Do Language Models Robustly Acquire New Knowledge?

Harshay Shah[*] MIT

Badih Ghazi Google

Yangsibo Huang Google

Ravi Kumar Google

Da Yu Google

Chiyuan Zhang Google

Abstract

Language models acquire vast knowledge during pretraining, but adding new knowledge to pre-trained models often lacks robustness—models can retrieve individual facts but struggle with multi-hop reasoning over newly acquired knowledge. To systematically study this robustness gap, we introduce RANK (Robust Acquisition of New Knowledge), a testbed using synthetic knowledge graphs to evaluate knowledge acquisition via k-hop reasoning tasks of increasing complexity. Our evaluation of supervised fine-tuning (SFT) and in-context learning (ICL) reveals that ICL performance degrades with reasoning complexity and knowledge scale, while SFT trained on simple facts fails completely at multi-hop reasoning. However, we find that increasing training data diversity induces a sharp phase transition of fine-tuned models from memorization to out-of-distribution generalization. RANK enables controlled experiments that reveal insights into knowledge acquisition robustness.

1 Introduction

Language models (LMs) acquire vast knowledge during pretraining [79, 15, 37], but incorporating new knowledge post-training remains crucial for adapting LMs to proprietary data. Moreover, rapidly evolving domains make static pretraining both costly and quickly outdated. To address these needs, researchers have developed approaches based on continual pretraining [39, 77], fine-tuning [69], model editing [53, 27], and context compression [56, 63]. However, these methods can fail to integrate large-scale knowledge bases robustly, struggle with update implications of new knowledge [81, 16], interfere with existing knowledge [59], and increase hallucinations [66, 30]. More generally, the robustness of knowledge acquisition—whether models can reason over new knowledge, not just retrieve it—remains understudied.

To study this robustness problem systematically, we define robust knowledge acquisition as the ability to both retrieve individual facts and reason over multiple facts to draw inferences. This motivates three key research questions:

Q1 *How can we systematically measure knowledge acquisition robustness?*

Q2 *How do standard knowledge acquisition methods compare in terms of robustness?*

Q3 *Which data factors most influence robust knowledge acquisition via fine-tuning?*

[*]Work done during an internship at Google.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Continual and Compatible Foundation Model Updates (CCFM).

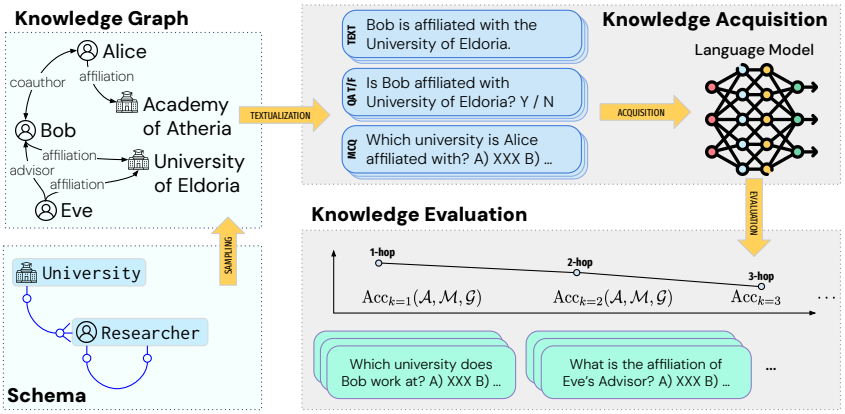

Figure 1: **RANK: A testbed for Robust Acquisition of New Knowledge.** We use our testbed RANK to evaluate robust knowledge acquisition in three steps: (1) generate synthetic knowledge graphs, (2) textualize the graph into natural language for different knowledge acquisition methods, and (3) evaluate their performance through k-hop performance metrics that jointly measure both basic knowledge retrieval and multi-hop compositional reasoning.

To answer the first question, we introduce RANK (Robust Acquisition of New Knowledge), a testbed that uses synthetic knowledge graphs to systematically evaluate robust knowledge acquisition. As shown in Figure 1, knowledge graphs naturally encode reasoning chains as paths, enabling us to generate questions of varying complexity (k-hop) that test both basic retrieval and multi-hop reasoning capabilities.

We use RANK to evaluate the knowledge acquisition robustness of two standard approaches: In-context Learning (ICL) and Supervised Fine-tuning (SFT). First, we find that ICL exhibits systematic performance degradation with both reasoning complexity (number of hops) and knowledge graph size, even for frontier models like Gemini 2.5. Second, models finetuned exclusively on one-hop data achieve perfect retrieval but fail completely on multi-hop reasoning, performing at random chance even after aggressive data augmentation.

To understand why SFT fails at multi-hop reasoning, we use RANK to analyze the role of training data composition, revealing two data-specific factors that significantly improve robustness: (1) increasing training data diversity by adding multi-hop data induces a shift from narrow in-distribution generalization to out-of-distribution generalization and (2) training with chain-of-thought rationales is essential—-removing them causes failure across all reasoning levels. With these changes, SFT can perform significantly better than ICL for robust knowledge acquisition, highlighting the critical role of dataset design for knowledge acquisition.

Organization. We present RANK and our evaluation framework in Section 2, followed by experimental results for SFT and ICL in Section 3. Due to space constraints, we defer detailed experimental setups, additional ablations, and related work to the appendix.

2 RANK: A Testbed for Robust Acquisition of New Knowledge

We now describe RANK, our testbed that leverages synthetic knowledge graphs for evaluating whether language models robustly acquire and reason over new knowledge.

2.1 Using Knowledge Graphs to Study Robust Knowledge Acquisition

Robust knowledge acquisition requires two capabilities: retrieval (learning and recalling individual atomic factual relationships) and reasoning (synthesizing or chaining multiple facts to arrive at an answer). While basic factual knowledge is necessary for reasoning, it is not sufficient—recent work demonstrates that methods successfully adding or updating individual facts often fail when models must reason over the acquired knowledge [81, 80, 16, 59]. Therefore, measuring robustness requires evaluation approaches that jointly assess both retrieval and reasoning over acquired knowledge.

Our testbed RANK operationalizes this study using knowledge graphs—structured representations with entities as nodes and relationships as edges—which naturally enable testing robust knowledge acquisition via k-hop questions. One-hop questions test retrieval of atomic facts, while multi-hop questions test reasoning by requiring models to chain multiple relations. For example, a model that robustly internalizes an affiliation-based knowledge graph should answer the two-hop question "Are Bob and Allen colleagues?" by retrieving and reasoning over: "Bob affiliated with Hogwarts" and "Allen affiliated with Hogwarts."

2.2 Evaluating Knowledge Acquisition Robustness with Multi-hop Questions

We formalize our setup as follows: Let $\mathcal{M}$ denote an instruction-tuned language model, $\mathcal{G}$ a knowledge graph of factual triplets $\{(e, r, e')\}$ unknown to $\mathcal{M}$, and $\mathcal{A} : (\mathcal{M}, \mathcal{G}) \to \mathcal{M}_G$ a knowledge acquisition method (e.g., fine-tuning or in-context learning). Knowledge graphs naturally support multi-hop evaluation: k-hop questions correspond to k-step paths through $\mathcal{G}$. We generate questions by sampling these paths via $\mathcal{Q}_k(\mathcal{G})$ and converting them to natural language.

Definition 1 (k-hop accuracy)**.** *Let $\mathcal{M}$ denote a language model, $\mathcal{G}$ a knowledge graph, and $\mathcal{A}$ a knowledge acquisition method. Let $\mathcal{Q}_k(\mathcal{G})$ denote a question generation process that samples k-hop walks from $\mathcal{G}$ and converts them to natural language questions with ground truth answers. The k-hop accuracy of method $\mathcal{A}$ applied to model $\mathcal{M}$ and graph $\mathcal{G}$ is:*

$$\mathrm{Acc}_k(\mathcal{A}, \mathcal{M}, \mathcal{G}) = \mathbb{E}_{(q,a)\sim\mathcal{Q}_k(\mathcal{G})}[\mathbf{1}[\mathcal{A}(\mathcal{M}, \mathcal{G})(q) = a]]$$

where $\mathbf{1}[\cdot]$ is the indicator function for correct answers.

The evaluation produces a performance profile over multiple values of k, providing a fine-grained measure of robustness. One-hop questions test retrieval of individual facts, while k-hop questions ($k > 1$) test reasoning over multiple facts. Additionally, RANK supports multiple question formats—multiple-choice, cloze, true-false—as detailed in Appendix B.

2.3 Using RANK in Practice

RANK comprises a four-step pipeline: (1) defining entity-relation schemas, (2) generating synthetic knowledge graphs, (3) textualizing graph relations into natural language via templates, and (4) constructing datasets (summaries, questions, solutions) for different knowledge acquisition methods. We defer this discussion to Appendix B.1 due to space constraints.

3 Evaluating Robust Knowledge Acquisition with RANK

We now evaluate the extent to which two standard approaches—Supervised Fine-tuning (SFT) and In-context Learning (ICL)—robustly acquire new knowledge using RANK.

3.1 Supervised Fine-tuning (SFT)

Setup. We begin by describing the setup—data, training, evaluation—used in our SFT experiments. Given a knowledge graph $\mathcal{G}$, we format SFT training data as single-turn (question, answer) conversations containing synthetically generated CoTs using the pipeline described in Section 2.3. We uniformly sample across all data formats—summaries, multiple choice, true-false, and cloze-style questions—and vary dataset size by varying the number of template-based rephrasals per relation (see Appendix B.1). Following prior work [77], we construct a replay buffer of general instruction-following data (using Dolly [19]) to mitigate catastrophic forgetting. We perform full-network fine-tuning on instruction-tuned Qwen3 models [76] for a single epoch and evaluate them with a k-hop performance profile (Definition 1), where $k \in \{1, ..., k_{\text{train}}, ..., k_{\text{test}}\}$. Specifically, we evaluate fine-tuned models based on (a) in-distribution performance on $[1, ...k_{\text{train}}]$-hop questions and (b) out-of-distribution performance with $(k_{\text{train}}, ..., k_{\text{test}}]$-hop questions. We defer details to Appendix D.

Results. We systematically examine three key aspects of robust knowledge acquisition with SFT:

- **Models fine-tuned on one-hop data lack robustness.** The first subplot in Figure 2 shows that fine-tuned models trained exclusively on 1-hop questions achieve near-perfect 1-hop performance (retrieval) with sufficient augmentation (50-100×). However, these models fail on out-of-distribution

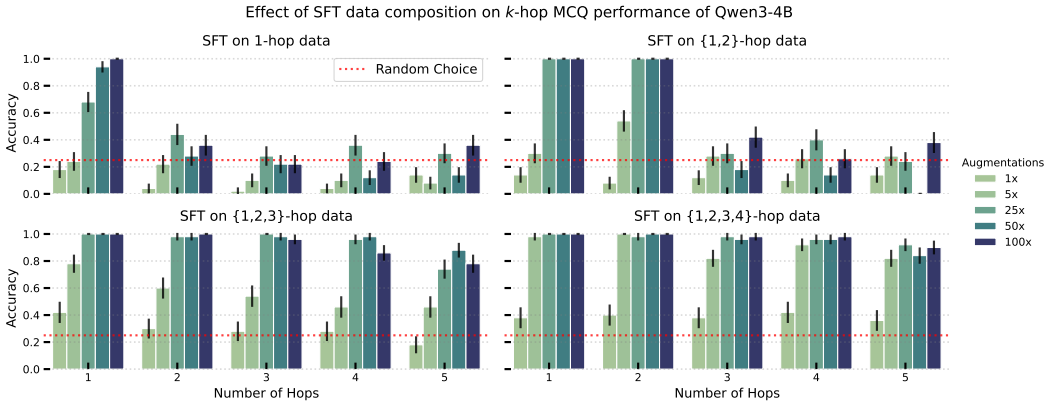

Figure 2: **Training on diverse k-hop questions improves robustness.** The four subplots show the effect of training Qwen3-4B on $\leq k_{\text{train}}$-hop data and testing on k_{test}-hop multiple-choice questions, where $k_{\text{train}} \in [1, 4]$ and $k_{\text{test}} \in [1, 5]$. Each subplot–one for each value of k_{train}—shows accuracy (y-axis) versus $k_{\text{test}} \in [1, 5]$ (x-axis), with darker bars representing more training data augmentation ($1\times$ to $100\times$) and the gray line indicating random chance. Our results reveal three key findings: (a) with sufficient augmentation, SFT achieves strong in-distribution performance when $k_{\text{test}} \leq k_{\text{train}}$, (b) out-of-distribution generalization ($k_{\text{test}} > k_{\text{train}}$) remains poor when training on ≤ 2-hop data but exhibits a critical transition at $k_{\text{train}} = 3$ that enables 4-hop and 5-hop performance, and (c) training on ≤ 4-hop data further improves 5-hop out-of-distribution performance.

multi-hop questions, performing at chance level on all ($k > 1$)-hop tasks regardless of augmentation level. Figure 14 in Appendix D validates this finding across different QA formats and model sizes.

- **Multi-hop training data improves robustness.** Figure 2 examines the effect of training Qwen3-4B on k_{train}-hop training data and evaluating on k_{test}-hop performance, with $k_{\text{train}} \in \{1, 2, 3, 4\}$ and $k_{\text{test}} \in [1, 5]$. With sufficient augmentation, training on $\{1\}$-hop and $\{1, 2\}$-hop data results in models with near-perfect in-distribution performance (i.e., when $k_{\text{test}} \leq k_{\text{train}}$) and close-to-random out-of-distribution performance on $(k_{\text{test}} > k_{\text{train}})$-hop questions. However, when $k_{\text{train}} \geq 3$, fine-tuning improves out-of-distribution generalization. Similarly, adding 4-hop questions to the training data further improves out-of-distribution performance on 5-hop data, indicating a shift from narrow in-distribution generalization to compositional out-of-distribution generalization.
- **Fine-tuning without CoT hurts robustness.** Figure 13 in Appendix D compares Qwen3-1.7B fine-tuned on $\{1, 2, 3\}$-hop data with and without synthetic CoTs. We show that removing CoT from the training data significantly degrades both in-distribution and OOD performance, suggesting that step-by-step decomposition of multi-hop is useful for robustness.

Discussion. Our analysis with RANK unifies key findings on SFT-based knowledge acquisition: the role of data augmentation on one-hop retrieval [3, 51, 60, 11], the role of CoTs for multi-hop knowledge manipulation [4], and more generally, the role of data diversity for compositional generalization [70, 82]. More broadly, while recent findings show that SFT amplifies memorization over generalization [14], our results in Figure 2 show that data augmentation, problem diversity, and CoT fine-tuning can, in fact, make fine-tuned models generalize better.

3.2 In-context Learning (ICL)

We evaluate ICL by textualizing RANK-generated graphs as factual statements in context and testing multi-hop reasoning via zero-shot CoT prompting. Our findings show that ICL performance worsens with an increase in (a) number of hops k (proxy for reasoning complexity) and (b) knowledge graph size (proxy for in-context length), well before reaching maximum context limits, even for frontier models like Gemini 2.5 [17]. We defer this analysis to Appendix C due to space constraints.

4 Conclusion

We introduced RANK, a testbed for systematically evaluating robust knowledge acquisition via multi-hop reasoning over synthetic knowledge graphs. Our experiments reveal limitations in current approaches: ICL performance degrades with reasoning complexity and knowledge scale, while SFT requires careful data design to achieve robust reasoning. We also find a sharp phase transition in SFT—training on diverse multi-hop examples with sufficient augmentation and CoT enables robust out-of-distribution generalization. RANK provides a controlled framework for future research on this critical challenge, enabling systematic evaluation of knowledge acquisition robustness across different methods and configurations.

References

[1] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

[2] Afra Feyza Akyürek, Ekin Akyürek, Leshem Choshen, Derry Wijaya, and Jacob Andreas. Deductive closure training of language models for coherence, accuracy, and updatability. *arXiv preprint arXiv:2401.08574*, 2024.

[3] Zeyuan Allen-Zhu and Yuanzhi Li. Physics of Language Models: Part 3.1, Knowledge Storage and Extraction. In *Proceedings of the 41st International Conference on Machine Learning*, ICML '24, July 2024. Full version available at `https://ssrn.com/abstract=5250633`.

[4] Zeyuan Allen-Zhu and Yuanzhi Li. Physics of Language Models: Part 3.2, Knowledge Manipulation. In *Proceedings of the 13th International Conference on Learning Representations*, ICLR '25, April 2025. Full version available at `https://ssrn.com/abstract=5250621`.

[5] Zeyuan Allen-Zhu and Yuanzhi Li. Physics of Language Models: Part 3.3, Knowledge Capacity Scaling Laws. In *Proceedings of the 13th International Conference on Learning Representations*, ICLR '25, April 2025. Full version available at `https://ssrn.com/abstract=5250617`.

[6] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[7] Deniz Bayazit, Negar Foroutan, Zeming Chen, Gail Weiss, and Antoine Bosselut. Discovering knowledge-critical subnetworks in pretrained language models. *arXiv preprint arXiv:2310.03084*, 2023.

[8] Lucas Caccia, Alan Ansell, Edoardo Ponti, Ivan Vulić, and Alessandro Sordoni. Training plug-n-play knowledge modules with deep context distillation. *arXiv preprint arXiv:2503.08727*, 2025.

[9] Tianyu Cao, Neel Bhandari, Akhila Yerukola, Akari Asai, and Maarten Sap. Out of style: Rag's fragility to linguistic variation. *arXiv preprint arXiv:2504.08231*, 2025.

[10] Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. How do large language models acquire factual knowledge during pretraining? In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/6fdf57c71bc1f1ee29014b8dc52e723f-Paper-Conference.pdf`. NeurIPS 2024.

[11] Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. How do large language models acquire factual knowledge during pretraining? *Advances in neural information processing systems*, 37:60626–60668, 2024.

[12] Tong Chen, Hao Fang, Patrick Xia, Xiaodong Liu, Benjamin Van Durme, Luke Zettlemoyer, Jianfeng Gao, and Hao Cheng. Generative adapter: Contextualizing language models in parameters with a single forward pass. *arXiv preprint arXiv:2411.05877*, 2024.

[13] Xilun Chen, Ilia Kulikov, Vincent-Pierre Berges, Barlas Oğuz, Rulin Shao, Gargi Ghosh, Jason Weston, and Wen-tau Yih. Learning to reason for factuality. *arXiv preprint arXiv:2508.05618*, 2025.

[14] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

[15] 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 preprint arXiv:2110.14168*, 2021.

[16] Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298, 2024.

[17] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[18] Arthur Conmy, Augustine N Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *arXiv preprint arXiv:2304.14997*, 2023.

[19] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm. `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm`, 2023. Databricks Blog Post.

[20] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

[21] Hui Dai, Ryan Teehan, and Mengye Ren. Are llms prescient? a continuous evaluation using daily news as the oracle. *arXiv preprint arXiv:2411.08324*, 2024.

[22] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*, 2021.

[23] Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, et al. Loramoe: Alleviate world knowledge forgetting in large language models via moe-style plugin. *arXiv preprint arXiv:2312.09979*, 2023.

[24] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2023.

[25] Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6: 290–297, 1959.

[26] Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Will Tennien, Atri Rudra, James Zou, Azalia Mirhoseini, et al. Cartridges: Lightweight and general-purpose long context representations via self-study. *arXiv preprint arXiv:2506.06266*, 2025.

[27] Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Shi Jie, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. *arXiv preprint arXiv:2410.02355*, 2024.

[28] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*, 2023.

[29] Jiahai Feng, Stuart Russell, and Jacob Steinhardt. Extractive structures learned in pretraining enable generalization on finetuned facts. *arXiv preprint arXiv:2412.04614*, 2024. doi: 10.48550/arXiv.2412.04614. URL `https://arxiv.org/abs/2412.04614`.

[30] Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning LLMs on new knowledge encourage hallucinations? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.444. URL https://aclanthology.org/2024.emnlp-main.444/.

[31] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.

[32] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.751. URL https://aclanthology.org/2023.emnlp-main.751/.

[33] Gaurav Ghosal, Tatsunori Hashimoto, and Aditi Raghunathan. Understanding finetuning for factual knowledge extraction. *arXiv preprint arXiv:2406.14785*, 2024.

[34] Yuzhe Gu, Wenwei Zhang, Chengqi Lyu, Dahua Lin, and Kai Chen. Mask-dpo: Generalizable fine-grained factuality alignment of llms. *arXiv preprint arXiv:2503.02846*, 2025.

[35] Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*, 2023.

[36] Daphne Ippolito, Florian Tramer, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*, 2022.

[37] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

[38] Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. Unfamiliar finetuning examples control how language models hallucinate. *arXiv preprint arXiv:2403.05612*, 2024.

[39] Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. Continual pre-training of language models. *arXiv preprint arXiv:2302.03241*, 2023.

[40] Mikail Khona, Maya Okawa, Jan Hula, Rahul Ramesh, Kento Nishi, Robert Dick, Ekdeep Singh Lubana, and Hidenori Tanaka. Towards an understanding of stepwise inference in transformers: A synthetic graph navigation model. *arXiv preprint arXiv:2402.07757*, 2024.

[41] Jiyeon Kim, Hyunji Lee, Hyowon Cho, Joel Jang, Hyeonbin Hwang, Seungpil Won, Youbin Ahn, Dohaeng Lee, and Minjoon Seo. Knowledge entropy decay during language model pretraining hinders new knowledge acquisition. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*, 2025. URL https://openreview.net/forum?id=eHehzSDUFp.

[42] John Kirchenbauer, Janny Mongkolsupawan, Yuxin Wen, Tom Goldstein, and Daphne Ippolito. A fictional qa dataset for studying memorization and knowledge acquisition, 2025. URL https://arxiv.org/abs/2506.05639.

[43] Kalle Kujanpää, Pekka Marttinen, Harri Valpola, and Alexander Ilin. Efficient knowledge injection in LLMs via self-distillation. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=drYpdSnRJk.

[44] Yury Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Advances in Neural Information Processing Systems*, 37:106519–106554, 2024.

[45] Andrew K. Lampinen, Arslan Chaudhry, Stephanie C. Y. Chan, Cody Wild, Diane Wan, Alex Ku, Jörg Bornschein, Razvan Pascanu, Murray Shanahan, and James L. McClelland. On the generalization of language models from in-context learning and finetuning: a controlled study. *arXiv preprint arXiv:2505.00661*, 2025. doi: 10.48550/arXiv.2505.00661. URL https://arxiv.org/abs/2505.00661v1. Version 1, submitted May 1, 2025.

[46] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[47] Sheng-Chieh Lin, Luyu Gao, Barlas Oguz, Wenhan Xiong, Jimmy Lin, Wen-tau Yih, and Xilun Chen. Flame: Factuality-aware alignment for large language models. *Advances in Neural Information Processing Systems*, 37:115588–115614, 2024.

[48] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023.

[49] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1711.05101.

[50] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2025. URL https://arxiv.org/abs/2308.08747.

[51] Nick Mecklenburg, Yiyou Lin, Xiaoxiao Li, Daniel Holstein, Leonardo Nunes, Sara Malvar, Bruno Silva, Ranveer Chandra, Vijay Aski, Pavan Kumar Reddy Yannam, Tolga Aktas, and Todd Hendry. Injecting new knowledge into large language models via supervised fine-tuning. *arXiv preprint arXiv:2404.00213*, 2024. doi: 10.48550/arXiv.2404.00213. URL https://arxiv.org/abs/2404.00213.

[52] Alexander Meinke and Owain Evans. Tell, don't show: Declarative facts influence how llms generalize. *arXiv preprint arXiv:2312.07779*, 2023. doi: 10.48550/arXiv.2312.07779. URL https://arxiv.org/abs/2312.07779.

[53] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022.

[54] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022.

[55] Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR, 2022.

[56] Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352, 2023.

[57] Vaishnavh Nagarajan, Chen Henry Wu, Charles Ding, and Aditi Raghunathan. Roll the dice & look before you leap: Going beyond the creative limits of next-token prediction. *arXiv preprint arXiv:2504.15266*, 2025.

[58] Benjamin Newman, Abhilasha Ravichander, Jaehun Jung, Rui Xin, Hamish Ivison, Yegor Kuznetsov, Pang Wei Koh, and Yejin Choi. The curious case of factuality finetuning: Models' internal beliefs can improve factuality. *arXiv preprint arXiv:2507.08371*, 2025.

[59] Kento Nishi, Rahul Ramesh, Maya Okawa, Mikail Khona, Hidenori Tanaka, and Ekdeep Singh Lubana. Representation shattering in transformers: A synthetic study with knowledge editing. *arXiv preprint arXiv:2410.17194*, 2024.

[60] Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv preprint arXiv:2312.05934*, 2023. doi: 10.48550/arXiv.2312.05934. URL https://arxiv.org/abs/2312.05934.

[61] Core Francisco Park, Zechen Zhang, and Hidenori Tanaka. New news: System-2 fine-tuning for robust integration of new knowledge. *arXiv preprint arXiv:2505.01812*, 2025.

[62] Bryan Perozzi, Bahare Fatemi, Dustin Zelle, Anton Tsitsulin, Mehran Kazemi, Rami Al-Rfou, and Jonathan Halcrow. Let your graph do the talking: Encoding structured data for llms. *arXiv preprint arXiv:2402.05862*, 2024.

[63] Aleksandar Petrov, Mark Sandler, Andrey Zhmoginov, Nolan Miller, and Max Vladymyrov. Long context in-context compression by getting to the gist of gisting. *arXiv preprint arXiv:2504.08934*, 2025.

[64] Harshay Shah, Andrew Ilyas, and Aleksander Mądry. Decomposing and editing predictions by modeling model computation. 2024.

[65] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR, 2023.

[66] Chen Sun, Renat Aksitov, Andrey Zhmoginov, Nolan Andrew Miller, Max Vladymyrov, Ulrich Rueckert, Been Kim, and Mark Sandler. How new data permeates LLM knowledge and how to dilute it. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*, 2025. URL https://openreview.net/forum?id=NGKQoaqLpo. Spotlight.

[67] Pengwei Tang, Yong Liu, Dongjie Zhang, Xing Wu, and Debing Zhang. Lora-null: Low-rank adaptation via null space for large language models. *arXiv preprint arXiv:2503.02659*, 2025.

[68] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[69] Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*, 2023.

[70] Arnas Uselis, Andrea Dittadi, and Seong Joon Oh. Does data scaling lead to visual compositional generalization? *arXiv preprint arXiv:2507.07102*, 2025.

[71] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

[72] Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan Ö Arık. Astute rag: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. *arXiv preprint arXiv:2410.07176*, 2024.

[73] Rowan Wang, Avery Griffin, Johannes Treutlein, Ethan Perez, Julian Michael, Fabien Roger, and Sam Marks. Modifying llm beliefs with synthetic document finetuning. Alignment Science Blog, April 2025. URL https://alignment.anthropic.com/2025/modifying-beliefs-via-sdf/.

[74] Yike Wang, Shangbin Feng, Yulia Tsvetkov, and Hannaneh Hajishirzi. Sciencemeter: Tracking scientific knowledge updates in language models. *arXiv preprint arXiv:2505.24302*, 2025.

[75] Khurram Yamin, Gaurav Ghosal, and Bryan Wilder. Llms struggle to perform counterfactual reasoning with parametric knowledge. *arXiv preprint arXiv:2506.15732*, 2025.

[76] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[77] Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candes, and Tatsunori Hashimoto. Synthetic continued pretraining. *arXiv preprint arXiv:2409.07431*, 2024.

[78] Xunjian Yin, Baizhou Huang, and Xiaojun Wan. Alcuna: Large language models meet new knowledge. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1397–1414, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.87. URL https://aclanthology.org/2023.emnlp-main.87/.

[79] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[80] Eric Zhao, Pranjal Awasthi, and Nika Haghtalab. From style to facts: Mapping the boundaries of knowledge injection with finetuning. *arXiv preprint arXiv:2503.05919*, 2025.

[81] Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*, 2023.

[82] Xiang Zhou, Yichen Jiang, and Mohit Bansal. Data factors for better compositional generalization. *arXiv preprint arXiv:2311.04420*, 2023.

[83] Nicolas Zucchet, Jörg Bornschein, Stephanie Chan, Andrew Lampinen, Razvan Pascanu, and Soham De. How do language models learn facts? dynamics, curricula and hallucinations, 2025. URL https://arxiv.org/abs/2503.21676. Accepted at the 2nd Conference on Language Modeling (2025).

[84] Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. Self-adapting language models. *arXiv preprint arXiv:2506.10943*, 2025.

Appendix

A Related Work

We categorize work on knowledge acquisition into three areas: analyzing how models acquire knowledge, developing methods to update parametric knowledge, and evaluating knowledge acquisition performance across methods.

Analyzing knowledge acquisition. A growing line of work on understanding properties of knowledge acquisition leverages synthetic datasets to design controlled pretraining experiments. These studies highlight the role of data augmentation on knowledge extraction [3], CoT reasoning on knowledge manipulation [4], and model size on knowledge capacity [5]. More recent works study different aspects of factuality: learning dynamics [83, 10], role of data formatting and generation [29, 52, 58], and the interplay between in-context information and parametric knowledge [75]. Mechanistic interpretability studies use localization techniques tools [18, 71, 64] to identify knowledge-specific circuits [20, 7], edit factual knowledge [22, 53], and infer mechanisms responsible for factual recall [31, 32]. Recent works on knowledge injection highlight the (a) difficulty of adding *new* knowledge to pretrained models [30, 78], and (b) its interplay with "knowledge entropy" [41], hallucination [66], and existing knowledge [83]. More recently, Zhao et al. [80] show that (a) knowledge acquisition is primarily driven by train-evaluation format alignment and (b) models struggle to employ injected facts in multi-hop reasoning. Our work contributes to this direction by using RANK (Section 2)—a testbed based on synthetic knowledge graphs—to analyze whether language models *robustly* acquire new knowledge (Section 3).

Improving knowledge acquisition. Knowledge acquisition methods can be grouped by approach: fine-tuning, reinforcement learning, model editing, and in-context learning. Fine-tuning variants typically improve knowledge acquisition via data augmentation recipes based on synthetic documents [73, 77], paraphrasing [51, 60], implications [2], multi-agent debate [24], and in-context inferences [45, 61]. Methods based on reinforcement learning directly learn such data augmentation recipes [84] or tune the model for factuality via external verifiers [69], fine-grained loss masking [34], or multi-objective rewards [47, 13]. Model editing methods, which update or add atomic facts by modifying a small subset of model weights, typically rely on localization [53, 54, 27, 67], meta-learning [55] and adding new parameters [35, 23]. Methods that focus on in-context knowledge can be grouped into those that (a) learn reusable parametric modules (e.g., through context distillation [43, 8], generative adapters [12], and learned KV caches [26]), and (b) improve the in-context encoding of graph-structured knowledge [28, 62]. In contrast, the goal of our work is not to develop new methods for knowledge acquisition but rather to understand how representative approaches—in-context learning and fine-tuning—fare at robust knowledge acquisition.

Evaluating knowledge acquisition. Another line of work evaluates knowledge acquisition interventions and their failure modes. Studies that evaluate knowledge editing methods highlight issues such as representation shattering [59], unintended ripple effects that impact related knowledge [16], and limited effectiveness on compositional tasks [81]. Additionally, several works highlight unintended consequences of knowledge acquisition approaches: retrieval-augmented generation suffers from brittleness [9] and increased hallucinations [72], while fine-tuning exhibits sensitivity to fact popularity [33], hallucinations due to unfamiliar knowledge [38, 30], and catastrophic forgetting [50]. Recent works also develop synthetic testbeds for studying knowledge acquisition [3, 42] and realistic benchmarks for evaluating practical knowledge acquisition performance [74, 21]. Specifically, recent works also use synthetic graphs to study language model phenomena such as stepwise inference [40] and creativity vis-a-vis next-token prediction [57].

B RANK: Robust Acquisition of New Knowledge

This appendix provides additional details on the implementation of RANK, covering the complete pipeline from generating synthetic entities and knowledge graphs to creating multi-hop reasoning questions with chain-of-thought solutions.

B.1 Using RANK in Practice

We now outline a four-step pipeline for applying RANK to study knowledge acquisition: (1) defining the schema, (2) generating the knowledge graph, (3) textualizing the graph into natural language, and (4) constructing datasets for training and evaluation, as shown in Figure 1. In this work, we use RANK to generate knowledge graphs over researchers and their academic relationships, though the framework can be applied to any domain by specifying alternative schemas.

Step 1: Defining the schema. To construct synthetic knowledge graphs, we first need a schema specifying valid relations over a pool of entities. We generate five types of fictional entities—`people`, `universities`, `companies`, `cities`, and `research areas`—using instruction-tuned models in order to avoid overlap with knowledge acquired during pretraining. Then, we create a schema to specify valid relations—`graduate school`, `graduation year`, `research advisor`, `research area`, `current affiliation`, `location`, and `coauthors`—between pairs of entities. Each relation corresponds to an atomic fact (s, r, t), associating a source entity s to a target entity t via relation r. Together, these entities and relations form the nodes and edges of our knowledge graphs. We defer details to Appendix B.

Step 2: Generating the knowledge graph. Given the schema and entity pools, we generate synthetic knowledge graphs by first specifying the number of nodes (i.e., entities of each type) and the number of edges (i.e., total relations in the graph). We then construct the graph in two steps: (a) adding nodes by uniformly sampling entities of each type from the entity pools, and (b) adding edges by uniformly sampling[2] target entities for each (source entity, relation) pair. We also use rejection sampling to enforce logical consistency constraints, e.g., ensuring that an advisor's graduation year precedes their advisee's graduation year.

Step 3: Textualizing relations in the knowledge graph. To convert knowledge graphs to natural language, we start by textualizing the relations (i.e., edges) that encode atomic facts. Following prior work [3, 42], we create multiple templates for each relation type in different formats such as declarative statements, reversals, and questions. For example, the triplet (`Bob`, `graduate school`, `University of Eldoria`) can be textualized as "Bob graduated from the University of Eldoria", (statement),"The University of Eldoria is Bob's alma mater" (reversal), and "Where did Bob graduate from? University of Eldoria" (question). This approach supports data augmentation through multiple templates per relation while controlling for verbatim memorization [36] by using distinct template sets for training and evaluation. We provide the complete set of templates in Appendix B.

Step 4: Constructing datasets. We use the textualized relations from the previous step to generate training and evaluation data. Different approaches to knowledge acquisition require different data formats, so introduce building blocks that can be adapted to a given approach; for example, supervised fine-tuning requires question-answer pairs while in-context learning requires demonstrations. We outline three building blocks—summaries, questions, and solutions—below, each of which chains sequences of 1-hop relations to synthesize multi-hop information; we vary sequence length to control the number of hops.

- **Summaries**: We consider nodal and relational summaries. Nodal summaries aggregate all 1-hop relations from an entity to create descriptive text about that entity. Relational summaries sample pairs of entities, trace connecting paths between them, and generate descriptions of their relationships, similar to the augmentation approach in EntiGraph [77].

[2] Our uniform graph sampling approach in the second step (Section 2.2) resembles Erdős–Rényi graphs [25]. This can be swapped with more realistic graph models such as Barabási and Albert [6] to generate scale-free knowledge graphs with long-tailed degree distributions.

- **Questions**: We create question-answer pairs by sampling k-hop random walks from the knowledge graph and converting them into multiple choice, true-false, and cloze questions. For example, a 2-hop walk through (`Alice`, `advisor`, `Bob`) and (`Bob`, `affiliation`, `University of Eldoria`) generates the question "Where did Alice's advisor graduate from?".
- **Solutions**: Similar to reasoning traces, we provide algorithmically generated solutions to k-hop questions that decompose the reasoning into k steps to arrive at the correct answer. Each step resolves intermediate entities, showing how information chains across relations.

B.2 Additional details

Generating fictitious entities . To construct synthetic knowledge graphs with entities unknown to pre-trained models, we employ a two-stage generation and filtering process. First, we use instruction-tuned language models with manually crafted prompts to generate diverse pools of fictitious entities across five types: full names, universities, companies, cities, and research areas. Our prompts are designed to encourage realistic but uncommon names—for example, requesting entities that sound plausible but do not match any well-known real-world counterparts. Each entity type has specific constraints to ensure consistency and realism, such as requiring universities to either start or end with "University" and limiting company names to single words. Second, we filter the generated entities using a base language model's cross-entropy loss to identify candidates that are neither too familiar (potentially in training data) nor too unrealistic (difficult for models to process). By computing loss scores for each entity given a neutral context, we select entities that strike an optimal balance between realism and novelty, avoiding knowledge the confounding effects from pre-existing parametric knowledge.

Specifying the graph schema. Our schema defines the structure of academic knowledge graphs by specifying which entities can be connected and how. We model an academic domain with six entity types: people (researchers), universities, companies, cities, research areas, and years. The schema defines relations between these entities, such as "graduate school" connecting people to universities, "current affiliation" linking researchers to their institutions, and "coauthor" relating researchers to each other. Each relation specifies its domain (source entity types) and range (target entity types), along with its inverse relation—for example, if Alice graduated from University X, then University X has Alice as an alumnus. To avoid redundancy, we designate one relation in each inverse pair for sampling while the other is automatically inferred. Some relations have dependencies: advisor relationships require graduation years to be assigned first, ensuring that advisors graduated before their students. Our validation checks ensure the schema is consistent—inverse relations properly mirror each other, dependency chains don't create cycles, and exactly one relation in each pair is marked for sampling. The schema supports both unique relationships (each person has one graduation year) and multiple relationships (universities can have many alumni), enabling realistic academic networks that support multi-hop reasoning questions about researcher connections, institutional affiliations, and collaborative relationships.

Generating the graph. Given the schema and entity pools, we generate knowledge graphs through a systematic two-step process. First, we sample the specified number of entities for each type from the filtered entity pools described earlier. Second, we generate relations by processing relation types in dependency order—relations with prerequisites are sampled after their required relations have been established. For each entity and applicable relation type, we uniformly sample the specified number of target entities from valid candidates, applying constraint filters when necessary. For example, when assigning academic advisors, we filter potential advisors to ensure their graduation year precedes their student's graduation year, maintaining temporal consistency. The system automatically handles inverse relations by simultaneously creating both directions—when Alice graduates from University X, we add both the "graduate school" relation from Alice to University X and the inverse "alumni" relation from University X to Alice. This process generates realistic academic networks where researchers have graduation histories, institutional affiliations, collaborative relationships, and research specializations. We provide multiple standardized graph configurations ranging from smaller networks (30 researchers) to larger ones (500 researchers) to support reproducible experiments across different scales while maintaining consistent academic relationship patterns. As shown in Figure 3, even moderately-sized knowledge graphs can quickly exceed the context length limits of current language models when textualized, making controlled graph sizes essential for systematic evaluation.

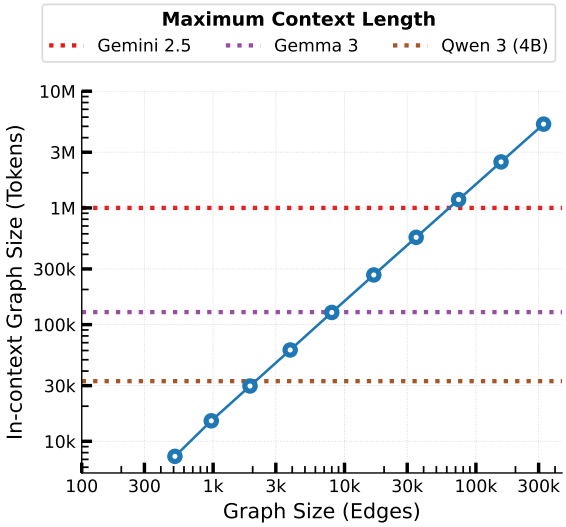

Figure 3: **Large knowledge graphs quickly exhaust context limits.** We plot the relationship between the size of RANK knowledge graphs (measured in edges or factual triplets) and the resulting context length in tokens after textualizing the graph. The horizontal dashed lines indicate maximum context lengths for different model families: Qwen 3 4B (32k tokens), Gemma 3 (128k tokens), and Gemini 2.5 (1M tokens). Even moderately-sized knowledge graphs with 30k-100k edges exceed the context limits of long-context model families such as Gemini.

Textualizing relations with templates We convert knowledge graph triplets into natural language using a template-based approach that naturally supports data augmentation. Our templates are stored in YAML configuration files, with each relation type having multiple template variants that use standardized placeholders: `{domain}` for the source entity, `{range}` for the target entity, and `{relation}` for the relation type. Figure 4 shows examples from our `augmented.yml` file, where a graduation relation can be textualized as "Bob completed their graduate studies at University of Eldoria" (formal) or "Bob's graduate alma mater is University of Eldoria" (descriptive variant). To ensure models generalize beyond specific phrasings, we implement a train-test template splitting system that reserves distinct template sets for training and evaluation while allowing some templates to be shared across both splits. This approach enables controlled data augmentation—we can generate multiple textual variants of the same factual triplet while ensuring that models must learn the underlying relationships rather than memorizing specific sentence structures. This approach also supports different augmentation strategies by categorizing templates into types (base statements, chain-of-thought reasoning, yes/no questions) and allows selective sampling from template subsets to run controlled experiments.

Generating multi-hop questions. We use the textualized relations to generate training and evaluation data through a modular building blocks approach. Different knowledge acquisition methods require different data formats—supervised fine-tuning needs question-answer pairs while in-context learning requires demonstrations—so RANK designs flexible components that adapt to specific experimental requirements.

- **Building blocks for multi-hop data generation.** RANK provides three core building blocks that chain sequences of 1-hop relations to synthesize multi-hop information:
  - **Summaries**: We generate nodal summaries that aggregate all outgoing relations from an entity, and relational summaries that describe connection paths between entity pairs. These support descriptive tasks ("Describe Alice and her background") and relationship queries ("How are Bob and Alice related?").
  - **Questions**: We sample k-hop random walks and convert them into three formats: multiple choice questions, true-false statements, and cloze questions with masked placeholders.

Example Templates from augmented.yml

Relation Type

`GRAD_SCHOOL` (domain: person, range: university)

Template Variations

- "{domain} completed their graduate studies at {range}."
- "{domain} earned a graduate degree at {range}."
- "{domain} received a graduate degree from {range}."
- "{domain} pursued graduate studies at {range}."
- "{domain}'s graduate alma mater is {range}."

Relation Type

`CURRENT_AFFILIATION` (domain: person, range: institution)

Template Variations

- "{domain} presently holds a position at {range}."
- "{domain}'s current professional home is {range}."
- "{domain} is associated with {range} at present."
- "{domain} is currently working at {range}."
- "{domain}'s employer is {range}."

Figure 4: **Templates for textualizing relations.** Example templates for two relation types from our `augmented.yml` file. Each relation has multiple templates that use placeholders (`{domain}`, `{range}`) to create different text versions of the same fact, supporting data augmentation while preventing memorization.

  - **Solutions**: We generate structured chain-of-thought traces that decompose k-hop reasoning into k explicit steps. Each step verbalizes intermediate triplets and builds toward the final answer, providing models with step-by-step guidance for multi-hop reasoning.

- **Converting k-length walks to k-hop questions.** RANK converts k-hop random walks into evaluation questions via entity masking. The approach identifies intermediate entities—those appearing as both domain and range within the walk sequence—and replaces them with typed placeholders while preserving the first domain and final range entities. Each placeholder receives a unique identifier based on entity type abbreviations (e.g., "Person PE1", "University UN2"), maintaining type information while preventing models from exploiting memorized entity associations. The masked walks are then converted into different question formats.

- **Generating chain-of-thought solutions.** For each multi-hop question, RANK generates structured CoT-like traces using question-specific CoT subroutines. The approach decomposes k-hop reasoning into k sequential steps, where each step verbalizes the corresponding knowledge graph triplet and establishes intermediate entity bindings (e.g., "From step 1, we know that Person PE1 is Alice"). Different CoT generators handle question types distinctly: multiple choice generators check each option against sets of valid entities derived from all possible paths, true-false generators verify statement validity by comparing claimed entities with computed valid sets, and cloze generators systematically resolve masked entities through step-by-step triplet evaluation.

Examples. As discussed in Appendix B.1, RANK generates multiple question formats with corresponding algorithmic solutions. Figures 5 and 6 provide examples of descriptive questions that aggregate entity relationships without requiring multi-hop reasoning. For tasks requiring compositional inference, Figures 7 to 9 provide examples of 2-hop multiple-choice, true-false, and cloze questions along with algorithmically generated CoT solutions. Each example contains a task description, a question, a step-by-step CoT-like solution that decomposes multi-hop inference into sequential steps, and the final answer.

An example of a nodal summary generated using RANK

Task Description

Read the instruction and describe the query entity and its direct relationships with other entities. Provide a coherent description of the entity's relations and connections.

Instruction

Give key facts about researcher Elspeth Rigel.

Answer

Elspeth Rigel conducts research in the field of Holographic Memory Encoding. Elspeth Rigel graduated from Zenith Vale University. Elspeth Rigel is employed at Talwood University. Elspeth Rigel completed their graduate studies in the year 1965. Elspeth Rigel's research advisor in graduate school was Tilbury Quill. Elspeth Rigel served as the research advisor for Jumoke Adebayo. Elspeth Rigel has been a co-author with Kellan Winters on research papers. Elspeth Rigel has been a co-author with Bastian Kirov on research papers. Elspeth Rigel has been a co-author with Caspian Vela on research papers. Elspeth Rigel has been a co-author with Milana Brubaker on research papers. Elspeth Rigel has been a co-author with Dorian Zephyr on research papers. Elspeth Rigel has been a co-author with Liesel Tiberius on research papers. Elspeth Rigel has been a co-author with Rafferty Blakely on research papers. Elspeth Rigel has been a co-author with Nyssum Tharen on research papers. Elspeth Rigel has been a co-author with Elden Harrow on research papers.

Figure 5: **An example of a nodal summary generated using RANK.** This figure demonstrates a descriptive task where the model aggregates all 1-hop relationships for a given entity, providing a comprehensive overview of the entity's direct connections within the knowledge graph.

An example of a 1-hop relational summary generated using RANK

Task Description

Read the instruction and describe how the two query entities are connected through a sequence of relationships. Provide an explanation of the path of connections between them.

Instruction

How does researcher Isolde Brine relate to year 1993?

Answer

Isolde Brine completed their graduate studies in the year 1993.

Figure 6: **An example of a 1-hop relational summary generated using RANK.** This figure demonstrates a relational task where the model describes the direct connection between two specified entities within the knowledge graph.

An example of a two-hop true-false question generated using RANK

Task Description

Read the statement and determine if it is True or False. The text may contain entity placeholders such as Person PE#, University UN#, City CT#, Research Area RA#, etc. Each placeholder has a prefix for the entity type and a number that uniquely identifies it inside this prompt. For example, PE1 always refers to the same hidden person, while PE1 and PE2 refer to different people. Figure out which entity each placeholder can represent (multiple valid assignments may exist) and reason over those assignments to answer the question. Do not include entity placeholders in your final answer. Output 'True' if all parts of the statement are factually correct. Output 'False' if any part of the statement is factually incorrect. Think step-by-step. In your final step, clearly indicate your answer on a new line starting with 'The answer is'.

Statement

Niles Calder is employed at University UN1. University UN1 has a base in the city of Draymont.

Determine whether the statement above is True or False.

Algorithmically generated CoT

My plan is to verify the statement by building a set of all possible correct answers step-by-step. I will start with the first known entity and, for each relationship in the chain, I will find all valid resulting entities. Finally, I will check if the entity claimed in the statement exists within my final set of possibilities to determine if the statement is True or False.
Here is the step-by-step reasoning:

- **Step 1:** Which entity could University UN1 be? Niles Calder is employed at Havenlight University. So, University UN1 is Havenlight University.
- **Step 2:** Is University UN1 located in Draymont?
  - Havenlight University is located in Morwen.
  - The set of valid entities is (Morwen). Draymont is not in this set.
  - So, the answer to "Is University UN1 located in Draymont?" is a No.

Therefore, the statement is False.

Final Answer

The answer is: False.

Figure 7: **An example of a two-hop true-false question generated using RANK.** This figure demonstrates the complete structure of a multi-hop reasoning question, including task instructions, the statement with masked entities (placeholders), step-by-step chain-of-thought reasoning that resolves entity relationships across multiple hops, and the final answer determination.

An example of a two-hop multiple-choice question generated using RANK

Task Description

Read the question and the answer choices first. Among the four choices, one is correct while the others are incorrect. The text may contain entity placeholders such as Person PE#, University UN#, City CT#, Research Area RA#, etc. Each placeholder has a prefix for the entity type and a number that uniquely identifies it inside this prompt. For example, PE1 always refers to the same hidden person, while PE1 and PE2 refer to different people. Figure out which entity each placeholder can represent (multiple valid assignments may exist) and reason over those assignments to answer the question. Do not include entity placeholders in your final answer. Output only the letter corresponding to the correct choice (e.g., A, B, C, D). Think step-by-step. In your final step, clearly indicate your answer on a new line starting with 'The answer is'.

Question

Prospera has a base in the city of City CT1. Out of the following institutions, which ones are located in City CT1?

- **A:** Technacle
- **B:** Silicodynamics
- **C:** Oakridge University
- **D:** Sagebrush College

The goal is to determine which of the given answer choices correctly answers the question.

Algorithmically generated CoT

My plan is to verify the answer by building a set of all possible correct answers step-by-step. I will start with the first known entity and, for each relationship in the chain, I will find all valid resulting entities. Finally, I will check which of the multiple choice options exists within my final set of possibilities to determine the correct answer.
Here is the step-by-step reasoning:

- **Step 1:** Which entity could City CT1 be? Prospera is in Moldenfell. So, City CT1 is Moldenfell.
- **Step 2:** We can now determine valid hosts of City CT1.
  - Moldenfell is a host of one or more offices of Oakridge University. Moldenfell hosts one or more offices of Prospera. Moldenfell hosts one or more offices of Talwood University. Moldenfell hosts one or more offices of Virtalith.
  - So, the list of valid hosts is: Prospera, Virtalith, Talwood University, Oakridge University.
  - Is choice A (Technacle) in this list? No. Is choice B (Silicodynamics) in this list? No. Is choice C (Oakridge University) in this list? Yes. Is choice D (Sagebrush College) in this list? No.

Therefore, the correct answer choice is C.

Final Answer

The answer is: C.

Figure 8: **An example of a two-hop multiple-choice question generated using RANK.** This figure demonstrates the complete structure of a multi-hop reasoning question, including task instructions, the question with masked entities (placeholders), step-by-step chain-of-thought reasoning that resolves entity relationships across multiple hops, and the final answer selection.

An example of a 2-hop cloze question generated using RANK

Task Description

Read the statement carefully. The statement ends with a [MASK] placeholder. The text may contain entity placeholders such as Person PE#, University UN#, City CT#, Research Area RA#, etc. Each placeholder has a prefix for the entity type and a number that uniquely identifies it inside this prompt. For example, PE1 always refers to the same hidden person, while PE1 and PE2 refer to different people. Figure out which entity each placeholder can represent (multiple valid assignments may exist) and reason over those assignments to answer the question. Do not include entity placeholders in your final answer. Your task is to replace [MASK] with exactly one correct entity name. Even if multiple entities could fit, you must choose and output only one of them. Think step-by-step. In your final step, clearly indicate your answer on a new line starting with 'The answer is'.

Statement

Liora Vale served as the research advisor for Person PE1. Person PE1 completed their studies in the year [MASK]

The goal is to find the correct entity to replace `[MASK]` in the statement above.

Algorithmically generated CoT

To find the correct entity to replace `[MASK]`, I will trace the relationships between the entities mentioned in the statement, starting from the first known entity and following the relationships until I can identify a correct entity for `[MASK]`.
Here is the step-by-step reasoning:

- **Step 1:** Liora Vale was the research advisor of Meera Bharti. So, Person PE1 is Meera Bharti.
- **Step 2:** From step 1, we know that Person PE1 is Meera Bharti. Meera Bharti completed their graduate studies in the year 2018.

Therefore, we can replace `[MASK]` with 2018.

Final Answer

The answer is: 2018.

Figure 9: **An example of a 2-hop cloze question generated using RANK.** This figure demonstrates a fill-in-the-blank task where the model must resolve entity relationships across multiple hops to determine the correct entity to fill the masked position.

C Evaluating In-context Learning with RANK

C.1 Findings

Setup. We evaluate ICL-based knowledge acquisition in two steps: (a) placing the knowledge graph $\mathcal{G}$ in context and (b) evaluating multi-hop compositional reasoning. First, using the template-based approach in Appendix B, we textualize the knowledge graph (generated via RANK) by converting every edge—a factual triplet—into factual statements and presenting them as a list in the context. Then, we generate multi-hop questions where every question includes the in-context knowledge graph, a task description with a prompt for zero-shot CoT reasoning, and the question itself, which is either a k-hop multiple choice question or true/false statement. To answer correctly, the language model must break down the question into k reasoning steps and use the in-context information at each step to arrive at the correct answer. Our experiments use instruction-tuned language models and knowledge graphs that entirely fit in the context of all models we consider. We defer details and examples to Appendix C.

Results. We now use RANK to evaluate whether ICL can robustly acquire new knowledge:

- **Effect of number of hops and model size.** Figure 10 evaluates models of size 3.8B to 14B from three families—Phi 4 [1], Gemma 3 [68], and Qwen 3 [76]—on k-hop multiple choice and true-false questions. For all models, we observe that increasing the number of hops systematically degrades performance. For example, Phi 4 achieves near-perfect accuracy (98%) on 1-hop multiple choice questions but drops substantially to 85% and 52% on 2-hop and 5-hop questions respectively. However, larger models within each family demonstrate better multi-hop reasoning robustness over the in-context knowledge graph. For instance, on 3-hop true-false questions, Phi 4 achieves 90% accuracy compared to 35% for Phi 4 mini.
- **Effect of context length.** Figure 11 examines the effect of context length on ICL performance by varying the size of the knowledge graph itself. We evaluate Gemini 2.5 models [17], which support context lengths up to 1M tokens, on knowledge graphs that when textualized range from 1000 tokens to 1M tokens. While these models excel on small knowledge graphs, their performance degrades as the knowledge graph size (and corresponding context length) increases, well before reaching the maximum context limit. All models except Gemini 2.5 Flash on 1-hop tasks suffer substantial accuracy drops as context length grows from 1000 tokens to 1M. The degradation is more pronounced for higher-hop questions, with 3-hop performance showing the most degradation across all model variants.

Our findings show that ICL struggles with robust knowledge acquisition, degrading as reasoning complexity (number of hops) and knowledge scale (context length) increase. Additionally, Appendix C shows that the response length, a proxy for test-time compute, increases with the number of hops k (Figure 12).

Discussion. Our analysis with RANK shows that robust knowledge acquisition via ICL is bottlenecked by basic challenges in long-context reasoning [44], such as sensitivity to irrelevant information [65] and the "lost-in-the-middle" phenomenon [48]. Additionally, ICL is constrained by (a) memory requirements that scale quadratically with context length and (b) context size limits—even moderately-sized knowledge graphs generated using RANK quickly exceed the long-context limits of frontier models like Gemini. These limitations suggest that alternative in-context strategies such as context compression [56, 63] and retrieval-augmented generation (RAG) [46] could be more amenable to large-scale knowledge acquisition.

C.2 Experiment setup

We evaluate ICL-based knowledge acquisition by placing entire knowledge graphs in context and testing multi-hop reasoning capabilities. Our experimental pipeline consists of context construction, question generation, and evaluation across different model families and graph sizes.

- **Converting knowledge graphs into in-context information.** We textualize knowledge graphs by extracting each entity's one-hop neighborhood and converting triplets to natural language using the template-based approach in Appendix B.1. These entity-centric descriptions are then concatenated

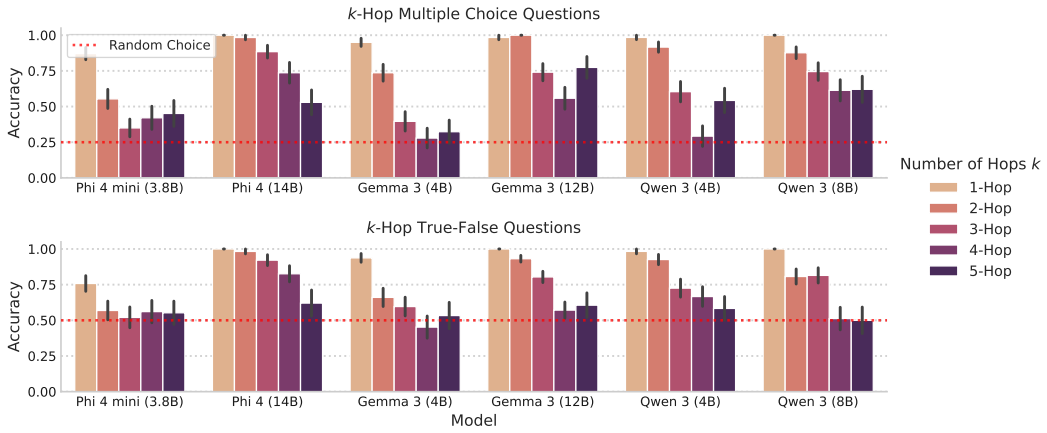

Figure 10: **Knowledge acquisition via ICL sensitive to model size and reasoning complexity.** We evaluate the effectiveness of ICL in robustly reasoning over knowledge graphs provided in context. We test models of different sizes from three families—Phi 4 [1], Gemma 3 [68], and Qwen 3 [76]—on k-hop multiple-choice (top row) and true-false (bottom row) questions, where the number of hops k serves as a proxy for reasoning complexity. Our results show that (a) performance consistently degrades as the number of hops k increases and (b) larger models within each model family exhibit better performance across all k-hop questions.

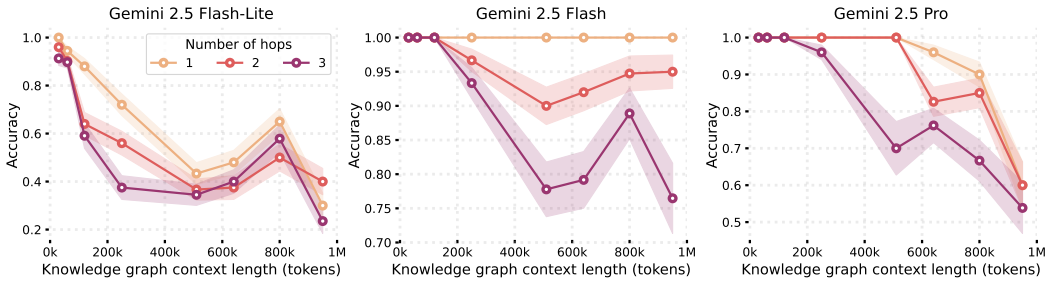

Figure 11: **ICL-based knowledge acquisition does not scale to large knowledge graphs.** We use the Gemini 2.5 family [17] to evaluate the effect of context length—the number of tokens needed to embed the knowledge graph in context—on knowledge acquisition via ICL. The subplots evaluate k-hop performance of Flash-Lite (left), Flash (middle), and Pro (right) as a function of context length. We observe that (a) performance degrades as context length increases, well before reaching the 1M context limit, and (b) higher-hop questions show more severe degradation.

into a single list containing all factual information in the graph. Our experiment in Figure 10 uses a knowledge graph comprising 100 researchers, 10 companies, 10 universities, 10 cities, and 10 research areas, with total context length scaling linearly with the number of relations in the graph.

- **Generating questions.** We sample k-hop random walks from the knowledge graph and convert them into multiple choice, true-false, and cloze questions. To prevent shortcut reasoning, intermediate entities in multi-hop questions are masked with typed placeholders (e.g., "Person PE1"). Our prompt structure presents the textualized graph under an "Information" header, followed by task-specific instructions and the target question, with instructions to encourage multi-hop reasoning via chain-of-thought reasoning.
- **Scaling context length.** We leverage RANK to systematically vary context length by generating knowledge graphs of different sizes, with entity counts ranging from 50 to 3,000 and edge density varying from 1 to 5 coauthors per researcher. When textualized, these graphs produce context lengths spanning from several thousand to one million tokens. We evaluate performance across multiple model families (Phi, Gemma, Qwen) to analyze how both model scale and context length jointly affect robust knowledge acquisition.

C.3 Additional experiments

We verify that reasoning complexity, as measured by the number of hops k, correlates with test-time compute. Figure 12 shows that for both ground-truth chain-of-thought and model-generated responses, the response length increases with k.

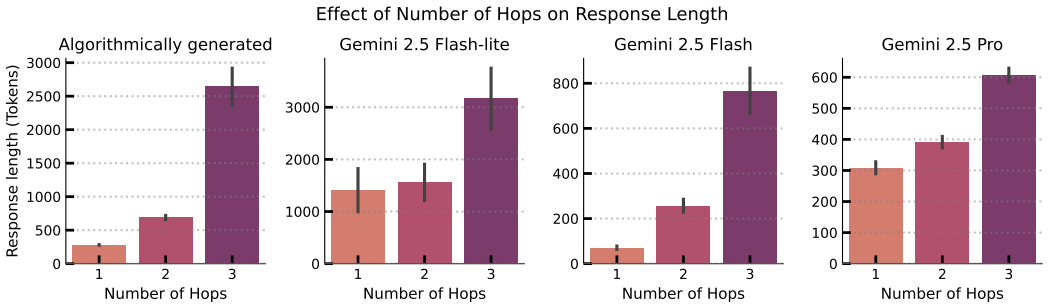

Figure 12: **Test-time compute increases with number of hops k.** We measure the average response length—a proxy for test-time compute–for k-hop multiple-choice questions. The leftmost subplot shows ground-truth CoTs generated using RANK, while the remaining subplots show responses from Gemini 2.5 Flash-Lite, Flash, and Pro models [17]. In all cases, response length consistently increases with the number of hops k, making it an effective proxy for reasoning complexity.

D Evaluating Supervised Fine-tuning with RANK

D.1 Experiment setup

In this section, we describe the knowledge graphs, training procedures, and experimental design used in our SFT experiments.

- **Knowledge graph and dataset configuration.** We use the smaller knowledge graph configuration (30 researchers, 10 companies, 10 universities, 10 cities, 10 research areas) described in Appendix B. Our SFT datasets combine all five data types—nodal descriptions, cloze questions, walk descriptions, multiple choice questions, and true-false statements—with systematic control over hop counts and template repetition factors. We generate datasets with maximum hop counts $k \in \{1, 2, 3, 4\}$ and augmentation factors from $1\times$ to $100\times$ to study both in-distribution and out-of-distribution generalization.
- **Training configuration.** We perform supervised fine-tuning on Qwen3 models (1.7B and 4B parameters) using completion-only loss. See Table 1 for details on training hyperparameters.
  - **Optimization**: We use the AdamW optimizer [49] with learning rate 1×10^{-4}, which we selected from a learning rate sweep conducted with our fixed effective batch size of 128. We employ cosine learning rate scheduling with 10% warmup and gradient clipping at norm 1.0.
  - **Infrastructure**: We conduct training on 4-8$\times$ A100 GPUs using distributed data parallel (DDP) with Accelerate and HuggingFace SFT Trainer. We employ gradient checkpointing and gradient accumulation to manage memory constraints across the distributed setup.
  - **Training setup**: We train for a single epoch with maximum sequence length of 2048 tokens and include a 5% Dolly replay buffer to mitigate catastrophic forgetting. We use full fine-tuning rather than LoRA, as preliminary experiments showed LoRA yielded slightly worse results with much slower convergence.
  - **Data processing**: We implement template train-test separation with 75% of templates allocated to training and 25% to testing. We also apply automatic deduplication to remove identical examples from the training data.

Hyperparameter	Value(s)
Model Size	{Qwen3-1.7B, Qwen3-4B}
Max Hop Count	$\{1, 2, 3, 4\}$
Template Augmentation	$\{1, 5, 25, 50, 100\}\times$
Learning Rate	$\{5 \times 10^{-6}, 1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}\}$
Training Epochs	1
Effective Batch Size	128
Max Sequence Length	2048
Optimizer	AdamW
LR Scheduler	Cosine with 10% warmup
Weight Decay	0
Gradient Clipping	1.0
Template Train Ratio	75%
Dolly Replay Buffer	5%
LoRA	Disabled
Completion-only Loss	Enabled

Table 1: **Hyperparameter configuration used in SFT experiments.** Training settings for Qwen3 models across different hop counts (1-4) and template augmentation factors (1×-100×), including optimization details, data processing parameters, and experimental design choices for multi-hop reasoning tasks.

D.2 Additional experiments

We conduct additional experiments to analyze the impact of chain-of-thought reasoning and single-hop data limitations on model robustness. Figure 13 shows that training without CoT leads to poor generalization, while Figure 14 demonstrates that models trained only on 1-hop data fail at multi-hop reasoning.

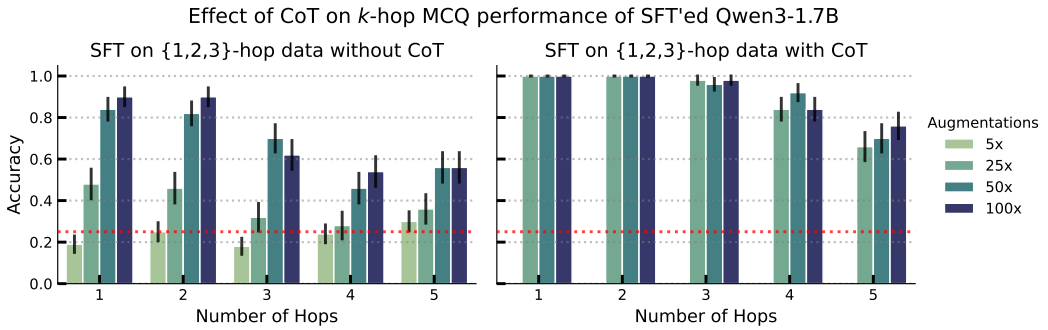

Figure 13: **SFT without CoT hurts robust knowledge acquisition.** We compare SFT using Qwen-1.7B models trained on $\{1, 2, 3\}$-hop data with (right) and without (left) CoT-like rationales. Each subplot shows accuracy (y-axis) versus number of hops $k \in [1, 5]$ (x-axis), with darker bars representing more augmentation ($5\times$ to $100\times$) and the gray line indicating random chance. Our results demonstrate that CoT solutions are crucial for robust acquisition, as training without them leads to poor performance on all k-hop questions.

Figure 14: **Models fine-tuned only on 1-hop data lack robustness.** We evaluate the effect of SFT on k-hop multiple-choice (top) and true-false (bottom) questions using two models: Qwen3-1.7B (left) and Qwen3-4B (right). Each subplot shows accuracy on the y-axis and number of hops $k \in [1, 5]$ on the x-axis, with darker colored bars representing more data augmentation ($1\times$ to $100\times$) and the horizontal line indicating random choice performance. Our results reveal two key findings: (a) models require substantial augmentation ($50\times$–$100\times$) to achieve near-perfect 1-hop retrieval performance, and (b) even with maximum augmentation, performance on multi-hop reasoning ($k > 1$) remains at random chance levels across both model sizes, revealing that training exclusively on atomic facts lacks the robustness needed for compositional reasoning.