A Controlled Study for Memorization and Emergence of Relational Semantics in LLMs

Yihua Zhu[1,3] **Qianying Liu**[3] **Jiaxin Wang**[1] **Fei Cheng**[1] **Chaoran Liu**[3]
Akiko Aizawa[2,3] **Sadao Kurohashi**[1,3] **Hidetoshi Shimodaira**[1,4]
[1]Kyoto University [2]University of Tokyo [3]NII LLMC [4]RIKEN
{zhu.yihua.22h@st, wang.jiaxin.77y@st, feicheng@i, kuro@i, shimo@i}.kyoto-u.ac.jp
{ying, cliu, aizawa}@nii.ac.jp

Abstract

Autoregressive LLMs often excel at relational tasks that connect entities through relation words (e.g., father/son, friend), yet it remains uncertain whether they actually internalize the underlying logical meaning of these relations, such as symmetry and inversion. To investigate this, we introduce a controlled, knowledge-graph-based synthetic setup that renders text from symmetric and inverse triples, trains GPT-style autoregressive models from scratch, and tests memorization, logical inference, and in-context generalization to previously unseen entities. Our results reveal an abrupt transition: with enough logic-relevant supervision, relational semantics emerge sharply even in shallow models with only 2 to 3 layers, and stronger generalization coincides with stable signals in intermediate layers.

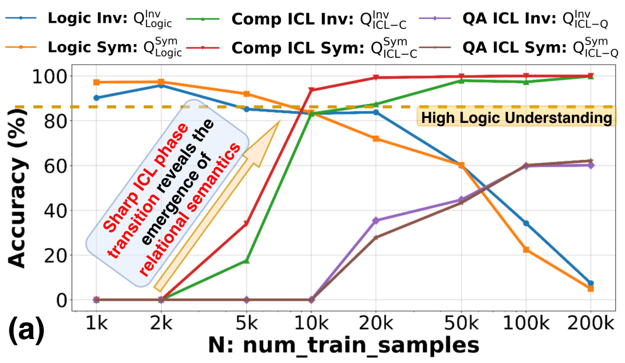

Figure 1: **Relational-word understanding in autoregressive language models.** Models can store symmetric and inverse relational facts and answer forward-format logic QA correctly. Moreover, both completion-based and QA-based in-context tests on unseen entities show an abrupt phase transition in performance once logic-relevant supervision is adequate, suggesting that relational semantics emerge at that point.

1 Introduction

Autoregressive (AR) large language models (LLMs) trained on web-scale data have delivered strong performance on many NLP tasks, such as question answering, information extraction, and reasoning [2, 3, 4, 5]. A core requirement underlying these tasks is the capacity to identify, encode, and operate over relations between entities. However, it is still an open question whether LLMs actually acquire the **logical semantics** conveyed by **relational words**, or whether their apparent competence mainly comes from shallow textual co-occurrence signals.

Relational words play a distinctive role in natural language: in addition to connecting entities within a sentence, they can express abstract and systematic logical structure. For example, some relations are **symmetric** (if *A* is a `friend` of *B*, then *B* is a `friend` of *A*), whereas others come in **inverse** pairs (if *A* is the `father` of *B*, then *B* is the `son` of *A*). In knowledge graphs (KGs), these regularities are explicitly captured as relation properties and function as a central organizing principle for relational reasoning [6, 7]. This leads to our research question: **Can auto-regressive language models memorize relational facts and internalize the logical semantics of relational words, and under what training conditions (e.g., data and model scale) does this ability emerge?**

Answering this question using standard web-scale pretrained LLMs is w. Relational words are often polysemous and context-dependent, and uncontrolled pretraining data

This paper is a revised version of [1].

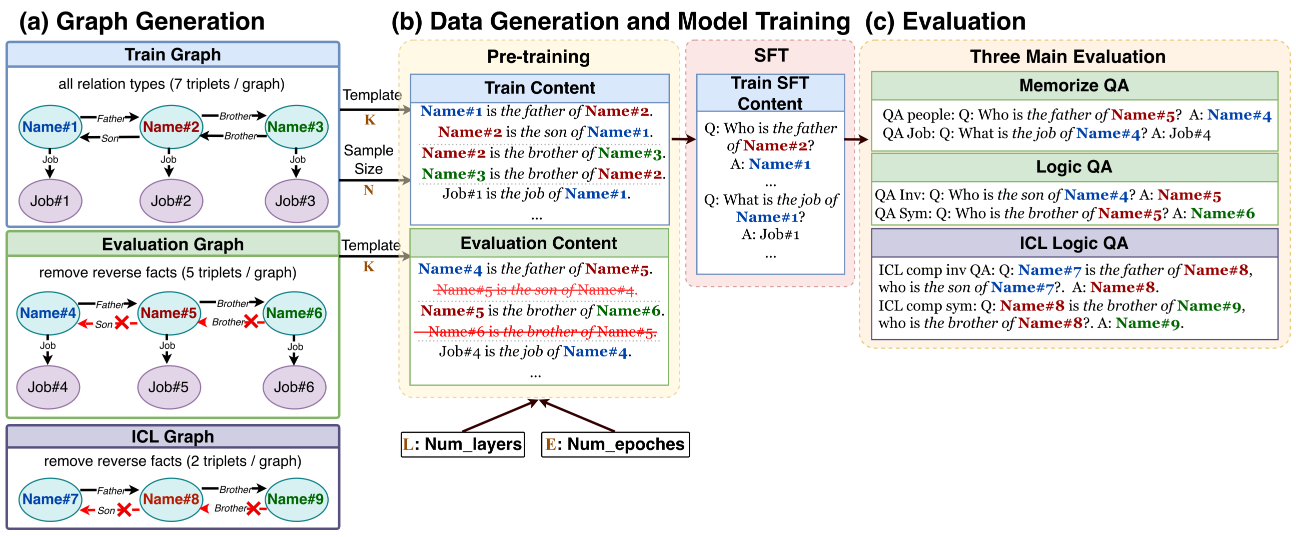
(a) Graph Generation
(b) Data Generation and Model Training
(c) Evaluation
Train Graph
all relation types (7 triplets / graph)
Name#1
Name#2
Name#3
Father
Son
Brother
Brother
Job
Job#1
Job#2
Job#3
Evaluation Graph
remove reverse facts (5 triplets / graph)
Name#4
Name#5
Name#6
Job#4
Job#5
Job#6
ICL Graph
remove reverse facts (2 triplets / graph)
Name#7
Name#8
Name#9
Template K
Sample Size N
Template K
Pre-training
Train Content
Name#1 is the father of Name#2.
Name#2 is the son of Name#1.
Name#2 is the brother of Name#3.
Name#3 is the brother of Name#2.
Job#1 is the job of Name#1.
...
Evaluation Content
Name#4 is the father of Name#5.
~~Name#5 is the son of Name#4.~~
Name#5 is the brother of Name#6.
~~Name#6 is the brother of Name#5.~~
Job#4 is the job of Name#4.
...
L: Num_layers
E: Num_epoches
SFT
Train SFT Content
Q: Who is the father of Name#2?
A: Name#1
...
Q: What is the job of Name#1?
A: Job#1
...
Three Main Evaluation
Memorize QA
QA people: Q: Who is the father of Name#5? A: Name#4
QA Job: Q: What is the job of Name#4? A: Job#4
Logic QA
QA Inv: Q: Who is the son of Name#4? A: Name#5
QA Sym: Q: Who is the brother of Name#5? A: Name#6
ICL Logic QA
ICL comp inv QA: Q: Name#7 is the father of Name#8, who is the son of Name#7?. A: Name#8.
ICL comp sym: Q: Name#8 is the brother of Name#9, who is the brother of Name#8?. A: Name#9.

Figure 2: Overview of the KG-synthetic data generation framework, model training process, and evaluation detail.

introduces the risk of contamination, making it difficult to distinguish genuine logical generalization from memorization [8].

To address these challenges, we propose a fully controlled experimental framework based on synthetic corpora constructed from knowledge-graph triples that explicitly encode relational properties such as symmetry and inversion. This framework enables a systematic examination of relational understanding under multiple evaluation settings. To examine question, under both cloze-style sentence completion and question answering.

Using our controlled framework, as shown in Figure 1, our main findings can be summarized as follows:

1. **Relational fact learning under autoregressive training.** We show that autoregressive LMs can memorize relational facts during pretraining and answer both memorize and logic questions, with memorization capacity increasing with model scale.
2. **Emergence of relational semantics and generalization.** We find that relational semantics emerge with sufficient logic-bearing supervision, enabling generalization to unseen entities. This emergence exhibits a sharp phase transition from zero to near perfect as training data increases, and can occur even in small, shallow (2–3 layers) models.
3. **Layer-wise correlates of relational generalization.** Through layer-wise analysis, we show that successful relational generalization is associated with stable, logic-relevant representations in intermediate layers.

2 Methodology

This section defines our controlled experimental setup, which shown in Figure 2.

2.1 Controllable KG Synthetic Corpus

We can refer to Figure 2 and Appendix B for the complete details of how the synthetic dataset is constructed from the knowledge graph.

2.2 Training Setup

Inspired by [8, 9], we train GPT2-style AR LMs [10] from scratch. Pre-training is conducted on the combined corpus of train and evaluation content. We control four experimental variables: K = `num_template`, N = `num_train_samples`, E = `num_training_epochs`, and L = `num_layers`.

After pre-training, we perform SFT [11] on the SFT corpus (generated from train content only) to equip the model with question-answering capability. No experimental variables are introduced during SFT. More details about training setup see Appendix B.1.

2.3 Evaluation Setup

As shown in Figure 2, we evaluate models with three complementary query sets: memorize QA queries facts explicitly present in Evaluation content, i.e., Q_{Mem}, to measure both memorization of pre-training text and basic QA capability after SFT. Logic QA Q_{Logic} evaluates whether

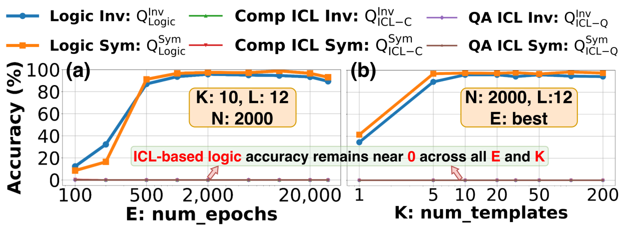

Figure 3: Logic and ICL evaluation. (a) Vary E with K, L, N fixed. (b) Vary K with N, L fixed.

the model can infer relational logic beyond memorization by querying facts that are absent from Evaluation content but logically implied by it. In-context learning (ICL) logic QA Q_{ICL} further tests whether the learned logic generalizes to unseen entities that never appear in Pre-train corpus.

3 Experiment and Result

We test whether GPT2-style auto-regressive LMs can learn the relational-word logical semantics inspired by KGs, such as inversion (e.g., `father/son`) and symmetry (e.g., `friend`).

3.1 Task Setting.

We ask three questions: (1) can the model learn the relational-word logical semantics; (2) under what training conditions (N, K, L, E) does this ability emerge; and (3) how do models that succeed differ internally from those that fail.

Task 1 (main sweeps). We use the same task setting as in Result 1 (the same sweeps over E, K, N, and L), but replace the evaluation with Q_{Logic}, $Q_{ICL\text{-}C}$, and $Q_{ICL\text{-}Q}$.

Task 2 (small-L study). To isolate the effect of model depth, we additionally evaluate shallow models with $L \in \{1, 2, 3\}$. Unless stated otherwise, we fix $K = 10$ and sweep N, reporting both Q_{Logic} and $Q_{ICL\text{-}C}$.

Task 3 (layer-wise analysis). For prompts in $Q_{ICL\text{-}C}$ (e.g., "A is the father of B. B is the son of __"), we extract, at each layer l, the next-token logit, softmax probability, and rank of the correct first token. We then compute layer-wise means over 1,800 prompts.

3.2 Observations.

Task 1. Figure 3 (a, b) shows that for fixed L and N, varying training epochs E or template count K yields a consistent pattern: Q^{Inv}_{Logic} and Q^{Sym}_{Logic} rise with training and reach stable peak accuracy above 95%, while both $Q_{ICL\text{-}C}$ and $Q_{ICL\text{-}Q}$ remain near 0%.

In contrast, when sweeping data scale N with fixed L and K (Figure 1 (a)), Q^{Inv}_{Logic} and Q^{Sym}_{Logic} decrease as N increases and drop below 5% at $N = 200{,}000$. Meanwhile, in-context performance improves nonlinearly: $Q_{ICL\text{-}C}$ exhibits a clear **phase transition** when N exceeds 2,000, and $Q_{ICL\text{-}Q}$ shows a phase transition once N exceeds 20,000.

Figure 4 (a) further shows that increasing model depth L (with fixed K and N) improves in-context performance: $Q_{ICL\text{-}Q}$ (both inversion and symmetry) increases with L and grows more slowly beyond $L \approx 12$. Notably, $Q_{ICL\text{-}C}$ improves sharply even for shallow models, reaching about 85% accuracy at $L = 3$ and around 10% at $L = 1$.

Task 2. Figure 4 (b, c, d) shows distinct behaviors for shallow models. With $L = 1$, both Q_{Logic} and $Q_{ICL\text{-}C}$ are low. With $L = 2$, Q_{Logic} stays near 0% across N, but $Q_{ICL\text{-}C}$ exhibits a **phase transition** once $N > 10{,}000$ and can rise to about 85%. With $L = 3$, Q_{Logic} can be high at $N = 2{,}000$ but decreases as N grows, and $Q_{ICL\text{-}C}$ follows a similar trend to the $L = 2$ case.

Task 3. Figure 5 shows layer-wise analysis for ICL completion $Q_{ICL\text{-}C}$. When $N = 2{,}000$ (Figure 5 (a)), the mean rank stays above $\sim$ 125 until layer 8, improves at layers 8–10, and then drops sharply at layers 11–12.

In contrast, when $N = 10{,}000$ and 20,000 (Figure 5 (b, c)), mean rank improves sharply starting around layer 5 and continues to improve, reaching near rank = 1 by layer 12; mean probability increases accordingly and approaches 100% at the final layer. Mean logits increase throughout and end substantially higher than in the $N = 2{,}000$ setting.

3.3 Takeaway.

AR LMs can learn logical semantics of relational word. Under appropriate training conditions, Q_{Logic} and $Q_{ICL\text{-}C}$ exceed 95%, and $Q_{ICL\text{-}Q}$ reaches ~85% and keeps improving with larger N without a clear plateau, as shown in Figure 1(a).

Relational semantics emerge with sufficient logic-bearing supervision, enabling generalization to unseen entities. This emergence exhibits a sharp phase transition from near-zero to near-perfect performance as the training data increases (Figure 1(a)), and it can arise even in small, shallow models with only 2–3 layers (Fig-

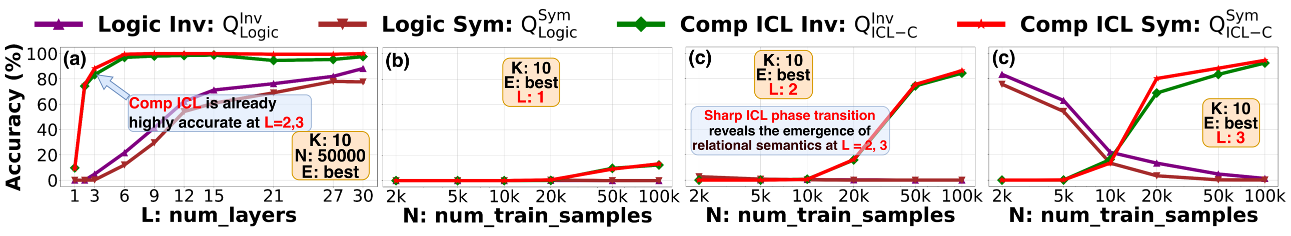

Figure 4: Effect of model depth on logic and ICL completion evaluations. (a) In-context performance as L varies under fixed K and N. (b–d) For shallow models ($L = 1, 2, 3$), evaluation performance across N with fixed K.

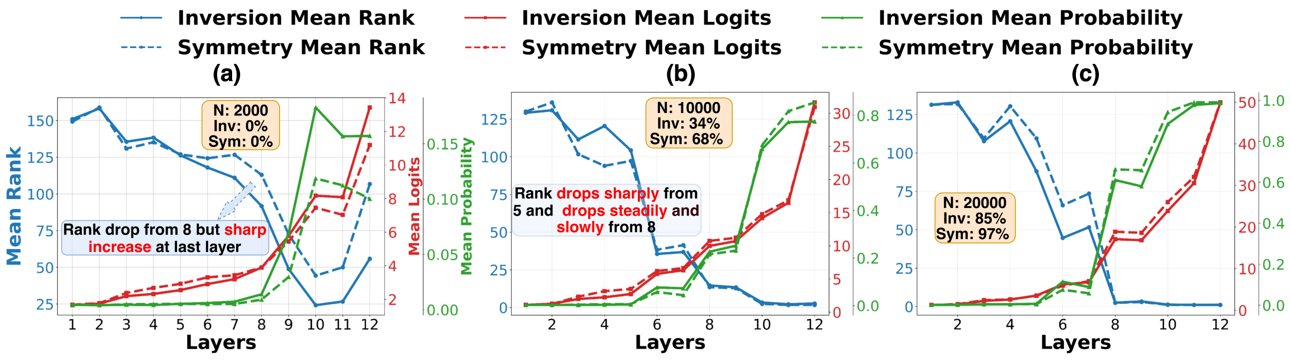

Figure 5: Layer-wise mean logit, mean probability, and mean rank of the first correct token on $Q_{\text{ICL-C}}$, for models trained with different N (with $K = 10$ and $L = 12$ fixed). Panels (a–c) correspond to $N = 2{,}000$, 10,000, and 20,000, respectively; the $Q_{\text{ICL-C}}$ accuracy for each setting is annotated in the figure.

ure 4(c,d)).

Successful relational generalization is associated with stable, logic-relevant representations in intermediate layers, whereas unsuccessful models exhibit late-stage, unstable signals that degrade at the final layer. From Figure 5, we observe that high $Q_{\text{ICL-C}}$ accuracy coincides with an early improvement of the first-correct-token signal around layers 5–8, whereas low-performing models show a delayed onset (around layers 9–11) followed by a sharp final-layer degradation, as indicated by the layer-wise rank and probability curves.

3.4 Grokking of Relational Semantics.

Figure 6 shows how Q_{Logic} and $Q^{\text{Comp}}_{\text{ICL}}$ change with training steps E under fixed $N = 20000$, $L = 12$, and $K = 10$. Panel(a) reveals a clear grokking transition: $Q^{\text{Inv}}_{\text{ICL-C}}$ and $Q^{\text{Sym}}_{\text{ICL-C}}$ abruptly jump to near-perfect accuracy around $E \approx 20$. Notably, this ICL Logic QA generalization, which is largely decoupled from direct memorization, emerges earlier than the improvement in memorization-dependent Logic QA. This ordering contrasts with the standard characterization of grokking as delayed generalization after memorization[12, 13, 14], and instead suggests that for relational semantics in our setting, generalization can precede memorization.

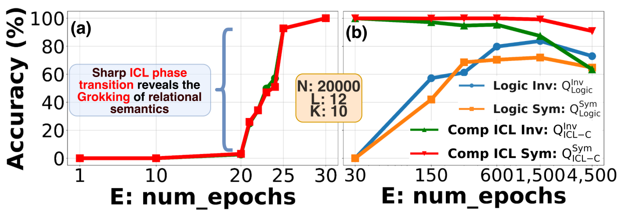

Figure 6: Learning dynamics of Q_{Logic} and $Q_{\text{ICL-C}}$ across training steps E under fixed N, L, K, with panels showing early and later 30 training epochs.

4 Conclusion

We train AR LMs from scratch with KG-based synthetic data to address question: Can AR LMs learn relational word logical semantics, and when? In our results, we observe a sharp phase transition in which relational semantics emerge with sufficient logic-bearing supervision, even in shallow (2–3 layer) models, and successful generalization aligns with stable intermediate-layer signals.

Acknowledgements

This work was partially supported by JST SPRING JPMJSP2110 (YZ), JSPS KAKENHI 22H05106, 23H03355, and JST CREST JPMJCR21N3 (HS).

Ethics Statement

This study complies with the ACL Ethics Policy.

References

[1] Yihua Zhu, Qianying Liu, Jiaxin Wang, Fei Cheng, Chaoran Liu, Akiko Aizawa, Sadao Kurohashi, and Hidetoshi Shimodaira. Memorization, emergence, and explaining reversal failures: A controlled study of relational semantics in llms. **arXiv preprint arXiv:2601.02931**, 2026.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. **Advances in neural information processing systems**, Vol. 30, , 2017.

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

[4] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. **arXiv preprint arXiv:2206.07682**, 2022.

[5] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. **IEEE Transactions on Knowledge and Data Engineering**, Vol. 36, No. 7, pp. 3580–3599, 2024.

[6] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. **arXiv preprint arXiv:1902.10197**, 2019.

[7] Yihua Zhu and Hidetoshi Shimodaira. 3D rotation and translation for hyperbolic knowledge graph embedding. In Yvette Graham and Matthew Purver, editors, **Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1497–1515, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

[8] Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. **arXiv preprint arXiv:2309.14316**, 2023.

[9] Charlie Zhang, Graham Neubig, and Xiang Yue. On the interplay of pre-training, mid-training, and rl on reasoning language models. **arXiv preprint arXiv:2512.07783**, 2025.

[10] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. **OpenAI blog**, Vol. 1, No. 8, p. 9, 2019.

[11] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. **Advances in neural information processing systems**, Vol. 35, pp. 27730–27744, 2022.

[12] Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokked transformers are implicit reasoners: A mechanistic journey to the edge of generalization. **arXiv preprint arXiv:2405.15071**, 2024.

[13] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. **arXiv preprint arXiv:2201.02177**, 2022.

[14] Qinyuan Wu, Soumi Das, Mahsa Amani, Bishwamittra Ghosh, Mohammad Aflah Khan, Krishna P Gummadi, and Muhammad Bilal Zafar. Rote learning considered useful: Generalizing over memorized data in llms. **arXiv preprint arXiv:2507.21914**, 2025.

A Appendix

B Controllable KG Synthetic Corpus

We construct a fully controllable synthetic corpus grounded in a KG schema.

KG and Triple Generation As shown in Figure 2, each KG triple is generated by independently sampling its entities from uniform distributions. Person names take the form [first,middle,last], where each part is sampled from one of three disjoint pools of 100 synthetic tokens, yielding up to 10^6 unique full names; job entities are sampled from 300 real-world occupations. Logic relations include (i) inversion kinship pairs (e.g., father/son, husband/wife, uncle/niece) and (ii) symmetric relations (e.g., brother, friend, spouse). The non-logic relation is (iii) a person–job relation.

We construct three graph types: Train Graphs contain all relation types and include both directions for inversion and symmetric relations plus job facts (7 triples per graph); Evaluation Graphs remove the reverse facts (e.g., keep "name#4 is the father of name#5" but drop "name#5 is the son of name#4"); ICL Graphs keep only person–person relations and also remove reverse facts. Notably, names are sampled independently across the three graph types, so entity sets do not overlap.

Pre-train Corpus Construction Triples are insufficient for training LLMs, so we verbalize each triple into a natural-language sentence by randomly choosing one of four surface formats. For instance, (Name#1, `father`, Name#2) is rendered as "Name#1 is the father of Name#2." For each KG, we sample K distinct paragraph templates (`num_template` = K) by selecting one format for each triple, which yields 4^7 possible realizations for Train Graphs (7 triples) and 4^5 for Evaluation Graphs (5 triples). We then randomly shuffle the sentence order within each paragraph.

This yields K templated paragraphs per graph. We then generate the train content from N independently sampled Train Graphs (denoted as `num_train_samples` = N). In contrast, the evaluation set is fixed to 500 samples and does not vary with the pre-train data size. Finally, the pre-train corpus is constructed by combining training content and evaluation content.

SFT Corpus Construction After constructing the synthetic KGs and the pre-training corpus (Figure 2), we further build a supervised fine-tuning (SFT) corpus to endow the model with question-answering capability. Specifically, we generate QA pairs only from the train content of the pre-training corpus. For each sentence derived from a triple, we create a corresponding question and its answer; for example, from "Name#1 is the father of Name#2" we derive "Q: Who is the father of Name#2? A: Name#1." Notably, we do not construct any SFT QA data from the evaluation content.

B.1 Training Setup

B.1.1 Model Architecture

We train a decoder-only, GPT-2–style model[10] from scratch on our KG-based synthetic corpus, using the standard GPT-2 tokenizer. Except for experiments where we vary the number of layers (L), we use a fixed architecture with 12 layers, 12 attention heads, and a 768-dimensional hidden size. On 8×A100 (40GB), training a model with ($N = 2000$) and ($K = 10$) (about 2Million tokens) to its best performance takes approximately 1.2 hours.

B.1.2 Hyperparameter

Pre-training. We use a batch size of 491,520 tokens per iteration, a learning rate of (6×10^{-4}) with weight decay 0.1, cosine decay to a minimum learning rate of (6×10^{-5}), and 500 warmup iterations. All models are trained in bf16 precision.

SFT. We fine-tune with standard supervised fine-tuning [11], using a learning rate of (3×10^{-5}) and a batch size of 32,768 tokens per iteration.

Inference. We generate outputs with temperature (0.8) and (top_k=100).