Do Larger Language Models Imply Better Generalization? A Pretraining Scaling Law for Implicit Reasoning

Xinyi Wang [1] **Shawn Tan** [2] **Mingyu Jin** [3] **William Yang Wang** [1] **Rameswar Panda** [2] **Yikang Shen** [2]

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks requiring complex reasoning. However, the effects of scaling on their reasoning abilities remain insufficiently understood. In this paper, we introduce a synthetic multihop reasoning environment designed to closely replicate the structure and distribution of real-world large-scale knowledge graphs. Our reasoning task involves completing missing edges in the graph, which requires advanced multi-hop reasoning and mimics real-world reasoning scenarios. To evaluate this, we pretrain language models (LMs) from scratch solely on triples from the incomplete graph and assess their ability to infer the missing edges. Interestingly, we observe that overparameterization can impair reasoning performance due to excessive memorization. We investigate different factors that affect this U-shaped loss curve, including graph structure, model size, and training steps. To predict the optimal model size for a specific knowledge graph, we find an empirical scaling that linearly maps the knowledge graph search entropy to the optimal model size. This work provides new insights into the relationship between scaling and reasoning in LLMs, shedding light on possible ways to optimize their performance for reasoning tasks.

1. Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks. Recently, the reasoning capacity of LLMs has drawn a lot of attention as it is highly correlated with LLMs' performance on many complex real-world tasks (Wei et al., 2022a; Guo et al., 2025). While the reasoning capability is usually enhanced during the post-training stage, it is reasonable to assume that LLMs have already acquired the capability during the pretraining stage, as the post-training is on a significantly smaller scale than pretraining. Some recent work has investigated the possible mechanism of LLMs learning to reason through next-token prediction pretraining (Zhu et al., 2024; Wang et al., 2024a;b). However, the impact of pretraining scaling on LLMs' reasoning ability remains insufficiently understood.

The general scaling behavior of language models has been investigated, including the well-known exponential scaling laws for testing loss and compute proposed by (Kaplan et al., 2020) and the training compute-optimal scaling studied by (Hoffmann et al., 2022a). Recent work has also examined the scaling of specific capabilities like machine translation (Ghorbani et al., 2022) and knowledge capacity/memorization (Allen-Zhu & Li, 2025; Lu et al., 2024). According to these existing scaling laws, it is in general believed that larger models imply better testing loss or task performance when trained on more data.

In this paper, however, we find that the reasoning ability of language models under pretraining scaling can behave differently from normal power-law scaling, in a simplified pretraining environment. More specifically, when given enough compute, we find that the testing loss scaling curve is U-shape and there exists an **optimal model size** that produces the best reasoning performance/testing loss. This implies that overparameterization might hurt reasoning capability during pre-training. We first observe this phenomenon with real-world knowledge graph data, and then systematically study it through synthetically generated data.

We choose to mimic the real-world knowledge structure and distribution with synthetic knowledge graphs (KGs). We define

*Equal contribution [1]UC Santa Barbara [2]MIT-IBM Watson AI Lab [3]Rutgers University. Correspondence to: Xinyi Wang <xinyi wang@ucsb.edu>, Yikang Shen <yikang.shn@gmail.com>.

Accepted at Methods and Opportunities at Small Scale (MOSS), ICML 2025, Vancouver, Canada.

reasoning over world knowledge as completing missing edges in an incomplete knowledge graph, which requires multiple jumps on the knowledge graph according to some pre-defined rules that are latently encoded into the graph generation process. To analyze this, we pretrain LMs from scratch using only triples from the incomplete graph and evaluate their ability to infer missing connections.

As we observed that the **optimal model size** is likely solely determined by the training knowledge graph, we then aim to find an empirical scaling law that can predict the optimal model size with knowledge graph statistics. We then discover a linear relationship between the **optimal model size** and a newly proposed **graph search entropy**, which measures the entropy of performing random searches on a knowledge graph. Roughly, 124 additional parameters in the optimal model size are required per 1-bit entropy increase in searching a knowledge graph.

Our work contributes to the broader understanding of LLM reasoning by shedding light on the intricate relationship between scaling and reasoning capability. Our proposed empirical reasoning scaling law provides possible practical insights for optimizing LLMs' reasoning ability at pretraining time.

2. Preliminaries

While the real-world LLMs are pretrained on large scale text corpus, this corpus can be viewed as encoding a wide range of world knowledge, which in our experiments is simplified to a knowledge graph.

Formally, a knowledge graph G consists of $|G| = N$ triples (e^h, r, e^t), where $e^h \in \mathcal{E}$ is the head entity, $e^t \in \mathcal{E}$ is the tail entity, and $r \in \mathcal{R}$ is a relation. A simple example of knowledge triple is `(DC, is the capital of, USA)`. These knowledge triples naturally form a graph, with nodes as the entities and each edge labeled with a relation type. We denote the total number of entities or nodes by $|\mathcal{E}| = N_e$ and the total number of edge or relation types by $|\mathcal{R}| = N_r$. Then a corpus constructed from this knowledge graph would consist of N data points. The objective of a language model with parameter θ trained on this corpus is then:

$$L(\theta) = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} -\log P_\theta(e_i^h, r_i, e_i^t).$$

To evaluate the language model's capability of reasoning over the knowledge graph, we test the language models on a held-out set of triples that are not seen in the training time. Note that all entity and relation types should have been seen during training time and the language model is only tasked to connect missing edges. To eliminate the need to generate the correct form of relation and entity IDs, and to handle the case where multiple correct answers exist, we design the testing set to be 10-option multiple-choice questions: the language model is tasked to choose the correct tail entity given the head entity and the relation. We ensure that there is only one correct answer among the given 10 options. Suppose there are M questions in the testing set.[1] For a ground truth triple (e^h, r, e^t), we design 9 distracting options $e^{(1)}, e^{(2)}, ..., e^{(9)}$. Then we use the test accuracy $\text{Acc}(\theta, G)$ and testing loss $\ell(\theta, G)$ to evaluate the reasoning capability of a language model θ over the knowledge graph G:

$$\hat{e}_i = \arg \max_{e \in \{e_i^t, e_i^{(1)}, e_i^{(2)}, ..., e_i^{(9)}\}} P_\theta(e | e_i^h, r_i),$$

$$\text{Acc}(\theta, G) = \sum_{i=1}^{M} \mathbb{1}[\hat{e}_i = e_i^t]/M, \qquad \ell(\theta, G) = \sum_{i=1}^{M} -\log P_\theta(e_i^t | e_i^h, r_i)/M.$$

3. Real-world Experiments

In our initial sets of experiments, we investigate the reasoning scaling effect using a real-world knowledge graph, FB15K-237 (Toutanova & Chen, 2015). FB15K-237 is sampled from FB15K (Bordes et al., 2013), which is a dataset adapted from the Freebase knowledge base (Bollacker et al., 2007), a web-scale knowledge base released by Google. FB15K-237 contains $N_e = 14,505$ entities, $N_r = 237$ relations, and $N = 310,116$ knowledge triples.

In Figure 1, we show different-sized language models trained on FB15K-237 with different numbers of steps. When trained with the same number of steps, we observe a slight reasoning performance drop when using larger models. So we then look

[1] We fix $M = 1000$ for all of our experiments.

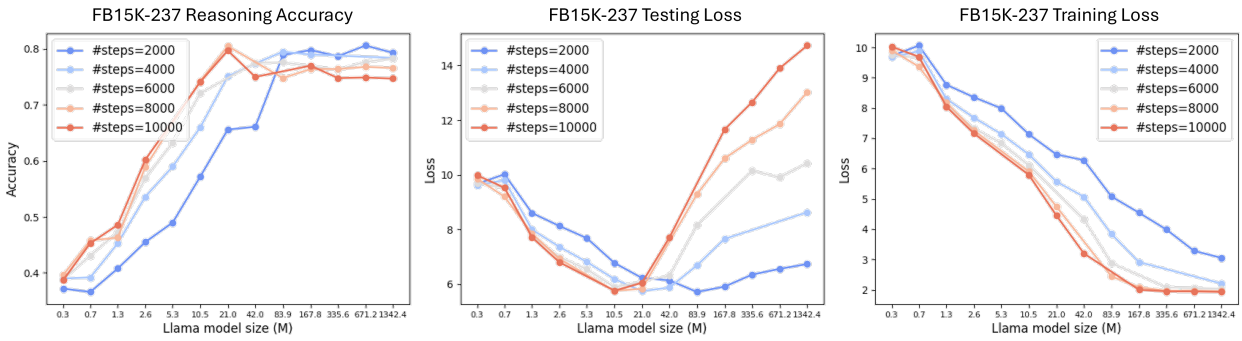

Figure 1. The multiple-choice accuracy/loss on unseen triples of different-sized language models trained on a real-word knowledge graph FB15K-237. The left panel (trained with 10k steps) shows that the testing accuracy decreases after a certain model size. The middle panel shows U-shape loss curves of language models trained with different number steps. The right panel shows Note that the model size on x-axis is in log scale.

at the testing loss on these datasets and observe a U-shape trend with respect to the model size. This observation contradicts the previous belief that larger models yield a smaller testing loss. The training loss decreases monotonically with respect to model size.

This implies that a language model can overfit to the training data when it is overparameterized for the underlying reasoning structure. Such deviation from the traditional scaling law has also been reported in broken neural scaling law (Caballero et al., 2023) which proposed a double-descent-like (Nakkiran et al., 2020) function form instead of a monotonic power-law form. There have also been observations of tasks with inverse scaling (Wei et al., 2023) for large language models.

In this paper, we mainly focus on the scaling of model size. Instead of only scaling the size of the training data, we explore different possible ways of generating the knowledge graph and studying the effect of overall graph complexity on the model reasoning performance. In the following sections, we will mostly focus on understanding the "turning point" of the reasoning loss. More specifically, we want to understand what is the **optimal model size** that can obtain the smallest possible reasoning testing loss. Due to the space limitation, we defer the description of synthetic knowledge graph generation process into Appendix A, and the detailed analysis of the effect of different components in the data generation process into Appendix C.

4. Scaling Laws

In this section, we investigate the scaling law of language models trained on different synthetic knowledge graphs. We first propose an information-theoretical way to measure the overall reasoning complexity of a knowledge graph, which we call the **graph search entropy**, and then relate this linearly with the **optimal model size**. i.e. the model size that obtains the lowest possible testing loss.

From Appendix C, we hypothesize that the optimal model size is positively related to the overall complexity of the knowledge graph. Thus, we propose that we measure the complexity of a knowledge graph by quantifying the amount of information that can be obtained from the graph by exploring the graph through a random search. From our task definition, to reason over the knowledge graph, the language model needs to (a) identify the set of logic rules by observing repetitive patterns; (b) traverse the graph using one or more specific logic rules to locate the tail entity. So we define the **graph search entropy** as the maximum amount of information that can be obtained when randomly traversing the graph.

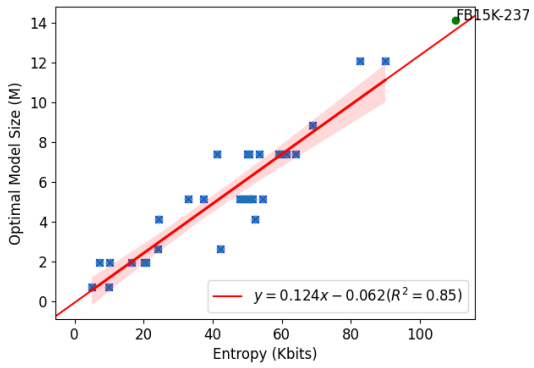

Figure 2. The optimal model size with the lowest possible testing loss v.s. the graph search entropy. The red line is the linear regression line using data from the synthetic experiments (blue squares), with a 95% confidence interval. We also plot the graph search entropy and optimal model size from the real-world FB15K-237 experiment (green dot) to verify the accuracy of the obtained linear scaling law.

To simplify the problem, we first focus on the average amount of information we can observe at one node of the graph. If we consider a random walk over a knowledge graph, then we refer to the entropy produced by each step/node on the walk trace for an infinitely long random walk as the *entropy rate* of this random walk. For a graph G, the maximum entropy rate is equal to the log of the largest eigenvalue of the adjacency matrix A. Note that only consider the entropy rate with respect to the entity, without considering the entropy rate with respect to the relation. We can compute the relation entropy rate with the stationary distribution and transition matrix induced by the maximal entropy rate random walk. If we denote the dominating eigenvalue by $\lambda \in \mathbb{R}$ and the corresponding eigenvector by $\psi \in \mathbb{R}^{N_e}$, then the stationary distribution $\rho \in \mathbb{R}^{N_e}$ can be written as $\rho_i = \psi_i / ||\psi||_2^2$.

The transition matrix $S \in \mathbb{R}^{N_e \times N_e}$ of the maximal entropy random walk can be written as $S_{ij} = (A_{ij}/\lambda)(\psi_j/\psi_i)$. We can then transform the entity-to-entity transition matrix $S \in \mathbb{R}^{N_e \times N_e}$ into an entity-to-relation transition matrix $S^r \in \mathbb{R}^{N_e \times N_r}$ by merging the entries with the same relation together:

$$S^r_{ij} = \sum_{k=1}^{N_e} \mathbb{1}[(i,j,k) \in G] S_{ik}.$$

Finally, the relation entropy rate $H^r(G)$ can be written as:

$$H^r(G) = -\sum_{i=1}^{N_e} \rho_i \sum_{j=1}^{N_r} S^r_{ij} \log(S^r_{ij}).$$

The overall **graph search entropy** $H(G)$ can then be written as the sum of the entity entropy rate and the relation entropy rate multiplied by the number of nodes:

$$H(G) = N_e(\log(\lambda) + H^r(G)).$$

We empirically investigate the relation between the optimal model and the graph search entropy by plotting them against each other in Figure 2, and perform linear regression. The optimal model sizes are obtained from the synthetic experiments conducted in the ablation studies. In the ablation studies we only report the results for exponentially increasing model sizes for clarity. In this study to better capture the optimal model size, we make the model sizes near the optimal model size more fine-grained.

We find a strong linear relation between the optimal model size and the graph search entropy with $R^2 = 0.85$. Note that there are a few sources of noise for locating the optimal model size for a specific knowledge graph. First, we only train language model with selected sizes due to compute and time limitations, and the quantization of the model size would disrupt the smoothness of the scaling law. Second, the exact location of the optimal model size is dependent on the training steps, which we did not thoroughly traverse but choose to inspect at the training step 10k.

After fitting a linear regression line using the data from our synthetic experiments, we check the validity of this empirical scaling law against our real-world knowledge graph, FB15K-237. We calculate the graph search entropy for FB15K-237, and find the predicted optimal model size is very close to the observed optimal model size, shown as a green dot in Figure 2.

From our scaling law, we can see that roughly 124 additional parameters in the optimal model size are required per 1-bit entropy increase in the knowledge graph. That is a language model can only reliably (not perfectly) reason over 0.008 bit information per parameter. Due to the space limit, we defer the discussion of our proposed scaling law in comparison to the knowledge scaling law proposed by Allen-Zhu & Li (2025) to **??**.

This is very different from the knowledge capacity scaling law concluded by (Allen-Zhu & Li, 2025), which shows that the language model can store 2 bits of knowledge per parameter. This discrepancy is due to two reasons: first, our scaling law is not only about memorizing the knowledge, but also about reasoning over the learned knowledge, which is significantly harder. Second, the way we compute the graph search entropy is fundamentally different from the way (Allen-Zhu & Li, 2025) computes the knowledge entropy. While (Allen-Zhu & Li, 2025) describes the entropy of the knowledge generation process, our graph search entropy describes the entropy of randomly traversing a fixed knowledge graph. In this way, we did not directly measure the amount of information that a language model needs to memorize, but measured the complexity of traversing, and therefore, reasoning over a graph. It is hard, if not impossible, to obtain the data generation process of real-world data, but it is possible to get an estimate of the underlying knowledge graph of a corpus through automated

knowledge graph construction algorithms (Zhong et al., 2023). Thus, it is possible to predict the optimal reasoning model size for real-world pretraining, by first constructing a knowledge graph from the pretraining corpus, and then computing its graph search entropy, and finally using a similar scaling law to calculate the optimal model size.

5. Limitations

We want to highlight that this study is only conducted on simplified pretraining data from knowledge graphs, and the results are likely not directly applicable to real-world language model pretraining with large text corpus. The setting of our study provides a reasonable analogy to the real-world language model pretraining, and the obtained insight might be found useful in the real world when the compute is abundant with very large models and very large datasets that exhaustively traverse the underlying knowledge graph. We leave the work of verifying our scaling law in the real word to future research due to its resource-demanding nature.

6. Related Work

Language Model Scaling Laws (Kaplan et al., 2020) first observed a power-law relationship between LLM perplexity, model parameter count, and training data size, laying the foundation for scaling law research. Subsequently, (Hoffmann et al., 2022b) explored optimal training strategies under constrained computational resources and discovered that LLM parameter size and the number of training tokens should scale proportionally to achieve optimal compute efficiency under a fixed budget. Beyond pretraining performance, researchers further confirmed that downstream task performance can also be reliably predicted based on model size and training data volume (Hernandez et al., 2021; Isik et al., 2024). (Allen-Zhu & Li, 2025; Lu et al., 2024) have turned to exploring more specific capability dimensions, focusing particularly on the scaling laws of factual memory in LLMs and their behavioral patterns when memorizing different types of facts. Most recently, (Roberts et al., 2025) have confirmed that scaling laws are skill-dependent, and found that knowledge-intensive tasks are more parameter-hungry while reasoning-intensive tasks are more data-hungry. (Springer et al., 2025) challenge a core assumption in scaling research—that more pretraining invariably leads to better downstream performance. Our paper identifies a different U-shaped scaling curve under the specific scenario of knowledge graph reasoning and reveals that the search complexity of the knowledge graph determines the optimal model size. This echoes the discovery of (Pandey, 2024) and (Yin et al., 2024) that classic scaling laws are highly dependent on the data complexity or the compression ratio of the data. (Havrilla & Liao, 2024) also confirmed from both theoretical and empirical perspectives that the power of the power scaling law depends on the intrinsic dimension of the training data.

Language Model Reasoning Our paper focuses on the reasoning capability of language models which has drawn a lot of attention recently (Zhang et al., 2023; Chen et al., 2023; Yao et al., 2023a;b; Wang et al., 2023; Guo et al., 2025; Jin et al., 2024; Yeo et al., 2025; Team et al., 2025; Li et al., 2025). LLMs usually reason in a step-by-step manner in real-world tasks like math word problems (Wei et al., 2022b). In our experiments, we do not ask language models to generate a step-by-step solution for its answer, but ask the language model to directly choose the correct answer from the given options, because our pretrain-only language models are not trained to give a step-by-step solution for a query. Our synthetic reasoning environment is the most similar to (Wang et al., 2024b), which also use the knowledge graph completion task as a testbed to understand how language models learn to reason at pretraining time. They propose that language models are able to aggregate random walk paths sampled from the knowledge graph. (Wang et al., 2024a; Zhu et al., 2024) also employ a graph structure to ground their synthetic reasoning tasks to explain how LLMs reason, but their reasoning is defined as concatenations of relations: A is r_1 to B and B is r_2 to C implies A is $r_1 r_2$ to C. The knowledge graph completion task we employ is more complex than simple concatenation of relations as the language model needs to find out which relation $r_1 r_2$ corresponds to from the knowledge graph.

7. Conclusion

This paper investigates reasoning scaling in language models trained on knowledge graphs. Our results reveal a U-shaped relationship between model size and reasoning performance, where overparameterization leads to excessive memorization and degraded reasoning ability. We identify key factors that determine the optimal model size, such as the number of training triples and graph complexity. Notably, we propose an empirical scaling law linking optimal model size to graph search entropy, offering a quantitative guide for model design. While our experiments are conducted in controlled settings, these insights pave the way for future work in real-world pretraining scenarios and improved reasoning capabilities in LLMs.

References

Allen-Zhu, Z. and Li, Y. Physics of language models: Part 3.3, knowledge capacity scaling laws. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=FxNNiUgtfa.

Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

Bollacker, K., Cook, R., and Tufts, P. Freebase: a shared database of structured general human knowledge. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pp. 1962–1963. AAAI Press, 2007. ISBN 9781577353232.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

Caballero, E., Gupta, K., Rish, I., and Krueger, D. Broken neural scaling laws. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sckjveqlCZ.

Chen, W., Ma, X., Wang, X., and Cohen, W. W. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023.

Ghorbani, B., Firat, O., Freitag, M., Bapna, A., Krikun, M., Garcia, X., Chelba, C., and Cherry, C. Scaling laws for neural machine translation. In *International Conference on Learning Representations*, 2022.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Havrilla, A. and Liao, W. Understanding scaling laws with statistical and approximation theory for transformer neural networks on intrinsically low-dimensional data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=N2wYPMpifA.

Hernandez, D., Kaplan, J., Henighan, T., and McCandlish, S. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022a.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 30016–30030, 2022b.

Isik, B., Ponomareva, N., Hazimeh, H., Paparas, D., Vassilvitskii, S., and Koyejo, S. Scaling laws for downstream task performance of large language models. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*, 2024.

Jin, M., Yu, Q., Shu, D., Zhao, H., Hua, W., Meng, Y., Zhang, Y., and Du, M. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 1830–1842, 2024.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Li, Z.-Z., Zhang, D., Zhang, M.-L., Zhang, J., Liu, Z., Yao, Y., Xu, H., Zheng, J., Wang, P.-J., Chen, X., et al. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*, 2025.

Lu, X., Li, X., Cheng, Q., Ding, K., Huang, X.-J., and Qiu, X. Scaling laws for fact memorization of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 11263–11282, 2024.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1g5sA4twr.

Pandey, R. gzip predicts data-dependent scaling laws. *arXiv preprint arXiv:2405.16684*, 2024.

Roberts, N., Chatterji, N., Narang, S., Lewis, M., and Hupkes, D. Compute optimal scaling of skills: Knowledge vs reasoning. *arXiv preprint arXiv:2503.10061*, 2025.

Springer, J. M., Goyal, S., Wen, K., Kumar, T., Yue, X., Malladi, S., Neubig, G., and Raghunathan, A. Overtrained language models are harder to fine-tune. *https://arxiv.org/abs/2503.19206*, 2025.

Team, K., Du, A., Gao, B., Xing, B., Jiang, C., Chen, C., Li, C., Xiao, C., Du, C., Liao, C., et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Toutanova, K. and Chen, D. Observed versus latent features for knowledge base and text inference. In Allauzen, A., Grefenstette, E., Hermann, K. M., Larochelle, H., and Yih, S. W.-t. (eds.), *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pp. 57–66, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-4007. URL https://aclanthology.org/W15-4007/.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Wang, B., Yue, X., Su, Y., and Sun, H. Grokking of implicit reasoning in transformers: A mechanistic journey to the edge of generalization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=D4QgSWxiOb.

Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K.-W., and Lim, E.-P. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2609–2634, 2023.

Wang, X., Amayuelas, A., Zhang, K., Pan, L., Chen, W., and Wang, W. Y. Understanding reasoning ability of language models from the perspective of reasoning paths aggregation. In *Forty-first International Conference on Machine Learning*, 2024b.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022a. ISSN 2835-8856. URL https://openreview.net/forum?id=yzkSU5zdwD. Survey Certification.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in neural information processing systems*, volume 35, pp. 24824–24837, 2022b.

Wei, J., Kim, N., Tay, Y., and Le, Q. V. Inverse scaling can become u-shaped. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=19sGqVUxQw.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. R. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023b.

Yeo, E., Tong, Y., Niu, M., Neubig, G., and Yue, X. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

Yin, M., Wu, C., Wang, Y., Wang, H., Guo, W., Wang, Y., Liu, Y., Tang, R., Lian, D., and Chen, E. Entropy law: The story behind data compression and llm performance. *arXiv preprint arXiv:2407.06645*, 2024.

Zhang, Z., Zhang, A., Li, M., and Smola, A. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhong, L., Wu, J., Li, Q., Peng, H., and Wu, X. A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 56(4):1–62, 2023.

Zhu, H., Huang, B., Zhang, S., Jordan, M., Jiao, J., Tian, Y., and Russell, S. J. Towards a theoretical understanding of the'reversal curse'via training dynamics. *Advances in Neural Information Processing Systems*, 37:90473–90513, 2024.

A. Synthetic Data Construction

To investigate how the underlying knowledge structure influences language models' reasoning performance, we propose an algorithm to generate synthetic knowledge graphs that mimic real-world knowledge graphs. More specifically, we assume that the knowledge graph generation process is governed by a set of logical rules.

For example, a rule for inferring the `locatedIn` relation can be (e_1, `locatedIn`, e_2) $\leftarrow$ (e_1, `neighborOf`, e_3) $\wedge$ (e_3, `locatedIn`, e_2). Formally, for a target relation r, we consider logic rules with conjunctive form. For $\forall\{e_i\}_{i=0}^{n} \subset \mathcal{E}$,

$$(e_0, r, e_n) \leftarrow (e_0, r_1, e_1) \wedge ... \wedge (e_{n-1}, r_n, e_n),$$

where $(e_{i-1}, r_i, e_i) \in \mathcal{G}$. We abbreviate such rule by $h(r) = [r_1, r_2, ..., r_n]$. We randomly generate a set of logical rules $\mathcal{H}$ and ensuring there is no cycles in the set. To grow a graph that follows these rules, we enforce sparsity of the possible relation types connecting to and branching out each entity. More specifically, we define *node types* based on the possible relation types connecting to and branching out each entity, based on the generated rules, as illustrated in Figure 3. Such sparsity is also observed in real-world knowledge graphs.

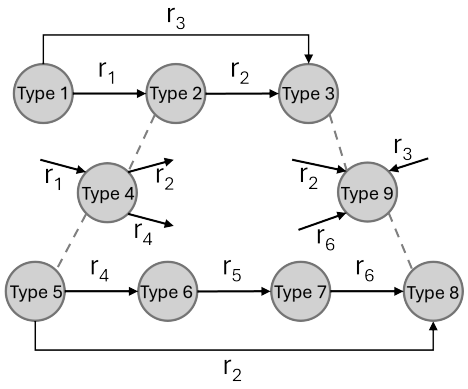

Figure 3. Nine possible node types generated by two logical rules. Each entity position in a rule would create a new entity type. Each relation shared between two rules would also create two new entity types.

Our random graph generation process is inspired by the preferential attachment process (Barabási & Albert, 1999), which is used for generating scale-free networks with a power-law distribution for the degrees of the nodes. Intuitively, preferential attachment implies a "the rich get richer" approach to edge placement in the graph. Each time a new node is added to the graph, there is a 'preference' to connect to the nodes that are already highly connected, with a probability proportional to the target node's degree. Since we have observed the scale-free property in real-world knowledge graphs and the internet is known to be a scale-free network, we adopt a preferential attachment based graph generation process. To accommodate different relation types assigned to each edge, we maintain a degree distribution for each relationship and add new edges according to preferential attachment.

The code for our random graph generation algorithm is shown in the Jupyter notebook. In summary, we first randomly generate a set of rules $\mathcal{H}$, with the number of rules $|\mathcal{H}| = N_h$ and the range of rule length $[L_{min}, L_{max}]$ as hyperparameters. Then we generate all possible node types as illustrated in Figure 3, with the maximum number of relations per node M_r as a hyperparameter. We generate a seed graph by instantiating each rule with a set of new entities. To this, we incrementally add one new entity until the number of entities reaches N_r, by first randomly assigning a node type to it, and then randomly sampling the m relation types from the set of relations defined by the node type. We choose the target of these m new edges by preferential attachment. After adding every K entities, we search through the current graph to add any edges that can be inferred through the logic rules defined in $\mathcal{H}$. We call the triples that can be deduced through a logic rule by *deductible triples*, otherwise *atomic triples*.

Finally, we limit the number of training triples to N and ensure that the the ratio between the number of deductible triples and atomic triples to γ by subsampling the generated graph. We also further ensure that the triples in the held-out test set are all deductible through the training triple. In this way, we can generate synthetic knowledge graphs with specific sizes and complexity.

B. Experiment Details

To eliminate confounding variables and information contained in the lexical form of the entity and relation names, we label each entity and relation with a random ID and tokenize the IDs by characters. We use the LlaMA (Touvron et al., 2023) model architecture to implement LMs of different sizes by adjusting the hidden dimensions and the number of layers, as shown in Table 1. In all experiments, we keep the training hyperparameter the same, with 10k train steps, as shown inTable 2.

Note that, at training time, we repeat the training triples for many epochs (e.g. 30 times for FB15K-237) to find the optimal model size. This graph epoch is different from the real-world cases where we repeat the whole pretraining corpus for certain

epochs. Because we can view each triple in the graph as a piece of factual knowledge (e.g. Barack Obama's wife is Michelle Obama), this knowledge is usually repeated many times in a pretraining text corpus, in many different forms. Therefore, although our models have seen the same triple many times during training, the same piece of factual knowledge could also have been repeated several times in one pass of a real-world pretraining corpus.

Model size	hidden size	MLP size	#attention heads	#layers
0.3M	128	256	2	2
0.7M	128	256	2	4
1.3M	256	512	4	2
2.6M	256	512	4	4
5.3M	256	512	4	8
10.5M	512	1024	8	4
21.0M	512	1024	8	8
42.0M	512	1024	8	16
83.9M	1024	2048	16	8
167.8M	1024	2048	16	16
335.6M	1024	2048	16	32
671.2M	2048	4096	32	16
1342.4M	2048	4096	32	32

Table 1. Language model (Llama) size details

batch size	lr	lr scheduler	warmup ratio	weight decay	max length
1024	1e-4	cosine	0.2	0	128

Table 2. Hyperparameter settings for language model pretraining.

	N	N_e	N_r	N_h	γ
(a)	100k	10k	100	50	0.5
(b)	10k/20k/.../100k	10k	100	50	0.5
(c)	100k	10k	100	5/10/.../50	0.5
(d)	100k	10k	10/20/.../100	50	0.5
(e)	100k	10k	100	50	0.1/0.5/.../0.9
(f)	10k/20k/.../100k	1k/2k/.../10k	10	5	0.5

Table 3. Knowledge graph hyperparameter settings for Figure 4 experiments. We keep $L_{min} = 2$ and $L_{max} = 4$ for all experiments. Here N denotes the number of triples, N_e denotes the number of entities, N_r denotes the number of relations, N_h denotes the number of rules, γ denotes the ratio between deductible triples and atomic triples, L_{min} denotes the minimum rule length, and L_{max} denotes the maximum rule length.

C. Graph Generation Ablation

We investigate important factors that affect the U-shape scaling of reasoning loss versus language model size. Our important findings can be summarized as follows:

- The minimum reasoning loss/maximum reasoning accuracy that a language model can reach is capped by the training data, regardless of the training steps and model size.
- The optimal model size for a training corpus is largely fixed regardless of the training steps when the number of training steps is large enough.
- When the underlying knowledge graph is fixed, training on more data sampled from the graph increases the optimal model size and reasoning performance.

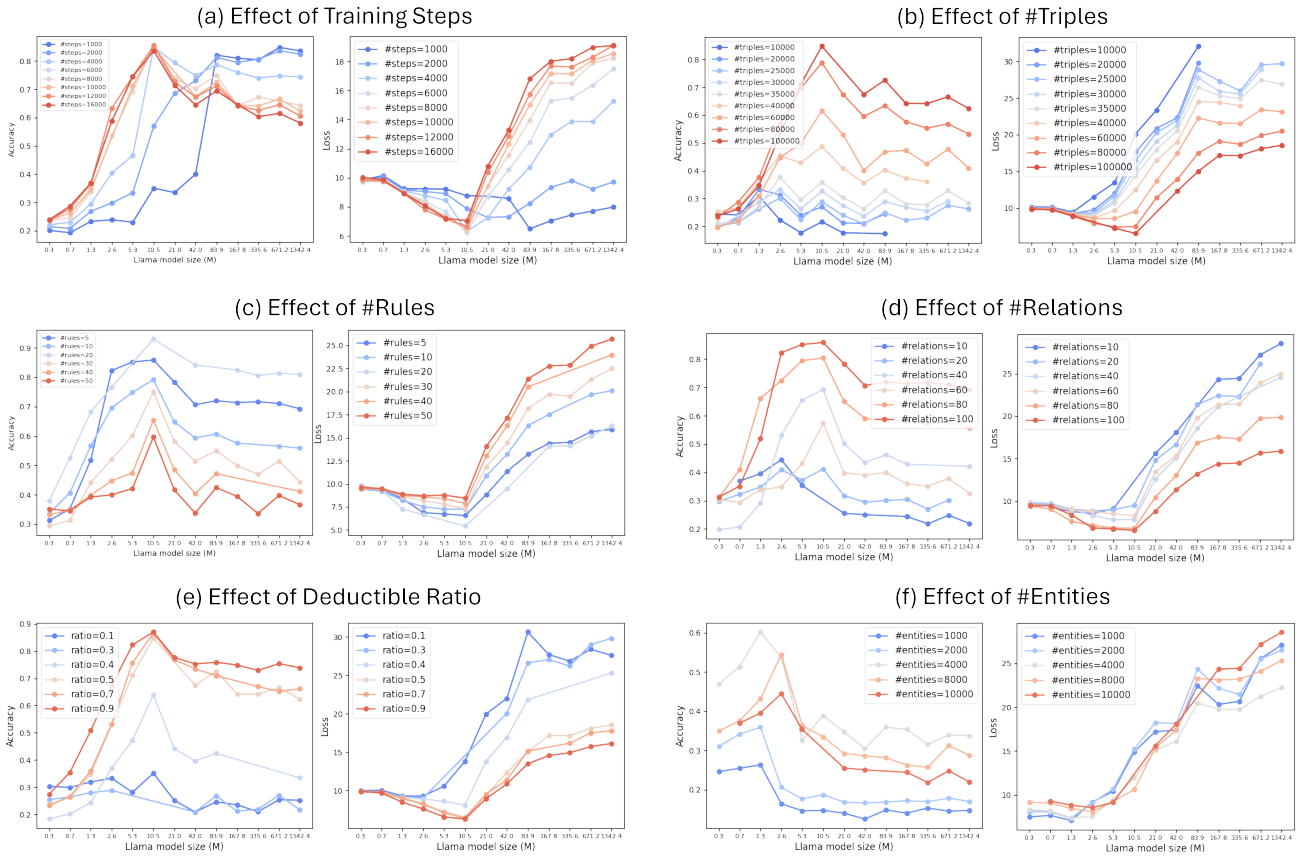

Figure 4. We show the effect of different hyperparameters of the synthetic knowledge graph generation process. In each experiment, we keep all other parameters the same and only change one hyperparameter. We show the effect with both the testing accuracy (left) and the testing loss (right) as the y-axis, with different model sizes as the x-axis in log scale.

- More complex knowledge graph implies a larger optimal model size.

We study the effects of the following four hyperparameters of graph data generation: the number of triples N, the number of entities N_e, the number of relations N_r, and the number of rules N_h. We fix all training hyperparameters as specified in the Appendix B but study the effect of training steps, as according to our preliminary experiments it has the largest effect on the optimal model size. The detailed data generation configuration for each set of experiments can also be found in the Appendix B.

Stable optimal model size with respect to training steps. In Figure 4 (a), we show the effect of training language models on the same knowledge graph with different numbers of training steps. As mention in the last part of Section 2, the optimal model size becomes smaller when the number of training steps increases, and then becomes stable after 4k steps. Another observation is regardless of the number of training steps, the maximum accuracy or minimum loss is stable. While we have ensured that all testing triples can be deduced through the training triples, there seems to be a performance cap determined solely by the knowledge graph data, which is unaffected by model size. In all following experiments, we train all models for 10k steps.

More triples implies a larger optimal model size. In Figure 4 (b), we show the effect of the number of unique triples N sampled after the same knowledge graph generation process. This setting is arguably the most similar to the real-world pretraining of language models: the underlying world knowledge graph of all the pretraining corpora is largely stable, and training data are realizations of the underlying knowledge graph and so the sizes of different corpora are simply a result of subsampling/upsampling the knowledge in the existing graph. We can see that a larger number of training triples results in a larger optimal model size and a better reasoning performance. This observation aligns with the classic scaling laws. However, there exists an optimal model size for the full knowledge graph: after sampling beyond the size of the full

knowledge graph, you can only sample previously seen knowledge. In this case, the optimal model size would be stable no matter the training data size.

Number of rules does not impact optimal model size. In Figure 4 (c), we show the effect of generating knowledge graphs of the same size with different numbers of rules N_h. More rules mean that the testing triples need to be solved in more ways. The number of rules does not have a significant effect on the optimal model size, but affects the reasoning performance. There appears to be an optimal number of rules (20) that results in the best performance. This is because more rules increases the complexity of solving the test set while fewer rules increases the ambiguity in the training set. i.e. a relation may be be deduced through correlations outside of the predefined rules. The reason why the number of rules does not affect the optimal model size is likely because it does not significantly impact the graph search entropy. This is discussed in detail in Section 4.

More relations imply a larger optimal model size. In Figure 4 (d), we show the effect of generating knowledge graphs of the same size and the same number of rules with different numbers of relations N_r. While the rules used for deducing the testing set remain the same for all experiments, there are additional relations that may not be used by any of the rules. We construct knowledge graphs with an excessive number of relations by adding additional relation patterns. In general, more relations improves the best reasoning performance while increasing the optimal model size. More relations increases the complexity of the knowledge graph, and thus increases the optimal model size. On the other hand, as discussed in the previous experiment, a small number of rules along with a small number of relations increases the ambiguity in the training set. By adding dummy relations that are not used for reasoning, the language model can better distinguish between the logic rules and spurious correlations between relations. Thus the reasoning performance improves with more relations.

The optimal model size increases with the deductible ratio when the ratio is small. In Figure 4 (e), we show the effect of generating knowledge graphs with different ratios between deductible triples and atomic triples, γ, while keeping the number of entities and the number of triples unchanged. A larger ratio implies that the language model can see more rule patterns at training time, thus improving the reasoning performance. The increase in performance and optimal model size stops after a ratio threshold.

More entities imply a larger optimal model size. In Figure 4 (f), we show the effect of generating knowledge graphs with different numbers of nodes/entities N_e. In this experiment, we also scale the number of triples to keep all other hyperparameters unchanged. Increasing the number of entities increases the optimal model size while also increasing the testing loss. More entities imply a larger graph which increases the graph complexity, thus increasing the optimal model size. As in this experiment, we use a small number of rule ($N_h = 5$) and relations ($N_r = 10$), an excessive number of entities and triples will create more ambiguity thus hurting the reasoning performance.