were sufficient to fit fine-tuning data of all sizes, and the combinations of the learning rate and batch size were chosen to ensure numerical stability for each benchmark. During inference with all Mistral-7B models, we set the temperature to 0.01 and top-k to 1 to minimize randomness during generation, consequently enhancing reproducibility.

C Prompt Design

In this section, we list the prompts that we have meticulously designed to tailor LLMs for specific reasoning tasks, including CLUTRR family relationship prediction and StepGame spatial relation reasoning. For each prompt, we designed the instruction part through trial and error iterations until we confirmed that both models (Mistral-7B and GPT-4o) could

[6] We include this model to benchmark our methods against a state-of-the-art closed-source system, which is treated as a measure to assess the upper bound of performance.

[7] We use the `gpt-4o-2024-05-13` version of the model from https://chatgpt.com/

follow the instructions and generate outputs in the required format. In the few-shot setting, we set the number of in-context examples, that were picked up from training set of both datasets, to 5 as we did not see any improvement in adding more examples. We did not observe any benefits from using more in-context examples, as the outputs remained mostly stable, with minor to no changes in the model responses.

For both CLUTTER and StepGame, we design a standard (`STD-P`) and an 'extract then answer' `ETA-P` prompt. As described in Section 2.3, the `STD-P` prompts the LLM to directly generate the answer, while `ETA-P` requires first extracting the relational graph before producing the answer. Our proposed `ETA-P` is similar to the step-by-step Chain of Thought (COT) prompting techniques Wei et al. (2022a), but it is specifically tailored for reasoning tasks, as it explicitly instructs the model to extract the relational graph before providing an answer.

We do not position `ETA-P` as an alternative or competitor to Chain-of-Thought (CoT) prompting for more general tasks. Rather, ETA-P is a specialized adaptation, which aims to improve performance on graph-based reasoning tasks. While ETA-P can be viewed as a task-specific variant of CoT, it offers a key advantage in its ability to facilitate fine-tuning for graph-based tasks that require step-by-step reasoning. By leveraging synthetic triples as gold-standard intermediate steps, `ETA-P` enables a more structured and targeted approach to reasoning, eliminating the need of collecting and annotating intermediate steps as is typically required when finetuning models with CoT-like reasoning strategy OpenAI (2024b).

In our prompt, we use the following placeholders: `[STORY]` for the provided natural language story; `[QUERY]` for the natural language query; `[TRIPLES]` for the expected relational graph to extract in triplet format; `[ANSWER]` for the expected answer to be generated. Each of the prompts listed below has a few-shot version, which consists of placing five in-context examples in the same format (### Story: ... `[ANSWER]`) just before the example of interest. Tables 8 to 13 show concrete examples of model prediction case analyses for illustrative purposes.

C.1 CLUTRR `STD-P`

```
You are given a narrative describing the familial relationships between several individuals.
Analyze the narrative and determine the familial relationship between two specified
individuals. The relationship between the characters must be the following: ['aunt',
'brother', 'daughter', 'daughter-in-law', 'father', 'father-in-law', 'granddaughter',
'grandfather', 'grandmother', 'grandson', 'mother', 'mother-in-law', 'nephew', 'niece',
'sister', 'son', 'son-in-law', 'uncle']

### Story:
[STORY]
### Query:
[Query]

### Output:
[ANSWER]
```

C.2 CLUTRR `ETA-P`

```
You are given a narrative describing the familial relationships between several individuals.
First break down the narrative into ordered structured triples, then attempt to answer
the question. The relationship between the characters must be the following: ['aunt',
'brother', 'daughter', 'daughter-in-law', 'father', 'father-in-law', 'granddaughter',
'grandfather', 'grandmother', 'grandson', 'mother', 'mother-in-law', 'nephew', 'niece',
'sister', 'son', 'son-in-law', 'uncle']
### Story:
[STORY]
### Query:
[Query]
```

Output:
The ordered structured triples are: [TRIPLES].
Therefore, [ANSWER]

C.3 StepGame STD-P

```
You are given a narrative describing the spatial relationships between several individuals. Analyze the narrative and determine the spatial relationship between two specified individuals. The relationship between the characters must be chosen from the following options: ["above", "below", "left", "lower-left", "lower-right", "right", "upper-left", "upper-right", "overlaps"]
```

Story:
[STORY]
Query:
[Query]

Output:
[ANSWER]

C.4 StepGame ETA-P

```
Prompt:
You are given a narrative describing the spatial relationships between several individuals. First break down the narrative into ordered structured triples, then attempt to answer the question. The relationship between the characters must be the following: ["above", "below", "left", "lower-left", "lower-right", "right", "upper-left", "upper-right", "overlaps"]
```

Story:
[STORY]
Query:
[Query]

Output:
The ordered structured triples are: [TRIPLES].
Therefore, [ANSWER]

D Results

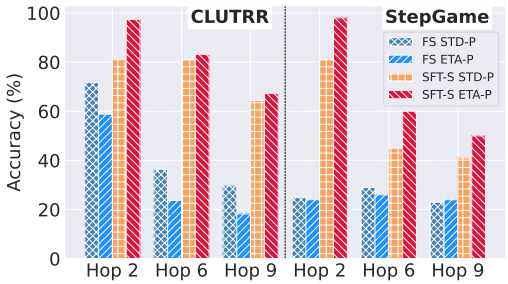


Figure 3: Mistral-2-7B performances on CLUTRR (left) and StepGame (right) datasets under **FS** and **SFT-S** settings when using STD-P and ETA-P prompting.

Hop	CLUTRR Mistral-7B-Instruct					GPT-4o	StepGame Mistral-7B-Instruct					GPT-4o
	FS	**SFT-S**	**SFT-S**+2*k*	**SFT-S**+5*k*	**SFT-S**+10*k*	**FS**	**FS**	**SFT-S**	**SFT-S**+1*k*	**SFT-S**+2*k*	**SFT-S**+5*k*	**FS**
2	58.8	97.4	100.0	100.0	99.1	98.3	24.0	98.0	99.0	100.0	98.0	78.0
3	15.3	93.4	96.4	95.3	95.7	97.1	26.0	90.0	95.0	96.0	97.0	47.0
4	33.2	84.4	87.2	88.1	88.1	85.8	27.0	74.0	90.0	92.0	92.0	45.0
5	25.8	86.1	88.0	90.9	91.9	88.0	24.0	76.0	87.0	94.0	95.0	54.0
6	23.7	83.2	87.6	87.6	91.0	87.6	26.0	60.0	82.0	91.0	87.0	50.0
7	20.1	72.8	82.1	79.7	83.7	79.3	17.0	59.0	73.0	82.0	83.0	50.0
8	21.8	68.6	76.7	77.6	78.5	75.0	12.0	48.0	75.0	76.0	79.0	39.0
9	18.2	67.3	69.6	74.4	74.4	73.8	24.0	50.0	66.0	76.0	80.0	41.0
10	21.3	58.6	65.6	64.4	68.7	68.1	26.0	37.0	61.0	78.0	76.0	43.0

Table 3: Performance, in terms of accuracy, on the CLUTRR and StepGame benchmarks for the few-shot setting (**FS**) with the GPT-4o model, as well as the Mistral-7B-Instruct under **FS**, story supervised fine-tuning (**SFT-S**), and when using our synthetic data (**SFT-S**+*k*). We report the performance using when `ETA-P` for all models.

	Hop	Qwen2.5-7B-instruct					Llama3-8B-Instruct				
		FS	**SFT-S**	**SFT-S**+2*k*	**SFT-S**+5*k*	**SFT-S**+10*k*	**FS**	**SFT-S**	**SFT-S**+2*k*	**SFT-S**+5*k*	**SFT-S**+10*k*
CLUTRR	2	52.6	99.1	100.0	100.0	98.3	44.7	71.9	92.9	89.5	88.6
	3	45.4	83.0	96.4	93.5	97.1	32.8	59.0	82.1	91.7	95.2
	4	52.1	65.8	87.2	83.1	84.9	36.1	60.7	69.4	66.7	73.0
	5	55.5	63.0	88.0	80.5	83.4	37.3	56.8	64.0	64.3	70.8
	6	59.6	65.7	87.6	81.5	82.6	43.3	69.1	72.5	72.5	73.0
	7	52.0	61.4	82.1	72.0	70.3	39.8	60.6	65.5	65.5	72.8
	8	52.6	55.7	76.7	73.7	70.6	32.9	57.5	64.9	66.2	64.5
	9	45.9	49.4	69.6	64.0	57.0	34.3	58.7	58.3	58.1	58.1
	10	54.6	57.1	65.6	63.2	54.0	34.4	61.4	58.3	52.8	61.9
StepGame	2	15.0	40.0	64.0	81.0	82.0	21.0	42.0	72.0	80.0	84.0
	3	21.0	29.0	44.0	54.0	69.0	21.0	34.0	50.0	75.0	76.0
	4	21.0	23.0	36.0	39.0	42.0	21.0	31.0	32.0	56.0	57.0
	5	20.0	31.0	44.0	42.0	50.0	15.0	28.0	42.0	59.0	65.0
	6	21.0	18.0	32.0	31.0	31.0	13.0	29.0	39.0	55.0	50.0
	7	16.0	16.0	33.0	40.0	33.0	12.0	28.0	40.0	47.0	48.0
	8	13.0	15.0	20.0	35.0	38.0	9.0	26.0	36.0	51.0	48.0
	9	18.0	15.0	24.0	37.0	25.0	7.0	23.0	29.0	44.0	45.0
	10	18.0	22.0	26.0	31.0	33.0	9.0	22.0	30.0	34.0	39.0

Table 4: Performance, in terms of accuracy, on the CLUTRR and StepGame benchmarks under few-shot setting (**FS**), story supervised fine-tuning (**SFT-S**), and when using our synthetic data (**SFT-S**+*k*) for both Qwen2.5-7B-instruct and Llama3-8B-Instruct. We report the performance using when `ETA-P` for both models.

Hop	CLUTRR						StepGame					
	Mistral FS		Mistral SFT-S		GPT-4o FS		Mistral FS		Mistral SFT-S		GPT-4o FS	
	STD-P	ETA-P	STD-P	ETA-P	STD-P	ETA-P	STD-P	ETA-P	STD-P	ETA-P	STD-P	ETA-P
2	71.8	58.8	81.6	97.4	90.4	98.3	25.0	24.0	81.0	98.0	55.0	78.0
3	14.7	15.3	93.0	93.4	95.1	97.1	28.0	26.0	66.0	90.0	52.0	47.0
4	30.3	33.2	80.2	84.4	80.4	85.8	26.0	27.0	60.0	74.0	49.0	45.0
5	37.2	25.8	76.1	86.1	71.8	88.0	29.0	24.0	62.0	76.0	51.0	54.0
6	36.4	23.7	80.9	83.2	68.5	87.6	29.0	26.0	45.0	60.0	46.0	50.0
7	35.7	20.1	63.0	72.8	64.6	79.3	31.0	17.0	48.0	59.0	47.0	50.0
8	34.5	21.8	66.8	68.6	55.7	75.0	23.0	12.0	43.0	48.0	45.0	39.0
9	29.6	18.2	64.4	67.3	50.6	73.8	23.0	24.0	41.0	50.0	47.0	41.0
10	35.3	21.3	65.1	58.6	54.0	68.1	31.0	26.0	40.0	37.0	45.0	43.0

Table 5: Performance, in terms of accuracy, on the CLUTRR and StepGame benchmarks for 3 models: **GPT-4o FS**, **Mistral-7B-Instruct FS**, and **Mistral-7B-Instruct SFT-S**. For each model, we ablate when using standard (STD-P) and our extract then answer (ETA-P) prompting strategies.

	Hop	Qwen2.5 FS		Qwen2.5 SFT-S		Llama3 FS		Llama3 SFT-S	
		STD-P	ETA-P	STD-P	ETA-P	STD-P	ETA-P	STD-P	ETA-P
CLUTRR	2	80.7	52.6	99.1	99.1	67.5	44.7	89.5	71.9
	3	61.1	45.4	66.4	83.0	38.0	32.8	89.1	59.0
	4	54.3	52.1	61.2	65.8	37.9	36.1	69.4	60.7
	5	52.9	55.5	59.1	63.0	30.5	37.3	65.6	56.8
	6	54.5	59.6	56.2	65.7	27.5	43.3	59.6	69.1
	7	54.0	52.0	52.0	61.4	23.6	39.8	63.0	60.6
	8	54.0	52.6	50.4	55.7	24.1	32.9	54.8	57.5
	9	47.7	45.9	46.5	49.4	21.5	34.3	47.7	58.7
	10	57.1	54.6	50.9	57.0	23.3	34.4	50.9	61.4
StepGame	2	17.0	15.0	17.0	40.0	16.0	21.0	37.0	42.0
	3	23.0	21.0	17.0	29.0	17.0	21.0	43.0	34.0
	4	16.0	21.0	17.0	23.0	16.0	21.0	40.0	31.0
	5	11.0	20.0	18.0	31.0	14.0	15.0	45.0	28.0
	6	17.0	21.0	6.0	18.0	10.0	13.0	41.0	29.0
	7	9.0	16.0	8.0	16.0	12.0	12.0	38.0	28.0
	8	13.0	13.0	15.0	15.0	10.0	9.0	30.0	26.0
	9	17.0	18.0	12.0	15.0	10.0	7.0	36.0	23.0
	10	12.0	18.0	7.0	22.0	10.0	9.0	27.0	22.0

Table 6: Performance, in terms of accuracy, on the CLUTRR and StepGame benchmarks for **Qwen2.5-7B-Instruct FS**, **Qwen2.5-7B-Instruct SFT-S**, **Llama3-8B-Instruct FS**, and **Llama3-8B-Instruct SFT-S**. For each model, we ablate when using standard (STD-P) and our extract-then-answer (ETA-P) prompting strategies.

Benchmark	Qwen2.5-7B-Instruct			Llama3-7B-Instruct		
	w.o. SFT	CLUTRR	StepGame	w.o. SFT	CLUTRR	StepGame
$\text{MMLU}_{\text{0-shot}}$	17.64%	24.56%	25.47%	61.66%	63.42%	59.63%
$\text{GPQA}_{\text{0-shot}}$	32.14%	33.26%	33.48%	31.03%	29.46%	28.35%
$\text{GSM8K}_{\text{8-shot}}$	69.29%	71.34%	71.424%	76.35%	74.98%	76.8%

Table 7: Performance comparison of the Qwen2.5-7B-Instruct and Llama3-7B-Instruct models in their original form, **SFT-S**+10k fine-tuning on CLUTRR, and **SFT-S**+5k fine-tuning on StepGame, evaluated across three benchmarks.

Prompt C.2 Placeholder Value	**Prediction**
[STORY] Morgan was making dinner when her daughter Frances walked into the kitchen. Frances asked her brother Donald if he wanted to help set the table. Donald's sister Danielle was busy finishing her homework. Kevin, Danielle's grandfather, was reading a newspaper in the living room. Valerie, Kevin's daughter, was knitting a scarf. Harry, Valerie's brother, was watching a football game on TV. Kenneth, Harry's brother, called from the other room asking for help with a project. Kenneth's son Robert was busy working on his science project. Elizabeth, Robert's sister, was reading a book. Brittney, Elizabeth's sister, was playing a board game. [QUERY] What is the relationship of Brittney to Morgan? [TRIPLES] Brittney is the sister of Elizabeth. Elizabeth is the sister of Robert. Robert is the son of Kenneth. Kenneth is the brother of Harry. Harry is the brother of Valerie. Valerie is the daughter of Kevin. Kevin is the grandfather of Danielle. Danielle is the sister of Donald. Donald is the brother of Frances. Frances is the daughter of Morgan. [ANSWER] Brittney is the niece of Morgan	**GPT-4o:** The ordered structured triples are: - Frances is the daughter of Morgan. - Donald is the brother of Frances. - Danielle is the sister of Donald. - Kevin is the grandfather of Danielle. - Valerie is the daughter of Kevin. - Harry is the brother of Valerie. - Kenneth is the brother of Harry. - Robert is the son of Kenneth. - Elizabeth is the sister of Robert. - Brittney is the sister of Elizabeth. Analyzing the relationships: - Frances, Donald, and Danielle are siblings and children of Morgan. - Kevin is the grandfather of Danielle, which means Kevin is the father of one of Morgan's parents. - Valerie is the daughter of Kevin, making her Morgan's mother. - Harry and Kenneth are Valerie's brothers, making them Morgan's uncles. - Robert, Elizabeth, and Brittney are Kenneth's children, making them Morgan's cousins. Therefore, Brittney is the cousin of Morgan. **SFT-S**: The ordered structured triples are: Brittney is the granddaughter of Kevin, who is the father of Valerie, who is the sister of Harry, who is the brother of Kenneth, whose son Robert is the brother of Elizabeth, who is the sister of Brittney. Therefore, Brittney is the granddaughter of Morgan **SFT-S**+10k: The ordered structured triples are: Brittney is the sister of Elizabeth. Elizabeth is the sister of Robert. Robert is the son of Kenneth. Kenneth is the brother of Harry. Harry is the brother of Valerie. Valerie is the daughter of Kevin. Kevin is the grandfather of Danielle. Danielle is the sister of Donald. Donald is the brother of Frances. Frances is the daughter of Morgan. Therefore, Brittney is the niece of Morgan

Table 8: Case analysis from the CLUTTR dataset showing the output of the GPT-4o, SFT-S, and SFT-S+10k systems when using our prompt in Appendix C.2. The left column displays placeholder filler values in the input prompt, as well as the ground truth labels [ANSWER] (which are not sent in the input prompt). For visualization purposes, the final output is underlined, and model errors are highlighted in red.

Prompt C.2 Placeholder Value	**Prediction**
[STORY] Nichole was excited because she was meeting her father, Christian, for lunch. Darren and his father Christian went to look at cars. Pennie took her brother Darren with her to eat pasta at a pasta buffet. Sean, Pennie's brother, joined them at the pasta buffet. Sean's mother, Evelyn, was disappointed that he failed his math class. [QUERY] What is the relationship of Evelyn to Nichole? [TRIPLES] Evelyn is the mother of Sean. Sean is the brother of Pennie. Pennie is the sister of Darren. Darren is the son of Christian. Christian is the father of Nichole. [ANSWER] Evelyn is the mother of Nichole	**SFT-S STD-P:** Evelyn is the grandmother of Nichole **SFT-S ETA-P:** The ordered structured triples are: Evelyn is the mother of Sean. Sean is the brother of Pennie. Pennie is the sister of Darren. Darren is the son of Christian. Christian is the father of Nichole. Therefore, Evelyn is the mother of Nichole

Table 9: Case analysis from the CLUTTR dataset showing the output of the STF-S STD-P and STF-S ETA-P systems when using our prompt in Appendix C.2. The left column displays placeholder filler values in the input prompt, as well as the ground truth labels [ANSWER] (which are not sent in the input prompt). For visualization purposes, the final output is underlined, and model errors are highlighted in red.

Prompt C.2 Placeholder Value	**Prediction**
[STORY] William and his brother James went fishing by the lake. Steven, the father of James, joined them later to share some stories. Steve, the son of Steven, brought along some snacks for everyone. Stanley, Steve's son, was excited to catch his first fish. James, being the fun uncle, helped Stanley bait his hook. Patrice, the sister of James, called to check on their fishing adventure. Darryl, Patrice's son, asked if he could join them next time. Eric, Darryl's brother, was already planning what to bring for the trip. Elsie, the grandmother of Eric, loved hearing about their plans and offered to bake some cookies. Constance, the daughter of Elsie, decided to help her mother with the baking. [QUERY] What is the relationship of Constance to William? [TRIPLES] Constance is the daughter of Elsie. Elsie is the grandmother of Eric. Eric is the brother of Darryl. Darryl is the son of Patrice. Patrice is the sister of James. James is the uncle of Stanley. Stanley is the son of Steve. Steve is the son of Steven. Steven is the father of James. James is the brother of William. [ANSWER] Constance is the sister of William	**GPT4o STD-P:** Constance is the aunt of William. **GPT4o ETA-P:** The ordered structured triples are: 1. William is the brother of James. 2. Steven is the father of James. 3. Steve is the son of Steven. 4. Stanley is the son of Steve. 5. James is the uncle of Stanley. 6. Patrice is the sister of James. 7. Darryl is the son of Patrice. 8. Eric is the brother of Darryl. 9. Elsie is the grandmother of Eric. 10. Constance is the daughter of Elsie. Analyzing the relationships: - William is the brother of James. - James is the brother of Patrice. - Patrice is the mother of Darryl and Eric. - Elsie is the mother of Constance. - Elsie is the grandmother of Eric and Darryl. - Therefore, Constance is the daughter of Elsie and the sister of Patrice. Since Patrice is the sister of James and William, Constance is also the sister of James and William. Therefore, Constance is the sister of William.

Table 10: Case analysis from the CLUTTR dataset showing the output of the GPT4o STD-P and GPT4o ETA-P systems when using our prompt in Appendix C.2. The left column displays placeholder filler values in the input prompt, as well as the ground truth labels [ANSWER] (which are not sent in the input prompt). For visualization purposes, the final output is underlined, and model errors are highlighted in red.

Prompt C.4 Placeholder Value	**Prediction**
[STORY] In the cozy living room on a rainy Sunday afternoon, Emma sprawled out on the soft carpet with her collection of colorful letter blocks. She carefully placed Block O at the very bottom of her play area. After some contemplation, she decided that Block A should be high above, watching over the rest. She nestled Block Q just above A, creating a hierarchy of letters. As she continued, Emma positioned Block S diagonally to the bottom left of Q, making sure it was perfectly aligned. She then placed Block F directly below S, forming a neat vertical line. Finally, she picked up Block M and placed it to the left of F, leaving a small gap between them as if M needed some personal space. Emma admired her work, satisfied with the arrangement of her playful, lettered world. [QUERY] What is the relation of the agent M to the agent O? [TRIPLES] M is directly to the left of F. F is directly below S. S is to the lower-left of Q. Q is directly above A. A is directly above O. [ANSWER] M is directly to the left of O.	**GPT-4o:** The ordered structured triples are: 1. O is at the very bottom of the play area. 2. A is high above O. 3. Q is directly above A. 4. S is to the lower-left of Q. 5. F is directly below S. 6. M is directly to the left of F. To determine the relation of M to O, we need to trace the spatial relationships: - M is directly to the left of F. - F is directly below S. - S is to the lower-left of Q. - Q is directly above A. - A is high above O. Therefore, M is to the lower-left of O. **SFT-S+5*K*:** Ordered structured triples are: M is directly to the left of F. F is directly below S. S is to the lower-left of Q. Q is directly above A. A is directly above O. Therefore, M is directly to the left of O. **SFT-S:** Ordered structured triples: M is directly to the left of F. F is directly below S. S is to the lower-left of Q. Q is directly above A. A is to the upper-right of O. Therefore, M is to the lower-left of O.

Table 11: Case analysis from the StepGame dataset showing the output of the GPT-4o, SFT-S, and SFT-S+5*k* systems when using our prompt in Appendix C.4. The left column displays placeholder filler values in the input prompt, as well as the ground truth labels [ANSWER] (which are not sent in the input prompt). For visualization purposes, the final output is underlined, and model errors are highlighted in red.

Prompt C.4 Placeholder Value	**Prediction**
[STORY] In a cozy corner of his room, young Alex decided to build an elaborate arrangement with his colorful toy blocks. He carefully placed Block U in the middle of his play mat, imagining it as the central point of his creation. With a gleeful grin, he positioned Block Y to the top-right of Block U, creating a small, vibrant tower. Next, Alex found Block X and placed it snugly at the lower left of Block U, its bright hues contrasting with the other blocks. He then took Block Q and aligned it parallel to Block X, making sure it was directly above, forming a neat, parallel structure. Determined to add more layers, Alex grabbed Block D and put it on the left side, with Block Q neatly set on the right. He stacked Block D above Block P, making sure they stood tall in a vertical line. Directly below Block P, he carefully positioned Block L, extending the vertical plane.Continuing his masterpiece, Alex placed Block B directly below Block L and then added Block O above B, maintaining the vertical alignment meticulously. Satisfied with his towering structure, he added Block H below and to the right of Block O, creating an intriguing diagonal extension. Lastly, with a final flourish, Alex placed Block C at the 9:00 position of Block H, completing his intricate design. He sat back and admired his colorful, geometric world, feeling a sense of accomplishment as the afternoon sun cast playful shadows over his creation. [QUERY] What is the relation of the agent C to the agent Y? [TRIPLES] C is directly to the left of H. H is to the lower-right of O. O is directly above B. B is directly below L. L is directly below P. P is directly below D. D is directly to the left of Q. Q is directly above X. X is to the lower-left of U. U is to the lower-left of Y. [ANSWER] C is to the lower-left of Y.	**SFT-S STD-P:** The agent C is to the lower-right of the agent Y. **SFT-S ETA-P:** Ordered structured triples: C is directly to the left of H. H is to the lower-right of O. O is directly above B. B is directly below L. L is directly below P. P is directly below D. D is directly to the left of Q. Q is directly above X. X is to the lower-left of U. U is to the lower-left of Y. Therefore, C is to the lower-left of Y.

Table 12: Case analysis from the StepGame dataset showing the output of the SFT-S STD-P and SFT-S ETA-P systems when using our prompt in Appendix C.4. The left column displays placeholder filler values in the input prompt, as well as the ground truth labels [ANSWER] (which are not sent in the input prompt). For visualization purposes, the final output is underlined, and model errors are highlighted in red.

Prompt C.4 Placeholder Value	**Prediction**
[STORY] In a cozy corner of her room, little Emily spread her colorful building blocks on the soft carpet. She carefully positioned Block L in front of her, a bright blue block that stood proudly at the center of her creation. She then gently placed Block T directly below it, lining them up with precision.With a thoughtful look, she picked up Block D, a vibrant yellow block, and placed it above and to the right of Block L. She smiled as she reached for Block G, placing it just to the left of Block D, ensuring they were perfectly aligned.Finally, she grabbed Block S, a cheerful red block, and positioned it above and slightly to the left of Block G. Emily gazed at her masterpiece with satisfaction, each block perfectly placed in her imaginative world of colors and shapes. [QUERY] What is the relation of the agent S to the agent T? [TRIPLES] S is to the upper-left of G. G is directly to the left of D. D is to the upper-right of L. L is directly above T. [ANSWER] S is to the upper-left of T.	**GPT4o STD-P:** S is to the upper-right of T. **GPT4o ETA-P:** The ordered structured triples are: 1. T is directly below L. 2. D is to the upper-right of L. 3. G is directly to the left of D. 4. S is to the upper-left of G. To determine the relation of S to T, we need to trace the spatial relationships: - S is to the upper-left of G. - G is directly to the left of D. - D is to the upper-right of L. - T is directly below L. Since D is to the upper-right of L, and G is directly to the left of D, G is to the upper-left of L. S is to the upper-left of G, which places S further to the upper-left of L. Given that T is directly below L, S is positioned above and to the left of T. Therefore, S is to the upper-left of T.

Table 13: Case analysis from the StepGame dataset showing the output of the GPT4o STD-P and GPT4o ETA-P systems when using our prompt in Appendix C.4. The left column displays placeholder filler values in the input prompt, as well as the ground truth labels [ANSWER] (which are not sent in the input prompt). For visualization purposes, the final output is underlined, and model errors are highlighted in red.