How do Transformers Learn Implicit Reasoning?

Jiaran Ye[♠†] **Zijun Yao**[♠†] **Zhidian Huang**[♠] **Liangming Pan**[♡‡] **Jinxin Liu**[♠]
Yushi Bai[♠] **Amy Xin**[♠] **Weichuan Liu**[◇] **Xiaoyin Che**[◇] **Lei Hou**[♠‡] **Juanzi Li**[♠]

[♠]DCST, BNRist; KIRC, Institute for Artificial Intelligence, Tsinghua University, *China*
[♡]MOE Key Lab of Computational Linguistics, Peking University, *China* [◇]Siemens AG, *China*

`{yejr23, yaozj20}@mails.tsinghua.edu.cn`
`{houlei, lijuanzi}@tsinghua.edu.cn`

Abstract

Recent work suggests that large language models (LLMs) can perform multi-hop reasoning implicitly—producing correct answers without explicitly verbalizing intermediate steps—but the underlying mechanisms remain poorly understood. In this paper, we study how such implicit reasoning emerges by training transformers from scratch in a controlled symbolic environment. Our analysis reveals a three-stage developmental trajectory: early memorization, followed by in-distribution generalization, and eventually cross-distribution generalization. We find that training with atomic triples is not necessary but accelerates learning, and that second-hop generalization relies on query-level exposure to specific compositional structures. To interpret these behaviors, we introduce two diagnostic tools: cross-query semantic patching, which identifies semantically reusable intermediate representations, and a cosine-based representational lens, which reveals that successful reasoning correlates with the cosine-base clustering in hidden space. This clustering phenomenon in turn provides a coherent explanation for the behavioral dynamics observed across training, linking representational structure to reasoning capability. These findings provide new insights into the interpretability of implicit multi-hop reasoning in LLMs, helping to clarify how complex reasoning processes unfold internally and offering pathways to enhance the transparency of such models.

`https://github.com/Jiaran-Ye/ImplicitReasoning`

1 Introduction

Large language models (LLMs) demonstrate strong performance on complex, multi-step reasoning tasks [9, 7, 29, 1, 32, 19, 14]. Typically, these reasoning abilities are elicited using chain-of-thought (CoT) prompting, which encourages models to explicitly articulate intermediate reasoning steps [28, 35, 6, 33, 30]. Beyond CoT, recent studies indicate that LLMs can also engage in ***implicit reasoning*** [37, 5, 11, 15], producing correct answers without verbalizing intermediate steps.

While implicit reasoning is widely acknowledged, the internal mechanisms that empower this ability remain unclear. In this paper, we aim to uncover the internal processes of implicit reasoning by examining a concrete, structured scenario: *multi-hop implicit reasoning*, where the model must answer compositional queries (*e.g.*, $(e_1, r_1, r_2) \rightarrow e_3$) by implicitly traversing an intermediate entity e_2, without explicitly verbalizing it. A fundamental question in this scenario is that: does the model genuinely conduct step-by-step reasoning internally, or is it merely recalling the answer from its memorized knowledge? Although both behaviors can produce correct outcomes, they reflect

[†] Equal Contribution.
[‡] Corresponding Author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

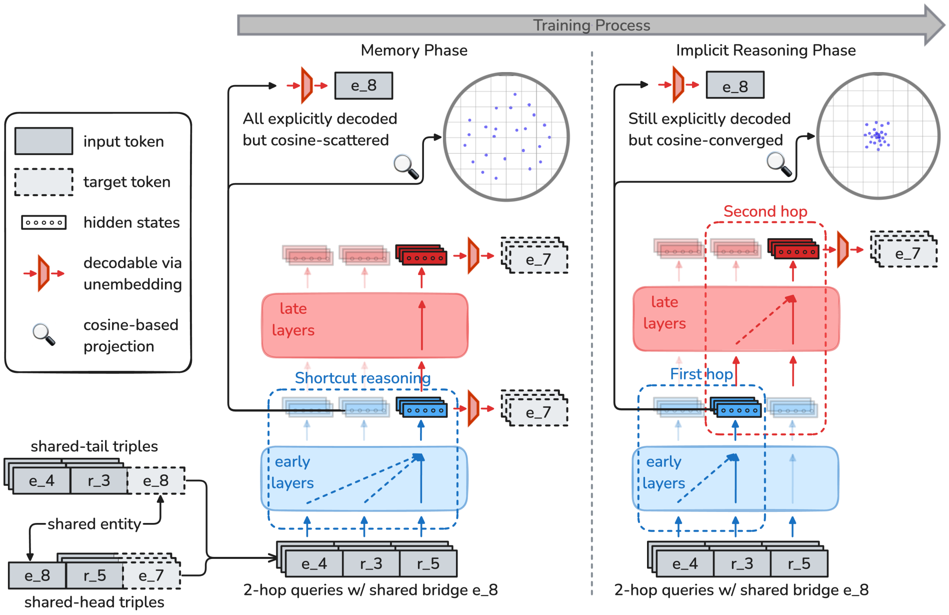


Figure 1: We track the evolution of bridge entity representations across training. In the memory phase (left), intermediate entities are explicitly decodable but geometrically scattered. In the implicit reasoning phase (right), representations converge in cosine space, supporting structured multi-hop reasoning through early-to-late layer transitions. Queries that share an bridge entity are composed for cosine-based representation analysis.

fundamentally distinct cognitive processes. This observation motivates our central research question: *How do LLMs acquire and perform implicit reasoning during training and inference?*

Existing studies that investigate implicit reasoning often rely on pretrained LLMs whose training data lacks precise experimental control, making it challenging to conclusively determine whether models have genuinely learned implicit multi-step reasoning or instead rely on its prior knowledge or shortcut solutions [12, 34, 5]. Symbolic datasets [25, 27, 26] partially alleviate this concern by training models from scratch, yet they still lack the fine-grained experimental control and behavioral granularity necessary for deeper analysis. To address these limitations, we construct *an extended symbolic environment*, featuring targeted omissions and query-level variations, to precisely identify whether implicit reasoning and generalization truly emerge.

To facilitate the analysis under our symbolic environment, we introduce two diagnostic tools that overcome specific limitations of prior methods: (1) *cross-query semantic patching*, which enhances causal interpretability by locating intermediate entity representations based on their semantic transferability across queries rather than solely their impact on final outputs; and (2) *a cosine-based representational lens*, which avoids assumptions inherent in decoding-based probing by examining structural consistency of internal representations across reasoning contexts. Together, these tools enable precise examination of the internal processes driving implicit reasoning.

Our empirical analysis begins with a behavioral study conducted under fine-grained experimental control (Section 2). Under a complete training configuration, we observe that multi-hop implicit reasoning emerges in three distinct stages: memorization, in-distribution generalization, and finally cross-distribution generalization. Through ablation studies, we further demonstrate that while exposure to in-distribution (ID) triples is not strictly necessary for achieving in-distribution generalization, its absence significantly delays the onset of this behavior. Additionally, we find that generalization to second-hop queries fails unless the model encounters exact compositional structures during training, revealing a strong dependency on query-level exposure.

These behavioral insights reveal previously unreported patterns, motivating us to revisit and probe the internal mechanisms of implicit reasoning. In Section 3, we first use cross-query semantic patching to localize intermediate entity representations, typically identifying them within the middle layers corresponding to the r_1 tokens. We then test the common assumption that intermediate entities are explicitly decodable from internal states and find this assumption inconsistent with our observed reasoning behavior. This disconnect leads us to adopt a geometric perspective, wherein successful reasoning strongly correlates with consistent clustering of intermediate representations within cosine similarity space (Figure 1).

In Section 4, we close the loop by explicitly connecting these internal representational mechanisms to external behavioral patterns. We demonstrate that successful generalization robustly correlates with the clustering structure of intermediate representations across diverse queries and training distributions. Although in-distribution (ID) triple supervision is not required to induce this clustering, it substantially accelerates its emergence by constraining the representational space early in training. Finally, we identify that what appears to be first-hop generalization to out-of-distribution (OOD) triples is actually an artifact arising from representational alignments induced by ID exposure, highlighting the fragile and data-dependent nature of implicit generalization.

Collectively, our results provide a comprehensive account of how implicit multi-hop reasoning emerges within LLMs — grounded in observable behaviors, elucidated through mechanistic analyses, and offering foundational insights for future studies on model interpretability.

2 Behavioral Signatures of Implicit Reasoning under Fine-Grained Control

Existing studies on implicit reasoning fall into two broad categories, each with notable limitations. (1) Analyses based on pretrained LLMs operate in an uncontrolled setting where the training data are opaque—making it difficult to distinguish genuine reasoning from memorization. (2) In contrast, recent works adopt symbolic datasets with synthetic training from scratch [25], but primarily focus on *dataset-level trends*, without isolating *what specific training signals are necessary* for solving each compositional query.

We argue that an ideal analysis setting should satisfy three key properties: (1) **compositional structure**, to support multi-step inference; (2) **fine-grained control**, to support query-level ablations and conditionally constructed variants; and (3) **behavioral resolution**, to distinguish between memorization, generalization, and reasoning. With these goals in mind, we construct a symbolic training environment that extends prior datasets with new configurations and targeted omissions, and reveals several behavioral phenomena not captured in prior work.

To achieve this, we adopt GPT-2 as our base model due to its balance of capacity and tractability, and verify the scalability of results using larger models. Full training details are provided in Appendix G.

2.1 Data Construction: Fine-Grained Control for Compositional Reasoning

To enable fine-grained behavioral analysis, we extend the symbolic reasoning setup of Wang et al. [25] with expressive query-level control configurations. The data comprises atomic triples and compositional queries:

- **Atomic Triples.** Each atomic fact is represented as a triple $(e_1, r_1) \rightarrow e_2$. This formulation mimics simple factual relations such as *(Alice, mother-of)* → *Beth* and *(Beth, sister-of)* → *Carol*, serving as the atomic unit of the reasoning environment. The triples are partitioned into two subsets: *In-Distribution (ID) Triples* are used in both standalone form and as components of multi-hop training queries; *Out-of-Distribution (OOD) Triples* appear in training data only in standalone form, and are excluded from multi-hop composition, enabling the creation of test queries involving out-of-distribution reasoning. Note that ID and OOD triples share the same set of entities and relations.
- **2-Hop Queries.** Each reasoning task takes the form of a compositional chain $(e_1, r_1, r_2) \rightarrow e_3$, where the model performs implicit reasoning over an bridge entity e_2. For instance, the model receives only the compositional query *(Alice, mother-of, sister-of)* and is expected to predict the correct target *Carol*, implicitly reasoning through the intermediate entity *Beth*. We distinguish: *Test-OI*: test queries where the first hop comes from an OOD triple and the second hop from an ID triple; *Train-II*: queries with both hops from ID triples used during training. Other query types, such as *Test-II*, *Test-IO*, and *Test-OO*, follow similar definitions.

Training Configurations. Our **base configuration** (Figure 6a) includes all atomic triples (ID and OOD) and the full set of Train-II queries, and evaluate on Test-II, Test-OI, Test-OO, and Test-IO, allowing comprehensive generalization assessment. To isolate the conditions for generalization, we define **a flexible family of training variants** that omit specific triples, restrict compositional roles, or remove entire subsets, allowing targeted query-level ablations. This design supports controlled investigations into the functional dependencies behind implicit reasoning behaviors.

Extension to 3-hop Reasoning. Although our main analyses focus on 2-hop queries, the same construction framework naturally applies to 3-hop settings, exhibiting consistent behavioral and mechanistic patterns. For more details, refer to Appendix C.

Together, these configurations serve as the foundation for our study. Further dataset construction details and illustrations are available in Appendix B.

2.2 Three-Stage Generalization

Leveraging the base configuration introduced in Section 2.1, we track model performance throughout training and observe a striking behavioral trajectory that unfolds in three distinct phases (Figure 2):

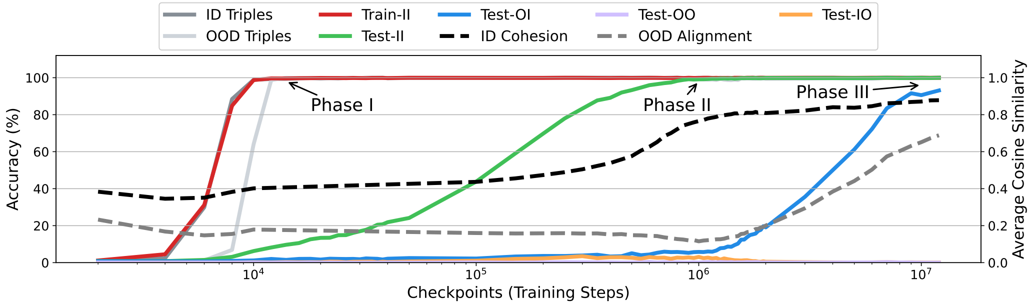


Figure 2: Training dynamics under the base configuration, revealing three distinct phases. Accuracy curves track model performance on different query types, while dashed lines plot the ID Cohesion and OOD Alignment scores (Section 3.3).

Phase I: Memorization. The initial stage involves quickly fitting the training data, including atomic facts and 2-hop compositions. The model memorizes these facts, but generalization to unseen queries remains minimal.

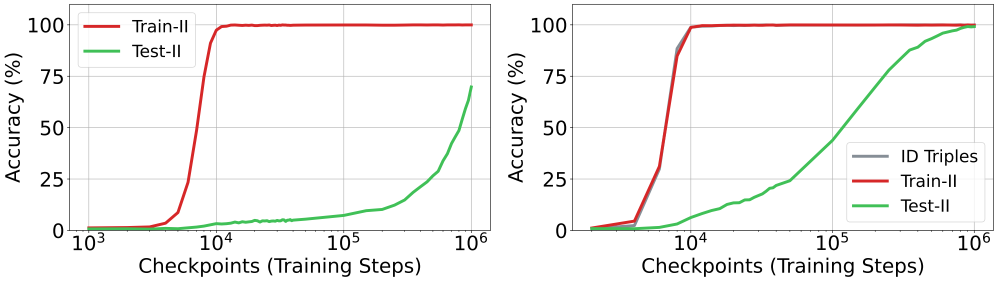
Phase II: ID Generalization. After memorization saturates, the model begins to generalize to Test-II queries (unseen ID-ID compositions), marking a shift from memorization to compositional generalization within ID, akin to the *grokking* phenomenon described by Wang et al. [25].

Phase III: Cross-Distribution Reasoning. The model next learns to generalize across distributions, gradually incorporating OOD triples in the first hop while maintaining the ID in the second. This transition is slower than Phase II and requires more training. Building on the grokking phenomenon, our analysis uncovers this additional phase of generalization across distributional boundaries.

Interestingly, generalization fails consistently when the second hop is from OOD triples, revealing a stronger bottleneck in the second relational step. These phases show that reasoning develops in structured stages, each with distinct patterns of success and failure, highlighting the need to treat reasoning not as a monolithic ability, but as a set of behaviors with separable developmental conditions.

2.3 ID Triples Are Not Required for ID Generalization—but Accelerate It

Prior work has repeatedly observed that while models can correctly answer individual atomic triples, they often fail to generalize to 2-hop queries constructed by composing those same triples[3, 31, 42]. Additionally, in Section 2.2, we observe that our model quickly memorized atomic triples (Phase I) but took longer to generalize to Test-II queries. These findings raise a natural question: *Are atomic ID Triples actually necessary for learning ID-based 2-hop reasoning?*



(a) Train-II only

(b) Train-II + ID triples

Figure 3: Training trajectories under two configurations used in Section 2.3. (a) Only Train-II queries are used, with no exposure to atomic ID triples. (b) Both Train-II queries and atomic ID triples are included in training.

To investigate this, we test a minimal training configuration that excludes both ID and OOD Triples, using only Train-II queries. Surprisingly, the model still generalizes to unseen ID combinations Test-II (Figure 3a). That is, training with Train-II alone is sufficient for ID-based generalization.

However, when comparing this minimalist setting to a variant where ID Triples are included[1] alongside Train-II, we found that generalization to Test-II occurs significantly earlier (Figure 3b), suggesting that while atomic facts are not required for generalization, they **accelerate** learning.

2.4 Second-Hop Generalization Requires Query-Level Training Match

While the model can generalize when OOD triples appear in the first hop (Section 2.2), it consistently fails when the OOD component is in the second hop. This raises a natural follow-up question: *What training data is necessary to enable second-hop generalization within the ID domain*?

Building on Section 2.3, where training with only Train-II queries still led to ID generalization, we perform a targeted ablation to isolate the role of second-hop coverage. Specifically, we remove a subset of atomic triples (e.g., (e_B, r_5, e_F)) from being used as second hops in any Train-II query. We then test whether the model could correctly answer Test-II queries that involved these **excluded triples as second-hop** (Figure 6b).

We find that the model consistently fails to answer these Test-II queries, while performance on other queries remained unaffected, even when the same atomic triples were used in the first hop. This confirms that **second-hop generalization requires query-level training match**: the model must encounter the specific second-hop composition during training to generalize over it. Exposure to the same facts in other structural roles is not sufficient.

To further validate this finding, we replicat the ablation under the full base configuration (Appendix D.1) and observe the same failure. Additionally, we analyze how second-hop exposure frequency impacts query acquisition order (Appendix D.2). We found that Test-II queries involving a particular atomic triple as the second hop were answered correctly earlier when that atomic triple appeared more frequently as a second hop during training.

3 Locating and Characterizing Reasoning Representations

In Section 2, we observed surprising behavioral phenomena that offer new insights into implicit reasoning in Transformers. These findings challenge prevailing assumptions, such as the necessity of exposure to atomic facts [40, 36] or the impossibility of OOD generalization [25].

To better understand these behaviors, we shift our focus from **what the model does** to **how the model achieves it internally**. Specifically, we examine the intermediate entity that connects the two relational steps. Any correct solution, at least implicitly, passes through this latent bridge, making it a key target for probing the model's reasoning process. To this end, we first introduce the causal

[1] We exclude OOD Triples from both configurations to ensure comparability: since the "Train-II only" configurations contains no OOD facts, generalization to Test-OI is inherently impossible.

probing method *Cross-Query Semantic Patching* (Section 3.1) to locate these intermediate entities in representations. We then revisit whether internal states are decodable using logit lens (Section 3.2). We finally explore how geometric regularity in these representations supports the model's ability to generalize across queries (Section 3.3).

3.1 Locating Intermediate Entity Representations via Cross-Query Patching

To analyze how transformers internally represent intermediate reasoning steps, a crucial first step is to identify **where** such representations are encoded in the model's hidden states.

Existing methods such as linear probing and causal patching offer only partial insight. Linear probing reveals correlations between hidden states and output tokens, but not their causal role in reasoning. Causal patching assesses causal influence, typically measures whether a random source activation affects the target's output, but doesn't assess what the activation semantically represents [25].

Cross-Query Semantic Patching. To go beyond correlation or superficial causal influence, we introduce *cross-query semantic patching*, a method designed to test whether a hidden representation encodes a semantically valid intermediate entity. Specifically, given a source query (e_1, r_1, r_2), we test a set of candidate hidden states from different layers and positions (*e.g.*, layer 3 at the r_1 position) that may contain the bridge entity representation. For each candidate, we insert its hidden vector into a structurally similar target query (e_5, r_6, r_7) at the same position, replacing the original hidden state.

If the patched model's prediction changes from the original reasoning path $r_7(r_6(e_5))$ to $r_7(r_1(e_1))$, this indicates that the inserted representation carries transferable semantic information corresponding to the bridge entity.

We apply this patching procedure across multiple layers and token positions, with three settings that differ in both the source of the intermediate entity and the model's training stage: (1) Phase II with ID-derived intermediate entities, (2) Phase III with ID-derived intermediate entities, and (3) Phase III with OOD-derived intermediate entities. This alignment ensures that patching is conducted under conditions where the model is capable of reasoning over the relevant intermediate entity type. For completeness, we also report Phase I results, where patching yields negligible success, confirming that reasoning-relevant representations emerge only after generalization.

Detailed per-setting results are presented in Appendix F. We report the average patching success rate across these three settings in Figure 4. This aggregated result shows that effective patching occurs primarily at the r_1 token position in the middle layers. In the following analyses, we use **layer 5 at the r_1 token position** of our 8-layer GPT-2 model as the reference point, denoting the hidden state as $\mathbf{h}^5_{\mathbf{r}_1}$.

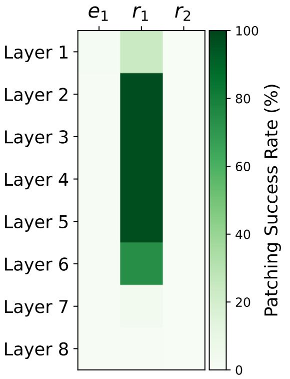


Figure 4: Average patching success rate across layers and token positions.

3.2 Explicitly Decodable $\neq$ Implicitly Informative

Having located the positions encoding intermediate entities, we next ask whether their internal role can be explained using existing interpretability tools. A key assumption in prior work is that if a hidden state encodes an intermediate entity, it should be decodable into a human-interpretable token, for example via the logit lens [18, 16, 33, 22, 40]. We test this assumption by measuring the decodability of $\mathbf{h}^5_{\mathbf{r}_1}$, and examining whether decodability aligns with the emergence of reasoning behavior across training phases.

Setup. We adopt the logit lens to evaluate decoding performance in two modes: (1) **Immediate probing**: projecting the extracted hidden state onto the output vocabulary directly. (2) **Full-run probing**: patching the extracted hidden state into a randomly selected query at the corresponding position, and decoding it after processing through the model's layers. These methods assess whether the hidden state contains a token-level signal or whether the model itself can internalize and recognize

Table 1: Success rates (%) of explicitly decoding intermediate entities from $\mathbf{h}^{5}_{\mathtt{r}_1}$ across reasoning phases and decoding methods.

Source	**Immediate Probing**			**Full-run Probing**		
	Phase I	Phase II	Phase III	Phase I	Phase II	Phase III
ID-derived	92.1	98.8	99.9	97.1	99.9	99.9
OOD-derived	67.7	81.3	99.8	83.7	98.6	99.7

it. For each phase, we compute the decoding success rate for intermediate entities grouped by their origin—either from ID or OOD triples[2].

Result 1: Decodability does not correlate with reasoning emergence. As shown in Table 1, decoding success remains high and stable for ID-derived representations across all phases. However, implicit reasoning capabilities only emerge after Phase II, suggesting that decodability alone does not explain reasoning emergence.

Result 2: No decodability gap between ID and OOD sources during cross-distribution generalization. In Phase II, while the model generalizes to ID-ID (Test-II) queries but fails on ID-OOD (Test-OI) queries, there is no significant difference in decoding success between ID-derived and OOD-derived representations. This further demonstrates that representations can be equally decodable yet differ in whether they are functionally recognized and utilized by the model.

Implication. These results indicate that explicit decodability alone cannot explain when or how a representation contributes to reasoning. Even when a representation can be decoded correctly, the model may not rely on it for reasoning.

To further probe the role of explicit decoding in reasoning, we constructed a controlled setting where the model was incentivized to represent intermediate entities in a decodable form. Interestingly, the model initially attempts this strategy but quickly abandons it, indicating that the model prefers non-explicit representations for generalization. We provide details of this experiment in Appendix E.

3.3 Geometric Regularity of Intermediate Representations via Cosine Lens

The gap between explicit decodability and actual usage motivates a different approach. Instead of asking *"Can we decode what this hidden state?"*, we ask *"How is this representation organized across different contexts?"*

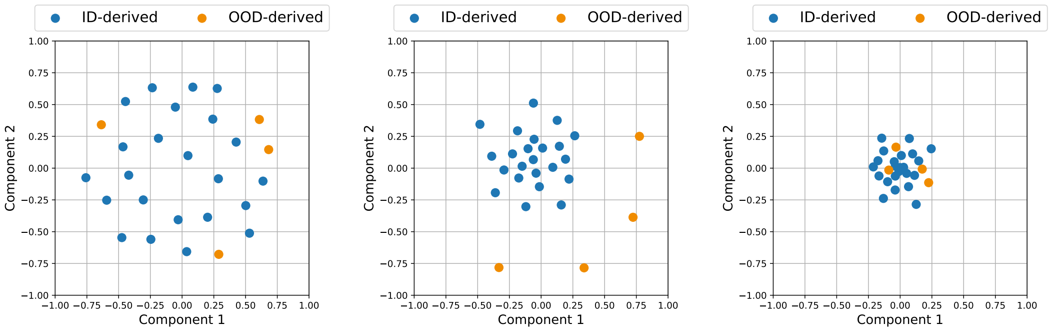
Most prior work focuses on decoding representations, but we take the reverse approach: **given a known intermediate entity**, can we identify recurring structure in how the model represents it across contexts to achieve consistent representations [24]? This reverse mapping is enabled by our earlier analysis in Section 3.1, which identifies the position $\mathbf{h}^{5}_{\mathtt{r}_1}$ encoding the intermediate entity. At this anchor, we collect hidden states from queries sharing the same intermediate entity (Figure 1), and examine whether these vectors reflect a consistent internal pattern.

To assess consistency, we focus on structural alignment in the model's embedding space, using *cosine similarity*—a common metric for semantic proximity in high-dimensional representations[8, 21, 13]. This allows us to examine whether the model reuses internal abstractions through representational geometry, instead of relying on explicit decodability.

Case Study: Visualizing Representational Clustering. To gain an initial sense of the representational patterns that emerge during training, we visualize the hidden states of a randomly selected intermediate entity that appears in multiple two-hop queries. For this entity, we extract $\mathbf{h}^{5}_{\mathtt{r}_1}$ across relevant queries instances (where the intermediate entity is either ID-derived or OOD-derived) and compute pairwise cosine distances (defined as $1 -$ cosine similarity) between them. We then project these high-dimensional vectors into two dimensions using *Multidimensional Scaling (MDS)*

As shown in Figure 5, hidden states for a common intermediate entity form distinct geometric patterns across the phases. In Phase I, both ID-derived and OOD-derived representations are scattered; in Phase II, ID-derived representations form tight cosine-space clusters, marking the transition to

[2] Notably, the origin of an intermediate entity depends solely on the first hop and is independent of the second-hop configuration.



(a) Phase I

(b) Phase II

(c) Phase III

Figure 5: Cosine-space projection of a random intermediate entity across three training phases.

in-distribution reasoning; In Phase III, OOD-derived representations also begin to align with the ID-based cluster, signaling cross-distribution generalization. This suggests that latent variables are reused not by explicit decoding, but through the emergence of a consistent geometric structure.

Quantifying Representational Convergence. Encouraged by this observation, we next quantify the consistency of entity-level representations across the full dataset. We define two metrics: (1) **ID Cohesion Score:** the average cosine similarity between ID-derived representations and their centroid, reflecting in-distribution consistency. (2) **OOD Alignment Score:** the average cosine similarity between OOD-derived representations and the same ID centroid, reflecting how well the model unifies cross-distribution representations. These scores are computed on a per-entity basis and averaged across all intermediate entities.

Tracking these scores, we find that the **ID Cohesion Score** rises steadily, aligning with Test-II generalization, while the **OOD Alignment Score** starts increasing later, following the rise in Test-OI performance (Figure 2). This suggests that successful implicit reasoning relies on representational consistency across diverse contexts: only when hidden states for the same entity align closely in cosine space can they be consistently reused for multi-hop inference. Thus, **cosine-space clustering** emerges as the model's internal mechanism for semantic abstraction and generalization.

4 Closing the Loop: Explaining Behavioral Phenomena

Drawing on the mechanistic evidence in Section 3.3, we revisit our empirical observations in Section 2 and explain how they emerges from the internal dynamics of representation formation.

4.1 Clustering of OOD-Derived Representations Driven by ID Supervision

In Phase III, where the model successfully performs OOD reasoning at the first hop, the clustering of OOD-derived representations in cosine space plays a crucial role. However, while the clustering of ID-derived representations is expected due to direct supervision from Train-II queries, the alignment of OOD-derived representations with ID clusters is less intuitive. OOD-derived entities are never explicitly supervised as intermediate steps in multi-hop queries, making their eventual clustering with ID entities surprising.

We hypothesize that **frequent exposure to atomic triples** is the driving factor behind the observed alignment. This exposure leads the model to assimilate the OOD representations into the existing ID clusters, thereby stabilizing them. To validate this hypothesis, we designed an ablation study where we varied the ID/OOD ratio across three configurations: 0.8/0.2, 0.5/0.5, and 0.3/0.7. Only the 0.8/0.2 ratio demonstrates successful generalization into Phase III, effectively answering Test-OI queries, while the others failed to achieve this stage. Detailed results are provided in Appendix H.

4.2 ID Triple Queries Accelerate Generalization by Constraining the Representation Space

Knowing that generalization relies on the clustering of intermediate representations, we link the acceleration effect (Section 2.3) brought by ID triple to the property of autoregressive Transformers:

both ID triple queries (e_1, r_1) and two-hop queries (e_1, r_1, r_2) produce **the same hidden state at the r_1 token position** due to causal masking, where the model only attends to tokens preceding r_1.

Since our mechanistic analyses anchor at r_1 position ($\mathbf{h}^5_{\mathbf{r}_1}$), the hidden states we study are indistinguishable across both query formats. This means ID triple supervision directly shapes the same representations used in multi-hop reasoning. While the ID triple task optimizes for explicit decoding—mapping (e_1, r_1) to e_2—it doesn't guarantee functional reasoning (Section 3.2). However, it plays a crucial role by **constraining** the r_1 hidden state to lie within a subspace that supports entity decoding, thereby limiting the model's search space during generalization to a smaller region.

To validate this mechanism, we construct an ablation configuration that *removes* a subset of ID triple (*e.g.*, (e_A, r_1, e_B)) from training, while still including their corresponding two-hop compositions (*e.g.*, (e_A, r_1, r_2)) in Train-II. We then test whether the model can recover the held-out ID triples at test time (*e.g.*, given input (e_A, r_1), predict e_B). The results align with expectations: the model is able to correctly predict e_B (see Appendix I for details). This outcome demonstrates that the r_1 hidden state associated with (e_A, r_1) lies in the same region as other Train-II queries sharing latent (e_B)—a region already shaped by remaining ID triples involving e_B (*e.g.*, (e_X, r_7, e_B)):

$$\underbrace{(e_A, r_1)}_{\text{held-out triple query}} \xrightarrow{\text{share } r_1 \text{ position with}} \underbrace{\left\{\begin{matrix}(e_A, r_1, (e_B), r_2)\\ (e_X, r_7, (e_B), r_3)\end{matrix}\right\}}_{\text{Train-II with latent } (e_B)} \xrightarrow{\text{constrained by}} \underbrace{(e_X, r_7)}_{\text{retained ID supervision}} \xrightarrow{\text{decodable}} e_B$$

This supports the claim that ID triple supervision constrains the r_1 hidden state to a decodable region, facilitating clustering. Consequently, as the model enters Phase II and learns two-hop reasoning, it refines representations within an already structured subspace, speeding up representational convergence and behavioral generalization compared to a without ID configuration where clustering must be learned from scratch.

4.3 Why the First Hop, and Only the First Hop, Generalizes to OOD?

The intended supervision signal from Train-II queries is to teach the model a structured two-step reasoning process: token$(e_1) \xrightarrow{r_1^{\text{2hop}}}$ latent$(e_2) \xrightarrow{r_2^{\text{2hop}}}$ token(e_3), where both r_1^{2hop} and r_2^{2hop} are relations applied in a purely compositional context, independent of atomic triple learning (Section 2.3). In contrast, atomic triples expose the model to direct mappings of the form: token$(e_1) \xrightarrow{r_1^{\text{atomic}}}$ token(e_2), which train a shallow predictive behavior over observed fact pairs.

It is therefore surprising that models can correctly answer Test-OI queries, suggesting that the mapping r_1^{atomic} somehow transfers into r_1^{2hop}, allowing the model to reuse OOD-derived token(e_2) representations for implicit reasoning. However, this apparent generalization is in fact a side-effect of representational alignment induced by ID triples.

As established in Section 4.2, the hidden state at the r_1 token position is shared across atomic and 2-hop queries, encouraging the model to align latent(e_2) with token(e_2) for ID triples. Separately, as shown in Section 4.1, the model gradually pulls OOD-derived representations into the same cosine-space cluster as ID-derived representations. These two mechanisms together enables OOD-derived latent(e_2) to match the expected format of r_1^{2hop}:

$$\text{OOD-derived token}(e_2) \xrightarrow{\textit{pulled by}} \text{ID-derived token}(e_2) \xrightarrow{\textit{align with}} \text{latent}(e_2),$$

In this sense, the model doesn't truly generalize OOD r_1^{atomic} into r_1^{2hop}—it "cheats" by reusing shared representation scaffolds shaped by ID triples. The success on Test-OI queries is thus illusory: what appears to be a generalization is in fact a misalignment between model structure and supervision.

Viewed from this perspective, the failure of second-hop generalization is not an exception. Unlike the first hop, the model cannot rely on representational anchoring from shared prefixes and must learn behavior through direct query-level supervision. In contrast, first-hop OOD generalization is an exception, made possible by incidental alignment from overlapping input contexts, hence this does not extend to deeper reasoning. A similar pattern holds in 3-hop reasoning: generalization is only observed when the reasoning path beyond the first hop remains within ID (Test-III and Test-OII), highlighting the need for explicit supervision in later steps (Appendix C).

We hypothesize that without the representational anchoring effect induced by ID supervision, OOD triples fail to form functionally useful intermediate representations. To test this, we construct a configuration with only OOD triples and Train-II queries, removing ID triples. In this setup, the model fails on Test-OI generalization, confirming that in the absence of ID-based anchoring, OOD triples alone cannot support implicit multi-hop reasoning. See Appendix J for details.

5 Discussion

Our study, conducted in a controlled symbolic dataset environment, reveals key insights into the mechanisms of implicit reasoning in transformers, highlighting specific patterns and behaviors that clarify how multi-hop implicit reasoning emerges. These findings may provide valuable answers to existing questions about the implicit reasoning capabilities of LLMs. For instance, our observation regarding the requirement for query-level match offers a potential explanation for why knowledge learned from single-hop tasks does not easily transfer to multi-hop reasoning in LLMs [3, 39, 31]. However, it is important to note that LLMs operate with far richer and more complex knowledge bases, and their internal knowledge interaction mechanisms likely differ from those in our controlled environment. Therefore, while our findings offer useful insights, they should be regarded as preliminary guidance rather than a complete explanation of the reasoning dynamics in LLMs.

Acknowledgement

This work is supported by Beijing Natural Science Foundation (L243006), National Natural Science Foundation of China (62476150), and the Tsinghua University (Department of Computer Science and Technology)-Siemens Ltd., China Joint Research Center for Industrial Intelligence and Internet of Things (JCIIOT).

References

[1] Anthropic. Claude 3.7 sonnet, 2025. URL https://www.anthropic.com/claude/sonnet. Accessed: 2025-05-15.

[2] Anthropic. Reasoning models don't always say what they think, 2025. URL https://www.anthropic.com/research/reasoning-models-dont-say-think. Accessed: 2025-05-15.

[3] Mikita Balesni, Tomek Korbak, and Owain Evans. The two-hop curse: Llms trained on a-> b, b-> c fail to learn a–> c. *arXiv preprint arXiv:2411.16353*, 2024.

[4] Aryasomayajula Ram Bharadwaj. Understanding hidden computations in chain-of-thought reasoning. *arXiv preprint arXiv:2412.04537*, 2024.

[5] Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. Hopping too late: Exploring the limitations of large language models on multi-hop queries. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14113–14130, 2024.

[6] Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025.

[7] Google DeepMind. Gemini 2.5 pro: Our most intelligent ai model, 2025. URL https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/. Accessed: 2025-05-15.

[8] Valeriya Goloviznina and Evgeny Kotelnikov. I've got the "answer"! interpretation of llms hidden states in question answering. In *International Conference on Applications of Natural Language to Information Systems*, pages 106–120. Springer, 2024.

[9] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[10] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

[11] Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

[12] Tianjie Ju, Yijin Chen, Xinwei Yuan, Zhuosheng Zhang, Wei Du, Yubin Zheng, and Gongshen Liu. Investigating multi-hop factual shortcuts in knowledge editing of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8987–9001, 2024.

[13] William Jurayj, William Rudman, and Carsten Eickhoff. Garden path traversal in gpt-2. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 305–313, 2022.

[14] Mikail Khona, Maya Okawa, Jan Hula, Rahul Ramesh, Kento Nishi, Robert P. Dick, Ekdeep Singh Lubana, and Hidenori Tanaka. Towards an understanding of stepwise inference in transformers: A synthetic graph navigation model. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=8VEGkphQaK.

[15] KimiTeam, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[16] Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. Understanding and patching compositional reasoning in llms. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9668–9688, 2024.

[17] Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*, 2025.

[18] nostalgebraist. Interpreting gpt: The logit lens, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. Accessed: 2025-05-01.

[19] OpenAI. Introducing openai o3 and o4-mini, 2025. URL https://openai.com/index/introducing-o3-and-o4-mini/. Accessed: 2025-05-15.

[20] Jacob Pfau, William Merrill, and Samuel R Bowman. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*, 2024.

[21] Ella Rabinovich, Samuel Ackerman, Orna Raz, Eitan Farchi, and Ateret Anaby-Tavor. Predicting question-answering performance of large language models through semantic consistency. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, volume 10, page 138, 2023.

[22] Mansi Sakarvadia. Towards interpreting language models: A case study in multi-hop reasoning. *arXiv preprint arXiv:2411.05037*, 2024.

[23] Yuval Shalev, Amir Feder, and Ariel Goldstein. Distributional reasoning in llms: Parallel reasoning processes in multi-hop reasoning. *arXiv preprint arXiv:2406.13858*, 2024.

[24] Boshi Wang and Huan Sun. Is the reversal curse a binding problem? uncovering limitations of transformers from a basic generalization failure. *arXiv preprint arXiv:2504.01928*, 2025.

[25] Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokking of implicit reasoning in transformers: A mechanistic journey to the edge of generalization. *Advances in Neural Information Processing Systems*, 37:95238–95265, 2024.

[26] Siwei Wang, Yifei Shen, Shi Feng, Haoran Sun, Shang-Hua Teng, and Wei Chen. ALPINE: unveiling the planning capability of autoregressive learning in language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/d848cb2c84f0bba7f1f73cf232734c40-Abstract-Conference.html.

[27] Xinyi Wang, Alfonso Amayuelas, Kexun Zhang, Liangming Pan, Wenhu Chen, and William Yang Wang. Understanding reasoning ability of language models from the perspective of reasoning paths aggregation. In *International Conference on Machine Learning*, pages 50026–50042. PMLR, 2024.

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

[29] xAI. Grok 3 beta — the age of reasoning agents, 2025. URL https://x.ai/blog/grok-3. Accessed: 2025-05-15.

[30] Amy Xin, Jinxin Liu, Zijun Yao, Zhicheng Lee, Shulin Cao, Lei Hou, and Juanzi Li. Atomr: Atomic operator-empowered large language models for heterogeneous knowledge reasoning. *arXiv preprint arXiv:2411.16495*, 2024.

[31] Ruoxi Xu, Yunjie Ji, Boxi Cao, Yaojie Lu, Hongyu Lin, Xianpei Han, Ben He, Yingfei Sun, Xiangang Li, and Le Sun. Memorizing is not enough: Deep knowledge injection through reasoning. *arXiv preprint arXiv:2504.00472*, 2025.

[32] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

[33] Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10210–10229, 2024.

[34] Sohee Yang, Nora Kassner, Elena Gribovskaya, Sebastian Riedel, and Mor Geva. Do large language models perform latent multi-hop reasoning without exploiting shortcuts? *arXiv preprint arXiv:2411.16679*, 2024.

[35] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

[36] Yunzhi Yao, Jizhan Fang, Jia-Chen Gu, Ningyu Zhang, Shumin Deng, Huajun Chen, and Nanyun Peng. Cake: Circuit-aware editing enables generalizable knowledge learners. *arXiv preprint arXiv:2503.16356*, 2025.

[37] Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*, 2024.

[38] Mengqi Zhang, Bowen Fang, Qiang Liu, Pengjie Ren, Shu Wu, Zhumin Chen, and Liang Wang. Enhancing multi-hop reasoning through knowledge erasure in large language model editing. *arXiv preprint arXiv:2408.12456*, 2024.

[39] Xiao Zhang, Miao Li, and Ji Wu. Co-occurrence is not factual association in language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[40] Zhuoran Zhang, Yongxiang Li, Zijian Kan, Keyuan Cheng, Lijie Hu, and Di Wang. Locate-then-edit for multi-hop factual recall under knowledge editing. *arXiv preprint arXiv:2410.06331*, 2024.

[41] Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702, 2023.

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

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction introduces the central focus of the study, which is understanding how implicit multi-hop reasoning emerges in transformers during training.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We acknowledges in the Discussion section that the results in this paper should be viewed as preliminary and may not fully generalize to more complex, real-world large language model settings.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper primarily discusses empirical observations and provides mechanistic analyses through diagnostic tools (like cross-query semantic patching and cosine-based representational analysis) rather than formal theorems or mathematical proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper outlines a detailed experimental setup, including a symbolic training environment and specific configurations for data generation. Information such as the exact model parameters, hyperparameters, or training steps necessary is provided in the Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The data and code will be attached in the supplementary materials.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides sufficient experimental details to understand the results. Details of data construction are mentioned in Appendix B; hyperparameters like the learning rate and optimizer type are explicitly mentioned in Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: There are too many configurations in our experiments, error bars are thus not reported because it would be too computationally expensive.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Information on the computer resources is provided in Appendix G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We did reviewed the NeurIPS Code of Ethics and the research conducted in this paper adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

Justification: While the paper mentions potential positive implications for LLM interpretability (improved transparency of reasoning processes), it does not explicitly discuss negative societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We utilizes a symbolic dataset that does not contain any real sensitive information, and therefore poses no risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper uses open-source models, and the creators or original owners of these models have been properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will provide detailed documentation for all new assets, including datasets, models, and code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper doesn't involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper doesn't involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Related Work

Mechanistic Exploration of Implicit Reasoning. The initial focus on the mechanisms of implicit reasoning arose from the discovery that traditional single-hop knowledge editing methods are ineffective in the context of multi-hop implicit reasoning [41, 38, 12]. This phenomenon since gained more significant attention, prompting researchers to investigate the mechanisms underlying implicit reasoning. Some works suggest that the failure of implicit reasoning is due to the intermediate entities not being properly processed [22, 33]; Other works suggest that the reason lies in the intermediate entities being processed, but not being passed to the correct position [16, 36, 5]. The parallel exploration paths within the model are also a research direction [23, 27]. Recent works have found that the different knowledge composed as different hops used in implicit reasoning is stored in different layers of the model [40, 36]. Additionally, some studies propose that models rely on shortcuts to successfully complete implicit reasoning [34].

Conditions for Learning Implicit Reasoning. Some works have found that the conditions for models to learn implicit reasoning are quite demanding. Some have discovered through custom symbolic datasets that models generalize implicit reasoning abilities only after grokking, and that they require highly compositional data [25]. Other works have found that simply providing optical knowledge is insufficient for enabling models to perform implicit reasoning, it requires training with corresponding multi-hop reasoning data [31, 39, 42]. Moreover, it has been observed that when the two pieces of foundational knowledge used for 2-hop implicit reasoning appear in different paragraphs of the training corpus, the model struggles to combine them for implicit reasoning. However, when these pieces appear within the same paragraph, the model's accuracy in answering the corresponding 2-hop queries significantly improves [3].

Probing Intermediate Entities. As for the probing tools, most of the work uses decoding-based methods to probe intermediate entities. Many works assume that if a model encodes an intermediate entity, it should be able to explicitly decode it. They use the logit lens [18]. as evidence of the presence of intermediate entities in implicit reasoning [22, 33, 16]. A few studies have also trained a linear transformation layer to decode intermediate entities [23].

Latent CoT. Although current Chain-of-Thought (CoT) based reasoning models have shown impressive performance, some perspectives argue that the thought chains do not truly reflect the model's reasoning process [17, 2]. Some works have begun to explore Chain of Thought (CoT) methods that do not verbalize intermediate steps. One approach is to replace the Chain of Thought (CoT) with dots, adjusting the number of dots based on the original length of the thought chain [20, 4]. Another approach to avoid verbalizing is to bypass decoding the tokens in the CoT process and directly use the last hidden state from the previous step as the input vector for the next step [10]. These approaches have achieved results comparable to traditional CoT on specific tasks.

B Dataset Details

Our data construction pipeline is adapted from the open-source code released by Wang et al. [25], with modifications to support fine-grained query-level control.

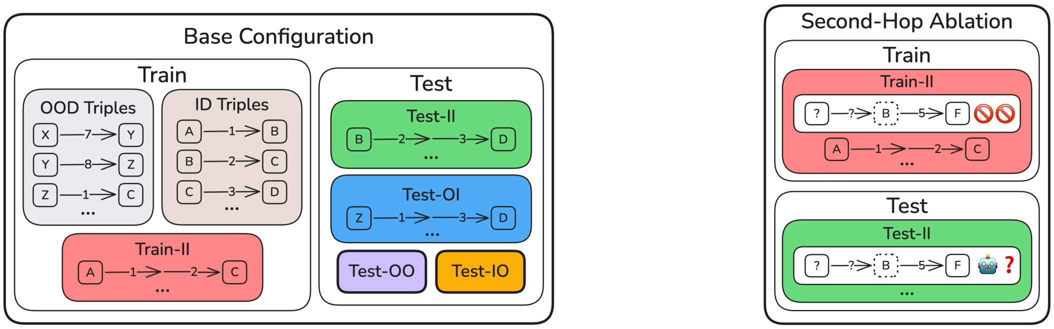
Entity and Relation Vocabulary. In all configurations, we construct a symbolic environment consisting of 2000 entities and 200 relations, each assigned a unique token with no inherent semantics. Since the model is trained from scratch, it has no prior knowledge about the symbols, and learning depends entirely on compositional supervision.

Atomic Triple Generation. We generate 40000 atomic triples of the form $(e_1, r_1) \to e_2$, where each entity e_1 is randomly assigned 20 outgoing relations, and for each such relation r_1, the tail entity e_2 is sampled uniformly from all entities. These triples are then randomly partitioned at the triple level into ID and OOD subsets, with a default OOD ratio of 5%, while entities and relations remain globally shared across all subsets.

2-Hop Query Construction. Two-hop queries are compositional chains of the form $(e_1, r_1, r_2) \to e_3$, where the model must implicitly reason through an intermediate entity e_2. These queries are constructed by pairing atomic triples: if both (e_1, r_1, e_2) and (e_2, r_2, e_3) exist, the corresponding query is eligible. Training queries (*Train-II*) are sampled from the set of ID–ID chains. The remaining queries are categorized as *Test-II*, *Test-IO*, *Test-OI*, or *Test-OO* depending on whether the first and second hops are ID or OOD.

We construct the training set with a 7.2:1 ratio of Train-II queries to ID atomic triples to ensure compositional supervision dominates, and sample a fixed test set of 3,000 examples for each type. Table 2 summarizes the key dataset statistics.

Illustrations of Data Construction Configurations. To aid in understanding the dataset construction process, we provide two representative configurations as illustrations (Figure 6).



(a) Base configuration

(b) Second-hop ablation

Figure 6: Two representative data construction configurations. **(a) Base configuration:** Contains all atomic triples and Train-II queries, with evaluation covering all test query types. **(b) Second-hop ablation configuration:** A targeted setup where a subset of atomic triples (e.g., (e_B, r_5, e_F)) are excluded from appearing as second hops in any training query, while corresponding second-hop queries remain present in the test set.

Table 2: Dataset Statistics

Data Type	**Split**	**Count**
Vocabulary	Entities Relations	2000 200
Atomic Triples	In-Distribution (ID) Out-of-Distribution (OOD)	38000 2000
2-hop Queries	Train-II (ID → ID) Test-II (ID → ID) Test-IO (ID → OOD) Test-OI (OOD → ID) Test-OO (OOD → OOD)	273600 3,000 3,000 3,000 3,000

Design Choices and Assumptions.

In designing our dataset, we made two key parameter choices to support our analysis focus and the intended analogy to real-world LLM behavior:

First, we fix a relatively high ratio of **Train-II to ID Triples** (7.2:1), ensuring that compositional supervision dominates over direct fact learning. While prior work [25] has shown that a high Train-II / ID ratio is necessary for generalization on Test-II queries, we do not treat this ratio as a variable of interest. Instead, we maintain it at a sufficient level to ensure the emergence of in-distribution reasoning, and shift our focus to more challenging phenomena such as cross-distribution generalization and the underlying mechanisms that support generalizations.

Second, we choose a high ratio of **ID to OOD atomic triples** (95% ID), which is critical for the representational alignment effects discussed in Section 3 and further validated in Appendix H. In particular, Phase III generalization relies on OOD-derived intermediate representations aligning with the subspace formed by ID-derived ones—an effect that only arises when ID triples are sufficiently dominant.

We acknowledge that these high-ratio settings may appear biased. However, they reflect plausible properties of real-world pretraining corpora: (1) compositional chains (analogous to our Train-II queries) are likely more common than isolated fact triples; and (2) most knowledge entities are learned in richly connected contexts (analogous to our ID triples), while only a minority are sparsely anchored or appear in isolation (analogous to our OOD triples). From this perspective, our symbolic setup not only enables controlled analysis, but also approximates realistic data imbalance patterns observed in LLM pretraining. Accordingly, we treat both ratios as default conditions rather than ablation variables.

C Extension to 3-Hop Reasoning

We extend our symbolic framework to support 3-hop queries and verify that the behavioral and mechanistic patterns observed in the 2-hop setting also emerge in deeper compositional regimes.

C.1 3-hop Dataset Construction

The overall construction methodology mirrors that of 2-hop queries, as described in Appendix B, with an additional relational step. These queries take the form $(e_1, r_1, r_2, r_3) \to e_4$, where the model must implicitly traverse three relational steps through two intermediate entities.

However, the increased compositional depth introduces new data balancing challenges. In particular, the combinatorial nature of 3-hop chaining (i.e., atomic composition scales cubically) leads to test set sparsity(especially Test-OOO) if the OOD triple proportion is too small. To mitigate this, we reduce the vocabulary size to 1000 entities and 100 relations, and increase the OOD ratio to 20%, while still maintaining a majority of ID supervision discussed in Appendix B. These adjustments ensure sufficient coverage across all evaluation regimes, including out-of-distribution settings. Table 3 summarizes the key statistics of 3-hop dataset.

Table 3: 3-hop Dataset Statistics

Data Type	Split	Count
Vocabulary	Entities	1000
	Relations	100
Atomic Triples	In-Distribution (ID)	8000
	Out-of-Distribution (OOD)	2000
3-hop Queries	Train-III (ID → ID → ID)	120000
	Test-III (ID → ID → ID)	1,000
	Test-IIO (ID → ID → OOD)	1,000
	Test-IOI (ID → OOD → ID)	1,000
	Test-IOO (ID → OOD → OOD)	1,000
	Test-OII (OOD → ID → ID)	1,000
	Test-OIO (OOD → ID → OOD)	1,000
	Test-OOI (OOD → OOD → ID)	1,000
	Test-OOO (OOD → OOD → OOD)	1,000

C.2 Training Dynamics and Generalization Patterns

We evaluate model behavior on the 3-hop dataset using the same model, training regime, and diagnostic tools as in the 2-hop setting. Figure 7 summarizes the accuracy trajectories across all test query types and clustering metrics of intermediate representations.

Compared to 2-hop queries, 3-hop reasoning introduces an additional intermediate entity, resulting in two latent steps: $e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} e_3 \xrightarrow{r_3} e_4$. To analyze how the model internally represents these latent entities, we extend our diagnostic metrics accordingly.

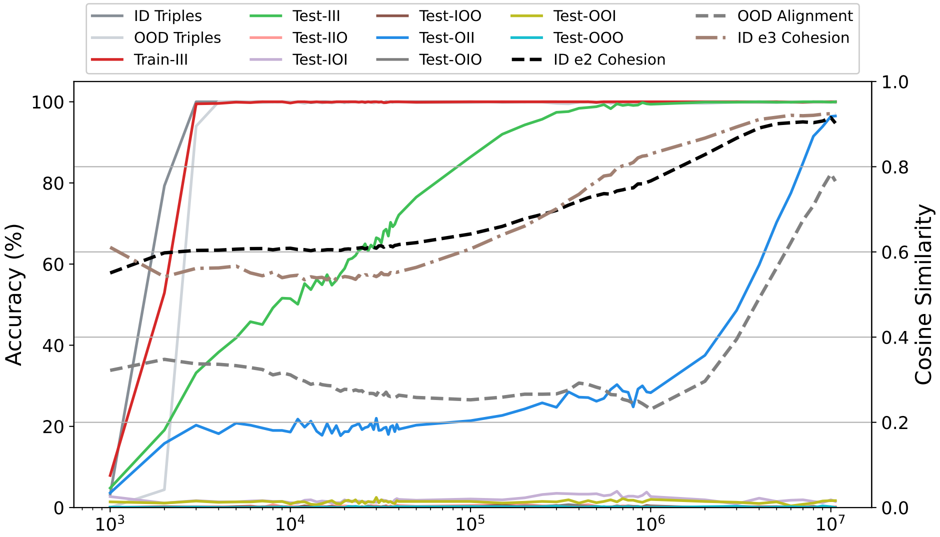


Figure 7: Training dynamics in the 3-hop setting. Accuracy and representation metrics exhibit consistent generalization patterns with the 2-hop case.

Specifically, we extract: (1) $\mathbf{h}^5_{\mathtt{r}_1}$, the hidden state at the r_1 token in layer 5, to represent the internal encoding of e_2 (same as in the 2-hop setting); (2) $\mathbf{h}^6_{\mathtt{r}_2}$, the hidden state at the r_2 token in layer 6, to represent the encoding of e_3.

Based on these, we compute the following representational clustering metrics: (1) ID e_2 Cohesion: average cosine similarity among representations of the same ID-derived e_2 entity; (2) ID e_3 Cohesion: the analogous metric computed over e_3 representations. (3) OOD e_2 Alignment: cosine similarity between OOD-derived e_2 representations and the corresponding ID-derived centroid. We do not compute alignment for e_3 because only the first hop (e_2) can successfully generalize from OOD inputs.

We observe two key generalization patterns consistent with the 2-hop results:

(1) In-distribution generalization emerges reliably: The model successfully generalizes to Test-III queries (ID → ID → ID), with accuracy rising steadily during training.

(2) Out-of-distribution generalization remains constrained to the first hop: Among all OOD-containing query types, only Test-OII (OOD → ID → ID) shows significant improvement. Other configurations where OOD triples appear in the second or third hop (*e.g.*, Test-IIO, IOI, OOO) fail to generalize. This reinforces the bottleneck observed in the 2-hop case: query-level exposure is necessary for downstream relational generalization.

Furthermore, OOD e_2 Alignment closely tracks Test-OII performance, while ID e_2 and e_3 Cohesion metrics rise in concert with Test-III accuracy. Together, these results highlight that representational clustering at multiple relational depths serves as the internal mechanism enabling successful multi-hop reasoning.

While we do not repeat all behavioral ablations from the 2-hop setting (*e.g.*, acceleration from ID triple exposure, second-hop query-level matching), we note that these phenomena are well explained by the clustering dynamics reported above. In particular, similar representational bottlenecks and alignment requirements arise at each reasoning step, such that second- and third-hop generalization depend on the same clustering dynamics as the first hop. As a result, the cohesion and alignment metrics we report suffice to capture the core generalization behaviors in the 3-hop case.

D Additional Validation of Query-Level Requirements in Second-Hop Generalization

D.1 Verifying Second-Hop Failure under Full Supervision

To ensure the observed failure of second-hop generalization is not an artifact of minimal training configurations, we repeat the second-hop ablation experiment under the full base setting described in Section 2.1. Specifically, we exclude a subset of atomic triples (*e.g.*, (e_B, r_5, e_F)) from appearing in any second-hop positions during training, while allowing them in atomic queries or first-hop usage (Figure 8).

Despite the model being exposed to these triples in other structural roles, it fails to generalize to Test-II queries that require them as second-hop compositions. Importantly, performance on all other query types remains unaffected. This confirms that **second-hop generalization requires direct query-level supervision**: exposure to a fact in other contexts is insufficient for enabling its role in compositional reasoning.

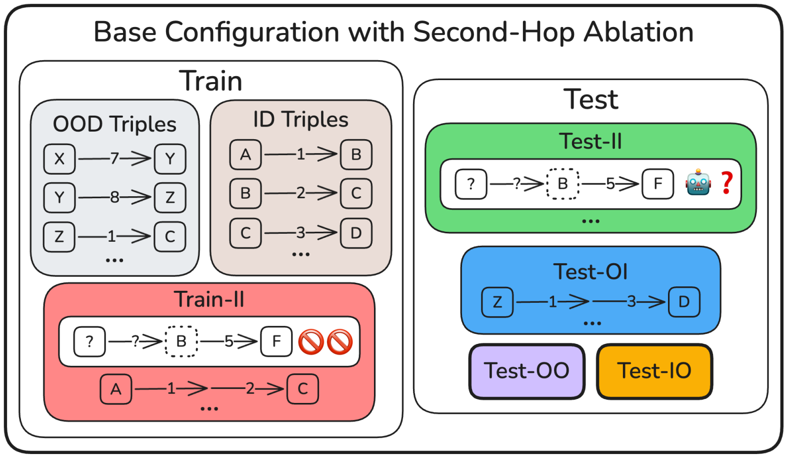


Figure 8: Illustration of the second-hop ablation under the full base configuration. A subset of atomic triples (*e.g.*, (e_B, r_5, e_F)) is excluded from appearing as second hops during training. The rest of the data remains unchanged, allowing controlled evaluation of second-hop generalization.

D.2 Accuracy vs. Second-Hop Exposure Frequency

To further investigate the role of query-level exposure in second-hop generalization, we examine whether second-hop triples that appear more frequently in training queries are learned earlier, using the full base configuration.

We select a fixed training checkpoint in the early part of Phase II—specifically when Test-OI accuracy reaches approximately 50%—and we group second-hop triples by their frequency of occurrence in Train-II queries. For each frequency group k, we compute the average accuracy over all Test-II queries whose second-hop triple appeared exactly k times in the training set.

The results reveal a clear trend: higher second-hop exposure frequency during training leads to greater accuracy on corresponding test queries at this intermediate phase (see Figure 9), reinforcing the causal role of second-hop participation in enabling generalization.

E Probing Preference for Explicit Decoding

To test whether the model prefers to encode intermediate entities in an explicitly decodable form, we construct a new configuration that reveals the model's decoding preference behaviorally.

This configuration (Figure 10a) removes all ID triples from the training data, but retains (i) all Train-II 2-hop queries, which require reasoning through ID-derived intermediate entities, and (ii) all OOD

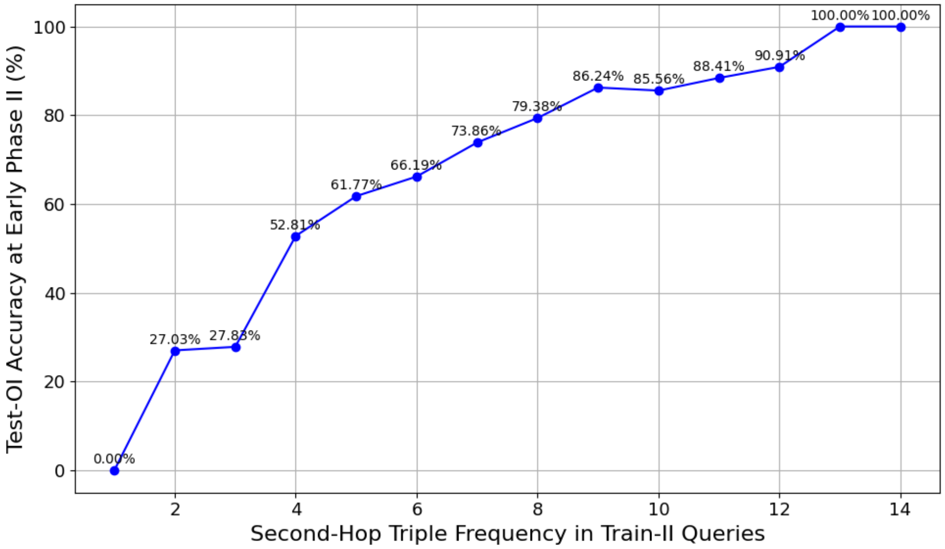
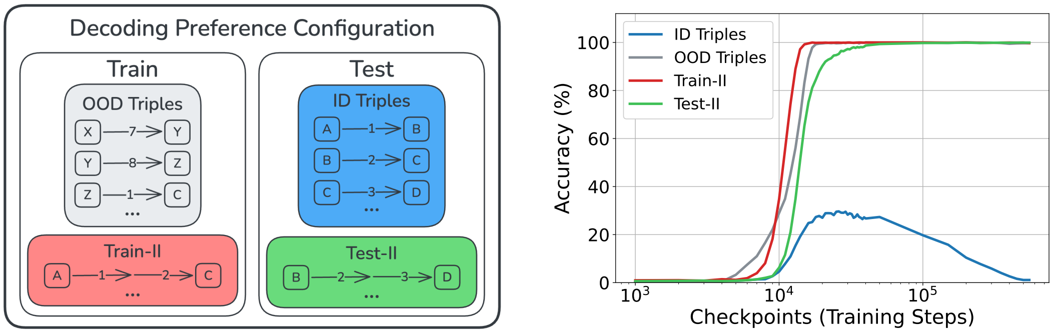


Figure 9: Average Test-OI accuracy at early Phase II (approx. 50%) grouped by second-hop triple frequency in Train-II queries.

triples, which still provide supervision for decoding at the r_1 position. Crucially, tail entities in both ID and OOD triples **share the same tail entity vocabulary**, encouraging the model to apply similar decoding strategies across domains.



(a) Decoding Preference Configuration

(b) Accuracy dynamics under this configuration.

Figure 10: **Decoding Preference Experiment.** (a) Experimental setup where ID triples are excluded from training and only used for testing., while the model is trained on OOD triples which share the same tail entity vocabulary as the ID triples, and all Train-II queries. (b) Accuracy over training steps shows that the model initially recovers held-out ID triples, suggesting an attempt at explicit decoding, but later abandons this strategy.

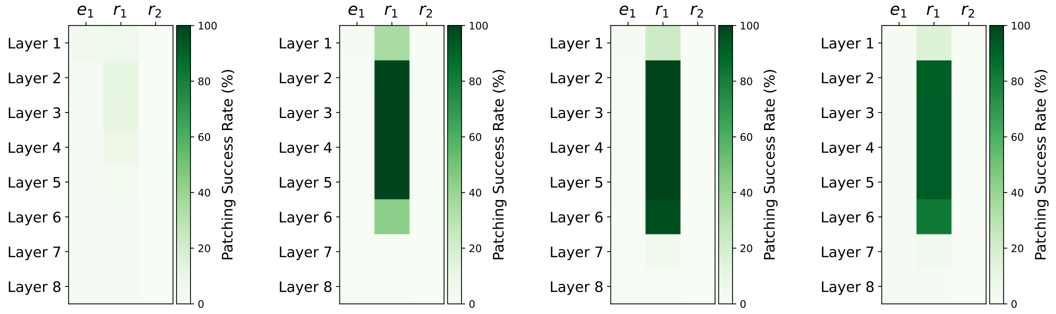
This setup leverages the shared hidden state structure between atomic and two-hop queries discussed in Section 4.2, implying that if the model encodes the intermediate entity in a decodable form during 2-hop reasoning, it should be able to recover ID triples (*e.g.*, given a ID query (e_A, r_1), predict e_B) even if these triples were never seen during training.

Figure 10b shows the result. Initially, the model answers some ID triples correctly, suggesting an early attempt for explicit decoding. However, this accuracy soon declines, while the performance on Test-II continues to rise. This indicates that the model abandons decodable representations in favor of internal ones that support reasoning but cannot be directly decoded. Explicit decoding, while initially attempted, is not sustained—suggesting it is not the model's preferred solution when alternatives are available.

F Additional Results for Cross-Query Causal Patching

We provide the individual patching success rates for each of the three settings: Phase II (ID-derived), Phase III (ID-derived), and Phase III (OOD-derived). These results complement the averaged trend shown in the main text (Figure 4) and confirm the consistency of intermediate entity localization across training stages and generalization regimes.

To provide a full developmental picture, we also include Phase I patching results in Figure 11a. These results exhibit near-zero success across all positions and layers, consistent with the absence of any generalization behavior at this stage. This reinforces our claim that meaningful intermediate representations emerge only after compositional reasoning capabilities begin to develop.



(a) Phase I: ID-derived (b) Phase II: ID-derived (c) Phase III: ID-derived (d) Phase III: OOD-derived

Figure 11: Patching success rate across layers and token positions for different phases and data sources.

G Training Details and Bigger Models

G.1 Training Details

Our training procedure largely follows the public implementation provided by Wang et al. [25], with a few modifications to accommodate our specific experimental setting.

The model is a decoder-only Transformer, identical in architecture to GPT-2, with 8 layers, 768 hidden dimensions, and 12 attention heads. Optimization is performed using AdamW with a learning rate of 1×10^{-4}, 2000 warm-up steps, weight decay of 0.1, and a batch size of 1024. All models are trained significantly beyond convergence to allow observation of late-stage generalization behavior (Section 2.2).

Training is conducted on NVIDIA RTX 3090 GPUs, and the maximum training duration is extended to 3 weeks to ensure stable cross-distributions generalization. All experiments are implemented using the same PyTorch and Huggingface Transformers framework as in the original codebase.

G.2 Scaling Analysis: Dynamics and Alignment in Larger Models

We extend our main experimental setup by training a larger model, Qwen2.5-1.5B, under the same base configuration. Our goal is to examine whether the developmental trajectory of multi-hop reasoning observed in smaller models persists at scale, and to further investigate the alignment between behavioral accuracy and representational metrics.

The training progresses successfully through Phase I (memorization) and Phase II (in-distribution generalization), and reaches Phase III (cross-distribution generalization). However, we observe increased instability during Phase III: although the model demonstrates the ability to generalize to Test-OI queries, the performance exhibits significant fluctuations. We report the Test-II and Test-OI accuracies, alongside the ID Cohesion and OOD Alignment metrics (Figure 12).

Interestingly, we find that the Test-II accuracy does not rise in lockstep with the ID Cohesion metric, in contrast to the strong correlation observed in smaller models (Figure 2). We interpret this

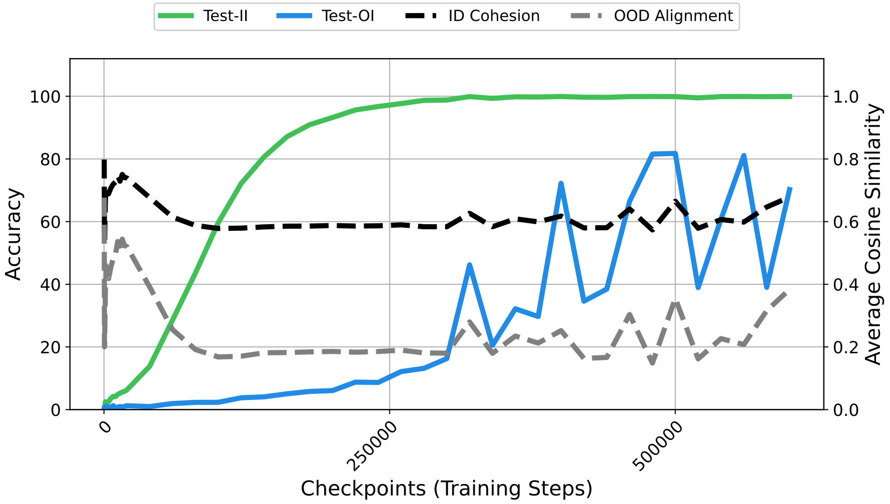


Figure 12: Test-II and Test-OI accuracy and ID Cohesion and OOD Alignment metrics over training steps in the Qwen2.5-1.5B model. Although the model reaches Phase III generalization, substantial variance is observed in both Test-OI accuracy and OOD Alignment, which nonetheless remain tightly coupled. In contrast, Test-II accuracy does not closely track the ID Cohesion metric, suggesting a representational bottleneck at the second relational step.

decoupling as evidence that ID Cohesion is a necessary but not sufficient condition for successful Test-II generalization. While a coherent latent space is required to support compositional reasoning, achieving high Test-II accuracy also depends on the model's ability to utilize these representations in **executing the second relational step**. In other words, beyond aligning representations, the model must also learn to map from an aligned intermediate state to the correct final output via the second-hop relation.

In smaller models, these two aspects—representation alignment and second-hop reasoning—tend to emerge together as part of a single learning phase, leading to tight coupling between ID Cohesion and Test-II performance. In contrast, larger models appear to decouple these processes: representational clustering may occur early, while second-hop reasoning capabilities require additional training to fully mature. As a result, the second relational step becomes the **dominant bottleneck** in Phase II generalization.

This interpretation is further supported by the close alignment between the OOD Alignment metric and Test-OI accuracy. Because second-hop reasoning over ID triples is already well established by Phase II, generalization on Test-OI becomes predominantly constrained by whether OOD-derived intermediate representations have successfully aligned with the ID-centric latent space. This tight correlation holds across model scales: in both the main 2-hop setup with smaller models and the 3-hop results in Appendix C (*e.g.*, alignment between Test-OII and OOD-derived clustering), the emergence of Phase III generalization closely tracks OOD Alignment. In our large-model experiment, although the Test-OI accuracy exhibits high variance, its fluctuations are closely mirrored by the OOD Alignment metric, reinforcing our hypothesis.

We leave the optimization of Phase III training strategies for larger models to future work. Our findings suggest that alignment-based representational diagnostics may serve as useful guides for tuning training schedules or data exposure in this regime, and we encourage future work to explore these directions further.

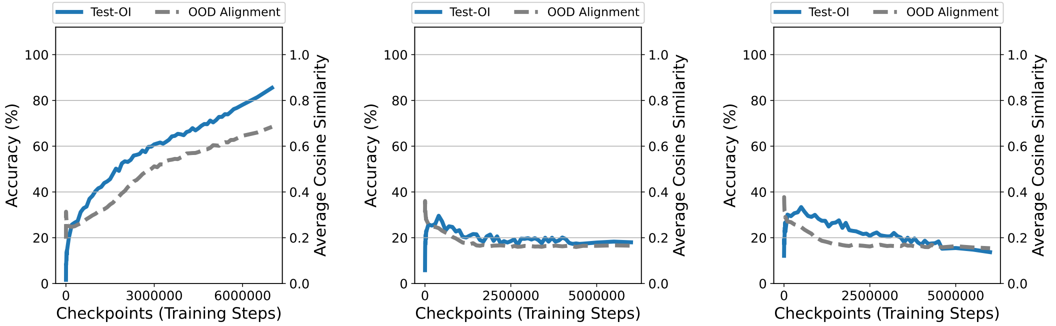
H Validation of Phase III Emergence via ID/OOD Ratio Ablation

To validate our hypothesis in Section 4.1 that the emergence of cross-distribution generalization (Phase III) depends on the dominance of in-distribution (ID) supervision, we conduct an ablation study by varying the ID/OOD ratio of atomic triples under the base configuration.

Experimental Setup. We begin with the full base configuration, which includes all atomic triples (both ID and OOD) and the complete set of Train-II queries. We fix the total number of atomic triples and vary the ID/OOD ratio while keeping all other components of training unchanged. Specifically, we test three settings: 80% ID / 20% OOD, 50% ID / 50% OOD, and 30% ID / 70% OOD. In all cases, the **Train-II / ID ratio** is held constant, as prior work Wang et al. [25] identifies this as a critical factor for Phase II generalization.

Results. All three ID/OOD configurations unsurprisingly reach Phase I and Phase II, to focus on the emergence of Phase III, we report the Test-OI accuracy and OOD Alignment Score, which capture the model's ability to reason across distributions and align OOD-derived intermediate representations with the ID-induced cluster structure.

As shown in Figure 13, in the 0.8/0.2 setting, both Test-OI accuracy and OOD Alignment Score increase together during training, indicating that the model successfully assimilates OOD-derived intermediate representations into the ID-induced subspace and is able to reuse them for cross-distribution reasoning. In contrast, in the 0.5/0.5 and 0.3/0.7 settings, both metrics remain consistently low, suggesting that the model fails to form aligned representations for OOD triples and consequently cannot generalize to Test-OI queries. This divergence across configurations highlights that strong ID supervision is essential for enabling Phase III generalization.



(a) ID/OOD = 0.8/0.2

(b) ID/OOD = 0.5/0.5

(c) ID/OOD = 0.3/0.7

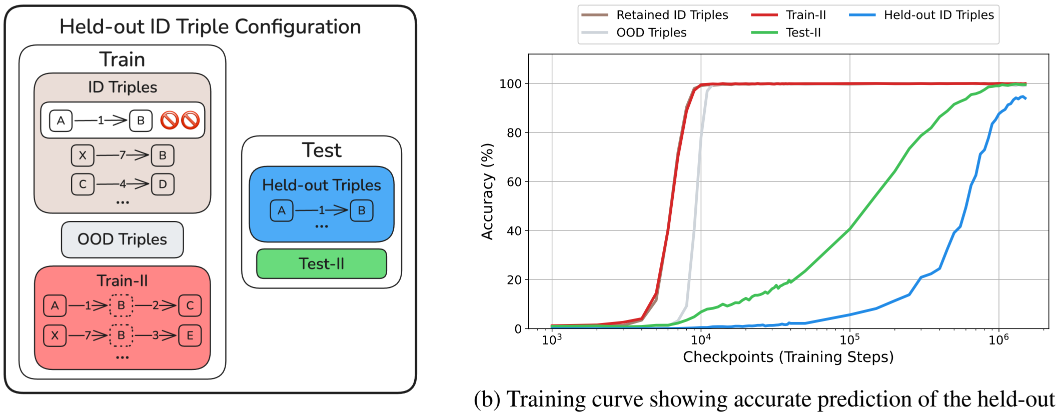
Figure 13: **Test-OI accuracy and OOD Alignment Score** under different ID/OOD splits. All configurations successfully reach Phase I and Phase II. Only the 0.8/0.2 setting supports Phase III generalization, as indicated by joint increases in Test-OI accuracy and OOD alignment. Lower-ID settings fail to align OOD-derived bridge entities, preventing cross-distribution reasoning.

I Evidence for Representation Clustering from the Decodable Subspace

To validate the representational mechanism discussed in Section 4.2 that ID triple supervision constrains the r_1 representation to lie in a decodable subspace, we design an ablation experiment to test whether held-out ID triples can be recovered solely through shared compositional contexts in Train-II queries.

Specially, We randomly select a subset of ID triples (e.g., $(e_A, r_1) \to e_B$) to exclude from the atomic triple training set. These held-out triples are removed from all atomic query contexts but remained involved in Train-II queries (*e.g.*, $(e_A, r_1, r_2) \to e_C$, where e_B serves as the intermediate entity). Crucially, the model retains exposure to other atomic triples that sharing the same tail entity, such as $(e_X, r_7) \to e_B$, which appear in both atomic and corresponding compositional queries (*e.g.*, $(e_X, r_7, r_3) \to e_Y$), Figure 14a illustrates the configuration. This configuration enables us to validate whether ID atomic triples supervision constrains the r_1 hidden state to a decodable region by testing whether these held-out triples could be recovered.

As illustrated in the Figure 14b, The model successfully recovers held-out triples despite their absence from atomic training. Crucially, this recovery capability emerges concurrently with Test-II generalization, confirming that the model leverages the same intermediate representation subspace for both atomic and compositional reasoning. The results indicate that ID triple supervision accelerates generalization not by providing explicit factual memorization, but by structurally constraining the model's representational space to align atomic and compositional reasoning pathways.



(a) Illustration of the data construction.

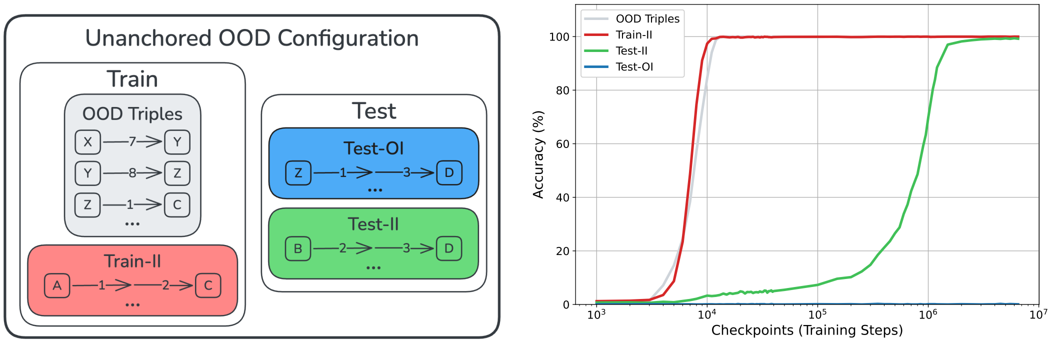
(b) Training curve showing accurate prediction of the held-out triples.

Figure 14: **Verification of ID triple constraint effect.** The model successfully recovers held-out ID triples by leveraging representational constraints from multi-hop supervision and structurally related retained triples, supporting the claim that ID triples accelerate generalization by shaping a decodable representational subspace.

J Removing ID Triples Breaks First-Hop OOD Generalization

To test whether ID supervision is necessary for cross-distribution (Test-OI) generalization, we constructed a simplified configuration that removes all ID triples from training. The model is trained only on OOD atomic triples and Train-II 2-hop queries, as illustrated in Figure 15a.

Despite having access to OOD facts and multi-hop supervision, the model fails to generalize to Test-OI queries where the first hop is from an OOD triple. As shown in Figure 15b, Test-OI accuracy remains near chance throughout training. This validates our claim in Section 4.3: without representational anchoring from ID triples, OOD-derived entities cannot support implicit multi-hop reasoning.



(a) Illustration of the Unanchored OOD Configuration.

(b) Training dynamics: Test-OI accuracy fails to improve

Figure 15: Validation experiment under ID-removed configuration. Without ID triples, the model fails to reach Phase III, confirming that representational anchoring from ID supervision is essential for OOD-based generalization.