Memorization, Emergence, and Explaining Reversal Failures: A Controlled Study of Relational Semantics in LLMs

Anonymous ACL submission

Abstract

Autoregressive LLMs perform well on relational tasks that require linking entities via relational words (e.g., father/son, friend), but it is unclear whether they learn the logical semantics of such relations (e.g., symmetry and inversion logic) and, if so, whether reversal-type failures arise from missing relational semantics or left-to-right order bias. We propose a controlled Knowledge Graph-based synthetic framework that generates text from symmetric/inverse triples, train GPT-style autoregressive models from scratch, and evaluate memorization, logical inference, and in-context generalization to unseen entities to address these questions. We find a sharp phase transition in which relational semantics emerge with sufficient logic-bearing supervision, even in shallow (2–3 layer) models, and that successful generalization aligns with stable intermediate-layer signals. Finally, order-matched forward/reverse tests and a diffusion baseline indicate that reversal failures are primarily driven by autoregressive order bias rather than deficient inversion semantics.

1 Introduction

Auto-regressive (AR) large language models (LLMs) trained on massive corpora have achieved remarkable progress across a wide range of NLP tasks, including question answering, information extraction, and reasoning (Vaswani et al., 2017; Wei et al., 2022b,a; Pan et al., 2024). Many of these tasks fundamentally rely on the ability to recognize, represent and manipulate relations between entities. Despite this success, it remains unclear whether LLMs truly internalize the *logical semantics* carried by *relational words*, or whether their performance primarily reflects learning surface-level co-occurrence patterns from text.

Relational words occupy a special role in natural language: beyond linking entities in a sentence, they often encode abstract and systematic logical properties. For instance, some relations are *symmetric* (if A is a `friend` of B, then B is a `friend` of A), while others form *inverse* pairs (if A is the `father` of B, then B is the `son` of A). In knowledge graphs (KGs), such regularities are formalized as relation properties and serve as a core organizing principle for relational reasoning (Sun et al., 2019;

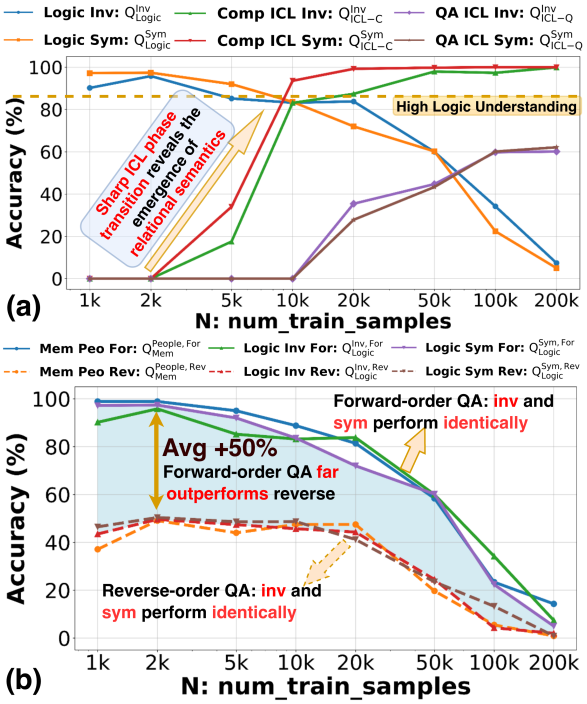

Figure 1: **Two-stage inquiry into relational-word understanding in autoregressive language models.** (a) Models can memorize symmetric and inverse relational facts and perform forward-order logic QA; both completion- and QA-based in-context evaluations on unseen entities exhibit a sharp phase transition with sufficient logic-bearing supervision, indicating the emergence of relational semantics. (b) While inverse relations perform comparably to their original under forward queries, reversed-order queries show a substantial performance drop, revealing the reversal curse of order sensitivity.

Zhu and Shimodaira, 2024). This motivates our first research question: **RQ1: 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?**

While RQ1 concerns whether relational semantics can be learned at all, RQ2 focuses on how such knowledge—when present—is deployed at inference time. Recent work reports the reversal curse (Berglund et al., 2023): a model trained on relational statements of the form "A is the father of B" often fails to answer the logically equivalent reversed query "B is the son of A." This raises a fundamental ambiguity: are such failures caused by missing explicit inversion relational representations, or by order bias induced by left-to-right AR decoding? This leads to our second research question: **RQ2: When a model has internalized relational semantics, are reversal-type failures primarily due to deficiencies in relational semantics, or to order bias in AR generation?** Complementary, RQ1 and RQ2 intersect at a central issue: in LLMs, whether relation logic, represented by relational words, is in a form that supports robust, direction-agnostic inference.

Answering these questions using standard web-scale pretrained LLMs is w. Relational words are often polysemous and context-dependent, and uncontrolled pretraining data introduces the risk of contamination, making it difficult to distinguish genuine logical generalization from memorization (Allen-Zhu and Li, 2023).

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. We verbalize each triple using a small set of templates and compose them into paragraph-level text with controlled linguistic variation, enabling systematic manipulation of training data scale. Using this corpus, we then train GPT-style AR models from scratch using next-token prediction, followed by question–answer style supervised fine-tuning. This framework enables a systematic examination of relational understanding under multiple evaluation settings. To examine RQ1, under both cloze-style sentence completion and question answering, we test whether models can perform (i) Memorize QA: memorize observed relational facts and answer questions, (ii) Logic QA: infer missing symmetric or inverse facts given partial information, and (iii) In-context Learning (ICL) Logic QA: generalize relational knowledge to unseen entities. To study RQ2, we further perform order-matched forward and reverse evaluations, and compare AR models with diffusion-based language models that do not rely on left-to-right decoding.

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 when queried in the same forward order seen during training, 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, whereas unsuccessful models exhibit late-stage, unstable signals that degrade at the final layer.

4. **Revisiting the reversal curse.** We provide evidence that reversal-type failures primarily stem from order bias in left-to-right autoregressive decoding rather than missing relational semantics, and show that this issue is mitigated by bidirectional exposure during training. In contrast, diffusion-based LMs do not exhibit such behavior.

2 Methodology

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

2.1 Controllable KG Synthetic Corpus

We construct a fully controllable synthetic corpus grounded in a KG schema. We show end-to-end text generation example in Table 3.

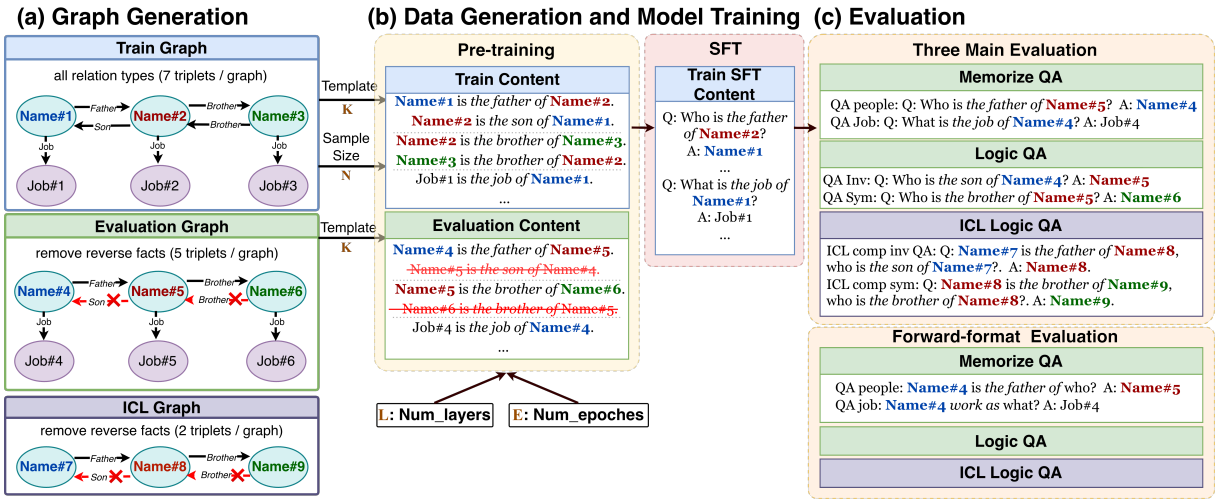

Figure 2: Overview of the KG-synthetic data generation framework, model training process, and evaluation detail.

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 (see Table 4). For instance, $(\text{Name\#1}, \texttt{father}, \text{Name\#2})$ is rendered as "Name#1 is the father of Name#2." For each KG, we sample K distinct paragraph templates ($\texttt{num\_template} = K$) 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 $\texttt{num\_train\_samples} = N$). 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.

2.2 Training Setup

Inspired by (Allen-Zhu and Li, 2023; Zhang et al., 2025), we train GPT2-style AR LMs (Radford et al., 2019) from scratch. Pre-training is conducted on the combined corpus of train and evaluation content. We control four experimental variables: $K = \texttt{num\_template}$, $N = \texttt{num\_train\_samples}$, $E = \texttt{num\_training\_epochs}$, and $L = \texttt{num\_layers}$.

After pre-training, we perform SFT (Ouyang et al., 2022) 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

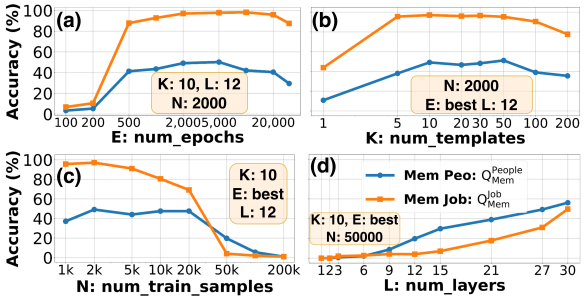

Figure 3: Memorize QA (Q_{Mem}) evaluation under varying training epochs E, template size K, training sample size N, and model depth L. (a) Vary E with K, L, N fixed. (b) Vary K with N, L fixed. (c) Vary N with K, L fixed. (d) Vary L with K, N fixed.

about training setup see Appendix A.1.

2.3 Evaluation Setup

As shown in Figure 2, we evaluate models with three complementary query sets: Memorize QA and Logic QA are derived from the Evaluation Graph, while in-context learning Logic QA is derived from the ICL Graph. Example queries are provided in Table 5.

Merorize QA 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 Logic QA Q_{Logic} evaluates whether the model can infer relational logic beyond memorization by querying facts that are absent from Evaluation content but logically implied by it. For instance, for an inversion pair (e.g., `father`/`son`), evaluation content may include only one direction such as "A is the father of B" but omit its inverse "B is the son of A"; we then ask an evaluation question "who is the son of A?", whose correct answer is B. Success on such queries indicates that the model has internalized the inverse relation between `father` and `son` and can derive the missing counterpart from the observed direction.

In-context learning logic QA 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. Each prompt provides a context instantiating a relation and requires the model to produce its logically implied counterpart. We consider both completion-based evaluation $Q_{ICL\text{-}C}$, performed on the pre-trained model before SFT (e.g., "A is the father of B. B is the son of __"), and QA-based evaluation $Q_{ICL\text{-}Q}$, performed after SFT (e.g., "A is the father of B. Who is the son of A?"). This separation allows us to probe relational reasoning both in pure language modeling and in QA mode, while minimizing confounding effects from memorized entity associations.

3 Result 1: Can auto-regressive LMs Memorize and Perform Basic QA?

We use Memorize QA Q_{Mem} as a prerequisite check to verify that models can memorize and answer factual queries, since failures on such queries would confound subsequent logic evaluations.

Task setting. We study four factors: L, N, K, and E. Unless otherwise stated, for each configuration we select the epoch E that yields the highest accuracy (since the optimal epoch differs across K). Specifically, we (1) fix L, N, K and sweep E; (2) fix L, N and vary K; (3) fix L, K and vary N; and (4) fix K, N and vary L. We evaluate two Memorize QA subsets, QA-people Q_{Mem}^{People} and QA-job Q_{Mem}^{Job}, whose answers are explicitly stated in evaluation content.

Observations. Figure 3 summarizes the results. (a) With $K = 10$ and $N = 2000$, Q_{Mem}^{People} and Q_{Mem}^{Job} exhibit a rise-then-fall trend as E increases; Q_{Mem}^{Job} peaks above 95% while Q_{Mem}^{People} peaks around 50%. (b) With $N = 2000$, varying K yields the same rise-then-fall pattern. (c) With $K = 10$, increasing N causes both accuracies to decrease monotonically, approaching zero for very large N (e.g., $> 10^5$). (d) Under $K = 10$ and $N = 50{,}000$, increasing depth L consistently improves both accuracies, with gains persisting beyond 30 layers.

Takeaway and Discussion. First, despite relatively low Q_{Mem}^{People} accuracy (about 50% at its best), Q_{Mem}^{Job} reaches above 95% under appropriate template diversity and training epochs, indicating that our pre-training plus SFT pipeline can support memorization and basic QA on the synthetic corpus, consistent with (Allen-Zhu and Li, 2023). Second, when scaling the train sample size N substantially, increasing model capacity (via larger L) becomes important to maintain and recover memorization and QA performance.

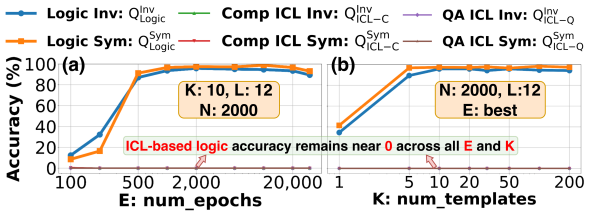

Figure 4: Logic and ICL evaluation. (a) Vary E with K, L, N fixed. (b) Vary K with N, L fixed.

4 Result 2: Can auto-regressive LMs learn relational word logical semantics, and when?

Having verified basic memorization and QA capability in Result 1, we now 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`).

4.1 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 $\mathrm{Q}_{\mathrm{Logic}}$, $\mathrm{Q}_{\text{ICL-C}}$, and $\mathrm{Q}_{\text{ICL-Q}}$.

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 $\mathrm{Q}_{\mathrm{Logic}}$ and $\mathrm{Q}_{\text{ICL-C}}$.

Task 3 (layer-wise analysis). For prompts in $\mathrm{Q}_{\text{ICL-C}}$ (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. (full setup are provided in the appendix A.4.2)

4.2 Observations.

Task 1. Figure 4 (a, b) shows that for fixed L and N, varying training epochs E or template count K yields a consistent pattern: $\mathrm{Q}^{\mathrm{Inv}}_{\mathrm{Logic}}$ and $\mathrm{Q}^{\mathrm{Sym}}_{\mathrm{Logic}}$ rise with training and reach stable peak accuracy above 95%, while both $\mathrm{Q}_{\text{ICL-C}}$ and $\mathrm{Q}_{\text{ICL-Q}}$ remain near 0%.

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

Figure 5 (a) further shows that increasing model depth L (with fixed K and N) improves in-context performance: $\mathrm{Q}_{\text{ICL-Q}}$ (both inversion and symmetry) increases with L and grows more slowly beyond $L \approx 12$. Notably, $\mathrm{Q}_{\text{ICL-C}}$ improves sharply even for shallow models, reaching about 85% accuracy at $L = 3$ and around 10% at $L = 1$.

Task 2. Figure 5 (b, c, d) shows distinct behaviors for shallow models. With $L = 1$, both $\mathrm{Q}_{\mathrm{Logic}}$ and $\mathrm{Q}_{\text{ICL-C}}$ are low. With $L = 2$, $\mathrm{Q}_{\mathrm{Logic}}$ stays near 0% across N, but $\mathrm{Q}_{\text{ICL-C}}$ exhibits a **phase transition** once $N > 10{,}000$ and can rise to about 85%. With $L = 3$, $\mathrm{Q}_{\mathrm{Logic}}$ can be high at $N = 2{,}000$ but decreases as N grows, and $\mathrm{Q}_{\text{ICL-C}}$ follows a similar trend to the $L = 2$ case.

Task 3. Figure 6 shows layer-wise analysis for ICL completion $\mathrm{Q}_{\text{ICL-C}}$. When $N = 2{,}000$ (Figure 6 (a)), the mean rank stays above ~ 125 until layer 8, improves at layers 8–10, and then drops sharply at layers 11–12. Mean probability mirrors this trend, rising only to about 15% before dropping in the final layers. Mean logits increase slowly before layer 8 and then rise rapidly afterward.

In contrast, when $N = 10{,}000$ and 20,000 (Figure 6 (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.

4.3 Takeaway.

AR LMs can learn logical semantics of relational word. Under appropriate training conditions, $\mathrm{Q}_{\mathrm{Logic}}$ and $\mathrm{Q}_{\text{ICL-C}}$ exceed 95%, and $\mathrm{Q}_{\text{ICL-Q}}$ reaches $\sim$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

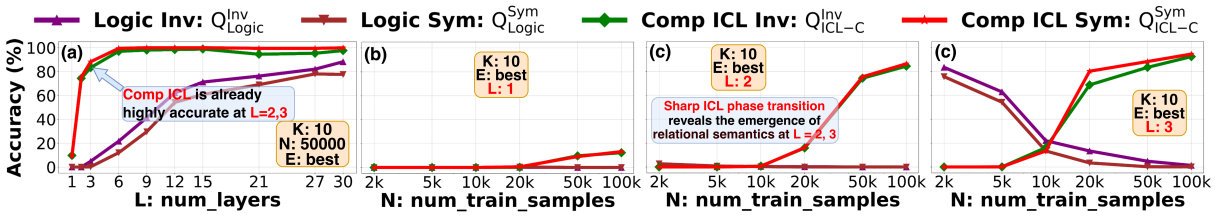

Figure 5: Effect of model depth on logic and ICL completion evaluations. (a) In-context performance as L varies under fixed K and N. (b–d) For shallow models ($L = 1, 2, 3$), evaluation performance across N with fixed K.

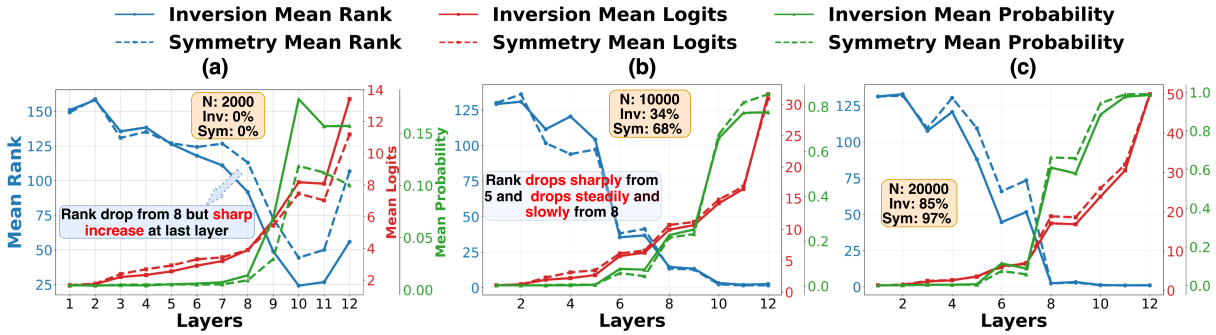

Figure 6: Layer-wise mean logit, mean probability, and mean rank of the first correct token on $Q_{ICL\text{-}C}$, for models trained with different N (with $K = 10$ and $L = 12$ fixed). Panels (a–c) correspond to $N = 2{,}000$, $10{,}000$, and $20{,}000$, respectively; the $Q_{ICL\text{-}C}$ accuracy for each setting is annotated in the figure.

(Figure 1(a)), and it can arise even in small, shallow models with only 2–3 layers (Figure 5(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 6, we observe that high $Q_{ICL\text{-}C}$ 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.

4.4 Discussion.

Emergence of Relational Semantics and Generalization We observe a phase transition phenomenon in Q_{ICL} once the number of inversion/symmetry training samples exceeds a threshold. This is consistent with prior reports of emergent capabilities with scale (Wei et al., 2022a), but we suggests that the effective trigger can be the scale of targeted logical evidence rather than model size alone. A plausible explanation is a strategy shift: below the threshold the model relies on local co-occurrence or weak memorization, whereas above it the logic signal is frequent and diverse enough to support a reusable rule mechanism, in line with grokking-style dynamics (Nanda et al., 2023). The opposite trends of Q_{Logic} (decreasing) and Q_{ICL} (increasing) with larger N are consistent with Q_{Logic} being more affected by entity-memorization interference, while Q_{ICL} more directly reflects relational-rule induction.

Layer-wise Correlates of Relational Generalization The Takeaway 4.3 from our layer-wise analysis matches the residual-stream/circuits view in which intermediate layers are the primary locus where features are written and composed into task mechanisms, while later layers mainly read out these mechanisms and align them to the output distribution (Elhage et al., 2021). Accordingly, when relational logic is truly learned, the corresponding mechanism can be formed early and then amplified consistently; when it is weak or not fully integrated, late layers may rely on unstable heuristics and the final readout can suppress the partial logic signal due to competition with other features or distributional alignment. Combined with mechanistic grokking accounts (Nanda et al., 2023), this suggests that failing models remain in a memorization/weak-mechanism regime, producing the observed "improve-then-drop" behavior near the output.

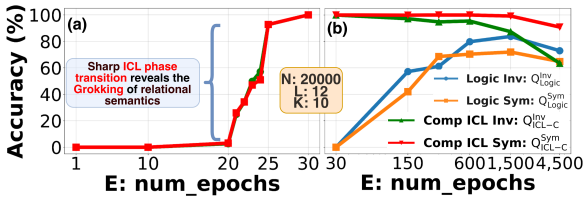

Figure 7: Learning dynamics of Q_{Logic} and $Q_{ICL\text{-}C}$ across training steps E under fixed N, L, K, with panels showing early and later 30 training epochs.

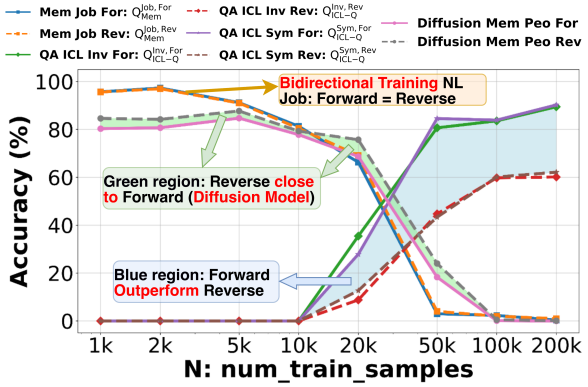

Figure 8: Forward–reverse accuracy comparisons for Q^{Job}_{Mem} and $Q_{ICL\text{-}Q}$, together with Memorize-people QA results under a diffusion model. All experiments vary N with K and L fixed.

Grokking of Relational Semantics with E. Figure 7 shows how Q_{Logic} and Q^{Comp}_{ICL} change with training steps E under fixed $N = 20000$, $L = 12$, and $K = 10$. Panel(a) reveals a clear grokking transition: $Q^{Inv}_{ICL\text{-}C}$ and $Q^{Sym}_{ICL\text{-}C}$ 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(Wang et al., 2024; Power et al., 2022; Wu et al., 2025), and instead suggests that for relational semantics in our setting, generalization can precede memorization.

5 Result 3: What Drives Reversal-Type Failures: Relational Semantics or Order Bias?

Prior work uncovered reversal curse(Berglund et al., 2023). However, it remains an open question whether this failure is primarily caused by order bias in AR decoding, or by the model's inability to internalize the inversion semantics between relational words (e.g., `father` vs. `son`). Most existing evaluations support the former explanation, but do not fully rule out the latter. Building on our Results 1–2, we design contrastive experiments that isolate order bias while controlling inversion relational semantics.

5.1 Task Setting

Forward vs. Reverse queries. Given a fact "A is the father of B" (order: A $\rightarrow$ `father` $\rightarrow$ B), a **Reverse** query conditions on B and asks for A ("Who is the father of B?" $\Rightarrow$ A), whereas a **Forward** query conditions on A and asks for B ("A is the father of who?" $\Rightarrow$ B).

Nuance (Q_{logic}). If the evaluation content provides only 'A is the father of B" ($A \rightarrow$ `father` $\rightarrow B$), the inversion logic query Q^{Inv}_{Logic} ('Who is the son of A?" $\Rightarrow$ B) matches the training order (A $\rightarrow \cdot \rightarrow$ B). Therefore, it is a **Forward** query with respect to information order, rather than a Reverse query like Q^{People}_{Mem} . The same logic applies to the symmetry query Q^{Sym}_{Logic}.

Task 1: One-directional training for relational attributes (people). All person–person evaluation facts use the one-directional template "A is the {relation} of B" ($A \rightarrow r \rightarrow B$). We evaluate (i) Memorize-people QA Q^{People}_{Mem}, (ii) inversion logic via Q^{Inv}_{logic} and $Q^{Inv}_{ICL\text{-}Q}$, and (iii) symmetry logic via Q^{Sym}_{logic} and $Q^{Sym}_{ICL\text{-}Q}$. All evaluations are conducted using both Forward and Reverse query formats.

Task 2: Bidirectional training for non-relational attributes (job). For job facts, we include both forward and reverse surface templates in training (e.g., "chef is the job of A" and "A's job is chef") and evaluate Q^{Job}_{Mem} using matched Forward/Reverse queries. Detail has been shown in Table 4.

Task 3: Comparison with Large Language Diffusion Model. We train a diffusion LM baseline on our synthetic data using a bidirectional Transformer under LLaDA (Nie et al., 2025) and compare its dynamics to an AR model (see Appendix A.5 for settings).

5.2 Observations.

Tasks 1. Across both Memorize-people and Logic (inversion and symmetry) queries, the Reverse accuracy curves are far below their Forward counterparts (Figure 1 (b) and Figure 8). Moreover, within the same query format (forward or

reverse), Symmetry and Inversion consistently perform at similar levels (e.g., $Q_{\text{Logic}}^{\text{Sym,For}}$ vs. $Q_{\text{Logic}}^{\text{Inv,For}}$, or $Q_{\text{Logic}}^{\text{Sym,Rev}}$ vs. $Q_{\text{Logic}}^{\text{Inv,Rev}}$).

Task 2. The forward and reverse accuracies on $Q_{\text{Mem}}^{\text{Job}}$ become nearly identical (Figure 8, $Q_{\text{Mem}}^{\text{Job,For}}$ vs. $Q_{\text{Mem}}^{\text{Job,Rev}}$).

Task 3 Figure 8 illustrates that, in the diffusion model, forward- and reverse-based Memorize-people query accuracies are nearly identical; intriguingly, the reverse setting is on average about 4% higher than the forward setting.

5.3 Takeaway

Reversal-type failures primarily stem from order bias in left-to-right AR decoding rather than missing relational semantics. Symmetry and inversion perform similarly under both formats, indicating comparable relational semantics. Since symmetry should not induce reversal failures, the gap is best explained by order sensitivity rather than deficient inversion semantics.

Bidirectional Training substantially mitigates reversal failures. With bidirectional training in Task 2 and diffusion model, Forward and Reverse accuracies on are nearly identical.

5.4 Discussion

Reversal Failures Reflect AR Order Bias, Not Semantics Prior work reports a consistent reversal curse (Berglund et al., 2023). Our controlled KG-based experiments further support the view that this failure is primarily due to order bias (Lv et al., 2024; Zhu et al., 2024). Crucially, unlike many studies on web-scale pre-trained models where semantic confounds are difficult to rule out, we disentangle order bias from inversion relational semantics and, for the first time, verify that the reversal curse is not caused by a failure to understand inversion semantics.

Bidirectional Training Mitigates the Reversal Curse We also observe a direct mitigation consistent with the order-based account. In Task 3, bidirectional training makes forward and reverse accuracies on $Q_{\text{non}}^{\text{job}}$ nearly identical, indicating that training exposure to both orders largely removes reversal failures. This complements prior mitigation efforts that adjust training signals (Lv et al., 2024) and is consistent with recent evidence that non-AR diffusion LMs are less sensitive to order reversal, further implicating left-to-right decoding bias as the root cause (Nie et al., 2025).

6 Related Work

Controlled Training from Scratch with Synthetic Data Controlled, from-scratch training on synthetic data is increasingly used to attribute LM behaviors under fully known data-generating processes and to avoid confounds such as data contamination (Zhang et al., 2025; Liu et al., 2025; Yuan et al., 2025). Such testbeds, exemplified by the Physics of LMs line and curated synthetic corpora like TinyStories, isolate specific capabilities and failure modes while making small-scale model development more interpretable (Allen-Zhu and Li, 2023; Eldan and Li, 2023).

Ability emergence phenomenon. Prior work reports threshold-like "emergent" gains in reasoning with scale and shows that training can exhibit phase-transition dynamics in generalization (Wei et al., 2022a; Nanda et al., 2023). Mechanistic analyses further suggest that transformers build computation by progressively composing features in the residual stream, with intermediate layers often forming stable rule-like circuits that explain abrupt behavioral shifts (Elhage et al., 2021).

Reversal curse The reverse curse phenomenon was introduced by Berglund et al. (2023). Subsequent analyses attribute it to directional training signals induced by causal masking and propose training-time interventions that expose alternative conditioning patterns (Lv et al., 2024; Zhu et al., 2024). In parallel, LLaDA(Nie et al., 2025) suggests that departing from the standard AR paradigm can help mitigate this effect.

7 Conclusion

We train AR LMs from scratch with KG-based synthetic data to address two questions: (i) Can AR LMs learn relational word logical semantics, and when? (ii) What drives reversal-type failures: deficient relational semantics or order bias? 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. Moreover, order-matched forward/reverse tests indicate that reversal failures are primarily driven by AR order bias rather than deficient inversion semantics.

Limitations

Limited relational word types Our study focuses on a narrow set of relation properties—primarily symmetric relations and inverse kinship pairs (plus a simple person–job attribute). While these cover two canonical forms of relational logic, they do not represent the broader landscape of relational semantics encountered in natural language and knowledge graphs. In particular, we do not test properties such as transitivity, antisymmetry, hierarchical relations, multi-hop compositional rules, or interactions among multiple properties, which may exhibit different emergence thresholds and failure modes.

Template-based language generation Although we use multiple verbalization templates and shuffle sentence order to increase surface diversity, the corpus is still template-generated and thus may not fully reflect the breadth of paraphrastic variation and contextual nuance in natural text. This limitation does not invalidate the controlled comparisons in our setting, but it may affect how directly the quantitative thresholds and emergence dynamics transfer to more naturalistic corpora.

Large Language Diffusion Model training Training diffusion LMs from scratch on our synthetic corpus is not yet fully mature in our implementation, particularly for the supervised finetuning stage and its associated procedures. Therefore, our diffusion results currently support only the qualitative conclusion that forward and reverse queries exhibit little gap under diffusion-style training. A comprehensive accuracy-level comparison between diffusion and autoregressive models across the full set of query types remains future work.

Ethics Statement

This study complies with the ACL Ethics Policy.

References

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

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*.

Ang Lv, Kaiyi Zhang, Shufang Xie, Quan Tu, Yuhan Chen, Ji-Rong Wen, and Rui Yan. 2024. An analysis and mitigation of the reversal curse. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13603–13615.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

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

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

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

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

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

Lifan Yuan, Weize Chen, Yuchen Zhang, Ganqu Cui, Hanbin Wang, Ziming You, Ning Ding, Zhiyuan Liu, Maosong Sun, and Hao Peng. 2025. From $f(x)$ and $g(x)$ to $f(g(x))$: Llms learn new skills in rl by composing old ones. *arXiv preprint arXiv:2509.25123*.

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

Hanlin Zhu, Baihe Huang, Shaolun Zhang, Michael Jordan, Jiantao Jiao, Yuandong Tian, and Stuart J Russell. 2024. Towards a theoretical understanding of the'reversal curse'via training dynamics. *Advances in Neural Information Processing Systems*, 37:90473–90513.

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

A Appendix

A.1 Training Setup

A.1.1 Model Architecture

We train a decoder-only, GPT-2–style model(Radford et al., 2019) 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.

A.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 (Ouyang et al., 2022), 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).

A.1.3 Dataset Statistics and Training Configuration

Table 1 summarizes the best-performing training configurations for each dataset size (N) under fixed ($L = 12$) and ($K = 10$), including the resulting token counts, iterations per epoch, and the approximate pre-training/SFT schedules (epochs or iterations) and wall-clock time on 8×A100 (40GB). Notably, both the optimal pre-training length and the overall runtime vary substantially with the corpus size, and all reported epochs and times are approximate.

Table 2 reports the number of evaluation questions for each task type when the evaluation set is fixed to 500 samples. Since forward and reverse formats are constructed to be balanced, we list only the reverse counts for clarity.

A.2 Supplementary Material

Unless otherwise stated, we report results from a single run due to computational constraints (training from scratch and fine-tuning are expensive and time-consuming under our GPU budget).

Moreover, all data are synthetically generated and do not contain personally identifying information. We also avoid generating offensive content by construction and manually spot-check a sample of the generated corpus. Finally, all datasets in this work are synthetically generated for research and reproducibility purposes. The released artifacts (code/data/model checkpoints, if any) are intended for research use only.

Our experiments were facilitated by leveraging PyTorch, Huggingface, and Numpy as essential tools. Furthermore, We use ChatGPT in our paper writing and programming.

We will release the code under the MIT License. The generated synthetic datasets (and trained checkpoints, if released) will be distributed under the same license terms specified in the repository. We use standard open-source libraries (e.g., PyTorch under BSD-3-Clause, NumPy under BSD-3-Clause, and Transformers under Apache-2.0) and comply with their licenses.

A.3 Data Generation Framework

We briefly described this in the Methodology 2, but the coverage was not sufficiently detailed or technical. In this section, we provide a more formal description of the KG-based corpus construction, along with detailed statistics for each dataset. In Table 3, we illustrate the end-to-end text generation pipeline with a concrete example.

A.3.1 KG and Triple Generation

As illustrated in Figure 2, each knowledge graph (KG) instance corresponds to a single semantic sample. We define a KG as a triple $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ denote the entity set, relation set, and triple set, respectively. The entity set is given by $\mathcal{E} = \mathcal{E}^N \cup \mathcal{E}^J$, where $\mathcal{E}^N = \{e_1^N, e_2^N, e_3^N\}$ consists of three name entities, and $\mathcal{E}^J = \{e_1^J, e_2^J, e_3^J\}$ consists of their corresponding job entities. The relation set $\mathcal{R}$ comprises three components: (i) three inversion relation pairs $\mathcal{R}_{\text{inv}} = \{(r_1^I, r_1^{I,-1}), (r_2^I, r_2^{I,-1}), (r_3^I, r_3^{I,-1})\} = \{(\texttt{father}, \texttt{son}), (\texttt{husband}, \texttt{wife}), (\texttt{uncle}, \texttt{niece})\}$; (ii) three symmetric relations $\mathcal{R}_{\text{sym}} = \{r_1^S, r_2^S, r_3^S\} = \{\texttt{friend}, \texttt{brother}, \texttt{spouse}\}$; and (iii) a single attribute relation $r_J = \texttt{job}$.

To construct the triple set $\mathcal{T}$, we randomly select one inversion pair $(r_k^I, r_k^{I,-1}) \in \mathcal{R}_{\text{inv}}$ and one symmetric relation $r_l^S \in \mathcal{R}_{\text{sym}}$, where $k, l \in \{1, 2, 3\}$. Each resulting KG contains exactly seven triples: an inversion-consistent pair (e_1^N, r_k^I, e_2^N)

and $(e_2^N, r_k^{I,-1}, e_1^N)$; a symmetry-consistent pair (e_2^N, r_l^S, e_3^N) and (e_3^N, r_l^S, e_2^N); and three job attribute triples (e_i^N, r_J, e_i^J) for $i \in \{1,2,3\}$.

A.3.2 Training Corpus Construction

In the previous subsection, we constructed a synthetic KG whose relations explicitly exhibit inversion and symmetry. We now define how the KG triples are transformed into the pre-training corpora:

$$
\begin{aligned}
D_\alpha &= \mathcal{P}_\alpha(N, K; \mathcal{F}) \\
&= \left\{ \left\{ \text{Shuffle}\left(\{ f^{(\tau_i^{(k)})}(T_i^{(n)}) \}_{i \in \mathcal{I}_\alpha} \right) \right\}_{k=1}^{K} \right\}_{n=1}^{N}, \\
&\quad \text{where } \tau_i^{(k)} \overset{\text{i.i.d.}}{\sim} \text{Unif}(\{1,2,3,4\}), \\
&\quad \mathcal{I}_\alpha = \begin{cases} \{1,2,3,4,5,6,7\}, & \text{if } \alpha = \text{train}, \\ \{1,3,5,6,7\}, & \text{if } \alpha = \text{evaluation}. \end{cases}
\end{aligned}
$$

Here D_α denotes the training corpus and is generated by $\mathcal{P}_\alpha$ parameterized by $N = \texttt{num\_train\_samples}$, $K = \texttt{num\_template}$, and the fixed set of four sentence formats $\mathcal{F} = \{f^{(1)}, f^{(2)}, f^{(3)}, f^{(4)}\}$(Detail examples has been shown in Table 4). The outer set $\{\cdot\}_{n=1}^N$ indexes the N semantic samples, where the n-th sample is defined by a tuple of several KG triples $T_i^{(n)}$. For each semantic sample, we generate K different paragraphs express same information, indexed by $\{\cdot\}_{k=1}^K$, each corresponding to a paragraph-level template choice. Within the k-th paragraph, each sentence $i \in \mathcal{I}_\alpha$ is produced by first drawing a format index $\tau_i^{(k)}$ independently and uniformly from $\{1,2,3,4\}$, then transforming triple $T_i^{(n)}$ with the selected format via $f^{(\tau_i^{(k)})}(T_i^{(n)})$. Finally, $\text{Shuffle}(\cdot)$ randomly permutes the resulting several sentences to form the paragraph text, yielding K shuffled paragraphs per semantic sample and N samples in total for D_α.

We create two pre-training corpora with different completeness. Train content D_{train} uses $\mathcal{I}_{\text{train}}$ (7 sentences) and thus contains full inversion/symmetry information; its sample count equals $N = \texttt{num\_train\_samples}$. Evaluation content D_{eval} uses $\mathcal{I}_{\text{eval}}$ (5 sentences) by removing one direction of each inversion/symmetry pair, serving for evaluation. Train and evaluation content are generated independently, and the number of evaluation samples is fixed to 500. Finally, we merge the train content D_{train} and the evaluation content D_{eval} to form the pre-training corpus for the LLM.

A.3.3 SFT Corpus Construction

As illustrated in Figure 2 and examples shown in Table 3, after constructing the synthetic KG and the pre-training corpora D_{train} and D_{eval}, we build a supervised fine-tuning (SFT) corpus to equip the model with question-answering capability. We generate QA pairs only from the train content:

$$
B_{\text{train}} = \mathcal{S}(D_{\text{train}}; \mathcal{U}) = \left\{ \{(\text{Q}_{n,i}, A_{n,i})\}_{i=1}^{7} \right\}_{n=1}^{N},
$$

where $N = \texttt{num\_train\_samples}$ and $\mathcal{U} = \{u^{(1)}, u^{(2)}, u^{(3)}, u^{(4)}\}$ is a set of four question templates. Concretely, for each train semantic sample n, we construct seven QA pairs $\{(\text{Q}_{n,i}, A_{n,i})\}_{i=1}^7$, each derived from one of its seven underlying triples, with answers deterministically extracted from the queried triple. Importantly, we do not generate any QA pairs from the evaluation content (i.e., $\mathcal{Q}_{\text{eval}} = \varnothing$); since evaluation queries are posed on D_{eval}, this avoids target leakage and ensures that evaluation-time performance reflects inference rather than supervised exposure.

A.4 Other Results

A.4.1 Effect of Increasing Templates Under Fixed Train Samples

With $(N = 2000)$ and $(L = 12)$ fixed, in addition to the $(K = 10)$ experiment that tracks evaluation accuracy as a function of training steps (E), we also ran experiments with $(K = 15, 30, 50,)$ and (100), corresponding to Figure 9(a)–(d). As shown in the figure, except for in-context Inversion and Symmetry which remain at 0 throughout, all other evaluation metrics increase at early stages and then decline as (E) grows. These results confirm the same training-step trend and further indicate that increasing only the number of templates (K) (without increasing the number of samples (N)) is insufficient to induce emergent logical understanding.

A.4.2 More Detail and Experiment of Layer-wise Analysis

Experiment setup for Layer-wise analysis for ICL completion In Result 2 (Task 4.1), we analyze how model confidence evolves across layers by measuring, for $N \in 2000, 10{,}000, 20{,}000$, the logit, probability, and vocabulary rank of the **first correct answer token** conditioned on an in-context prompt. For a 12-layer GPT-2 model, the final next-token distribution is obtained by applying the shared output projection (LM head) to the final

N	#Tokens (Million)	Num_iters/Epoch	Best PT Epochs	PT Time (h)	Best SFT Iters
2,000	2.0	4.16	2000	1.2	2000
5,000	4.7	9.56	1100	1.2	2000
10,000	9.14	18.59	1100	2.5	2000
20,000	18.0	36.6	1400	5.0	2000
50,000	53.8	109.0	460	5.0	3000
100,000	89.0	180.0	1000	20.0	3000

Table 1: Best training configurations under fixed $L = 12$ and $K = 10$ for different dataset sizes N. Token counts and training times are approximate estimates measured on 8×A100 (40GB). "PT" denotes Pre-train in the table.

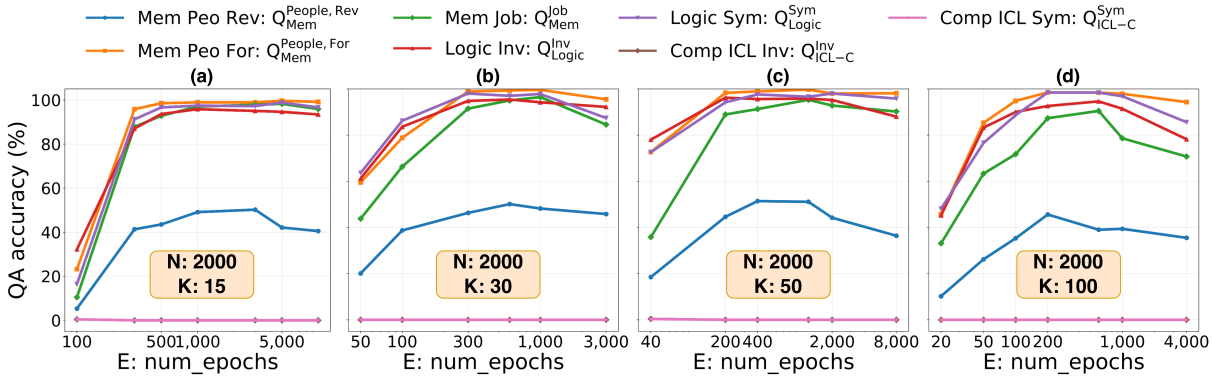

Figure 9: Evaluation accuracy across training steps E for $N = 2000$, $L = 12$, and template counts $K \in \{15, 30, 50, 100\}$. Increasing K alone does not induce in-context inversion/symmetry understanding.

Task / Question Type (Reverse)	Count
QA People (Memorize)	1000
QA Job (Memorize)	1500
QA Inversion (logic)	500
QA Symmetry (logic)	500
ICL Completion (Inversion)	1800
ICL Completion (Symmetry)	900
ICL QA (Inversion)	1800
ICL QA (Symmetry)	900

Table 2: Counts of evaluation questions under a fixed evaluation set of 500 samples. Forward and reverse formats are balanced; we report reverse counts only.

hidden state (after the last layer normalization), followed by softmax. To obtain layer-wise signals, for each layer l we take its intermediate hidden representation and decode it using the **same LM head** (and the corresponding normalization used for decoding), producing a full-vocabulary logit vector and the associated probabilities. We then extract the logit, probability, and rank of the gold answer's first token (e.g., for the answer "Noah Dylan Martinez", we track the token "Noah"). Repeating this procedure for all layers and all 1,800 prompts, we report per-layer mean logit, mean probability, and mean rank.

More experiments about layer-wise analysis As shown in Figure 13, we additionally report the layer-wise mean logit, mean probability, and mean rank for the model trained on the synthetic corpus with $N = 5000$ and $K = 10$. Although its final accuracies are relatively low (Inversion: 18%, Symmetry: 34%), the overall layer-wise dynamics closely match those of the higher-performing models trained with $N = 10{,}000$ and $N = 20{,}000$: the model begins to improve rapidly in the mid layers (approximately layers 5–8) and continues to strengthen steadily in later layers, without the last-layer collapse observed in smaller-data regimes.

To facilitate comparison, we aggregate the layer-wise mean logit, mean probability, and mean rank for $N \in 2000, 5000, 10{,}000, 20{,}000$ into a single figure, with Inversion and Symmetry reported separately. Figure 10 first shows in (a)(b) that the mean logits increase monotonically across layers for all N, while larger N yields higher logits in the final layer; in particular, the $N = 20{,}000$ model attains a substantially higher final-layer logit than the other settings, especially compared to

Stage	Example
Triples (Train Graph)	(1) (Edward Dane Burke, husband, Gabriel Bode Arnold) (2) (Gabriel Bode Arnold, wife, Edward Dane Burke) (3) (Gabriel Bode Arnold, friend, Michael Brett Ferguson) (4) (Michael Brett Ferguson, friend, Gabriel Bode Arnold) (5) (Edward Dane Burke, job, project manager) (6) (Gabriel Bode Arnold, job, consultant) (7) (Michael Brett Ferguson, entertainment manager)
Simple paragraph (canonical verbalization)	(1) Edward Dane Burke is the husband of Gabriel Bode Arnold. (2) Gabriel Bode Arnold is the wife of Edward Dane Burke. (3) Gabriel Bode Arnold is the friend of Michael Brett Ferguson. (4) Michael Brett Ferguson is the friend of Gabriel Bode Arnold. (5) project manager is the job of Edward Dane Burke. (6) consultant is the job of Gabriel Bode Arnold. (7) entertainment manager is the job of Michael Brett Ferguson.
Paragraph after template sampling + shuffle	(1) Edward Dane Burke serves as Gabriel Bode Arnold's husband. (2) Gabriel Bode Arnold serves as Michael Brett Ferguson's friend. (3) Michael Brett Ferguson acts in the role of friend to Gabriel Bode Arnold. (4) Edward Dane Burke is employed as a project manager. (5) Gabriel Bode Arnold is employed as an it consultant. (6) Michael Brett Ferguson works as an entertainment manager. (7) Gabriel Bode Arnold is the wife of Edward Dane Burke.
SFT corpus (QA generated from train content)	(1) Q: Who holds the relation of husband to Gabriel Bode Arnold? A: Edward Dane Burke (2) Q: Who is the friend of Michael Brett Ferguson? A: Gabriel Bode Arnold ... (6) Q: What does Michael Brett Ferguson work as? A: entertainment manager ...

Table 3: End-to-end text generation pipeline: from triples to paragraph realizations, and to SFT QA pairs (generated only from the training content).

$N = 2000$. In (c)(d), the mean rank curves for $N = 5000, 10{,}000$, and $20{,}000$ drop sharply starting around layer 5 and reach strong performance by layer 8, continuing to improve through layer 12. By contrast, the $N = 2000$ model improves more slowly in mid layers, only dropping markedly around layer 8, and then exhibits a late-layer degradation with rank increasing at layers 11–12. The mean probability curves in (e)(f) mirror the rank trends, with the $N = 2000$ model maintaining very low probabilities overall. This consolidated view highlights the clear gap between the weak logit-level understanding under $N = 2000$ and the substantially stronger, stable layer-wise dynamics for $N \geq 5000$, corroborating our earlier results and discussion.

A.4.3 Effect of SFT Data Fraction

To better understand the role of SFT, we conduct an additional experiment that varies the amount of SFT data. Our SFT corpus is derived from the Train Content: for each sentence verbalized from a triple, we construct a corresponding QA pair to equip the pre-trained model with question-answering capability. In all previous experiments, we fine-tuned on the full SFT corpus. Since the model pre-trained with $N = 2000$ and $K = 10$ already achieves strong QA performance after fine-tuning, we fix this pre-training setting and run SFT using only 10% to 100% of the SFT corpus. As shown in Fig-

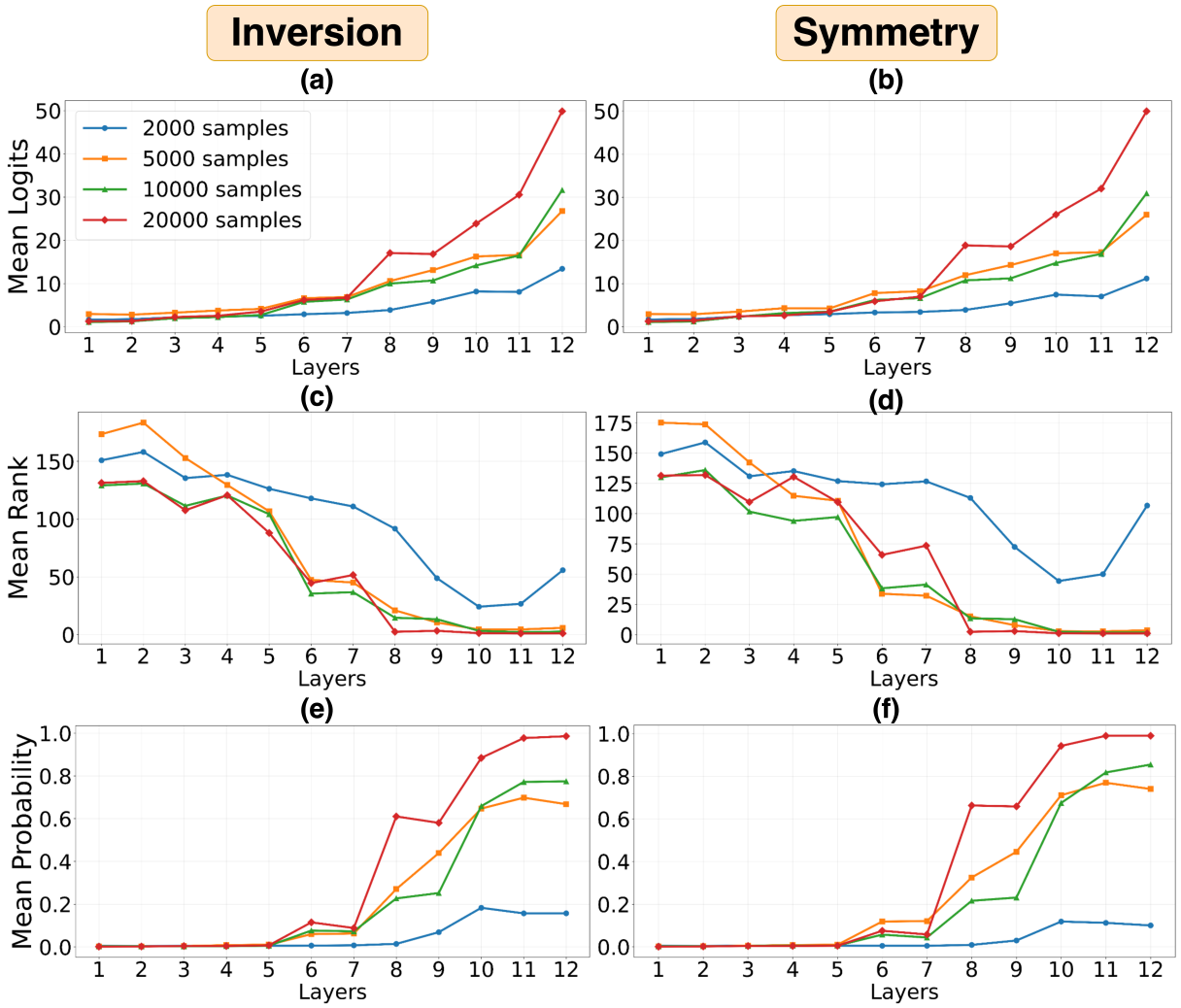

Figure 10: Layer-wise mean logit, mean rank, and mean probability of the first correct token for $N \in \{2000, 5000, 10{,}000, 20{,}000\}$ (Inversion and Symmetry shown separately), highlighting stronger and more stable late-layer behavior for larger N and late-layer degradation for $N = 2000$.

ure 11, we evaluate $Q_{\text{non}}^{\text{People}}$ (forward and reverse), $Q_{\text{non}}^{\text{Job}}$ (reverse), and Q_{Logic} (Inversion and Symmetry), and additionally test QA on training content excluded from SFT (e.g., when using 10% of the SFT corpus, we sample QA instances from the remaining 90% portion). We find that performance is modest at 10%, but starting from 20% the accuracies exhibit a clear turning point and already reach high values, indicating that even a small fraction of the SFT corpus is sufficient for effective fine-tuning.

A.5 Experiments on Diffusion Models

A.5.1 Implementation Details

To ensure a rigorous comparison with the AR baseline, our Diffusion LM is built upon the same **GPT-2 small** backbone ($L = 12$, hidden size 768, 12 heads). However, we introduce key architectural adaptations to support the non-autoregressive training objective, strictly following the LLaDA framework (Nie et al., 2025):

- **Bidirectional Attention**: We remove the causal masking in the self-attention layers. This allows every token to attend to the entire sequence context (both prefix and suffix) simultaneously, which is fundamental for learning joint probability distributions.
- **Training Objective**: The model is trained using a **Bernoulli masking** strategy. At each training step, tokens are independently masked with a probability $t \sim U(0, 1)$, and the model optimizes the re-weighted cross-entropy loss on the masked positions.
- **Inference Strategy**: We employ a **block-wise generation** strategy with low-confidence re-

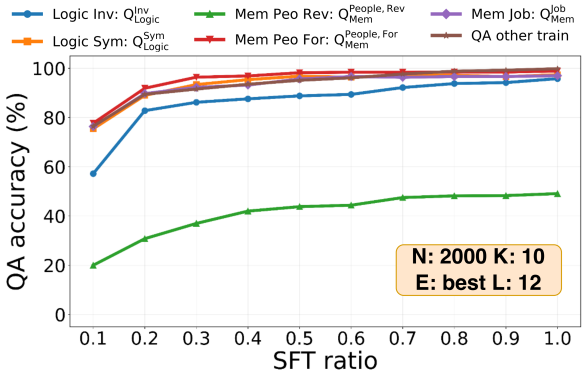

Figure 11: Effect of SFT data fraction (10%–100%) on evaluation accuracy for the $N = 2000$, $K = 10$ pre-trained model, showing a sharp improvement around 20% and strong performance with limited SFT data.

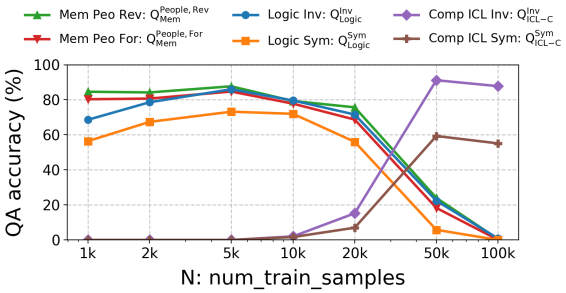

Figure 12: Performance of the Diffusion Model across different training scales (N).

masking. Specifically, the answer is generated in blocks of 4 tokens. Within each block, we iteratively refine the tokens by re-masking positions with the lowest confidence scores, allowing the model to self-correct based on the global bidirectional context.

A.5.2 Data Adjustment

A critical challenge in applying diffusion models to relational logic is the potential for "textual interpolation." In a standard paragraph containing coupled sentences, a uniform random mask rarely obscures one specific sentence entirely. Consequently, the model may learn to reconstruct missing tokens by interpolating from residual fragments of the symmetric sentence in the same context window, rather than learning the logical implication itself.

To address this, we restructured the pre-training corpus into three components with a ratio of **Raw : Single : Subset = 1 : 1 : 2**:

- **Raw (25%)**: Original complete paragraphs to maintain document-level coherence.
- **Single (25%)**: Randomly sampled individual sentences. This forces the model to learn atomic syntax and entity structures without reliance on external context.
- **Subset (50%)**: Random subsets containing 2 to $N - 1$ sentences. This component is crucial as it simulates "missing link" scenarios, compelling the model to internalize the underlying logical rules to recover the missing information during training.

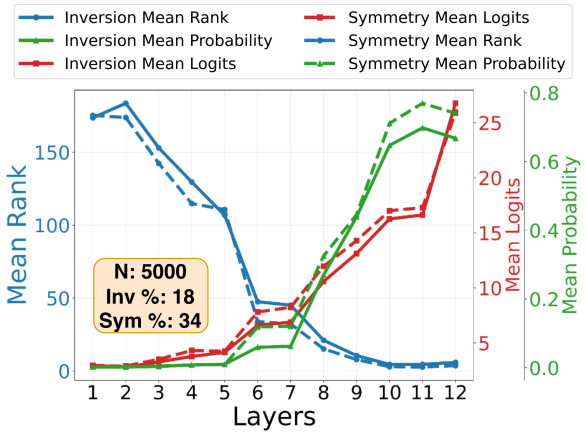

Figure 13: Layer-wise mean logit, mean probability, and mean rank of the first correct token for the model trained with $N = 5000$ and $K = 10$, showing mid-layer emergence (layers 5–8) and stable late-layer improvement without last-layer collapse.

A.5.3 Performance Across Different Scales

As shown in Figure 12, the diffusion model's In-context generalization improves with data scale, while rote memorization of Memorize QA facts fluctuates due to capacity constraints. Crucially, however, the immunity to the reversal curse persists across all scales, confirming that this robustness is intrinsic to the bidirectional architecture and holds regardless of data volume.

A.5.4 Discussion: Why does Reverse outperform Forward?

Figure 12 shows that reverse setting is on average about 4% higher than the forward setting. We hypothesize this stems from the **structural proximity of entity anchoring** in our prompts. In Reverse queries (e.g., "*Who serves as [Person B]'s niece?*"), the target entity [Person B] is explicitly provided as a local anchor immediately adjacent to the answer slot, offering a clear, low-entropy guidance signal for the diffusion process.

Category	Templates
People–People sentence templates $\mathcal{F}$ (person_a, rel, person_b)	(1) {person_a} is the {relationship} of {person_b}. (2) {person_a} serves as {person_b}'s {relationship}. (3) {person_a} acts in the role of {relationship} to {person_b}. (4) {person_a} holds the relation of {relationship} to {person_b}.
People–Job sentence templates $\mathcal{F}$ (person, Job, job)	(1) {job} is the job of {person}. (2) {person} works as {article} {job}. (3) {person}'s occupation is {job}. (4) {person} is employed as {article} {job}.
People–People QA question templates (reverse) (person, relationship, ?)	(1) Who is the {relationship} of {person}? (2) Who serves as {person}'s {relationship}? (3) Who acts in the role of {relationship} to {person}? (4) Who holds the relation of {relationship} to {person}?
People–People QA question templates (forward) (person, relationship, ?)	(1) {person} is the {relationship} of who? (2) {person} serves as whose {relationship}? (3) {person} acts in the role of {relationship} to who? (4) {person} holds the relation of {relationship} to who?
People–Job QA question templates (reverse) (person, Job, ?)	(1) What is the job of {person}? (2) What does {person} work as? (3) What is {person}'s occupation? (4) What is {person} employed as?
People–Job QA question templates (forward) (person, Job, ?)	(1) {person}'s job is what? (2) {person} works as what? (3) {person}'s occupation is what? (4) {person} is employed as what?
In-context learning completion templates (person_a, rel, person_b) (person_b, reverse_rel, ?)	(1) {person_a} serves as {person_b}'s {relationship}, {person_b} acts in the role of {reverse_relationship} to (2) {person_a} holds the relation of {relationship} to {person_b}, {person_b} is the {reverse_relationship} of (3) {person_a} acts in the role of {relationship} to {person_b}, {person_a} holds the relation of {reverse_relationship} to

Table 4: Sentence, QA, and in-context completion templates used for synthetic data generation. For symmetric relations, `reverse_relationship` is identical to `relationship`; for inversion relations, `reverse_relationship` is the corresponding inverse relation. The in-context learning QA templates are the union of the People–People sentence templates and the People–Job QA reverse question templates.

Evaluation content (paragraph example)	(1) Gregory Joel Henderson serves as Noah Dylan Martinez's husband. ~~(2) Noah Dylan Martinez is the wife of Gregory Joel Henderson~~ (3) Raymond Clayton Morris is the friend of Noah Dylan Martinez. ~~(4) Noah Dylan Martinez is the friend of Raymond Clayton Morris~~ (5) Gregory Joel Henderson is employed as a voice actor. (6) bus driver is the job of Noah Dylan Martinez. (7) Raymond Clayton Morris works as an event planner.
Evaluation	**QA Example**
Memorize QA (Reverse) $Q_{Mem}^{People,\ Rev}$ and $Q_{Mem}^{Job,\ Rev}$	People: Q: Who holds the relation of husband to Noah Dylan Martinez? A: Gregory Joel Henderson Job: Q: What is the job of Gregory Joel Henderson? A: voice actor
Memorize QA (Forward) $Q_{Mem}^{People,\ For}$ and $Q_{Mem}^{Job,\ For}$	People: Q: Gregory Joel Henderson holds the relation of husband to who? A: Noah Dylan Martinez Job: Q: Gregory Joel Henderson's job is what? A: voice actor
Logic QA (Forward) $Q_{Logic}^{Inv,\ For}$ and $Q_{Logic}^{Sym,\ For}$	Inversion: Q: Who holds the relation of wife to Gregory Joel Henderson? A: Noah Dylan Martinez Symmetry: Q: Who is the friend of Raymond Clayton Morris? A: Noah Dylan Martinez
Logic QA (Reverse) $Q_{Logic}^{Inv,\ Rev}$ and $Q_{Logic}^{Sym,\ Rev}$	Inversion: Q: Noah Dylan Martinez holds the relation of wife to who? A: Gregory Joel Henderson Symmetry: Q: Noah Dylan Martinez is the friend of who? A: Raymond Clayton Morris
ICL Logic QA (Completion) $Q_{ICL\text{-}C}^{Inv}$ and $Q_{ICL\text{-}C}^{Sym}$	Inversion: Q: Douglas Chance Holmes acts in the role of wife to Eric Braden Edwards.Eric Braden Edwards is the husband of A: Douglas Chance Holmes Symmetry: Q: Alan Logan Butler holds the relation of spouse to Jeffrey Brent Brown. Jeffrey Brent Brown holds the relation of spouse to A: Alan Logan Butler
ICL Logic QA (QA) $Q_{ICL\text{-}Q}^{Inv,\ For}$ and $Q_{ICL\text{-}Q}^{Sym,\ For}$	Inversion: Q: Douglas Chance Holmes acts in the role of wife to Eric Braden Edwards. Who is the husband of Douglas Chance Holmes? A: Eric Braden Edwards Symmetry: Q: Alan Logan Butler holds the relation of spouse to Jeffrey Brent Brown. Who holds the relation of spouse to Alan Logan Butler? A: Jeffrey Brent Brown
ICL Logic QA (QA) $Q_{ICL\text{-}Q}^{Inv,\ Rev}$ and $Q_{ICL\text{-}Q}^{Sym,\ Rev}$	Inversion: Q: Douglas Chance Holmes acts in the role of wife to Eric Braden Edwards. Eric Braden Edwards is the husband of who? A: Douglas Chance Holmes Symmetry: Q: Alan Logan Butler holds the relation of spouse to Jeffrey Brent Brown. Jeffrey Brent Brown is the spouse of who? A: Alan Logan Butler

Table 5: Examples of evaluation content and the corresponding evaluation question and answering.