Do Large Language Models Latently Perform Multi-Hop Reasoning?

Sohee Yang[1,2] **Elena Gribovskaya**[1] **Nora Kassner**[1] **Mor Geva**[3,4*] **Sebastian Riedel**[1,2*]

[1]Google DeepMind [2]UCL [3]Google Research [4]Tel Aviv University

{soheeyang,egribovskaya,norakassner,pipek,srriedel}@google.com

Abstract

We study whether Large Language Models (LLMs) latently perform multi-hop reasoning with complex prompts such as "The mother of the singer of 'Superstition' is". We look for evidence of a latent reasoning pathway where an LLM (1) latently identifies "the singer of 'Superstition"' as Stevie Wonder, the *bridge entity*, and (2) uses its knowledge of Stevie Wonder's mother to complete the prompt. We analyze these two hops individually and consider their co-occurrence as indicative of latent multi-hop reasoning. For the first hop, we test if changing the prompt to indirectly mention the bridge entity instead of any other entity increases the LLM's internal recall of the bridge entity. For the second hop, we test if increasing this recall causes the LLM to better utilize what it knows about the bridge entity. We find strong evidence of latent multi-hop reasoning for the prompts of certain relation types, with the reasoning pathway used in more than 80% of the prompts. However, the utilization is highly contextual, varying across different types of prompts. Also, on average, the evidence for the second hop and the full multi-hop traversal is rather moderate and only substantial for the first hop. Moreover, we find a clear scaling trend with increasing model size for the first hop of reasoning but not for the second hop. Our experimental findings suggest potential challenges and opportunities for future development and applications of LLMs.[1]

1 Introduction

Recent works have shown that Transformer-based (Vaswani et al., 2017) Large Language Models (LLMs) store and retrieve factual information in their parameters to complete simple prompts such as *"The mother of Stevie Wonder is"* (Petroni et al., 2019; Meng et al., 2022; Geva et al., 2021, 2022, 2023; Allen-Zhu and Li, 2023a). In addition, LLMs have demonstrated remarkable *in-context* reasoning abilities when the necessary information is explicitly given as part of the input (Wei et al., 2022b). For example, models can infer "Lula" as a possible completion of *"The mother of Stevie Wonder is Lula. The singer of 'Superstition' is Stevie Wonder. The mother of the singer of 'Superstition' is"*. These findings raise a question: Do LLMs retrieve factual information stored in their parameters and perform *latent multi-hop reasoning* when the information to reason from is *not* given as a part of the input? For instance, when LLMs process the two-hop prompt *"The mother of the singer of 'Superstition' is"*, do they (1) figure out that "the singer of 'Superstition"' refers to Stevie Wonder and (2) use their knowledge of who Stevie Wonder's mother is to complete the prompt?

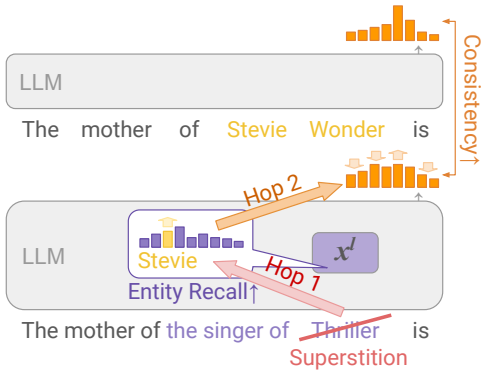


Figure 1: We investigate the latent multi-hop reasoning of LLMs. For the first hop, we change the input prompt to refer to the bridge entity (Stevie Wonder) and check how often it increases the model's internal recall of the bridge entity. For the second hop, we check if increasing this recall causes the model output to be more consistent with respect to what it knows about the bridge entity's attribute (mother of Stevie Wonder).

Answering this question is important. Evidence for such latent multi-hop reasoning would suggest that the LLM can *connect and traverse through implicit knowledge* stored in their parameters rather

[*]Corresponding authors.

[1]Our code and dataset are publicly available at https://github.com/google-deepmind/latent-multi-hop-reasoning

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10210–10229

than only storing information redundantly in its parameters. Future work could strengthen such paths of traversal, ultimately leading to more parameter-efficient and controllable models. Conversely, a lack of evidence would indicate more fundamental limitations of the Transformer architecture or training. It would also have critical implications for model editing: if complex facts are recalled instead of inferred, editing only base facts will never be enough since the changes cannot propagate (Onoe et al., 2023; Zhong et al., 2023; Cohen et al., 2023).

In this work, we limit ourselves to prompts that express a composition of two facts such as "*The mother of the singer of 'Superstition' is*" that humans can complete with two hops by (1) inferring a *bridge entity* (e.g., Stevie Wonder) and (2) inferring an attribute of that entity (e.g., who his mother is). Then, we study how often LLMs process the prompt using a similar latent two-hop reasoning pathway, although this pathway may not be the most salient pathway that largely determines the predicted output. To this end, we first study these hops individually, as shown in Figure 1. To study the first hop, we propose the *entity recall score* to approximate LLM's internal recall of the bridge entity by projecting specific hidden representations to vocabulary space. We test how changes to the input prompt affect this score. To study the second hop, we propose to measure the *consistency score* between the distributions for completions of the two-hop prompt and an equivalent recall-based one-hop prompt (e.g., "*The mother of Stevie Wonder is*"). We check how often an intervention to increase the entity recall score increases consistency as an indication of second-hop utilization. Finally, we investigate how frequently both steps coincide.

To study latent two-hop reasoning with diverse types of fact composition, we introduce TwoHopFact dataset, which is based on Wikidata (Vrandečić and Krötzsch, 2014) and consists of 45,595 two-hop prompts of 52 types of fact composition. We experiment with LLaMA-2 (Touvron et al., 2023) 7B, 13B, and 70B. Our findings can be summarized as follows. Across a wide range of fact composition types for the two-hop prompts, we find substantial evidence for the first hop of the multi-hop reasoning. In about 70% of the times where we change the prompt to indirectly mention the bridge entity, the later layers of the transformer show increased bridge entity recall. For the second hop and overall traversal, the evidence appears weaker: in 60% of the cases where we increase entity recall score, consistency goes up. Likewise, in about 40% of the time, both hops work together (compared to a random 25% baseline); changing the descriptive mention increases the entity recall score, and increasing this recall score increases consistency.

While the above aggregate statistics do not suggest a very prevalent use of the latent multi-hop reasoning pathway, it is worth pointing out that up to 23% of the fact composition types demonstrate strong evidence of latent multi-hop reasoning, occurring in more than 80% of the cases. This suggests that the pathway *exists* but is highly contextual. Additionally, we focus on a very narrow interpretation of the pathway – in reality, we expect it to be more distributed across layers and tokens. Hence, the effects we see might be a lower bound on the model's ability to perform latent two-hop reasoning. We also find striking scaling behavior: while the first hop clearly improves substantially with parameter count, the second hop (and the round-trip performance) remains relatively constant. This might indicate a fundamental limitation in today's architecture or pretraining.

Our contributions can be summarized as follows:

- We establish a **framework** for the investigation of *latent multi-hop reasoning in LLMs* and show its **existential evidence**.
- We construct the TwoHopFact **dataset** which consists of 45,595 two/one-hop prompts of 52 fact composition types, created using various types of entities and relations and diverse templates (§4).
- We propose two novel **metrics**, *internal entity recall score* and *consistency score*, as proxies of the degree of the LLM's recall of an entity for its descriptive mention (§5.1) and the degree of the LLM's utilization of its knowledge about the bridge entity's attribute (§6), respectively.
- We propose a **mechanism** to investigate a latent reasoning pathway even when it is not the most salient pathway determining the prediction, by measuring the relative frequency of the expected causal effects (§6.2).

2 Related Works

Recent works have shown that LLMs demonstrate remarkable in-context reasoning ability via prompting, which scales with model size (Brown et al., 2020; Wei et al., 2022a,b; Zhou et al., 2022). On the contrary, when the information to reason from

Notation	Example	Description
(e_1, r_1, e_2)	(Superstition, singer, Stevie Wonder)	fact triplets of named entities where e_i are named entities and r_i is a
(e_2, r_2, e_3)	(Stevie Wonder, mother, Lula)	relation function that maps e_i uniquely to e_{i+1}, such that $r_i(e_i) = e_{i+1}$
e_2	Stevie Wonder	bridge entity that connects the two fact triplets
$\tau_{1\text{H}}$	"The mother of Stevie Wonder is named"	one-hop prompt (requires one-hop reasoning)
$\tau_{2\text{H}}$	"The mother of the singer of 'Superstition' is named"	two-hop prompt (requires two-hop reasoning)
$\mu(r_1(e_1)))$	"the singer of 'Superstition'"	descriptive mention of the bridge entity e_2 created with e_1 and r_1
-	"mother of song's singer"	fact composition type

Table 1: Notations with corresponding examples from the dataset. The text in brown is the bridge entity e_2, Stevie Wonder (or the name of the bridge entity when presented as a substring in double quotation marks), and the text in purple is a descriptive mention of the bridge entity, $\mu(r_1(e_1)))$, "the singer of 'Superstition"'.

is not explicitly given as part of the input, LLMs often fail to correctly perform multi-hop reasoning even when they know the answer to the single-hop sub-step (Ofir Press et al., 2023; Dziri et al., 2023). While there have been wide investigations on how in-context reasoning works (Chan et al., 2022; Akyürek et al., 2023; Dai et al., 2023; Von Oswald et al., 2023; Prystawski and Goodman, 2023; Feng and Steinhardt, 2024), such an investigation has not been actively done to understand how latent multi-hop reasoning works.

While there have been works to investigate latent reasoning of LLMs, the exploration has been mostly done with simple single-hop reasoning tasks (Meng et al., 2022; Geva et al., 2023; Chanin et al., 2023; Hernandez et al., 2024) and/or controlled lightweight training/finetuning (Allen-Zhu and Li, 2023a,b; Saparov et al., 2023; Berglund et al., 2023, 2024). Also, many of the works that aim to identify latent reasoning pathways or circuits, have focused on finding the most salient reasoning pathway for simple synthetic tasks and/or toy models (Nanda et al., 2022; Olsson et al., 2022; Brinkmann et al., 2023; Wang et al., 2023; Conmy et al., 2023; Hou et al., 2023; Lieberum et al., 2023; McGrath et al., 2023). On the other hand, we study the existence of a latent multi-hop reasoning pathway, which may not be the most salient, in pretrained LLMs without further training, using diverse types of natural two-hop prompts.

Model editing examines ways to amend factual knowledge in LMs (De Cao et al., 2021; Mitchell et al., 2022; Meng et al., 2022; Zhang et al., 2024). However, recent works have shown that the existing editing approaches, largely focusing on single fact edits, fail to propagate the edits to facts that depend on the edited fact (Onoe et al., 2023; Zhong et al., 2023; Cohen et al., 2023). Our work explores the possibilities that such propagation could work. Moreover, our work investigates a pathway that affects the consistency at inference, whereas prior work in consistency has focused on quantifying inconsistency and improving consistency post-hoc (Ribeiro et al., 2019; Li et al., 2019; Asai and Hajishirzi, 2020; Elazar et al., 2021; Kassner et al., 2021, 2023; Jang et al., 2023). Sakarvadia et al. (2023) aim to improve multi-hop reasoning accuracy with a hypothesis that the errors stem from failure to recall the latent hop, while we investigate the foundations of this hypothesis of whether the model actually performs such a latent multi-hop reasoning. Li et al. (2024) is a concurrent work showing that a large portion of multi-hop reasoning failure cases can be attributed to incorrectly performing or utilizing the first hop of the latent multi-hop reasoning.

3 Problem Formulation

3.1 Preliminaries

We consider facts, such as *"The mother of Stevie Wonder is Lula"*, as triplets (e, r, e') of a subject entity e (e.g., Superstition), a relation r (e.g., mother), and an object entity e' (e.g., Lula). Specifically, in our analysis, we focus on triplets where e' is the only or the most well-known object entity for the relation r for e (e.g. the only mother of Stevie Wonder is Lula), and view r as a function $e' = r(e)$, where $r(e)$ is the function expression and e' is the value of the expression. We analyze how LLMs process the composition of two facts with a bridge entity e_2 connecting them, $((e_1, r_1, e_2), (e_2, r_2, e_3))$, of which the composition is represented as $r_2(r_1(e_1))$. An example is shown in Table 1.

To query LLMs, we use a template $\tau(\cdot)$ to convert expressions $r_2(e_2)$ or $r_2(r_1(e_1))$ into a prompt that can be completed correctly by the value of the given expression. For instance, the single-hop expression $\texttt{mother}(\text{Stevie Wonder})$ could be converted by $\tau(\texttt{mother}(\text{Stevie Wonder}))$ to the prompt *"The mother of Stevie Wonder is"*, which can be correctly completed with "Lula". Similarly, the two-hop expression

mother(singer(Superstition)) could be phrased by τ(mother(singer(Superstition))) as *"The mother of the singer of 'Superstition' is"* with the same correct completion. While $\tau(r_2(e_2))$ and $\tau(r_2(r_1(e_1)))$ have the same answer ("Lula"), the latter requires recalling two facts rather than one. Therefore, we call $\tau(r_2(e_2))$ a *one-hop prompt* and $\tau(r_2(r_1(e_1)))$ a *two-hop prompt*, and denote them as τ_{1H} and τ_{2H}, respectively.

We assume that the two-hop prompts yielded by $\tau(\cdot)$ for $r_2(r_1(e_1))$ always contain a noun phrase description of the bridge entity e_2 using e_1 and r_1, e.g., *"the singer of 'Superstition"'* for Stevie Wonder. We denote this description as $\mu(r_1(e_1)))$ and call it the *descriptive mention* of the bridge entity e_2.

Last, we denote the *type of the fact composition* of a two-hop prompt as "type(r_2) of type(e_1)'s type(r_1)", where "type(e_1)'s type(r_1)" represents the type of the bridge entity's descriptive mention in the prompt. For example, the fact composition type of τ(mother(singer(Superstition))) would be "mother of song's singer".

3.2 Latent Multi-Hop Reasoning in LLMs

Humans possess the deductive reasoning ability to infer conclusions from given premises, such as deducing that $r_2(r_1(e_1)) = e_3$ given a premise stating that $r_1(e_1) = e_2$ and another premise stating that $r_2(e_2) = e_3$. This multi-hop reasoning (Welbl et al., 2018; Yang et al., 2018) involves identifying the bridge entity (e.g., that "the singer of 'Superstition"' is Stevie Wonder) and using it to solve for the final answer (e.g., that Stevie Wonder's mother is Lula).

Our research explores the extent to which a pretrained Transformer-based Large Language Model (LLM) can perform similar multi-hop reasoning when completing a two-hop prompt. Given the complex nature of LLMs, which function through high-dimensional and distributed representations, it's unlikely for a single deterministic algorithm to govern their predictions except for under highly controlled and constrained setup (Nanda et al., 2022; Wang et al., 2023). Instead, LLMs may use aggregations from multiple inference pathways, ranging from shallow n-gram co-occurrence-based matching to deeper rule-based reasoning or even multi-hop reasoning, to make a prediction.

Therefore, to identify a pathway indicative of latent multi-hop reasoning, we focus on the internal dynamics of LLMs in processing two-hop prompts rather than the most salient pathway that contributes the most to the output. This involves analyzing how the LLM's recall and utilization of the knowledge $r_1(e_1)$ and $r_2(e_2)$ changes in response to certain alterations made while the LLM is processing a two-hop prompt, in what we consider as the first and second hop of reasoning, respectively.

Specifically, we investigate the following two key research questions (RQs):

RQ1. How often does an LLM perform the first hop of reasoning while processing two-hop prompts? We view the first-hop reasoning as the LLM's recall of the bridge entity for its descriptive mention. Therefore, we examine the frequency with which the LLM's internal recall of the bridge entity increases when it encounters a descriptive mention of the bridge entity within a prompt. For instance, we investigate whether altering the prompt from "The mother of the singer of *'Thriller'* is" to "The mother of the singer of *'Superstition'* is" increases the LLM's internal recall of Stevie Wonder.

RQ2. How often does an LLM perform the second hop of reasoning while processing two-hop prompts? We view the second-hop reasoning as the LLM's utilization of the first-hop reasoning for the second hop. Therefore, we examine the frequency with which enhancing the LLM's recall of the bridge entity for its descriptive mention improves its use of the knowledge about the bridge entity to answer the two-hop prompt. For example, we investigate if increasing the internal recall of Stevie Wonder for *"the singer of 'Superstition'"* makes the LLM better utilize its knowledge of Stevie Wonder's mother to complete the prompt.

By addressing these questions, we aim to identify evidence of LLMs leveraging a latent pathway for multi-hop reasoning.

4 TwoHopFact Dataset

To answer our questions with prompts of diverse fact composition types, we construct TwoHopFact using well-known named entities in Wikidata (Vrandečić and Krötzsch, 2014) and manually selected relations (Appendix A). TwoHopFact consists of 45,595 unique pairs of one-hop and two-hop prompts of 52 fact composition types constructed from the same number of fact triplet pairs $((e_1, r_1, e_2), (e_2, r_2, e_3))$ as in Table 1. Appendix Table 3 shows example two-hop prompts for each fact composition type, and Appendix B provides detailed data statistics.

5 First Hop of Multi-Hop Reasoning

In this section, we answer RQ1 of *how often an LLM performs the first hop of reasoning while processing two-hop prompts*. We first introduce EntRec as a metric to approximate the LLM's internal recall of the bridge entity upon its descriptive mention in a prompt (§5.1). Next, we propose to measure how often this recall increases when changing the input prompt to indirectly mention the bridge entity (§5.2). Then, we evaluate this using TwoHopFact and answer RQ1 (§5.3).

5.1 Internal Entity Recall Score

We define EntRec as a metric to measure the LLM's recall of the bridge entity e_2 within a two-hop prompt τ_{2H}. This is defined with respect to the hidden representation in a certain layer l, at the last position of the bridge entity's descriptive mention in the two-hop prompt. This hidden representation is projected to the vocabulary space to calculate the log probability of the first token of the entity's name (e.g., the first token of "Stevie Wonder"). Formally, let $e_2^{(0)}$ be the first token of e_2, then:

$$\begin{aligned} &\text{EntRec}^l(e_2, \tau_{2H}) \qquad (1)\\ &= \log \text{softmax}(\text{LayerNorm}(\mathbf{x}^l)W_U)_{\text{index}(e_2^{(0)})}, \end{aligned}$$

where $\mathbf{x}^l \in \mathbb{R}^h$ is the output from the l-th Transformer layer at the last token of the bridge entity's descriptive mention in the two-hop prompt τ_{2H}, and $\text{index}(e_2^{(0)}) \in [0, V-1]$ is the index of the token $e_2^{(0)}$ in the unembedding matrix $W_U \in \mathbb{R}^{h \times V}$. LayerNorm is the layer normalization used for the last layer output $\mathbf{x}^{L-1}$ before projecting it to the unembedding matrix to obtain the output next-token probability distribution. Applying this normalization makes $\text{EntRec}^{L-1}(e_2, \tau_{2H})$ compatible with the output probability of $e_2^{(0)}$ as the next token of the prefix of τ_{2H} ending at the descriptive mention (e.g., "The mother of the singer of 'Superstition"').[2] We interpret higher $\text{EntRec}^l(e_2, \tau_{2H})$ as stronger internal recall of the bridge entity e_2 at the l-th layer.

The proposed definition of EntRec is inspired by previous works which report that the representation constructed at the last token position of a subject often plays an important role in encoding information about the subject (Meng et al., 2022; Geva et al., 2023), the work of nostalgebraist (2020) that projects early-layer outputs to the vocabulary space, and the work of Geva et al. (2022) which shows that such projections at the last subject token position of one-hop prompts provide interpretable top-rank attributes that are semantically relevant to the subject. Although EntRec assesses the recall of an entity with respect to only the first token of its name, it is directly related to how autoregressive LLMs process the input text and prepare the next token to generate. A control experiment in Appendix C validates EntRec as a reasonable proxy for measuring the internal entity recall.

[2] We omit the bias term as it often models the frequency of the token (Kobayashi et al., 2023), which we do not want to consider for measuring the internal recall of an entity.

5.2 Experiment

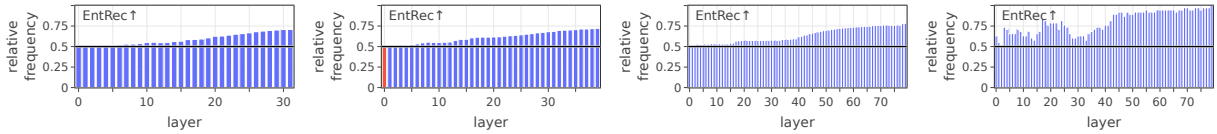
Given EntRec, we answer RQ1 by measuring how often the internal recall of e_2 improves at layer l when modifying a two-hop prompt from τ'_{2H} to τ_{2H}, where τ'_{2H} does not contain the descriptive mention of e_2 while τ_{2H} does. To be specific, we measure *the relative frequency* of τ_{2H} in TwoHopFact where $\text{EntRec}^l(e_2, \tau_{2H}) > \text{EntRec}^l(e_2, \tau'_{2H})$.

To construct τ'_{2H}, we alter the descriptive mention of the bridge entity in τ_{2H} in two ways: by replacing e_1 with e'_1 such that $\mu(r_1(e'_1))$ does not point to e_2, or r_1 with r'_1 to ensure $\mu(r'_1(e_1))$ does not refer to e_2. Examples include substituting "the singer of 'Superstition"' in τ_{2H} to "the singer of '*Thriller*"' or "*a plagiarist* of 'Superstition"'. These adjustments are termed *entity substitution* and *relation substitution*, respectively.

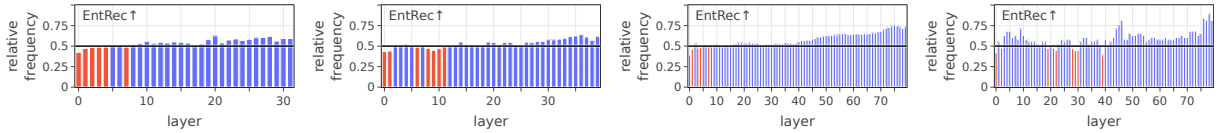
For each two-hop prompt τ_{2H} in TwoHopFact, we randomly select one e'_1 from the same fact composition type and one r'_1 from a set of predefined candidate relations (provided in Appendix Table 5) to create τ'_{2H}. We then measure the relative frequency of cases where replacing τ'_{2H} with τ_{2H} via entity or relation substitution increases the recall of e_2. A relative frequency above 0.5 suggests the LLM's chance to perform first-hop reasoning exceeds the random chance for these prompts.

5.3 Results

There is substantial evidence of the first hop of reasoning, which becomes stronger with increasing model size. Figure 2 shows the relative frequency of the cases that the entity recall at each layer increases with entity and relation substitution. LLaMA-2 7B entity substitution result (Figure 2a) shows that the evidence of first-hop reasoning becomes clearer with increasing layer depth, peaking at 0.71 in layer 31. Relation substitution exhibits a slightly noisier pattern with a peak at 0.63 in layer

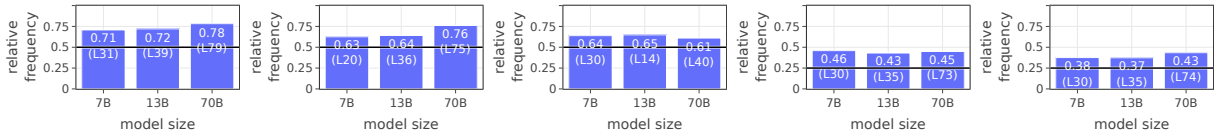


(a) 7B entity substitution (b) 13B entity substitution (c) 70B entity substitution (d) 70B entity substitution for "president of anthem's country"



(e) 7B relation substitution (f) 13B relation substitution (g) 70B relation substitution (h) 70B relation subst. for "president of anthem's country"

Figure 2: Relative frequency of the cases where the internal recall of the bridge entity of LLaMA-2 increases with entity substitution (top row) and relation substitution (bottom row). Bars are colored blue if the relative frequency is greater than or equal to 0.5 and red otherwise.



(a) RQ1 entity substitution result (§5) (b) RQ1 relation substitution result (§5) (c) RQ2 result (§6) (d) RQ1&2 entity substitution result (§7) (e) RQ1&2 relation substitution result (§7)

Figure 3: Experimental results with increasing scale of LLaMA-2. Technical details for all experiments in our work can be found in Appendix E.

20 (Figure 2e).

As model size increases from 7B to 13B and 70B, first-hop reasoning occurs more frequently for both entity substitution and relation substitution. For the former, the maximum relative frequency rises from 0.71 (7B) to 0.72 (13B) and 0.78 (70B) (Figure 3a). For the latter, it increases from 0.63 (7B) to 0.64 (13B) and 0.76 (70B) (Figure 3b).

Relatively strong evidence supports the first-hop reasoning in up to 73% of fact composition types. With LLaMA-2 7B-13B-70B, 18/25/34 and 21/27/38 out of 52 of fact composition types exhibit maximum relative frequencies exceeding 0.8 for entity and relation substitution, respectively. In addition, 11 out of 52 types demonstrate such strong first-hop reasoning evidence robustly across all model sizes and substitution types. For example, the maximum frequency of "president of anthem's country" ("The country with the national anthem 'Azat u ankakh Artsakh' is led by president") shows the maximum frequency of 0.97/0.92/1.0 (Figure 2d) and 0.87/0.87/0.89 (Figure 2h) with each model and substitution, respectively. Individual fact composition types exhibit diverse patterns of relative frequency across layers.

6 Second Hop of Multi-Hop Reasoning

In this section, we answer RQ2 of *how often an LLM performs the second-hop reasoning while processing two-hop prompts*. We view the second hop of reasoning as the LLM's utilization of what it knows about the bridge entity's attribute (Stevie Wonder's mother) to answer the two-hop prompt about the same attribute of the entity referred to by the descriptive mention (the singer of 'Superstition"s mother). Therefore, when an LLM performs the second hop, we expect to see a connection between its recall of the bridge entity (i.e. resolving the first hop) and its similarity in responding to a two-hop prompt and a corresponding one-hop prompt about the bridge entity's attribute, e.g., the two-hop prompt *"The mother of the singer of 'Superstition' is"* and the one-hop prompt *"The mother of Stevie Wonder is"*. Namely, the more strongly the model recalls the bridge entity (e.g., Stevie Wonder) while processing the two-hop prompt, the more similar the completion of this prompt should be to the completion of the one-hop prompt. In the following, we describe our approach for testing how often such a causal connection exists between entity recall and the *similarity* in the prompt completions, which we refer to as *consistency*.

6.1 Consistency Score

We define CNSTSCORE to measure how consistently an LLM responds to the two-hop and one-hop prompts. Let $\mathbf{p}_{\tau_{2H}}, \mathbf{p}_{\tau_{1H}} \in \mathbb{R}^V$ be the output

probability distributions for a two-hop prompt τ_{2H} and the corresponding one-hop prompt τ_{1H}, respectively. Denoting $\mathrm{H}(Q, P) = -\sum_{i=0}^{V-1} P_i \log Q_i$ as the cross-entropy between probability distributions P and Q, we define:

$$\begin{aligned}&\textsc{CnstScore}(\tau_{2H}, \tau_{1H}) \\ &= -0.5\mathrm{H}(\mathbf{p}_{\tau_{2H}}, \mathbf{p}_{\tau_{1H}}) - 0.5\mathrm{H}(\mathbf{p}_{\tau_{1H}}, \mathbf{p}_{\tau_{2H}}).\end{aligned} \tag{2}$$

This score evaluates the similarity between the two probability distributions by computing and averaging their cross-entropy, ensuring symmetry in the evaluation. The symmetry from averaging mitigates sensitivity to the individual distribution's entropy levels, aiming for equal treatment of divergences in both directions.

Note that we use consistency instead of two-hop prompt completion accuracy or the probability of the ground truth answer because the latter metrics are insufficient to capture the second-hop reasoning for the cases where the corresponding one-hop prompt completion is incorrect. In addition, these metrics inherit noise from the choice of the ground truth answer or the set of answer candidates. On the other hand, comparing the similarity of the output distributions is not affected by the choice of ground truth, and provides a way to capture the second-hop reasoning even when the ground truth answer is not in the top-1 generation of the one-hop prompt.

Also, we do not choose to compare the completion strings or their binary accuracy of the one/two-hop prompts because these metrics cannot capture subtle consistency differences in the probability distribution. We choose cross-entropy rather than Kullback–Leibler or Jensen-Shannon divergence because the latter metrics contain an entropy term that is irrelevant to consistency, but can dominate the score, diluting the cross-entropy signal. Higher consistency scores indicate greater similarity between the output distributions. In Appendix D, we provide empirical evidence for the consistency score being a reasonable approximation of the utilization of the model's knowledge about the bridge entity's attribute.

6.2 Experiment

Given EntRec and CnstScore, we answer RQ2 by measuring how often increasing the recall of the bridge entity e_2 at the l-th layer increases the LLM's consistency in answering the two-hop prompt with respect to the one-hop prompt. In other words, we examine whether increasing $\textsc{EntRec}^l(e_2, \tau_{2H})$ leads to increasing $\textsc{CnstScore}(\tau_{2H}, \tau_{1H})$.

We would have been able to use differential calculus to obtain the answer by calculating the direction of change if $\textsc{CnstScore}(\tau_{2H}, \tau_{1H})$ were directly dependent on $\textsc{EntRec}^l(e_2, \tau_{2H})$. However, there exists no direct functional dependency between the two values. Instead, we leverage the shared reliance of both metrics on $\mathbf{x}^l$ for computation where $l \in [0, L-1)$,[3] redefining them as $\textsc{EntRec}(\mathbf{x}^l)$ and $\textsc{CnstScore}(\mathbf{x}^l)$ relative to $\mathbf{x}^l$. This reparameterization allows us to change the question to: if $\textsc{EntRec}(\mathbf{x}^l)$ is increased by altering $\mathbf{x}^l$, does $\textsc{CnstScore}(\mathbf{x}^l)$ also increase?

To explore this, we adjust $\textsc{EntRec}(\mathbf{x}^l)$ in the direction of its steepest increase, represented by $\nabla_{\mathbf{x}^l}\textsc{EntRec}(\mathbf{x}^l)$, and observe the impact on $\textsc{CnstScore}(\mathbf{x}^l)$ by modifying $\mathbf{x}^l$ according to a magnitude of change α:

$$\hat{\mathbf{x}}^l(\alpha) = \mathbf{x}^l + \alpha\nabla_{\mathbf{x}^l}\textsc{EntRec}(\mathbf{x}^l).$$

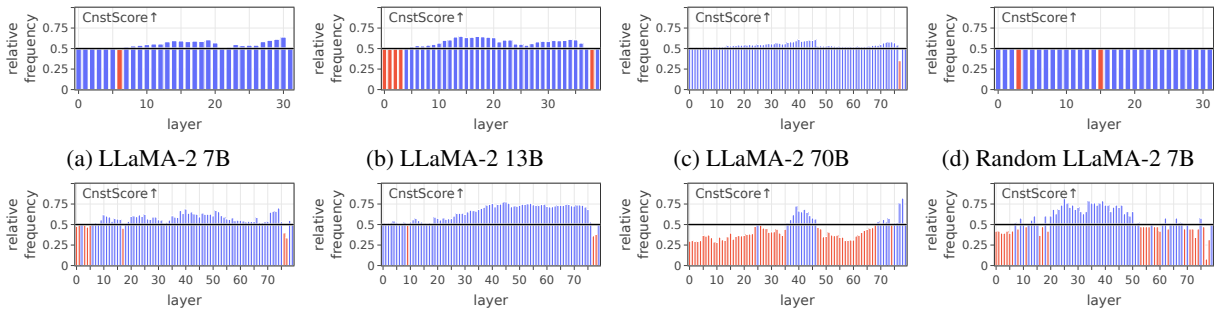
Subsequently, we calculate $\textsc{CnstScore}(\mathbf{x}^l)$ using $\hat{\mathbf{x}}^l(\alpha)$,[4] which allows us to express it as a function $\textsc{CnstScore}(\alpha)$ of α. Then, we examine its derivative, $\frac{d}{d\alpha}\textsc{CnstScore}(\alpha)\big|_{\alpha=0}$ to understand the direction of change at the current value. A positive derivative indicates that an increase in $\textsc{EntRec}(\mathbf{x}^l)$ leads to an increase in $\textsc{CnstScore}(\tau_{2H}, \tau_{1H})$, while a negative one suggests the opposite. By assessing *the relative frequency of positive gradients* among the two-hop prompts in TwoHopFact, we quantify how often the LLM performs the second hop of the reasoning, with frequencies above 0.5 suggesting that the LLM's chance to perform the second-hop reasoning exceeds random chance for these prompts.

6.3 Results

There is moderate evidence of the second-hop reasoning, which does not become stronger with increasing model size. Figure 4 shows the relative frequency of the cases where increasing the bridge entity recall increases the consistency. In LLaMA-2 7B, the middle and late layers exhibit a relative frequency higher than 0.5 (random chance) with statistical significance, peaking at 0.64 in layer

[3] $\textsc{CnstScore}(\tau_{2H}, \tau_{1H})$ utilizes $\mathbf{p}_{\tau_{2H}}$, which utilizes $\mathbf{x}^l$ for its calculation. However, only $\mathbf{x}^l$ where $l = 0, \cdots, L-2$ are used to calculate the attention outputs at layers $l = 1, \cdots, L-1$, respectively, to get $\mathbf{p}_{\tau_{2H}}$.

[4] We use activation patching (Wang et al., 2023) to implement the replacement of $\mathbf{x}^l$ with $\hat{\mathbf{x}}^l(\alpha)$.



(a) LLaMA-2 7B (b) LLaMA-2 13B (c) LLaMA-2 70B (d) Random LLaMA-2 7B

(e) 70B result of "stock exchange of game's developer" (f) 70B result of "mother of song's singer" (g) 70B result of "founder of person's undergrad university" (h) 70B result of "president of anthem's country"

Figure 4: Relative frequency that stronger recall of the bridge entity at the l-th layer increases the consistency of the LLM. Bars are colored blue if the relative frequency is greater than or equal to 0.5 and red otherwise. We manually set the value of 0.5 at the last layer because the intervention does not affect the consistency at that layer.

30. Test result with a randomly initialized model verifies 0.5 as the randomness baseline (Figure 4d).

However, unlike the first-hop reasoning (§5), the second-hop reasoning does not strengthen with increasing model size; when scaling from 7B to 13B and 70B, the maximum relative frequency remains relatively stable at 0.64 (7B), 0.65 (13B), and 0.61 (70B), as shown in Figure 3c. This observation does not change even when the test is conducted using the log probability of the ground truth answer instead of CNSTSCORE (Appendix F). It is worth noting that this finding aligns with the observation of Ofir Press et al. (2023), that the single-hop question answering performance improves faster than the multi-hop performance as the model size increases, and thus the *compositionality gap* (the ratio of how often models can correctly answer all sub-problems but not generate the overall solution) does not decrease with increasing model size.

Relatively strong evidence supports the second-hop reasoning in up to 19% of fact composition types. With LLaMA-2 7B-13B-70B, 10/7/5 out of 52 of fact composition types exhibit maximum relative frequencies exceeding 0.8, respectively. Among them, "founder of person's undergraduate university" and "president of anthem's country" demonstrate such strong second-hop reasoning evidence across all model sizes, with a maximum frequency of 0.86/0.81/0.82 (Figure 4g) and 0.84/0.89/0.82 (Figure 4h), respectively.

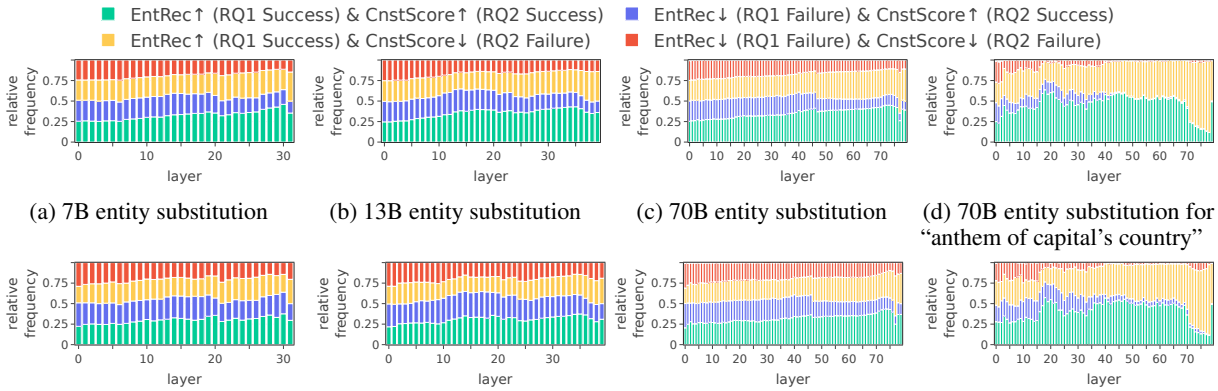
7 Latent Multi-Hop Reasoning

In this section, we measure *how often LLMs perform latent multi-hop reasoning while processing the two-hop prompt* by combining our answers to RQ1 and RQ2. For each two-hop prompt, we consider successful outcomes for RQ1 (an entity recall increase with entity/relation substitution) and RQ2 (a consistency increase with increased entity recall) as evidence of the first and second hops of reasoning, respectively. Four possible outcomes arise: (SS) success in both RQ1 and RQ2 that we view as the multi-hop reasoning; (FS) failure in RQ1 but success in RQ2; (SF) success in RQ1 but failure in RQ2; (FF) failure in both RQ1 and RQ2.

There is moderate evidence of the latent multi-hop reasoning, which sometimes becomes stronger with increasing model size. Figure 5 shows the relative frequency of the four cases, where green, blue, yellow, and red represent each of the cases of SS, FS, SF, and FF, respectively. LLaMA-2 7B exhibits a relative frequency for successful multi-hop reasoning (green) above random chance (0.25), peaking at 0.46 (entity substitution) and 0.38 (relation substitution). The likelihood of partial multi-hop reasoning (green + blue + yellow) exceeds 0.8 in later layers.

While entity substitution results do not show increased multi-hop reasoning with model size (Figure 3d), relation substitution exhibits a scaling trend. From 7B to 70B, the maximum relative frequency increases from 0.38 to 0.43, suggesting that larger models may facilitate multi-hop reasoning with relational changes (Figure 3e).

Relatively strong evidence supports latent multi-hop reasoning in up to 23% of fact composition types. Considering $0.8^2 = 0.64$ as the threshold, with respect to LLaMA-2 7B-13B-70B, 7/3/12 types exceed the threshold with entity substitution and 3/3/9 types do so with relation substitution. The maximum frequency of "anthem of capital's country" ("The national anthem of the country led by president Lazarus Chakwera is named") exceeds



(a) 7B entity substitution (b) 13B entity substitution (c) 70B entity substitution (d) 70B entity substitution for "anthem of capital's country"

(e) 7B relation substitution (f) 13B relation substitution (g) 70B relation substitution (h) 70B relation subst. for "anthem of capital's country"

Figure 5: Relative frequency of the four outcomes of RQ1 and RQ2 in LLaMA-2 models, with entity substitution (top row) and relation substitution (bottom row) for RQ1. Let the increase of the entity recall with the input substitution for the first hop reasoning be the *success* case of RQ1, and the increase of the consistency score with the increased entity recall for the second hop reasoning be the *success* case of RQ2. The green, blue, yellow, and red bars show the cases of SS (success-success), FS, SF, and FF for RQ1 and RQ2, respectively. We manually set the value of the last layer as 0.5 multiplied by the relative frequency for RQ1 because the intervention does not affect the consistency at that layer.

this threshold across all models and substitutions with 0.68/0.82/0.66 (Figure 5d) and 0.74/0.82/0.68 (Figure 5h), respectively. Individual types show diverse patterns distinct from the overall dataset.

8 Discussion and Conclusion

Our work studies the latent multi-hop reasoning abilities of LLMs. We find strong evidence of latent multi-hop reasoning for certain fact composition types with the reasoning pathway utilized in more than 80% of the cases. However, the utilization is highly contextual; there are also fact composition types where we see weak or almost no evidence of reasoning. The evidence of second and multi-hop reasoning across the whole set of prompts is rather moderate and only substantial in the first hop.

Moreover, while we see a clear scaling trend with the first hop of the latent multi-hop reasoning pathway with increasing model size, we do not see such scaling evidence for the second-hop reasoning pathway. This could be the reason behind the observation of Ofir Press et al. (2023) that the compositionality gap (the ratio of how often models can correctly answer all sub-problems but not generate the overall solution) does not decrease with increasing model size.

Although our analysis is based on LLaMA-2 family of models of up to 70B parameters, our findings suggest potential limitations in the current scaling paradigm for promoting latent multi-hop reasoning. Thus, we may need to study the choice of pretraining data, loss functions that promote knowledge retrieval and utilization, or model architectures with a stronger inductive bias towards internal knowledge representation for LLMs' stronger latent reasoning abilities. However, analyzing the subset of prompts with strong evidence of multi-hop reasoning with respect to pretraining dynamics and data may give insights into the emergence of such abilities even in the context of the current pretraining and scaling paradigm.

Overall, our findings advance the understanding of LLM capabilities and can guide future research aiming to promote and strengthen latent multi-hop reasoning which is relevant for parameter efficiency, generalization, and controllability.

9 Limitations

Latent Multi-Hop Reasoning Pathway While we study one pathway for latent multi-hop reasoning (e.g., we test the use of the second hop by means of entity recall), considering the potential redundancy of inference pathways in LLMs (McGrath et al., 2023), other pathways might exist; the same information might be retrieved in different ways. Also, we don't measure multi-hop reasoning end-to-end and track only the changes that occur in the first and the second hop with respect to a single layer, while the effect of the first hop of reasoning could possibly propagate to other layers. Hence, the effects we see might be a lower bound on the model's ability to perform latent two-hop reasoning.

Dataset We aim to collect fact triplets (e, r, e') such that $e' = r(e)$ is the only or the most famous object for the relation r for e. Although we use the entities with the most number of reference links and ensure that e' is the only object entity at least among the collected fact triplets for this purpose, there are noises introduced from Wikidata. Besides, in reality, it is difficult to strictly satisfy the condition of "only" due to the vast amount of real-world knowledge that changes rapidly and dynamically.

Metrics Our measure of internal entity recall is an approximation as we use only the first token of the entity, although it is directly related to how LLMs process the input text and prepare the next token to generate. Moreover, the internal entity recall score is based on logit lens (nostalgebraist, 2020) which has shortcomings such as representation drift, bias, and brittleness (Belrose et al., 2023; Timkey and van Schijndel, 2021). However, these limitations have minimal effect on our analysis because our focus is not on making the prediction accurate in early layers as studied for adaptive computation methods such as early exit (Din et al., 2023), but to study the LLM's internal dynamics as-is.

Acknowledgements

We would like to thank Sang-Woo Lee, Jasmijn Bastings, William Cohen, and Owain Evans for the valuable feedback and discussions.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? investigations with linear models. In *ICLR*.

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

Zeyuan Allen-Zhu and Yuanzhi Li. 2023b. Physics of language models: Part 3.2, knowledge manipulation. *arXiv*.

Akari Asai and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering. In *ACL*.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv*.

Lukas Berglund, Asa Cooper Stickland, Mikita Balesni, Max Kaufmann, Meg Tong, Tomasz Korbak, Daniel Kokotajlo, and Owain Evans. 2023. Taken out of context: On measuring situational awareness in llms. *arXiv*.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2024. The reversal curse: LLMs trained on "a is b" fail to learn "b is a". In *ICLR*.

Jannik Brinkmann, Abhay Sheshadri, Victor Levoso, Paul Swoboda, and Christian Bartelt. 2023. A mechanistic analysis of a transformer trained on a symbolic multi-step reasoning task.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. In *NeurIPS*.

David Chanin, Anthony Hunter, and Oana-Maria Camburu. 2023. Identifying linear relational concepts in large language models. *arXiv*.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2023. Evaluating the ripple effects of knowledge editing in language models. *arXiv*.

Arthur Conmy, Augustine N Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *NeurIPS*.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of ACL*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *EMNLP*.

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2023. Jump to conclusions: Short-cutting transformers with linear transformations. *arXiv*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D

Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. In *NeurIPS*.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *TACL*.

Jiahai Feng and Jacob Steinhardt. 2024. How do language models bind entities in context? In *ICLR*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *EMNLP*.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *EMNLP*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *EMNLP*.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2024. Linearity of relation decoding in transformer language models. In *ICLR*.

Yifan Hou, Jiaoda Li, Yu Fei, Alessandro Stolfo, Wangchunshu Zhou, Guangtao Zeng, Antoine Bosselut, and Mrinmaya Sachan. 2023. Towards a mechanistic interpretation of multi-step reasoning capabilities of language models. In *ACL*.

Myeongjun Jang, Bodhisattwa Prasad Majumder, Julian McAuley, Thomas Lukasiewicz, and Oana-Maria Camburu. 2023. Know how to make up your mind! adversarially detecting and alleviating inconsistencies in natural language explanations. In *ACL*.

Nora Kassner, Oyvind Tafjord, Ashish Sabharwal, Kyle Richardson, Hinrich Schuetze, and Peter Clark. 2023. Language models with rationality. In *EMNLP*.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief. In *EMNLP*.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2023. Transformer language models handle word frequency in prediction head. In *ACL*.

Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. 2019. A logic-driven framework for consistency of neural models. In *EMNLP*.

Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. 2024. Understanding and patching compositional reasoning in llms. *arXiv*.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. 2023. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv*.

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. 2023. The hydra effect: Emergent self-repair in language model computations. *arXiv*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *NeurIPS*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In *ICLR*.

Neel Nanda and Joseph Bloom. 2022. Transformerlens. https://github.com/neelnanda-io/TransformerLens.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2022. Progress measures for grokking via mechanistic interpretability. In *ICLR*.

nostalgebraist. 2020. interpreting gpt: the logit lens.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of EMNLP*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *arXiv*.

Yasumasa Onoe, Michael J Q Zhang, Shankar Padmanabhan, Greg Durrett, and Eunsol Choi. 2023. Can LMs learn new entities from descriptions? challenges in propagating injected knowledge. In *ACL*.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory

Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report. *arXiv*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? In *EMNLP*.

Ben Prystawski and Noah D Goodman. 2023. Why think step-by-step? reasoning emerges from the locality of experience. In *NeurIPS*.

Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. 2019. Are red roses red? evaluating consistency of question-answering models. In *ACL*.

Mansi Sakarvadia, Aswathy Ajith, Arham Khan, Daniel Grzenda, Nathaniel Hudson, André Bauer, Kyle Chard, and Ian Foster. 2023. Memory injections: Correcting multi-hop reasoning failures during inference in transformer-based language models. *arXiv*.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Seyed Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the general deductive reasoning capacity of large language models using OOD examples. In *NeurIPS*.

William Timkey and Marten van Schijndel. 2021. All bard and no bite: Rogue dimensions in transformer language models obscure representational quality. In *EMNLP*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *ICML*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *ICLR*.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. In *NeurIPS*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *TACL*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing. *arXiv*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. 2024. A comprehensive study of knowledge editing for large language models. *arXiv*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. MQAKE: Assessing knowledge editing in language models via multi-hop questions. In *EMNLP*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. In *ICLR*.

A Dataset construction

We construct TWOHOPFACT using Wikidata (Vrandečić and Krötzsch, 2014) with the following data construction pipeline.

A.1 Data Selection

We select relations and entities that are well-known and result in sufficient numbers of samples per relation. Relations are selected manually. At the time of querying Wikidata, we constrain entities to singular entities with natural language Wikipedia titles and select entities with a maximal number of reference links. We also exclude the cases of $e_1 = e_2$ that might allow trivial recall of e_2 by directly copying from the input. In addition, we make sure that bridge entities e_2 are unique among the facts of the same fact composition type to mitigate the imbalance in the bridge entity. Finally, we apply down-sampling to mitigate the imbalance in the fact composition type.

Relation Selection First, we determine the type of the bridge entity's descriptive mention by selecting the type of entities e_1 and relation r_1 to collect $r_1(e_1) = e_2$. The bridge entities we select have types like "song's singer" (the singer of a specific song), "country's anthem" (the country with a specific national anthem), "founder's organization" (the organization founded by a specific person), and "organization's ceo" (the CEO of a specific organization). For example, while there can be many authors for some novels, "author's novel" is selected as a type of descriptive mention of the bridge entity because we can use only the novels with a single author. We determine 19 types of bridge entity's descriptive mention with this process.

Now that we have "$\mathrm{type}(e_1)$'s $\mathrm{type}(r_1)$" determined, we determine the type of relations r_2 to determine the type of the fact composition, "$\mathrm{type}(r_2)$ of $\mathrm{type}(e_1)$'s $\mathrm{type}(r_1)$". Note that "$\mathrm{type}(e_1)$'s $\mathrm{type}(r_1)$" determined in the previous step falls into the category of country, organization (organization, undergraduate university, game developer), real person (author, president, CEO, spouse, singer), fictional character (main character), movie, novel, or city (headquarters city). Note that "$\mathrm{type}(e_1)$'s $\mathrm{type}(r_1)$" is also the bridge entity itself that the descriptive mention refers to. Therefore, we select r_2 that are likely to give us a sufficient number of (e_2, r_2, e_3) where e_3 is the only object entity satisfying the relation r_2 for these categories of e_2. As in the previous step, we select common relations as r_2. Using the selected types of r_2, we create 52 fact composition types including "mother of song's singer" (the city where the novel of a specific novel was born), "headquarterscity of video game's developer" (the city where the headquarters of the developer of a specific video game is located), and "director of main character's movie" (the director of the movie which has a specific character as the main character).

Querying Wikidata We collect the fact triplets of the selected fact composition types through Wikidata Query Service[5] with one handcrafted query for each of the 52 fact composition types. When there are too many results for the API call to bring before a timeout occurs, we reduce the number of the results by filtering the results with the number of reference links and/or adding other conditions to the query.

For the relations that are subject to change by nature, e.g., CEO of a company, we retrieve the information at the time of January 1, 2022. We choose this timestamp considering the training time of LLaMA-2 (Touvron et al., 2023) models that we use for our study. To analyze LLMs trained at different times, filtering the dataset with model accuracy before the analysis or using the prompts of the fact composition types that are less likely to change over time (e.g., "founder of videogame's developer") would resolve potential issues from the temporality of the dataset.

A.2 Natural Language Templates

We manually create natural language templates. To this end, we first create descriptive mentions of the bridge entity. To create the descriptive mentions, we manually write r_1-specific *mention-constructing templates* $m_{r_1}(\cdot)$. For example, $m_{\text{singer}}(\cdot)$ = "the singer of '$\cdots$'" creates $\mu(r_1(e_1)))$ = "the singer of 'Superstition'".

Next, we create one/two-hop prompt templates. We manually write r_2-specific *prompt-constructing templates* $t_{r_2}(\cdot)$ that take a mention of the bridge entity e_2 and form a prompt querying about e_2's relational attribute r_2 in a way that the prompt can be correctly answered with a mention of e_3. For example, $t_{\text{mother}}(\cdot)$ = "The mother of $\cdots$ is" is used to create the one-hop prompt "The mother of Stevie Wonder is" and also the two-hop prompt "The mother of the singer of 'Superstition' is".

[5] https://query.wikidata.org

Term	Notation	Example
fact composition type	"type(r_2) of type(e_1)'s type(r_1)"	"birth city of novel's author"
first fact triplet	(e_1, r_1, e_2)	(Ubik, author, Philip K. Dick)
second fact triplet	(e_2, r_2, e_3)	(Philip K. Dick, birth city, Chicago)
mention-constructing template	$m_{r_1}(\cdot)$	$m_{\text{author}}(\cdot)$ = "the author of the novel $\cdots$"
prompt-constructing template	$t_{r_2}(\cdot)$	$t_{\text{birth city}}(\cdot)$ = "$\cdots$ was born in the city of"
descriptive mention of e_2	$\mu(r_1(e_1))) = m_{r_1}(n_{e_1})$	$m_{\text{author}}(n_{\text{Ubik}})$ = "the author of the novel Ubik"
two-hop prompt	$\tau(r_2(r_1(e_1))) = t_{r_2}(m_{r_1}(n_{e_1}))$	$t_{\text{birth city}}(m_{\text{author}}(n_{\text{Superstition}}))$ = "The author of the novel Ubik was born in the city of"
one-hop prompt	$\tau(r_2(e_2)) = t_{r_2}(n_{e_2})$	$t_{\text{birth city}}(n_{\text{Philip K. Dick}})$ = "Philip K. Dick was born in the city of"
fact composition type	"type(r_2) of type(e_1)'s type(r_1)"	"director of main character's movie"
first fact triplet	(e_1, r_1, e_2)	(Dominick Cobb, movie, Inception)
second fact triplet	(e_2, r_2, e_3)	(Inception, director, Christopher Nolan)
mention-constructing template	$m_{r_1}(\cdot)$	$m_{\text{movie}}(\cdot)$ = "the movie featuring $\cdots$ as the main character"
prompt-constructing template	$t_{r_2}(\cdot)$	$t_{\text{director}}(\cdot)$ = "The name of the director of $\cdots$ is"
descriptive mention of e_2	$\mu(r_1(e_1))) = m_{r_1}(n_{e_1})$	$m_{\text{movie}}(n_{\text{Dominick Cobb}})$ = "the movie featuring Dominick Cobb as the main character"
two-hop prompt	$\tau(r_2(r_1(e_1))) = t_{r_2}(m_{r_1}(n_{e_1}))$	$t_{\text{director}}(m_{\text{movie}}(n_{\text{Dominick Cobb}}))$ = "The name of the director of the movie featuring Dominick Cobb as the main character is"
one-hop prompt	$\tau(r_2(e_2)) = t_{r_2}(n_{e_2})$	$t_{\text{director}}(n_{\text{Inception}})$ = "The name of the director of Inception is"
fact composition type	"type(r_2) of type(e_1)'s type(r_1)"	"stock exchange of video game's developer"
first fact triplet	(e_1, r_1, e_2)	(Assassin's Creed: Lost Legacy, developer, Ubisoft)
second fact triplet	(e_2, r_2, e_3)	(Ubisoft, stock exchange, Euronext Paris)
mention-constructing template	$m_{r_1}(\cdot)$	$m_{\text{developer}}(\cdot)$ = "the developer of the game '$\cdots$'"
prompt-constructing template	$t_{r_2}(\cdot)$	$t_{\text{stock exchange}}(\cdot)$ = "$\cdots$ is listed on a stock exchange named"
descriptive mention of e_2	$\mu(r_1(e_1))) = m_{r_1}(n_{e_1})$	$m_{\text{developer}}(n_{\text{Assassin's Creed: Lost Legacy}})$ = "the developer of the game 'Assassin's Creed: Lost Legacy'"
two-hop prompt	$\tau(r_2(r_1(e_1))) = t_{r_2}(m_{r_1}(n_{e_1}))$	$t_{\text{stock exchange}}(m_{\text{developer}}(n_{\text{Assassin's Creed: Lost Legacy}}))$ = "The developer of the game 'Assassin's Creed: Lost Legacy' is listed on a stock exchange named"
one-hop prompt	$\tau(r_2(e_2)) = t_{r_2}(n_{e_2})$	$t_{\text{stock exchange}}(n_{\text{Ubisoft}})$ = "Ubisoft is listed on a stock exchange named"

Table 2: Examples from TwoHopFact. The name of the bridge entity n_{e_2} is shown in brown font, and a descriptive mention of the bridge entity $\mu(r_1(e_1)))$ constructed with $m_{r_1}(n_{e_1})$ is shown in purple font.

We write one representative template for each m_{r_1} and t_{r_2} in a way that two-hop prompts are natural. Some examples of how the templates are used to construct the prompts are shown in Table 2. Afterward, we translate the collected fact triplets to pairs of two-hop prompts and one-hop prompts using the manually written templates. To represent entities in a string, we use the title of the entity's Wikidata page. We ensure that the generated prompts are grammatically correct. Table 3 shows the actual examples of the two-hop prompts and the bridge entity for each fact composition type.

Figure 6: Statistics of the dataset of TwoHopFact. The inner part shows the percentage of two-hop prompts with the type of descriptive mention of the bridge entity: "type(e_1)'s type(r_1)". The outer part shows the percentage of the two-hop prompts with the fact composition type: "type(r_2) of type(e_1)'s type(r_1)" (only type(r_2) of is shown as the annotation) in TwoHopFact. The expanded forms of the abbreviations used for the fact composition types are listed in Table 4.

B Dataset Statistics

TwoHopFact consists of 45,595 unique pairs of fact triplets $((e_1, r_1, e_2), (e_2, r_2, e_3))$ of 52 fact composition types, translated into 45,595 one/two-hop prompts. Figure 6 shows the distribution of the fact composition types. The distribution of the fact composition type is relatively balanced, with the type that has the largest portion covering only 7.41% of the dataset ("birth city of novel's author").

Figure 7a shows the percentage of the majority bridge entity e_2, i.e., e_2 that is utilized the most to construct the one-hop prompt that corresponds to each two-hop prompt. The highest percentage of majority bridge entity among all fact composition types is only 15%, showing that the dataset is not biased as favorable towards certain e_2. Figure 7b shows the percentage of majority e_3 that serve as the ground truth answer for the two-hop prompts. Table 3 shows the number of two-hop prompts for

Fact Composition Type	Two-Hop Prompt τ_{2H}	Bridge Entity e_2	e_3	Count	Percentage
actor of movie's mainchar	The main character of the movie Dream of the Red Chamber, Part 1 was played by an actor named	Lin Daiyu	Tao Huimin	73	0.16
anthem of capital's cntry	The national anthem of the country with Zagreb as its capital is named	Croatia	Lijepa naša domovino	204	0.45
anthem of president's cntry	The national anthem of the country led by president Lazarus Chakwera is named	Malawi	Mulungu dalitsa Malaŵi	50	0.11
author of mainchar's novel	The novel with 'Shere Khan' as the main character was written by an author named	The Jungle Book	Rudyard Kipling	308	0.68
birthcity of cntry's president	The president of South Korea was born in the city of	Moon Jae-in	Geoje	36	0.08
birthcity of novel's author	The author of the novel Hadrian the Seventh was born in the city of	Frederick Rolfe	London	3,379	7.41
birthcity of orgz's ceo	The CEO of Moderna was born in the city of	Stéphane Bancel	Marseille	189	0.41
birthcity of person's spouse	The spouse of Hiromi Suzuki was born in the city of	Koji Ito	Kobe	2,376	5.21
birthcity of song's singer	The singer of 'Rêver' was born in the city of	Mylène Farmer	Pierrefonds	1,453	3.19
birthcntry of cntry's president	The president of Somalia was born in the country of	Mohamed Abdullahi Mohamed	Somalia	36	0.08
birthcntry of novel's author	The author of the novel Christine was born in the country of	Stephen King	United States of America	3,358	7.36
birthcntry of orgz's ceo	The CEO of X was born in the country of	Parag Agrawal	India	189	0.41
birthcntry of person's spouse	The spouse of Vladimir Pyshnenko was born in the country of	Natalya Meshcheryakova	Russia	2,382	5.22
birthcntry of song's singer	The singer of 'Let's Get It In' was born in the country of	Lloyd	United States of America	1,434	3.15
capital of anthem's cntry	The capital of the country with the national anthem 'Fatshe leno la rona' is	Botswana	Gaborone	131	0.29
capital of president's cntry	The capital of the country led by president Ali Bongo Ondimba is	Gabon	Libreville	47	0.10
cntry of person's birthcity	The city where Aleksandăr Nikolov was born is in the country of	Tours	France	2,751	6.03
cntry of univ's hqcity	The city where the headquarters of Aichi Shukutoku University is located is in the country of	Nagakute	Japan	1,499	3.29
creator of novel's mainchar	The main character of the novel I Capture the Castle was created by	Cassandra Mortmain	Dodie Smith	141	0.31
director of mainchar's movie	The name of the director of the movie featuring Golden harp as the main character is	Mickey and the Beanstalk	Hamilton Luske	94	0.21
father of novel's author	The father of the author of the novel The Tale of Two Bad Mice is named	Beatrix Potter	Rupert William Potter	2,026	4.44
father of orgz's ceo	The father of the CEO of HarperCollins UK is named	Charles Redmayne	Richard Charles Tunstall Redmayne	49	0.11
father of person's spouse	The father of the spouse of Elsa Zylberstein is named	Nicolas Bedos	Guy Bedos	421	0.92
father of song's singer	The father of the singer of 'Étienne' is named	Guesch Patti	Jean Porrasse	602	1.32
founder of ceo's orgz	The organization led by CEO Vasily Levanov was founded by the person named	Visual Organization	Vasily Levanov	164	0.36
founder of person's uguniv	John Tien's undergrad university was founded by the person named	United States Military Academy	Thomas Jefferson	1,122	2.46
founder of vdgame's dev	The developer of the game 'Armour-Geddon' was founded by the person named	SCE Studio Liverpool	Ian Hetherington	3,503	7.68
hqcity of ceo's orgz	The organization led by CEO John Perry has its headquarters in the city of	Bluefin Payment Systems LLC	Atlanta	306	0.67
hqcity of founder's dev	The company founded by Stephen B. Streater has its headquarters in the city of	Eidos Interactive	London	406	0.89
hqcity of founder's univ	The university founded by John Wilson has its headquarters in the city of	University of Mumbai	Mumbai	93	0.20
hqcity of person's uguniv	Retta's undergrad university has its headquarters in the city of	Duke University	Durham	1,811	3.97
hqcity of vdgame's dev	The developer of the game 'The House of Da Vinci' has its headquarters in the city of	Blue Brain Games	Bratislava	2,310	5.07
hqcntry of ceo's orgz	The organization led by CEO Ties Carlier has its headquarters in the country of	VanMoof	Netherlands	525	1.15
hqcntry of founder's dev	The company founded by Anne-Laure Fanise has its headquarters in the country of	DigixArt	France	537	1.18
hqcntry of founder's univ	The university founded by Joseph Chamberlain has its headquarters in the country of	University of Birmingham	United Kingdom	94	0.21
hqcntry of person's uguniv	D. L. Waidelich's undergrad university has its headquarters in the country of	Lehigh University	United States of America	1,815	3.98
hqcntry of vdgame's dev	The developer of the game 'Terroir' has its headquarters in the country of	General Interactive Co.	Singapore	3,761	8.25
mother of novel's author	The mother of the author of the novel The Heat of the Day is named	Elizabeth Bowen	Florence Isabella Pomeroy Colley	1,443	3.16
mother of person's spouse	The mother of the spouse of Malaika Arora is named	Arjun Kapoor	Mona Shourie Kapoor	238	0.52
mother of song's singer	The mother of the singer of 'I Wanna Be Down' is named	Brandy	Sonja Norwood	533	1.17
origcntry of mainchar's movie	The movie featuring Juliane Klein as the main character was released in the country of	Marianne and Juliane	Germany	102	0.22
president of anthem's cntry	The country with the national anthem 'Azat u ankakh Artsakh' is led by president	Republic of Artsakh	Arayik Harutyunyan	38	0.08
president of capital's cntry	The country with Warsaw as its capital is led by president	Poland	Andrzej Duda	55	0.12
spouse of cntry's president	The spouse of the president of Ivory Coast is named	Alassane Ouattara	Dominique Folloroux-Ouattara	33	0.07
spouse of novel's author	The spouse of the author of the novel The Train Was on Time is named	Heinrich Böll	Annemarie Böll	1,597	3.50
spouse of orgz's ceo	The spouse of the CEO of Tethys is named	Jean-Pierre Meyers	Françoise Bettencourt Meyers	31	0.07
spouse of song's singer	The spouse of the singer of 'Last Night' is named	Snoop Dogg	Shante	407	0.89
stockexch of ceo's orgz	The organization led by CEO Luis von Ahn is listed on a stock exchange named	Duolingo	Nasdaq	74	0.16
stockexch of founder's dev	The company founded by Hae-Jin Lee is listed on a stock exchange named	Naver Corporation	Korean Stock Exchange	48	0.11
stockexch of vdgame's dev	The developer of the game 'Strider' is listed on a stock exchange named	Capcom	Tokyo Stock Exchange	946	2.07
ugmajor of novel's author	In college, the author of the novel The Masks of God majored in	Joseph Campbell	English literature	92	0.20
uguniv of novel's author	As an undergrad, the author of the novel Aiiieeeee! An Anthology of Asian-American Writers attended the university named	Shawn Wong	University of California, Berkeley	283	0.62

Table 3: Count of two-hop prompts for each fact composition type with examples. The text in purple indicates the descriptive mention $\mu(r_1(e_1))$ of the bridge entity. One-hop prompts τ_{1H} are constructed by replacing the descriptive mention with the bridge entity's name. The expanded forms of the abbreviations used for the fact composition types are listed in Table 4.

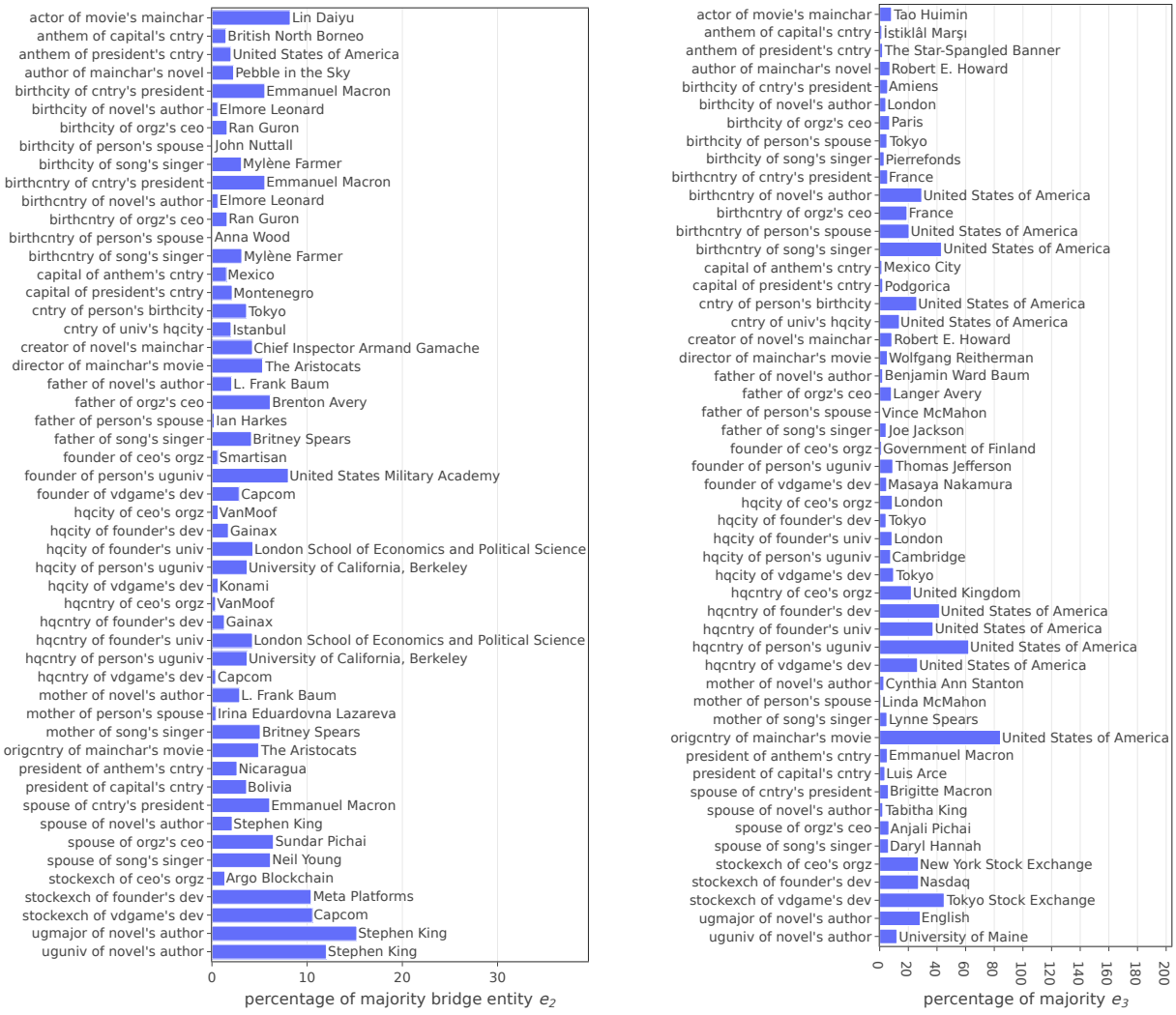
Abbreviation	Full Term
hq	headquarters
ug	undergrad
orig	origin
univ	university
stockexch	stock exchange
orgz	organization
mainchar	main character
vdgame	videogame
cntry	country
dev	developer

Table 4: Abbreviations used for the fact composition types.

each fact composition type with examples. We ensure that the number of prompts for a fact composition type exceeds at least 30 for statistically significant results.

C Justification of Internal Entity Recall Score: Appositive Generation Experiment

Experiment We demonstrate that ENTREC is a reasonable approximation of the internal recall of the bridge entity with indirect evidence. Note that $\text{ENTREC}^l(e_2, \tau_{2H})$ is calculated not at the last token of τ_{2H} but at the last token of the bridge entity's descriptive mention, where it is grammatically natural to prepend a comma followed by the name of e_2 (e.g., "The mother of the singer of 'Superstition', *Stevie Wonder*"). In the resulting string, grammatically $\mu(r_1(e_1))$ becomes the *antecedent* and e_2 becomes the *appositive*; an appositive is a noun phrase that follows another noun phrase in opposition to it and provides information that further identifies or defines it, and the antecedent is the noun phrase that the appositive describes. Then, if $\text{ENTREC}^l(e_2, \tau_{2H})$ reasonably approximates the internal recall of the bridge entity e_2, it is expected



(a) Percentage of majority bridge entity e_2

(b) Percentage of majority e_3

Figure 7: Percentage of the most frequent entities for each fact composition type of TwoHopFact. The expanded forms of the abbreviations used for the fact composition types are listed in Table 4.

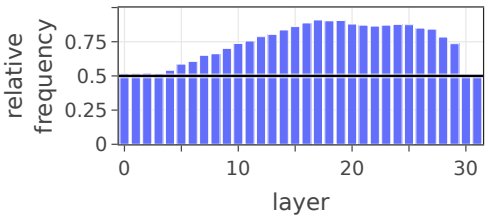


Figure 8: The relative frequency of the cases where increasing the entity recall score at a layer increases the probability of the model to output $e_2^{(0)}$ as the next token of a comma following the prefix of τ_{2H} ending at the descriptive mention (*"The mother of the singer of 'Superstition',"*), for LLaMA-2 7B.

that *there will be at least some layers l where increasing* $\text{EntRec}^l(e_2, \tau_{2H})$ *increases the relative frequency of the LLM to generate $e_2^{(0)}$ with a relative frequency higher than random chance*. In other words, we check the relative frequency of the cases where increasing the entity recall score at a layer increases the probability of the model to output $e_2^{(0)}$ as the next token of a comma following the prefix of τ_{2H} ending at the descriptive mention (*"The mother of the singer of 'Superstition',"*). We calculate this relative frequency as described in Section 6.2 but using the probability instead of CnstScore.

Result Figure 8 demonstrates that, in most of the mid-late layers, increasing the latent recall of the bridge entity when the LLM processes $\mu(r_1(e_1))$ also increases the relative frequency of the LLM to output $e_2^{(0)}$ to generate the appositive of $\mu(r_1(e_1))$ followed by a comma.[6] The result indicates that

[6]For this analysis, we exclude the cases where the descriptive mention ends with one of the following: ?', .', !', ,', ",)', "', where appending a comma introduces changes in the tokenization results for LLaMA-2.

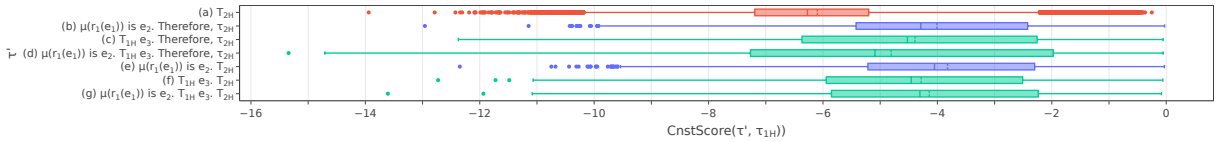


Figure 9: Distribution of CNSTSCORE calculated for different styles of prompts τ' for LLaMA-2 7B.

ENTREC at the n-th token has controllability of the token to be generated as the $n+2$-th token to make it more likely to be the first token of the appositive, serving as an indirect evidence that $\text{ENTREC}^l(e_2, \tau_{2H})$ is a reasonable proxy of the internal recall of the bridge entity.

D Justification of Consistency Score: Comparative Experiment with Chain-of-Thought Cases

Experiment We demonstrate that the proposed definition of $\text{CNSTSCORE}(\tau_{2H}, \tau_{1H})$ is a reasonable proxy of the utilization of what the LLM knows about the bridge entity's attribute – the latent recall of its answer to τ_{1H} – with indirect evidence. If the information to reason with is given as part of the input, e.g., if the given prompt is *"The singer of 'Superstition' is Stevie Wonder. The mother of Stevie Wonder is named Lula. The mother of the singer of 'Superstition' is"*, the LLM would not need to internally perform the multi-hop reasoning to refer to what its output to the one-hop prompt *"The mother of Stevie Wonder is"* is, but just copy the answer from the input. Therefore, CNSTSCORE of such a case will be lower than the case where the LLM needs to internally figure out what its answer to the one-hop prompt given the hint of who the descriptive mention refers to, e.g., *"The singer of 'Superstition' is Stevie Wonder. The mother of the singer of 'Superstition' is"*. Therefore, to check whether this is the case, we compare CNSTSCORE computed with the several Chain-of-Thought (CoT) style prompts τ', i.e., $\text{CNSTSCORE}(\tau', \tau_{1H})$.

Result Figure 9 shows the distribution of CNSTSCORE computed with different styles of prompts τ' as written in the y-axis. The red case is the consistency score of the two-hop prompt that we mainly study in our work, which requires full multi-hop reasoning. Because no information to reason from is given in the input, CNSTSCORE is significantly lower than the cases of other CoT-style prompts. The blue case is where what the descriptive mention refers to is given as the input, but what the LLM knows about the bridge entity's attribute needs to be internally recalled and referred to. The green cases are where the bridge entity's attribute, i.e., the answer to the prompt, is explicitly given in the input, and thus, the LLM does not need to refer to its answer to the one-hop prompt. The result demonstrates that the mean of CNSTSCORE is higher for the blue cases where the model is forced to refer to its answer to the one-hop prompt than in the green cases where the model does not need to refer to the answer. The difference between the red and the blue cases would have come from the existence of the information of the descriptive mention's identity in the input prompt, which would have helped the LLM to use the connection to refer to what it knows about the bridge entity.

E Technical Details

We modify the codebase of Nanda and Bloom (2022) to run the experiments before refactoring. We use 1-8 40GB A100 GPUs for the experiments. All experiments run in less than 24 hours. We use the model weights from HuggingFace Transformers (Wolf et al., 2020) and use full precision for LLaMA-2 7B and 13B and half-precision for 70B. The SPARQL queries for querying Wikidata are written with the help of GPT-4 (OpenAI et al., 2023).

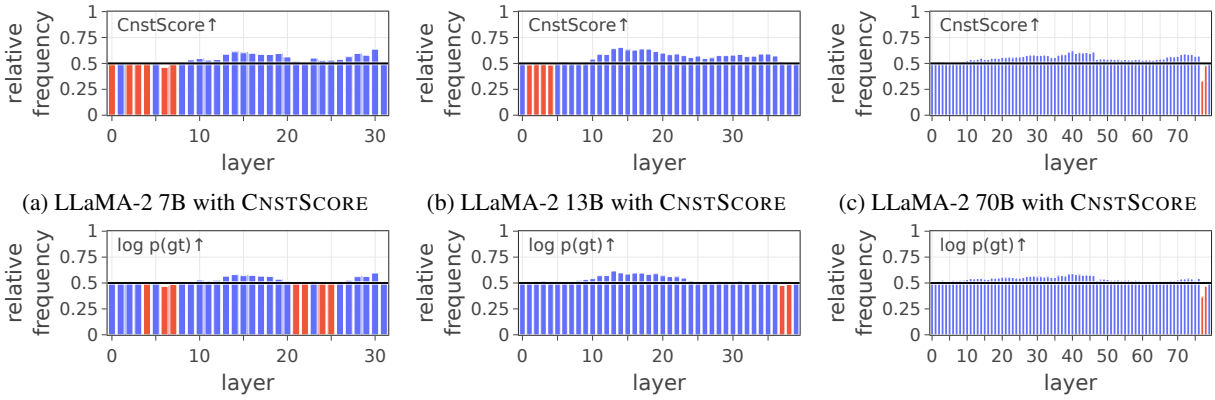
F Accuracy-based Analysis for the Second Hop of Multi-Hop Reasoning

We perform consistency-based analysis instead of an accuracy-based analysis because solely relying on the answer correctness has limitations in answering our research questions, as explained in Section 6.1. For further analysis, we present accuracy-based results in this section.

Using log probability of the ground truth answer instead of the consistency score does not affect our findings. We perform the RQ2 experiment described in Section 6.2 not with CNSTSCORE, but with the output log probability of the first token of the ground truth answer, e.g., *"Lula"* for the two-hop prompt *"The mother of the singer*

Descriptive Mention Type	0	1	2	3
novel's author	a critic of n_{e_1}	the filmmaker of n_{e_1}	the main character of n_{e_1}	a fan of n_{e_1}
person's birth city	the city where n_{e_1} never visited	the city where n_{e_1} is abandoned	the city where n_{e_1} is banned	the city where n_{e_1} never lived in
orgz's ceo	the COO of n_{e_1}	the rival of n_{e_1}	the CTO of n_{e_1}	the CFO of n_{e_1}
capital's cntry	the country which does not have n_{e_1} as its city	the country which does not have n_{e_1} as its capital	the country which does not have n_{e_1} as its largest city	
president's cntry	the country where n_{e_1} is not the president	the country where n_{e_1} is not the head of state	the country where n_{e_1} is a rival of the president	the country where n_{e_1} is a critic of the president
anthem's cntry	the country which does not have n_{e_1} as its anthem	the country which banned singing n_{e_1}	the country where n_{e_1} is blacklisted	the country where n_{e_1} is banned
vdgame's dev	a competitor of n_{e_1}	a plagiarist of n_{e_1}	a critic of n_{e_1}	a rival of n_{e_1}
founder's dev	the company n_{e_1} criticizes	a critic of n_{e_1}	a competitor to n_{e_1}	the company n_{e_1} is a rival of
univ's hqcity	the city where n_{e_1} is not located	the city where n_{e_1} is not headquartered	the city where n_{e_1} is not founded	the city where n_{e_1} is not established
movie's mainchar	the antagonist in n_{e_1}	a sidekick in n_{e_1}	an extra in n_{e_1}	a critic of n_{e_1}
novel's mainchar	the antagonist in n_{e_1}	a sidekick in n_{e_1}	an extra in n_{e_1}	a critic of n_{e_1}
mainchar's novel	the novel where n_{e_1} is not the main character	the novel where n_{e_1} does not appear	the novel where n_{e_1} is not the protagonist	the novel where n_{e_1} is not the antagonist
mainchar's movie	the movie where n_{e_1} is not the main character	the movie where n_{e_1} does not appear	the movie where n_{e_1} is not the protagonist	the movie where n_{e_1} is not the antagonist
ceo's orgz	the company n_{e_1} criticizes	a critic of n_{e_1}	a competitor to n_{e_1}	the company n_{e_1} is a rival of
cntry's president	a critic of n_{e_1}	a protester against n_{e_1}	a rival of n_{e_1}	a competitor to n_{e_1}
song's singer	a critic of n_{e_1}	a singer covering n_{e_1} without permission	a plagiarist of n_{e_1}	a rival of n_{e_1}
person's spouse	the father of n_{e_1}	the mother of n_{e_1}	a child of n_{e_1}	a sibling of n_{e_1}
person's uguniv	the university where the application of n_{e_1} was rejected	the university where n_{e_1} never went to	the university where n_{e_1} was not accepted	the university where n_{e_1} was not admitted
founder's univ	the university where n_{e_1} graduated from	the alma mater of n_{e_1}	the university where n_{e_1} was admitted to	the university where n_{e_1} was accepted to

Table 5: Candidate templates of r'_1 for each type of descriptive mention of the bridge entity. The expanded forms of the abbreviations used for the fact composition types are listed in Table 4.



(a) LLaMA-2 7B with CNSTSCORE (b) LLaMA-2 13B with CNSTSCORE (c) LLaMA-2 70B with CNSTSCORE

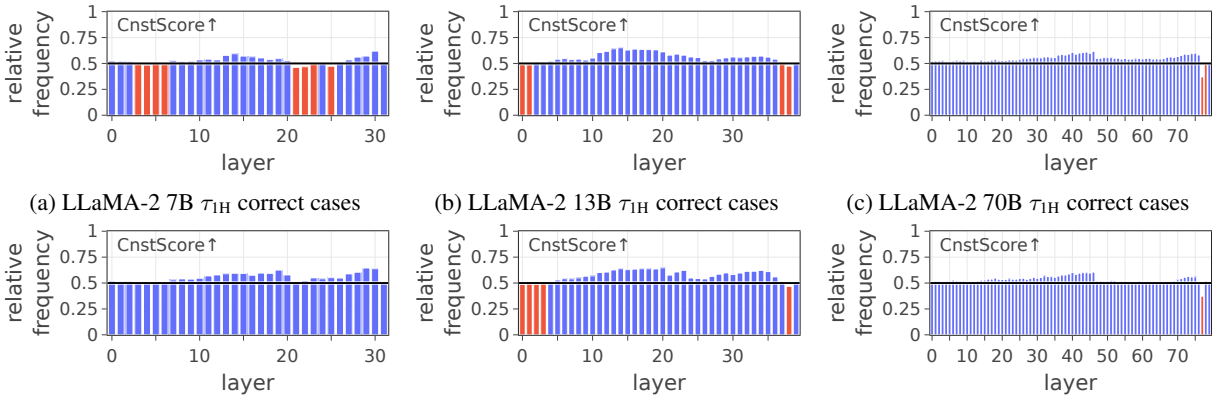
(d) LLaMA-2 7B with log probability (e) LLaMA-2 13B with log probability (f) LLaMA-2 70B with log probability

Figure 10: Relative frequency that stronger recall of the bridge entity at the l-th layer increases the consistency score (top row) or the log probability of the first token of the ground truth answer (bottom row) of the LLM. The relative frequency is calculated only for the cases where the one-hop prompt is completed with the ground truth answer. Bars are colored blue if the relative frequency is greater than or equal to 0.5 and red otherwise. We manually set the value of 0.5 at the last layer because the intervention does not affect the consistency at that layer.

of 'Superstition' is". We measure the relative frequency using the 17,231/45,595 two-hop prompts of which the corresponding one-hop prompts, e.g., *"The mother of Stevie Wonder is"*, are completed with the ground truth answer by all of the 7B, 13B, and 70B models.

Figure 10 shows the results, where the top row contains the relative frequency with CNSTSCORE and the bottom row contains the relative frequency with the log probability of the ground truth answer. When scaling from 7B to 13B and 70B, the maximum relative frequency with CNSTSCORE is 0.64 in layer 30 (7B), 0.65 in layer 14 (13B), and 0.62 in layer 40 (70B). The maximum relative frequency with the log probability of the ground truth answer is 0.60 in layer 30 (7B), 0.62 in layer 13 (13B), and 0.59 in layer 40 (70B). While the values are slightly lower for the log probability than those for the consistency score, the overall trends are alike. Also, as also observed in Section 6.3 with consistency, the second-hop reasoning with the log probability does not strengthen with increasing model size.

Filtering based on the accuracy of the one-hop prompt does not affect our findings. We test whether the second hop of the latent multi-hop reasoning is stronger for the two-hop prompts of which the corresponding one-hop prompts are completed correctly with one of the ground truth answer candidates. For this analysis, we filter the two-hop prompts into two sets: those where the corresponding one-hop prompts are correctly completed for all model scales (τ_{1H} correct) and those where the corresponding one-hop prompt is not completed with any of the ground truth answer candidates for all model scales (τ_{1H} incorrect). Since the trend of the relative frequency significantly varies for different fact composition types (Figure 4), for fair comparison, we make the distribution of the fact composition types of the two sets become the same by sampling. This results in two sets of 9,734 two-hop prompts with the same distribution of fact com-



(a) LLaMA-2 7B τ_{1H} correct cases (b) LLaMA-2 13B τ_{1H} correct cases (c) LLaMA-2 70B τ_{1H} correct cases

(d) LLaMA-2 7B τ_{1H} incorrect cases (e) LLaMA-2 13B τ_{1H} incorrect cases (f) LLaMA-2 70B τ_{1H} correct cases

Figure 11: Relative frequency that stronger recall of the bridge entity at the l-th layer increases the consistency of the LLM, compared between the cases where the one-hop prompt is correctly completed with one of the ground truth answer candidates (τ_{1H} correct; top row) and not (τ_{1H} incorrect; bottom row). Bars are colored blue if the relative frequency is greater than or equal to 0.5 and red otherwise. We manually set the value of 0.5 at the last layer because the intervention does not affect the consistency at that layer.

position types.

Figure 11 shows that the overall trends are similar for the two sets, τ_{1H} correct (top row) and τ_{1H} incorrect (bottom row), and that the maximum relative frequency of the two sets do not differ much across different model sizes. When scaling from 7B to 13B and 70B, the maximum relative frequency for the τ_{1H} correct set is 0.62 in layer 30 (7B), 0.65 in layer 14 (13B), and 0.62 in layer 46 (70B). The maximum relative frequency for the τ_{1H} incorrect set is 0.65 in layer 29 (7B), 0.65 in layer 20 (13B), and 0.61 in layer 46 (70B).