# Do Large Language Models Perform Latent Multi-Hop Reasoning *without* Exploiting Shortcuts?

**Anonymous ACL submission**

## Abstract

We evaluate how well Large Language Models (LLMs) latently recall and compose facts to answer multi-hop queries like *"In the year Scarlett Johansson was born, the Summer Olympics were hosted in the country of"*. One major challenge in such evaluation is that LLMs may have developed shortcuts by encountering the head entity *"Scarlett Johansson"* and the answer entity *"United States"* in the same training sequences or merely guess the answer based on frequency-based priors. To prevent shortcuts, we exclude test queries where the head and answer entities might have co-appeared during training. Through careful selection of relations and facts and systematic removal of cases where models might guess answers or exploit partial matches, we construct an evaluation dataset SOCRATES (SHORTCUT-FREE LATENT REASONING). We observe that LLMs demonstrate promising latent multi-hop reasoning abilities without exploiting shortcuts, but only for certain types of queries. For queries requiring latent recall of countries as the intermediate answer, the best models achieve 80% latent composability, but this drops to just 5% for the recall of years. Comparisons with Chain-of-Thought highlight a significant gap between the ability of models to reason latently versus explicitly. Analysis reveals that latent representations of the intermediate answer are constructed more often in queries with higher latent composability, and shows the emergence of latent multi-hop reasoning during pretraining.

## 1 Introduction

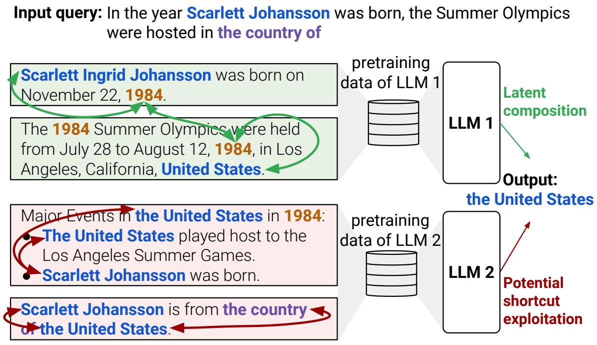

**Input query:** In the year Scarlett Johansson was born, the Summer Olympics were hosted in the country of
Scarlett Ingrid Johansson was born on November 22, 1984.
The 1984 Summer Olympics were held from July 28 to August 12, 1984, in Los Angeles, California, United States.
pretraining data of LLM 1
LLM 1
Latent composition
Output: the United States
Major Events in the United States in 1984: The United States played host to the Los Angeles Summer Games. Scarlett Johansson was born.
Scarlett Johansson is from the country of the United States.
pretraining data of LLM 2
LLM 2
Potential shortcut exploitation

Figure 1: Evaluation of latent multi-hop reasoning should exclude cases where LLMs can bypass the process of latently composing the single-hop facts by exploiting shortcuts. LLMs can develop shortcuts when they frequently encounter the head entity (*"Scarlett Johansson"*) or the relation pattern in the query (*"the country of"*) with the answer entity (*"United States"*). We propose desiderata for shortcut-free evaluation of latent multi-hop reasoning ability.

Latent multi-hop reasoning in Large Language Models (LLMs), or latently recalling and composing single-hop facts to answer multi-hop queries like *"In the year Scarlett Johansson was born, the Summer Olympics were hosted in the country of"*, has been of growing interest in recent years. First, this ability can be a measure towards better *localization* and *controllability* of factual knowledge in LLMs, as it can signal learning of a compressed representation of facts and their latent composition (Yang et al., 2024b). This would provide more hope towards locate-then-edit or unlearn paradigm of LLMs (Meng et al., 2022; Hong et al., 2024). For instance, if complex facts are redundantly learned and recalled, edits with only single-hop facts would not propagate to the relevant multi-hop facts (Onoe et al., 2023; Zhong et al., 2023; Cohen et al., 2024; Ju et al., 2024). In addition, the ability to provide accurate answers without explicit Chain-of-Thought (CoT) generation (Kojima and Gu, 2022) could reduce inference costs. At the same time, whether LLMs can spontaneously develop latent reasoning abilities during pretraining is of interest from a safety perspective, as latent reasoning is less visible and hard to monitor given the opaque computations in LLMs (Berglund et al., 2023; Treutlein et al., 2024; Chan et al., 2024). Taken together, these incentives raise the question of *How well do today's widely-used LLMs perform latent multi-hop reasoning over factual knowledge?*

We evaluate latent multi-hop reasoning abilities by assessing models' performance in answering multi-hop queries. While prior works have suggested that pretrained LLMs develop this ability (Ofir Press et al., 2023; Yang et al., 2024b; Biran et al., 2024; Li et al., 2024), they have not adequately addressed the possibility of models exploiting *shortcuts* (Elazar et al., 2022; Xu et al., 2022; Tang et al., 2023; Kang and Choi, 2023; Ju et al., 2024). Shortcuts from frequent co-occurrences of subject-object or relation-object in the training data can allow models to answer the multi-hop queries correctly without going through a true latent reasoning process. For instance, in a query about *"Scarlett Johansson"* (i.e., the *head entity*) where the answer entity is *"United States"*, LLMs may simply learn a subject-object entity shortcut if these entities frequently co-occur in training (Elazar et al., 2022; Kang and Choi, 2023; Zhang et al., 2024b; Ju et al., 2024). Similarly, LLMs can develop a relation-object shortcut from the frequency of *"United States"* appearing as a country and guess the answer based on the frequency-based prior (Elazar et al., 2022; Xu et al., 2022; Tang et al., 2023).

To overcome this challenge, we outline desiderata for shortcut-free evaluation of latent multi-hop reasoning in LLMs, which we address through dataset construction and evaluation procedure. For the dataset, we only use test queries where the head and answer entities are estimated to never co-occur in pretraining sequences, thus minimizing subject-object shortcuts. We further minimize shortcuts by carefully selecting relation types and removing queries where the answers are easy to guess from a substring of the head entity. For evaluation, we measure *latent composability* as the rate of correct multi-hop answers when single-hop facts are known, without explicitly generating the intermediate answer (i.e., the *bridge entity*). Additionally, we reduce relation-object shortcuts by excluding queries where the model may guess the answer without considering the head entity.

The main challenge in satisfying our desiderata is that most LLMs' pretraining data is inaccessible, making it impossible to directly check entity co-occurrences. To tackle this, we use a *proxy corpus* of roughly 4.8B unique documents by utilizing six publicly available training corpora, selecting only test queries where the head and answer entities never co-occur. This approximation's effectiveness is validated by showing that strong latent composability for specific query types persists even when extending our entity co-occurrence check to the whole web via Google Search. Our resulting dataset, SOCRATES (SHORTCUT-FREE LATENT REASONING), consists of 7,232 pairs of single-hop and multi-hop queries of 17 types of relation compositions with 4 types of bridge entities. Comparative experiments with a dataset constructed from the same data distribution but without careful fact selection, co-occurrence-based filtering, and rigorous evaluation show that latent composability can be overestimated without satisfying the desiderata.

Our results for 41 LLMs from 9 families reveal that there are successful cases of latent multi-hop reasoning, but the performance varies substantially according to the type of bridge entity that connects the facts. Notably, state-of-the-art models demonstrate strong latent composability of over 80% when the bridge entity is a country. However, the number is only around 6% for year-based queries, highlighting the importance of considering the distribution of relation composition types when evaluating LLMs' latent reasoning abilities. Models that know more single-hop facts tend to reason better latently, and the ability marginally improves with model scale. On the contrary, CoT composability effectively increases with the number of known facts and model size, with much higher and consistent performance across bridge entity types. Additional analysis shows that the latent representation of the bridge entity is clearly constructed more often for queries with higher latent composability, and reveals the emergence of latent multi-hop reasoning during pretraining.

In summary, our contributions are as follows:

- We present SOCRATES and evaluation procedure for latent multi-hop reasoning with minimal risk of shortcut exploitation, which is corroborated to be important through a comparative analysis.
- We show that latent composability in LLMs significantly varies according to the bridge entity type.
- We show that latent reasoning *marginally* improves with the number of known single-hop facts and model scale and identify a significant gap between latent and CoT composability.
- We present additional analysis results that help better understand LLMs' mechanisms for latent multi-hop reasoning.

## 2 Related Work

Studies have shown that LLMs' predictions often rely on shortcuts, shallow heuristics, and co-occurrence biases (Chen and Durrett, 2019; Jiang and Bansal, 2019; Geirhos et al., 2020; Elazar et al., 2022; Zhang et al., 2022a; Xu et al., 2022; Liu et al., 2023; Tang et al., 2023; Kang and Choi, 2023; Bachmann and Nagarajan, 2024; Ju et al., 2024). For instance, Elazar et al. (2022) have found that single-hop knowledge predictions can be influenced by subject-object co-occurrences or relation-object co-occurrences. Similarly, Kang and Choi (2023) and Zhang et al. (2024b) show that frequent co-occurrences can lead LLMs to favor high-frequency words over correct responses. Lastly, Ju et al. (2024) demonstrate that head-answer entity co-occurrence frequencies in multi-hop facts are correlated with factual shortcuts, which can cause failures in multi-hop knowledge editing.

Prior works on latent factual multi-hop reasoning have not fully addressed potential shortcuts (Ofir Press et al., 2023; Yang et al., 2024b; Biran et al., 2024; Li et al., 2024). While Ofir Press et al. (2023) attempt to create multi-hop questions unlikely to appear in training, their approach relies on assumptions rather than co-occurrence statistics (leaving shortcuts from subject-object co-occurrences exploitable) and does not address shortcuts from relation-object co-occurrences. Similarly, Biran et al. (2024) address relation-object shortcuts but overlook subject-object shortcuts.

Construction of a shortcut-free evaluation dataset of latent factual multi-hop reasoning ability of *any pretrained LLM* presents a unique challenge, as the pretraining data of most of the widely used LLMs is not accessible, making it difficult to verify if certain single-hop facts or their composition appeared in the same training sequence. Our need to consider the knowledge LLMs learn during pretraining makes our work distinct from prior works that aim for shortcut-free evaluation on the tasks where the training dataset is fully accessible (Min et al., 2019; Chen and Durrett, 2019; Ho et al., 2020; Trivedi et al., 2022; Gregucci et al., 2024).

Other studies have attempted to circumvent these issues by fine-tuning LLMs on synthetic or counterfactual tasks to control for single-hop knowledge (Jiang et al., 2022; Kassner et al., 2020; Allen-Zhu and Li, 2023; Saparov et al., 2023; Hou et al., 2023; Berglund et al., 2023; Petty et al., 2024; Treutlein et al., 2024; Wang et al., 2024). However, these studies do not address our target question of how much latent multi-hop reasoning ability *naturally* emerges in training. Moreover, finetuning may introduce side effects, such as hallucinations, reduced knowledge learning, and utilization efficiency (Yin et al., 2023; Kang et al., 2024; Ghosal et al., 2024; Gekhman et al., 2024; Gottesman and Geva, 2024). Works on latent compositional reasoning with algorithmic or mathematical tasks (Dziri et al., 2023; Chen et al., 2023; Deng et al., 2024) do not address our target question of latent multi-hop reasoning ability with factual knowledge, and may still suffer from data contamination that inflates performance (Zhang et al., 2024a). Ko et al. (2024) examine performance gaps between different numbers of reasoning hops but focus on more general reasoning capabilities rather than latent reasoning with factual knowledge.

Peng et al. (2024) study the theoretical limitations of compositional abilities. Consistent with our findings, they prove that for high arity relations (like relations with year-type bridge entity in our work), multi-hop reasoning is more difficult, albeit for the specific case of Transformers (Vaswani et al., 2017) with a single layer.

## 3 Shortcut-Free Evaluation of Latent Multi-Hop Reasoning

**Terms and Notations** We represent single-hop facts as $r_1(e_1) = e_2$ and $r_2(e_2) = e_3$, and multi-hop facts as their composition $r_2 \circ r_1(e_1) = e_3$. In the aforementioned example, the entities *"Scarlett Johansson"*, *"1984"*, and *"United States"* are respectively represented by $e_1$, $e_2$, $e_3$, and connected via relations $r_1$ (person-birthyear), $r_2$ (year-eventcountry), and $r_2 \circ r_1$ (person-birthyear-eventcountry). The set of aliases (names that the entity is also known as), of $e_1$, $e_2$, and $e_3$ are represented as $E_1$, $E_2$, and $E_3$, respectively. The answer set of the single-hop queries $q(r_1(e_1))$, $q(r_2(e_2))$, and multi-hop query $q(r_2 \circ r_1(e_1))$ is $E_2$, $E_3$, and $E_3$, respectively. For instance, the answer set corresponding to the query *"The year Scarlett Johansson was born in is"* is $\{$*"1948"*$\}$. We call each tuple $(q(r_1(e_1)), q(r_2(e_2)), q(r_2 \circ r_1(e_1)), E_1, E_2, E_3)$ a *test case*, where $e_2$ is a *bridge entity* that connects the two facts, $e_1$ is the *head entity*, and $e_3$ the *answer* entity. Also, we call *"the year Scarlett Johansson was born"* the *descriptive mention* $\mu$ of the bridge entity.

**Desideratum 1: Latent multi-hop reasoning** We define the latent multi-hop reasoning ability of LLMs as the ability to latently recall and compose learned single-hop facts to answer multi-hop queries. We evaluate this ability by assessing a model's performance in answering multi-hop queries. For example, if a model learned the correct answer to *"The year Scarlett Johansson was born in is"* and *"In 1984, the Summer Olympics were hosted in the country of"*, we evaluate whether it recalls and composes these facts latently to correctly answer a multi-hop query like *"In the year Scarlett Johansson was born, the Summer Olympics were hosted in the country of"*. Therefore, to exclusively evaluate *latent* as opposed to *explicit* reasoning, we require models to answer the query directly without generating intermediate results (e.g., *"1984"*), e.g., without using Chain-of-Thought. Therefore, **the evaluation should exclude the cases where the model generates the intermediate answers before generating the final answer.**

**Desideratum 2: Shortcut-free** A model has a chance of exploiting shortcuts when it can correctly answer a multi-hop query by observing only part of the query (e.g., without $e_1$ or $r_1$ in the input). Shortcut exploitation is problematic for evaluating latent multi-hop reasoning abilities because it allows the model to bypass the need to latently recall and compose single-hop facts. Following Elazar et al. (2022), we consider two types of shortcuts: *subject-object shortcuts*, where the model predicts objects that frequently co-occur with certain subjects or substrings of subjects, regardless of the relation semantics, and *relation-object shortcuts*, where the model predicts objects that frequently appear with certain surface form text of a relation, regardless of the subject. Therefore, **the evaluation should exclude the queries prone to subject (or substring)-object shortcuts or relation (or paraphrase)-object shortcuts.**

## 4 Evaluation Dataset

In this section, we describe our dataset construction process that minimizes the chance of subject-object shortcut (satisfying Desideratum 2), resulting SOCRATES (SHORTCUT-FREE LATENT REASONING), a dataset for evaluation of shortcut-free evaluation of latent multi-hop reasoning.

### 4.1 Dataset Construction

We generate test cases from a knowledge graph $G$, where facts are represented as subject-relation-object triplets $\langle s, r, o\rangle$. Specifically, we collect pairs of facts $r_1(e_1) = e_2$ and $r_2(e_2) = e_3$ and their composition $r_2 \circ r_1(e_1) = e_3$ by considering pairs of triplets with a shared bridge entity, i.e., $\langle e_1, r_1, e_2\rangle$, $\langle e_2, r_2, e_3\rangle$. We choose Wikidata (Vrandečić and Krötzsch, 2014) as $G$.

**Step 1: Selection of fact pairs** We select single-hop facts that are likely well-known, but their composition is unlikely to naturally appear in general text corpora, to minimize the change of the model developing a shortcut between $e_1$ and $e_3$. We observe that such cases typically occur when the set of possible options for $e_2$ is large, there are numerous $e_1$'s that map to the same $e_2$, and the set of possible options for $e_3$ is not too small (e.g., not `person-bloodtype`). This should lower the chance of the LLM getting the answer correct by mere guessing (Desideratum 2).

We exclude the following cases: (a) relation compositions where $e_1$ and $e_3$ are likely to be directly associated, e.g., `novel-maincharacter-creator`, (b) facts where the head and answer entities can be directly connected via popular single-hop relations other than the tested multi-hop relation, e.g., `person-birthyear-eventcountry(Scarlett Johansson) = United States = person-birthcountry(Scarlett Johansson)`, (c) queries with trivially inferrable bridge entities, e.g., `university-locationcountry (University of Washington) = United States`, which could enable answer prediction via entity substring-based shortcuts, (d) relations where there are likely to be many entities with 1:n relation, such as `person-children`, and (e) single-hop facts with more than one answer (details in §A.1).

**Step 2: Test case construction** We convert the selected fact pairs into a set of test cases $\{(q(r_1(e_1)), q(r_2(e_2)), q(r_2 \circ r_1(e_1)), E_1, E_2, E_3)\}$. To create the two single-hop queries $q(r_1(e_1))$, $q(r_2(e_2))$ and their corresponding multi-hop query $q(r_2 \circ r_1(e_1))$, we follow the common practice of using diverse handcrafted natural language templates (Petroni et al., 2019; Yang et al., 2024b). For each relation, we use 16 templates (4 for each of the two single-hop queries) and randomly sample one template for each query, resulting in approximately 100K test cases. We construct the queries as incomplete sentences, instead of questions, so that the test query can be naturally completed by any pretrained model to derive the answer without finetuning.

**Step 3: Test case filtering using training co-occurrence statistics** We filter out cases where

| $e_2$ type | relation composition type | count | example multi-hop query |
|---|---|---|---|
| city | person-birthcity-eventyear | 33 | $e_1$'s birth city hosted the Eurovision Song Contest in the year |
| country | university-locationcountry-anthem | 101 | The country where $e_1$ is in has the national anthem named |
| | university-locationcountry-isocode | 30 | The ISO 3166-1 numeric code of the country where $e_1$ is located is |
| | university-locationcountry-year | 7 | The founding year of the location country of $e_1$ is |
| | person-birthcountry-anthem | 22 | The name of the national anthem of $e_1$'s country of birth is |
| | person-birthcountry-isocode | 6 | The ISO 3166-1 numeric code used for the country where $e_1$ was born is |
| university | person-undergraduniversity-founder | 33 | The person who founded $e_1$'s undergrad university is named |
| | person-undergraduniversity-year | 25 | The year when the university where $e_1$ studied as an undergrad was founded is |
| year | person-birthyear-winner | 4,484 | The winner of the Nobel Prize in Literature in $e_1$'s birth year was |
| | city-eventyear-winner | 2 | In the year when the G7 Summit were hosted in $e_1$, the Nobel Prize in Chemistry was awarded to |
| | university-inceptionyear-winner | 632 | In the founding year of $e_1$, the Nobel Prize in Literature was awarded to |
| | university-inceptionyear-hostleader | 9 | The person who was the host leader of the G7 Summit in the founding year of $e_1$ is |
| | university-inceptionyear-eventcountry | 13 | In the year $e_1$ was founded, the host country of the G7 Summit was |
| | university-inceptionyear-eventcity | 62 | In the founding year of $e_1$, the host city of the G7 Summit was |
| | person-birthyear-eventcity | 1,389 | In the birth year of $e_1$, the Winter Olympics were hosted in the city of |
| | person-birthyear-hostleader | 260 | The leader of the host of the G7 Summit in $e_1$'s birth year is |
| | person-birthyear-eventcountry | 124 | The country where the Eurovision Song Contest took place in the birth year of $e_1$ is |
| | | 7,232 | |

Table 1: Dataset statistics and example queries of SOCRATES. The head entities are replaced with $e_1$ to prevent potential data leakage. A more granular breakdown with the relation composition subtypes is in Appendix Table 3.

any aliases of the head entity $e_1$ and answer entity $e_3$ co-occur in the same sequence that the evaluated LLM has seen during pretraining, preventing subject-object shortcuts. However, since pretraining sequences and/or corpora of most LLMs are often inaccessible, we approximate co-occurrences by checking document-level co-occurrences across a *proxy corpus* of 4.8B unique documents (details in §A.2). While this approximation cannot guarantee complete exclusion of co-occurring entities without access to exact pretraining corpora, we validate our approach using Google Search for web-wide co-occurrence verification (§C.3).

**The SOCRATES Dataset** SOCRATES contains 7,232 test cases of 17 types of relation compositions connected by 4 types of bridge entities, as shown in Table 1. Note that the distribution of relation compositions is imbalanced as $e_1$ and $e_3$ of some relation compositions frequently appear together in the same document and most of test cases are removed by the co-occurrence-based filtering.

## 5 Evaluation Procedure

We introduce an evaluation procedure that satisfies part of Desideratum 2 by minimizing the chance of the model exploiting the relation-object shortcut and Desideratum 1 that the evaluation should exclude cases where the model performs explicit reasoning (§5.1). Then, we define our evaluation metric, latent composability (§5.2).

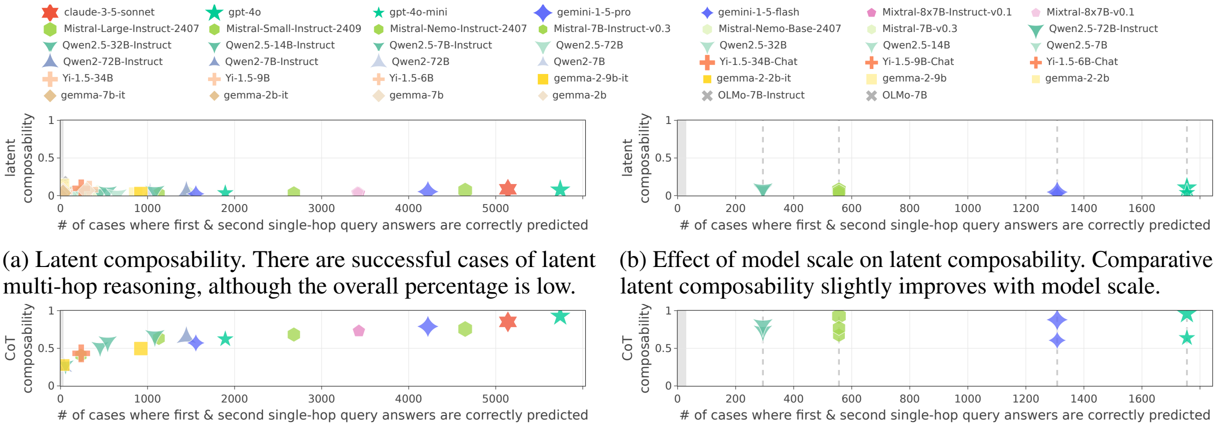
### 5.1 Filtering Guessable and Unusable Queries

**Excluding *guessable* cases** Even when the prediction of the model for a multi-hop query is correct, there is still a chance that the LLM might have guessed the answer correctly by chance, meaning that Desideratum 2 is not satisfied. For instance, when the answer is a popular entity among the potential answer set (e.g., *"United States"* as a country), the model might exploit a relation-object shortcut where the model makes the prediction based on the prior from the textual pattern of the relation (e.g., *". . . is from the country of"*).

To filter out the cases where it is indistinguishable whether models are guessing the answer, we adopt the method of Biran et al. (2024) which checks the model's prediction for a set of *ablated prompts* $Q_\emptyset = \{q(r_2 \circ r_1(\emptyset)), q(r_2(\emptyset))\}$, where the specific information of $e_1$ and $r_1(e_1)$ is ablated from the multi-hop query $q(r_2 \circ r_1(e_1))$, respectively (e.g., {*"In the year the person was born, the Summer Olympics were hosted in the country of"*, *"In the year, the Summer Olympics were hosted in the country of"*}). When the model answers the multi-hop query correctly, but also answers any of $q_\emptyset \in Q_\emptyset$ correctly, we exclude the test case from the evaluation. Namely, we detect and remove the cases where models may be exploiting a relation-object shortcut between $r_2 \circ r_1 / r_2$ and $e_3$.

**Excluding *unusable* cases** When the model correctly predicts the answer for a test multi-hop query but the LLM has just enumerated multiple potential answer candidates[1] or the LLM has performed an explicit reasoning (e.g., *"1984, United States"*), we view these test cases as *unusable* for the evaluation of latent multi-hop reasoning ability and exclude the test case from the evaluation, following Desideratum 1 (details in §B.1).

[1] *"A. United States B. Canada C. United Kingdom"*



(a) Latent composability. There are successful cases of latent multi-hop reasoning, although the overall percentage is low.

(b) Effect of model scale on latent composability. Comparative latent composability slightly improves with model scale.

(c) CoT composability. CoT composability is significantly higher than latent composability.

(d) Effect of model scale on CoT composability. Comparative CoT composability increases much more effectively with model scale compared to latent composability.

Figure 2: Latent **(upper row)** and CoT **(lower row)** composability on SOCRATES.

**Suppressing CoT for instruction-tuned LLMs** Since instruction-tuned LLMs tend to perform CoT-style reasoning by default[2], we formulate the task as a fill-in-the-blank task using a CoT-suppressing instruction as described in §C.2.[3]

### 5.2 Latent Composability

We assess *latent composability* as *the ability of the LLM to latently compose the already-learned single-hop facts* by calculating the ratio of the cases where the LLM correctly answers the multi-hop query while correctly answering both of the corresponding single-hop queries, excluding the *guessable* and *unusable* cases.To check the correctness of model outputs, we use a standard Exact Match with string normalization (details in §B.2).

When comparing latent composability between different models, it is misleading to compare latent composability calculated using different subsets of queries. Therefore, we calculate the ratio using the same test case subset where all of the compared LLMs correctly answer both single-hop queries where the test cases are *guessable* or *unusable* for neither of the models. We call this value *comparative* latent composability.

## 6 Experiments

We use SOCRATES to evaluate the latent multi-hop reasoning ability of LLMs. Our results show that models can perform latent reasoning without exploiting shortcuts, but their ability depends on the type of bridge entity connecting the two facts. Table 4 in the Appendix shows exemplifying test cases, model predictions, and results.

### 6.1 Experimental Setting

We assess 41 LLMs of different families and sizes. Among the proprietary LLMs, we evaluate Claude 3.5 Sonnet (Anthropic, 2024), GPT-4o (OpenAI, 2024b), GPT-4o mini (OpenAI, 2024a), and Gemini 1.5 Pro and Flash (Gemini Team et al., 2024). Among the open-source LLMs, we evaluate pre-trained and/or instruction-tuned models of 2B to 123B parameters from the model families of Mistral (Jiang et al., 2023), Mixtral (Jiang et al., 2024), Qwen 2.5 and 2 (Qwen Team, 2024; Yang et al., 2024a), Yi 1.5 (01.AI et al., 2024), Gemma 1 and 2 (Mesnard et al., 2024; Gemma Team et al., 2024), and OLMo (Groeneveld et al., 2024). Refer to §C.1 for model and inference details.

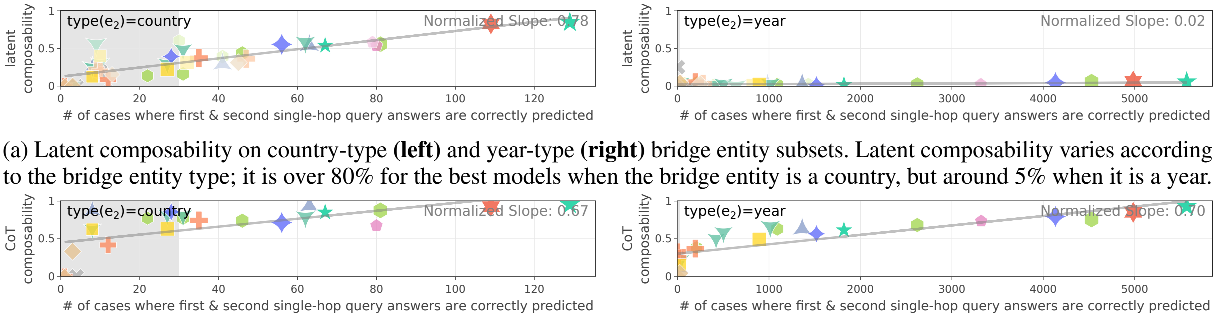
### 6.2 Evaluation Results on SOCRATES

**There are successful cases of latent multi-hop reasoning, although the overall percentage is low.** Figure 2a shows the latent composability of 41 models on SOCRATES. While there is a meaningful number of successful latent multi-hop reasoning cases (434 for Claude 3.5 Sonnet and 438 for GPT-4o, the best performing models), the percentage of such cases of the whole dataset is low; latent composability of Claude 3.5 Sonnet and GPT-4o is only 8.4% and 7.6%, respectively.

**Model scaling marginally improves overall performance.** Figure 2b shows *comparative* latent composability among models with different num-

[2] *"Scarlett Johansson was born in 1984, and the Summer Olympics that year were hosted by the United States."*

[3] While it is possible to use few-shot learning to restrict the format of the answer (Ofir Press et al., 2023), we use instructions to avoid potential biases in the selection of the few-shot demonstrations (Zhang et al., 2022b).



(a) Latent composability on country-type **(left)** and year-type **(right)** bridge entity subsets. Latent composability varies according to the bridge entity type; it is over 80% for the best models when the bridge entity is a country, but around 5% when it is a year.

(b) CoT composability on country-type **(left)** and year-type **(right)** bridge entity subsets. CoT composability does not fluctuate as dramatically as latent composability according to the type of bridge entity.

Figure 3: Latent **(upper row)** and CoT **(lower row)** composability measured for subsets of the test queries in SOCRATES, grouped according to the type of the bridge entity.

bers of parameters within the same model family[4]. We observe a consistent trend across all model families, where larger models answer more multi-hop queries correctly than smaller models, although the difference is not large in number. The gap in latent composability is 6.7% (118) and 2.4% (31) for GPT-4o vs. GPT-4o mini and Gemini 1.5 Pro vs. Gemini 1.5 Flash, respectively.

**Latent composability performance varies across bridge entity types.** Figure 3a shows the latent composability for two out of the four bridge entity types where the size of the subset of the queries is statistically significant (the results for all four types are in Appendix Figure 6). Notably, the latent composability of Claude 3.5 Sonnet and GPT-4o reaches 82.6% and 84.5%, respectively, when the single-hop facts are connected with country-type bridge entities, but only 6.7% and 5.7% when the facts are connected with year-type bridge entities. The rate of improvement in latent composability with the number of known single-hop facts also varies across different bridge entity types. Our finding implies that it is important to consider the dataset distribution and perform per-relation-composition analysis when evaluating latent multi-hop reasoning (explanation in §D.1). Drawing from prior works, we speculate that the high composability of country-related queries might stem from more frequent exposure to learning country-related facts together or in composition during pretraining (explanation in §D.2).

**There exist significant disparities between CoT and latent composability.** While GPT-4o achieves 92.8% composability with CoT reasoning, it is only 7.6% with latent reasoning (Figure 2c, 2a), with almost no cases where latent reasoning succeeds but CoT fails (Appendix Figure 8). Models that know more single-hop facts and larger models show dramatic improvements in composability for CoT reasoning compared to latent reasoning (Figures 2d, 2b), suggesting that merely increasing parameter count cannot effectively enhance latent multi-hop reasoning. Furthermore, CoT composability remains relatively consistent across bridge entity types (Figure 3b, 3a). Drawing from prior works, we speculate that the explicit generation of the bridge entity is the main factor behind high CoT composability (explanation in §D.3).

### 6.3 Additional Analysis

**Shortcut-free evaluation is important.** To check the importance of addressing shortcuts, we perform a comparative experiment with a shortcut-prone dataset and evaluation procedure that does not follow the proposed desiderata. Specifically, we construct a dataset with almost exactly the same distribution of the relation composition types (and thus the bridge entity types) of SOCRATES, but without applying any measure to remove potential shortcuts such as the entity co-occurrence-based filtering or relation-specific heuristics. As the evaluation procedure, we only check whether both of the single-hop facts are known by the model, and do not exclude the *guessable* and *unusable* cases.

Latent composability measured with shortcut-free data and evaluation procedure is three times lower than the shortcut-prone counterpart (Appendix Figure 10), and the former is consistently lower than the latter across all models (Appendix Figure 9). These results imply that overlooking

[4] GPT-4o vs. GPT-4o mini, Gemini 1.5 Pro vs. Gemini 1.5 Flash, Mistral Large (123B) vs. Small (22B) vs. Nemo (12B) Instruct, and Qwen 2.5 72B vs. 32B Instruct

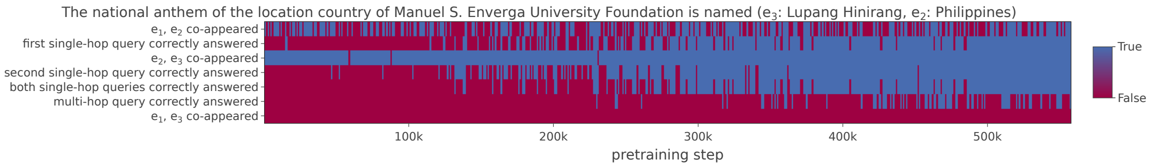


Figure 4: One successful case of OLMo 7B that demonstrates the emergence of latent multi-hop reasoning ability even when $e_1$ and $e_3$ have never co-appeared in any sequence throughout pretraining. While being pretrained for 557K steps, the model starts to correctly predict the answer to the single-hop queries after consistently seeing ($e_1$, $e_2$) and ($e_2$, $e_3$) together across multiple pretraining steps. After the model starts to learn to correctly answer the single-hop queries, the model starts to learn to correctly answer the multi-hop query.

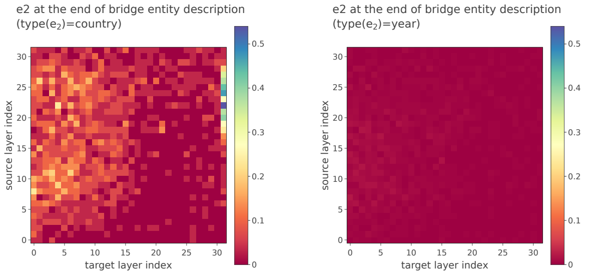


Figure 5: Experimental results with Mistral 7B v0.3 using Patchscopes (Ghandeharioun et al., 2024) to examine whether the model constructs latent representations of the bridge entity (e.g., *"1984"* for *"the year Scarlett Johansson was born"*) during the multi-hop query processing. Latent representations of bridge entities are constructed more often for queries with country-type bridge entities (that have higher latent composability).

shortcut exploitation can overestimate the latent composability of the model.

**Latent representation of the bridge entity appears more often for query types with higher latent composability.** We perform an experiment using Patchscopes (Ghandeharioun et al., 2024) following Biran et al. (2024) using Mistral 7B v0.3, which examines whether the model constructs latent representations of the bridge entity (e.g., *"1984"*) for its descriptive mention (*"the year Scarlett Johansson was born"*) encountered during multi-hop query processing. Figure 5 shows how often the hidden states at the last token of the descriptive mention taken from different layers (y-axis) generate the bridge entity when patched into appropriate contexts at different layers (x-axis), for the queries with country/year-type bridge entities where both single hop facts are known by the model. The bridge entity is generated more often (which suggests that the latent bridge entity representations are constructed more often) for the queries with higher latent composability (queries with country-type bridge entities). Details are in §C.4.

**Emergence of latent multi-hop reasoning during pretraining.** Our analysis of OLMo 7B's intermediate checkpoints (557 checkpoints from 1K to 557K pretraining steps) shows that for a subset of prompts, OLMO first learns to predict the single-hop answers, and then begins to correctly answer the respective multi-hop query. Figure 4 illustrates one such case. This set is small: 12 out of 13 cases where the model successfully performs multi-hop reasoning at a point, among 110 cases where the model is correct on both single-hop facts at some point and the model is not likely to be guessing the answer at any point during pretraining. That said, for OLMo, we have access to all the pretraining sequences in order and can hence guarantee that head entities have not been seen together with the answer entities in any of the pretraining sequences. Together with our other filtering to reduce the probability that answers can be correct by chance and guesswork, this indicates that even a small model like OLMo 7B can perform some level of latent reasoning, albeit only occasionally. More details of the experiment are provided in §C.5.

## 7 Conclusion

We outline desiderata for shortcut-free evaluation of LLMs' ability to latently recall and compose learned single-hop facts to answer multi-hop queries. By filtering entity co-occurrences and systematic removal of potential shortcuts, we construct the SOCRATES dataset, enabling a rigorous assessment of latent multi-hop reasoning. Our analysis reveals that while models can perform latent reasoning effectively in specific scenarios, their ability varies dramatically across different types of queries. This fluctuation, along with the significant gap between latent and explicit CoT reasoning, suggests substantial room for improvement in how LLMs internally compose their knowledge. Our work provides resources and insights for precise evaluation, understanding, and improvement of latent multi-hop reasoning of LLMs.

## Limitations

We do not test other forms of compositional reasoning such as comparisons because if the answer is binary, it is hard to rule out the cases of guessing. We do not test more than two hops because latent two-hop composability is already quite low, and adding more complexity to the problem may lower it to zero success cases. We do not consider facts that are subject to frequent change over time to compare models trained at different corpus cutout times. While we cannot know whether the LLMs we evaluate have not been trained on synthetic data generated from a knowledge graph, the low latent composability and high CoT composability obtained with our dataset suggest that the chance is very low for the evaluated models. While we cannot guarantee that the head and answer entities of every test query of SOCRATES would have never been learned in a single pretraining sequence for every model that we evaluate, we believe that our dataset construction that utilizes document co-occurrence counts of multiple pretraining corpora provides a tight approximation, which is also supported by the experimental results in §C.3.

## References

01.AI, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.AI. *arXiv [cs.CL]*.

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

Anthropic. 2024. Introducing Claude 3.5 Sonnet. *Anthropic blog*.

Gregor Bachmann and Vaishnavh Nagarajan. 2024. The pitfalls of next-token prediction. In *ICML*.

Shay Banon. 2010. Elasticsearch: A distributed, RESTful search and analytics engine.

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*.

BigScience Workshop et al. 2022. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv [cs.CL]*.

Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. 2024. Hopping too late: Exploring the limitations of large language models on multi-hop queries. In *EMNLP*.

Alan Chan, Carson Ezell, Max Kaufmann, Kevin Wei, Lewis Hammond, Herbie Bradley, Emma Bluemke, Nitarshan Rajkumar, David Krueger, Noam Kolt, Lennart Heim, and Markus Anderljung. 2024. Visibility into AI agents. In *FAccT*.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. 2024. How do large language models acquire factual knowledge during pretraining? In *NeurIPS*.

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. In *NAACL*.

Jiuhai Chen, Lichang Chen, Heng Huang, and Tianyi Zhou. 2023. When do you need chain-of-thought prompting for ChatGPT? *arXiv [cs.AI]*.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *TACL*.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. 2024. Implicit chain of thought reasoning via knowledge distillation. In *ICLR*.

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, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hanna Hajishirzi, Noah A Smith, and Jesse Dodge. 2024. What's in my big data? In *ICLR*.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Amir Feder, Abhilasha Ravichander, Marius Mosbach, Yonatan Belinkov, Hinrich Schütze, and Yoav Goldberg. 2022. Measuring causal effects of data statistics on language model's 'factual' predictions. *arXiv [cs.CL]*.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *EMNLP*.

Gemini Team et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv [cs.CL]*.

Gemma Team et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv [cs.CL]*.

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

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *ICML*.

Gaurav Ghosal, Tatsunori Hashimoto, and Aditi Raghunathan. 2024. Understanding finetuning for factual knowledge extraction. In *ICML*.

Daniela Gottesman and Mor Geva. 2024. Estimating knowledge in large language models without generating a single token. In *EMNLP*.

Cosimo Gregucci, Bo Xiong, Daniel Hernandez, Lorenzo Loconte, Pasquale Minervini, Steffen Staab, and Antonio Vergari. 2024. Is complex query answering really complex? *arXiv [cs.LG]*.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A Smith, and Hannaneh Hajishirzi. 2024. OLMo: Accelerating the science of language models. In *ACL*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *COLING*.

Yihuai Hong, Lei Yu, Shauli Ravfogel, Haiqin Yang, and Mor Geva. 2024. Intrinsic evaluation of unlearning using parametric knowledge traces. *arXiv [cs.CL]*.

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 *EMNLP*.

Hamish Ivison. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *Ai2 blog*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv [cs.CL]*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *arXiv [cs.LG]*.

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *ACL*.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2022. Understanding and improving zero-shot multi-hop reasoning in generative question answering. In *COLING*.

Tianjie Ju, Yijin Chen, Xinwei Yuan, Zhuosheng Zhang, Wei Du, Yubin Zheng, and Gongshen Liu. 2024. Investigating multi-hop factual shortcuts in knowledge editing of large language models. In *ACL*.

Cheongwoong Kang and Jaesik Choi. 2023. Impact of co-occurrence on factual knowledge of large language models. In *Findings of EMNLP*.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar finetuning examples control how language models hallucinate. *arXiv [cs.LG]*.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are pretrained language models symbolic reasoners over knowledge? In *CoNLL*.

Miyoung Ko, Sue Hyun Park, Joonsuk Park, and Minjoon Seo. 2024. Investigating how large language models leverage internal knowledge to perform complex reasoning. In *EMNLP*.

Takeshi Kojima and Shixiang Shane Gu. 2022. Large language models are zero-shot reasoners. In *NeurIPS*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *SOSP*.

Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. 2024. Understanding and patching compositional reasoning in LLMs. In *Findings of ACL*.

Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. 2023. Transformers learn shortcuts to automata. In *ICLR*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *arXiv [cs.CL]*.

Thomas Mesnard et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv [cs.CL]*.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. In *ACL*.

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*.

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. 2022. Introducing chatgpt. *OpenAI blog*.

OpenAI. 2024a. Gpt-4o mini: advancing cost-efficient intelligence. *OpenAI blog*.

OpenAI. 2024b. Hello GPT-4o. *OpenAI blog*.

Binghui Peng, Srini Narayanan, and Christos Papadimitriou. 2024. On limitations of the transformer architecture. In *COLM*.

Jordan Peterson, Ruairidh Battleday, Thomas Griffiths, and Olga Russakovsky. 2019. Openwebtext corpus. *GitHub repository*.

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

Jackson Petty, Sjoerd van Steenkiste, Ishita Dasgupta, Fei Sha, Dan Garrette, and Tal Linzen. 2024. The impact of depth on compositional generalization in transformer language models. In *NAACL*.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of EMNLP*.

Qwen Team. 2024. Qwen2.5: A party of foundation models! *Qwen Blog*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

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*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. In *ACL*.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *ACL*.

Ruixiang Tang, Dehan Kong, Longtao Huang, and Hui Xue. 2023. Large language models can be lazy learners: Analyze shortcuts in in-context learning. In *Findings of ACL*.

Johannes Treutlein, Dami Choi, Jan Betley, Cem Anil, Samuel Marks, Roger Baker Grosse, and Owain Evans. 2024. Connecting the dots: LLMs can infer and verbalize latent structure from disparate training data. In *NeurIPS*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *TACL*.

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

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural nlp: The case of gender bias. In *NeurIPS*.

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

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

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. Transformers: State-of-the-art natural language processing. In *EMNLP System Demonstrations*.

Nan Xu, Fei Wang, Bangzheng Li, Mingtao Dong, and Muhao Chen. 2022. Does your model classify entities reasonably? diagnosing and mitigating spurious correlations in entity typing. In *EMNLP*.

An Yang et al. 2024a. Qwen2 technical report. *arXiv [cs.CL]*.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024b. Do large language models latently perform multi-hop reasoning? In *ACL*.

Xunjian Yin, Baizhou Huang, and Xiaojun Wan. 2023. ALCUNA: Large language models meet new knowledge. In *EMNLP*.

Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van den Broeck. 2022a. On the paradox of learning to reason from data. In *IJCAI*.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue. 2024a. A careful examination of large language model performance on grade school arithmetic. *arXiv [cs.CL]*.

Xiao Zhang, Miao Li, and Ji Wu. 2024b. Co-occurrence is not factual association in language models. *arXiv [cs.CL]*.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022b. Active example selection for in-context learning. In *EMNLP*.

Yu Zhao, Yuanbin Qu, Konrad Staniszewski, Szymon Tworkowski, Wei Liu, Piotr Miłoś, Yuxiang Wu, and Pasquale Minervini. 2024. Analysing the impact of sequence composition on language model pre-training. In *ACL*.

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*.

# A Details of Dataset Construction

## A.1 Implementation of steps 1-2

We choose Wikidata (Vrandečić and Krötzsch, 2014) as the knowledge graph and collect facts in English. To ensure fair evaluation across models with different knowledge cutoff dates, we abstain from using relations subject to change over time (`organization-CEO` or `person-spouse`). We select 11 types of $r_1(e_1) = e_2$ and 10 types of $r_2(e_2) = e_3$ that are connected to each other with four types of $e_2$ – *"country"*, *"city"*, *"university"*, and *"year"* – resulting in a total of 21 relation composition types, although the number reduces to 17 through filtering during the implementation of step 3. Table 3 shows the 17 relation compositions that consist SOCRATES. As shown in the table, the relation compositions are divided further into subtypes for events[5] and award winners[6].

To use only the entities where $e_1$ and $e_3$ are not likely to be connected through other popular single-hop relations, we apply a relation-specific heuristic filtering. For instance, for `person-birthyear-eventcountry` relation (e.g., *"The country where the Eurovision Song Contest took place in the birth year of $e_1$ is"*), we exclude the cases where the country in which the event took place is the same with the person's birth country. A list of such heuristics is provided in Appendix Table 2.

Additionally, we exclude cases where $e_2$ can be easily inferred from the surface form of $e_1$, such as `university-locationcountry(University of Washington) = United States`, as these cases are where the multi-hop query is reduced into more like a single-hop query and also the cases where a substring of $e_1$ has relatively higher chance to frequently co-appear with $e_3$ in the same training sequences. A more detailed explanation is provided in §A.3. Through this process, we collect about 100K tuples $(e_1, e_2, e_3, r_1, r_2, E_1, E_2, E_3)$, where $e_1$, $e_2$, $e_3$ are the Wikidata entity titles, and $E_1$, $E_2$, $E_3$ are their corresponding sets of Wikidata aliases.

We manually construct four natural language templates for each subtype of $r_2$ and $r_1$ and randomly select one of the templates for each test case. Since multi-hop queries are constructed with the combinations of the templates chosen for $r_2$ and $r_2$, 16 templates are used for each type of relation composition. We feed $e_2$ to a template for $r_2$ to create $q(r_2(e_2))$, and feed the descriptive mention $\mu$ of the bridge entity (e.g., *"the year Scarlett Johansson was born"*) to create $q(r_2 \circ r_1(e_1))$, that both take

[5] Olympics Summer, Olympics Winter, Eurovision, Champions League Final, G7, European Capital of Culture

[6] Nobel Prize for Peace, Chemistry, Literature, Physiology or Medicine, and Masters Championship

| relation composition | heuristic conditions to meet |
|---|---|
| person-birthcity-year | city ≠ capital of a country<br>person's birth year ≠ event year |
| person-undergraduniversity-year | person's birth year ≠ university's inception year |
| person-birthyear-winner | person's birth/citizenship country ≠ winner's birth/citizenship country |
| person-birthyear-event country/city/leader | person's birth/citizenship country ≠ event country |
| university-inceptionyear-winner | university's location country ≠ winner birth/citizenship country<br>university ≠ winner's university for any degree |
| city-eventyear-winner | event country ≠ winner's birth/citizenship country |
| university-inceptionyear-event country/city/leader | university's location country ≠ event country |

Table 2: Relation-specific heuristic conditions that the collected test cases need to satisfy to prevent using the test queries where spurious correlations of $e_1$ and $e_3$ exist.

$E_3$ as the answer set (e.g., {*"United States"*, *"US"*, $\cdots$}). The descriptive mention of the bridge entity is created with a template for $r_1$. The same descriptive mention $\mu$ is also used to create $q(r_1(e_1))$ in the form *"$\mu$ is"* which takes $E_2$ as the answer set (e.g., {*"1948"*}). We filter out the dataset further during quality assurance (§A.4).

### A.2 Implementation of step 3 with document co-occurrence counts

Step 3 can be directly implemented when one has access to the LLM's pretraining sequences. However, we cannot use the approach as-is since LLMs are trained on different pretraining data, and this data is often not publicly available. Moreover, even if the data is available, information on how it has been broken into training sequences is rarely provided. To overcome these challenges, instead of computing the exact co-occurrence counts, we approximate them using two simplifications.

The first simplification is that we check the co-occurrence of the aliases of $e_1$ and $e_3$ *within a document instead of a training sequence*. To be specific, we use only test cases where there is no pretraining document in which any of the possible combinations of the aliases of $e_1$ and $e_3$ appear together. Filtering out test cases with non-zero document co-occurrence count imposes a stricter condition than doing so with sequence co-occurrence count because training sequences are substrings of a document in most LLMs trained with document boundaries, which is the standard approach in pretraining LLMs (Zhao et al., 2024).

The second simplification is that we utilize a *proxy corpus* to check the document co-occurrence since even the document-level information of the pretraining data of most models is not available.We use six different training corpora: Dolma v1.5, v1.7 (Soldaini et al., 2024), Tulu v2 (Ivison, 2023), OSCAR (Suárez et al., 2020), C4 (Raffel et al., 2020), and OpenWebText (Peterson et al., 2019), each of which contains 4,367M, 2,532M, 326K, 432M, 365M, 8M documents, used to train OLMo, OLMo 0724, OLMo Instruct (Groeneveld et al., 2024), BLOOM (BigScience Workshop et al., 2022), T5 (Raffel et al., 2020), and GPT-2 (Radford et al., 2019), respectively. The number of unique documents from the proxy corpus is roughly 4.8B.[7] In other words, we only use the test cases where none of any possible combination of the aliases of $e_1$ and $e_3$ appears together in 4.8B unique documents, which imposes a highly restrictive condition that enables obtaining a tight approximation of the test queries where the head and answer entities do not co-appear in the single pretraining sequence. While we cannot guarantee the exclusion of all entity co-occurring cases without access to exact pretraining corpora, we further validate our approximation using Google Search to check for co-occurrences across the whole web (§C.3).

We use the WIMBD (Elazar et al., 2024) API to get the document co-occurrence counts of these pretraining corpora which utilize Elasticsearch (Banon, 2010) as the backend with case-insensitive string match and exclude the test cases with non-zero co-occurrence count. After this filtering process, we obtain about 32K test cases of 17 relation composition types in total. Note that the distribution of the relation compositions is forced to be imbalanced as $e_1$ and $e_3$ of some relation compositions frequently appear together in the same document and most of the test cases are removed by the co-occurrence-based filtering. We down-sample the test cases with year-type bridge entities as the queries of these types outweigh other types.

**Data statistics** Our dataset for shortcut-free evaluation of latent multi-hop reasoning ability con-

[7] 4,367M (Dolma v1.5) + 432M (OSCAR) + 8M (OpenWebText) - 10M (OSCAR-Dolma overlap from WIMBD (Elazar et al., 2024) Demo Page)

tains 7,232 test cases of 17 types of relation compositions connected by 4 types of bridge entities, as shown in Table 1. Note that the distribution of relation compositions is imbalanced as $e_1$ and $e_3$ of some relation compositions frequently appear together in the same document and most of the test cases are removed by the co-occurrence-based filtering.

### A.3 Filtering Out the Cases with Easily Inferrable Bridge Entities

To prevent the multi-hop query from being reduced to be more similar to a single-hop query, we filter out the cases where the bridge entity is *easily inferrable* from the surface form of $e_1$ (Poerner et al., 2020), e.g., `university-locationcountry(University of Washington) = United States`, since Washington is a geographical location in the United States. Note that they also correspond to the cases where a *substring* of $e_1$ is likely to co-appear with $e_3$ in the same training sequence, e.g., `university-locationcountry-anthem(University of Washington) = The Star-Spangled Banner` where *"Washington"* likely co-occurs with *"The Star-Spangled Banner"*.

There are two such $r_1$ in our dataset: `university-locationcountry` and `person-birthcountry`. We use the prompt to GPT 3.5 turbo (OpenAI, 2022) and Claude 3 Haiku to guess the bridge entity solely from the name of the head entity for the first single-hop facts of these relation types (instruction is provided in §C.2), and if any of these models correctly predict the answer, we exclude it from the dataset.

As a result, for the cases with country as the bridge entity type, we only use the cases where the country of location of a university or the birth country of a person is hard to guess solely from the name without knowing the correct fact, such as `university-locationcountry(The International Graduate School of English) = South Korea` and `person-birthcountry(Natalie Portman) = Israel`.

### A.4 Dataset Quality Assurance

We apply several heuristic filterings to enhance the quality of the dataset such as excluding the cases without a natural language Wikidata title, excluding the cases with non-Unicode characters in the entity, excluding $e_1$ that contain double quotation marks or slashes, removing country flag emojis from the aliases of countries, and excluding the cases where each of $e_1$, $e_2$, and $e_3$ is a substring of the others. HTML characters are escaped and normalized. We discover that when all the open-source LLMs we evaluate fail to correctly answer a single-hop query, it is either because there is an error or noise in the answer set (Wikidata aliases) or the single-hop fact is not popular, and thus discard such cases from the dataset. Additionally, we use only the test cases where any alias combination of $e_1$ and $e_2$, and $e_2$ and $e_3$ appear together in Dolma v1.5 at least once.

## B Details of Evaluation Procedure

### B.1 Excluding *unusable* Cases

We observe that there are test cases such that the LLM generation is evaluated as correct, but the test cases are actually *unusable* for correct evaluation of the latent multi-hop reasoning due to the way the model generates the answer. The first type of such case is when the model completes the query as if constructing the answer choices of a multiple-choice question, such as completing *"National anthem of Woodie Flowers's country of birth:"* with *"1. "O Canada" 2. "The Star-Spangled Banner" 3. "God Save the Queen""*.[8] When this is the case for the completion of a single-hop or multi-hop query, we mark the case *unusable*.

The second type of *unusable* only applies to multi-hop queries. Especially for instruction-tuned models, even though we explicitly instruct the model to directly generate the answer without the bridge entity (§5.1), the models sometimes generate the bridge entity before generating the answer, such as completing *"The name of the national anthem of the country where Rishi Bankim Chandra Colleges is based is"* with *"\nThe correct answer is India.\nThe national anthem of India is Jana Gana Mana."* Such cases should be excluded from the evaluation of the latent multi-hop reasoning ability. Therefore, for the multi-hop queries with the EM score of 1, we additionally check if the LLM completion contains any of $e_2 \in E_2$ before the earliest $e_3 \in E_3$. If it is the case, we mark the case as *unusable*.

### B.2 Normalized Exact Match Score

To check whether an LLM has correctly predicted the answer to the given test query, we use the binary score of normalized exact match score (EM).

[8] This occurs the most often with pretrained Qwen2 models possibly due to training on the corpus where a large portion consists of multiple-choice questions and answers.

| relation composition type | relation composition subtype | count | example multi-hop query |
|---|---|---|---|
| person-birthcity-eventyear | person-birthcity-g7year | 9 | The G7 Summit was hosted in $e_1$'s birth city in the year |
| | person-birthcity-capitalofcultureyear | 3 | The year the birth city of $e_1$ was declared as the European Capital of Culture was |
| | person-birthcity-olympicswinteryear | 5 | The city where $e_1$ was born hosted the Winter Olympics in the year |
| | person-birthcity-eurovisionyear | 16 | The year when the birth city of $e_1$ hosted the Eurovision Song Contest was |
| person-birthcountry-anthem | person-birthcountry-anthem | 22 | The name of the national anthem of the birth country of $e_1$ is |
| person-birthcountry-isocode | person-birthcountry-isocode | 6 | The ISO 3166-1 numeric code of the country where $e_1$ was born is |
| university-locationcountry-anthem | university-locationcountry-anthem | 101 | The country where $e_1$ is based has the national anthem named |
| university-locationcountry-isocode | university-locationcountry-isocode | 30 | The ISO 3166-1 numeric code used for the country where $e_1$ is located is |
| university-locationcountry-year | university-locationcountry-year | 7 | The country where $e_1$ is located was established in the year |
| person-undergraduniversity-founder | person-undergraduniversity-founder | 33 | The person who founded the university where $e_1$ studied as an undergrad is named |
| person-undergraduniversity-year | person-undergraduniversity-year | 25 | The establishment year of the university where $e_1$ studied as an undergrad is |
| city-eventyear-winner | city-eurovisionyear-nobelchem | 1 | In the year the Eurovision Song Contest took place in $e_1$, the laureate of the Nobel Prize in Chemistry was |
| | city-g7year-nobelchem | 1 | In the year when the G7 Summit were hosted in $e_1$, the Nobel Prize in Chemistry was awarded to |
| person-birthyear-eventcity | person-birthyear-championsleaguecity | 196 | In $e_1$'s year of birth, the host city of the Champions League final was |
| | person-birthyear-capitalofculturecity | 264 | In the year $e_1$ was born, the city that was named the European Capital of Culture was |
| | person-birthyear-olympicswintercity | 288 | In $e_1$'s year of birth, the Winter Olympics were hosted in the city of |
| | person-birthyear-eurovisioncity | 391 | In $e_1$'s birth year, the host city of the Eurovision Song Contest was |
| | person-birthyear-g7city | 167 | In the year $e_1$ was born, the host city of the G7 Summit was |
| | person-birthyear-olympicssummercity | 83 | The city where the Summer Olympics took place in $e_1$'s year of birth is |
| person-birthyear-eventcountry | person-birthyear-olympicssummercountry | 7 | In $e_1$'s birth year, the Summer Olympics were hosted in the country of |
| | person-birthyear-championsleaguecountry | 35 | In the birth year of $e_1$, the Champions League final was hosted in the country of |
| | person-birthyear-olympicswintercountry | 8 | In $e_1$'s birth year, the Winter Olympics were hosted in the country of |
| | person-birthyear-g7country | 6 | The country that hosted the G7 Summit in $e_1$'s birth year is |
| | person-birthyear-eurovisioncountry | 68 | The country where the Eurovision Song Contest took place in the birth year of $e_1$ is |
| person-birthyear-hostleader | person-birthyear-hostleader | 260 | The person who was the host leader of the G7 Summit in $e_1$'s year of birth is |
| person-birthyear-winner | person-birthyear-nobelpsymed | 655 | In the birth year of $e_1$, the Nobel Prize in Physiology or Medicine was awarded to |
| | person-birthyear-nobelphysics | 675 | The winner of the Nobel Prize in Physics in the year $e_1$ was born is |
| | person-birthyear-nobelchem | 853 | In the birth year of $e_1$, the Nobel Prize in Chemistry was awarded to |
| | person-birthyear-nobellit | 777 | In the birth year of $e_1$, the Nobel Prize in Literature was awarded to |
| | person-birthyear-masterschampion | 931 | In the year $e_1$ was born, the winner of the Masters Tournament was |
| | person-birthyear-nobelpeace | 593 | The Nobel Peace Prize in the year $e_1$ was born was awarded to |
| university-inceptionyear-eventcity | university-inceptionyear-championsleaguecity | 14 | In the year $e_1$ was founded, the Champions League final was hosted in the city of |
| | university-inceptionyear-eurovisioncity | 17 | The city that hosted the Eurovision Song Contest in $e_1$'s inception year is |
| | university-inceptionyear-olympicswintercity | 7 | The city that hosted the Winter Olympics in the inception year of $e_1$ is |
| | university-inceptionyear-g7city | 9 | In the inception year of $e_1$, the host city of the G7 Summit was |
| | university-inceptionyear-olympicssummercity | 2 | In the year $e_1$ was founded, the host city of the Summer Olympics was |
| | university-inceptionyear-capitalofculturecity | 13 | The city that became the European Capital of Culture in the founding year of $e_1$ was |
| university-inceptionyear-eventcountry | university-inceptionyear-eurovisioncountry | 7 | The country that hosted the Eurovision Song Contest in the inception year of $e_1$ is |
| | university-inceptionyear-championsleaguecountry | 5 | The country where the Champions League final took place in the inception year of $e_1$ is |
| | university-inceptionyear-g7country | 1 | In the year $e_1$ was founded, the host country of the G7 Summit was |
| university-inceptionyear-hostleader | university-inceptionyear-hostleader | 9 | In the year $e_1$ was founded, the host leader of the G7 Summit was |
| university-inceptionyear-winner | university-inceptionyear-nobellit | 159 | In the inception year of $e_1$, the laureate of the Nobel Prize in Literature was |
| | university-inceptionyear-nobelchem | 156 | The winner of the Nobel Prize in Chemistry in the year $e_1$ was founded is |
| | university-inceptionyear-nobelpeace | 109 | The Nobel Peace Prize in the inception year of $e_1$ was awarded to |
| | university-inceptionyear-nobelphysics | 103 | The winner of the Nobel Prize in Physics in the founding year of $e_1$ is |
| | university-inceptionyear-nobelpsymed | 105 | In the year $e_1$ was founded, the Nobel Prize in Physiology or Medicine was awarded to |
| | | 7,232 | |

Table 3: Dataset statistics and example multi-hop test queries. The head entities are replaced with $e_1$ to prevent potential data leakage.

For each single-hop query, we apply string normalization to the completion of the LLM and each of the answer candidates in the answer set, which is the alias of the entity. The normalization consists of applying lowercase, removing accents, articles, and spaces in abbreviations, and replacing punctuation marks with spaces. The EM is 1 (correct) if any of the answer candidates is included in the generation respecting the word boundaries, and 0 (incorrect) otherwise. For the completion of the multi-hop queries, the calculation of EM goes through one more step of checking if the test case is *unusable*, as detailed below.

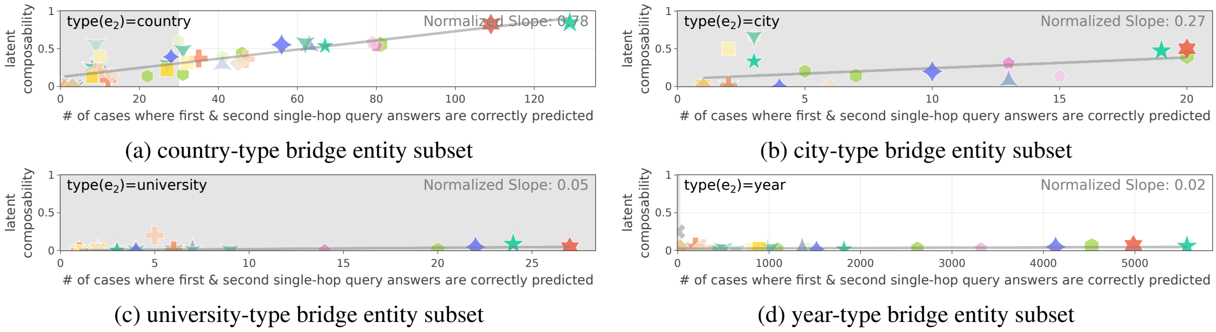
# C Details of Experiments

## C.1 Details of Experimental Setting

Among the open-source LLMs, we evaluate Mistral Large 2407 Instruct (123B), Small 2409 Instruct (22B), and all the pretrained and instruction-tuned models of Mistral Nemo 2407 (12B), Mistral 7B v0.3 (Jiang et al., 2023), Mixtral 8x7B v0.1 (Jiang et al., 2024), Qwen 2.5 (7B, 14B, 32B, 72B) (Qwen Team, 2024), Qwen 2 (7B, 72B) (Yang et al., 2024a), Yi 1.5 (6B, 9B, 34B) (01.AI et al., 2024) Gemma (2B, 7B) (Mesnard et al., 2024), Gemma 2 (2B, 9B) (Gemma Team et al., 2024), and OLMo (7B) (Groeneveld et al., 2024).

For all open-source LLMs, we use vLLM (Kwon et al., 2023) or HuggingFace Transformers (Wolf et al., 2020) to run the inference and greedy decoding[9] All experiments are performed with 1 to 8 40GB A100s using half precision. The proprietary LLM APIs are run with the default decoding parameters.

[9] We have also tried using random seed 0 and the decoding parameters specified in `generation_config.json` of each model in the HuggingFace Model Hub (https://hf.co). However, the performance difference was not large; in general, pretrained models performed slightly better with greedy decoding, and instruction-tuned models performed slightly worse with greedy decoding. Therefore, we chose to evaluate the models with greedy decoding for simplification and reproducibility.



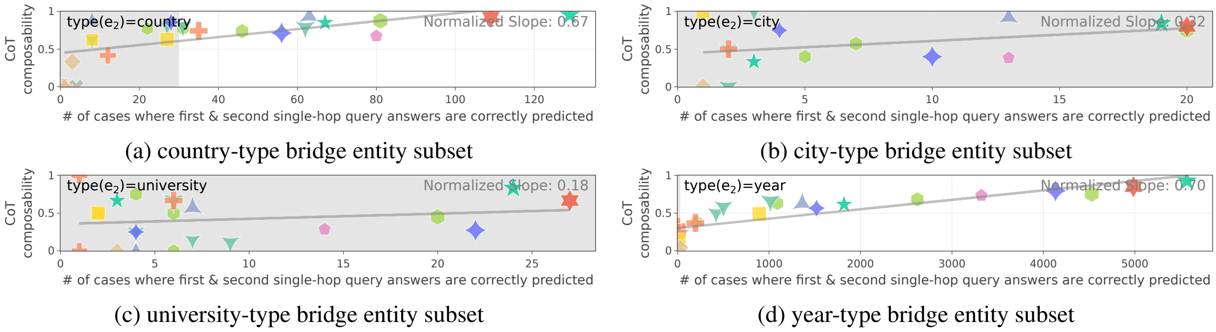
(a) country-type bridge entity subset

(b) city-type bridge entity subset

(c) university-type bridge entity subset

(d) year-type bridge entity subset

Figure 6: Latent composability measured for subsets of the test queries in SOCRATES, grouped according to the type of the bridge entity. Latent composability varies according to the bridge entity type; it is over 80% for the best models when the bridge entity is a country, but it is around 6% when it is a year.



(a) country-type bridge entity subset

(b) city-type bridge entity subset

(c) university-type bridge entity subset

(d) year-type bridge entity subset

Figure 7: CoT composability measured for subsets of the test queries in SOCRATES, grouped according to the type of the bridge entity. CoT composability does not fluctuate as dramatically as latent composability according to the type of the bridge entity, although the result is noisy for the city and university-type bridge entity subsets due to the small denominator.

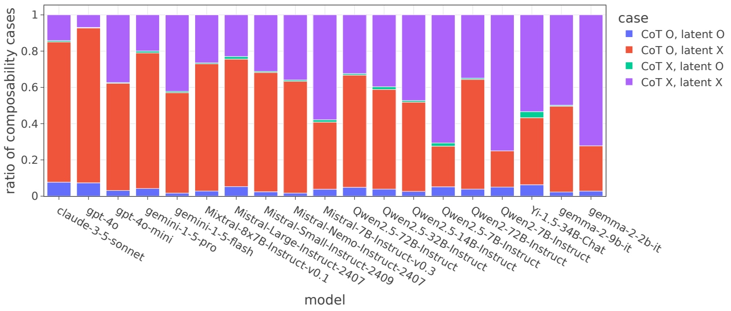


Figure 8: Ratio of whether each model successfully composes the facts with latent or CoT reasoning, among the cases where the models correctly predict the answer to both single-hop questions, excluding *guessable* and *unusable* cases. The results are shown only when the denominator used to calculate the composability is greater or equal to 30. There are almost no cases where latent reasoning succeeds but CoT reasoning fails.

### C.2 Instruction Details

For the LLMs that support custom system instruction (Claude, GPT, Mistral, Qwen), we provide the instruction as the system instruction. For other models, we use the instruction at the beginning of the prompt with a separator of "$\backslash n \backslash n$".

**CoT-suppressing instruction** To suppress the default CoT behavior of instruction-tuned LLMs, we use the instruction *"Fill in the blank. Write down only what goes in the blank. Do not explain your answer. The answer can consist of multiple words."* and postpend " ___" to all test queries to measure latent composability. This prompt has the most effectively prevented the CoT-style reasoning among several different task formulations that have been manually tested.

**CoT-triggering instruction** To trigger CoT of instruction-tuned LLMs, we use the following instruction: *"Fill in the blank. First, write the step-*

*by-step explanation necessary to get the solution with the prefix "EXPLANATION:". After that, write down the final answer with the prefix "ANSWER:". For the final answer, write down only what goes in the blank. The answer can consist of multiple words.”* and postpend “ ___” to the tested multi-hop query.

**Internal think-step-by-step instruction** We use the following instruction: *“Fill in the blank. Write down only what goes in the blank. Think step-by-step, but do it only internally and do not explain it in the answer. The answer can consist of multiple words.\n\nWhen is $e_1$’s birth year? Use the information.\n”* and postpend “ ___” to the tested multi-hop query.

**Guessing the bridge entities** For the first single-hop facts with `university-locationcountry` relations, we use the following instruction: *“Guessing from the name, what are the candidates of the country where “The University of Washington” is likely to be located? To be more specific, does “The University of Washington” contain the name of a location? If so, which country is the location in? Moreover, if “The University of Washington” contains a word of a language other than English that is used in specific countries, what are the names of those countries? Make sure to list the names of the countries guessed solely from the name.”*

For the first single-hop facts with `person-birthcountry` relations, we use the following instruction: *“Guessing from the name, what are the candidates of the country where “Shohei Ohtani” was likely to be born? To be more specific, what are the candidates of the country where someone with the first name “Shohei” was likely to be born? Likely, what are the candidates of the country where someone with the last name “Ohtani” was likely to be born? Make sure to list the names of the countries guessed solely from the person name.”*

### C.3 Additional Google Search Filter

For the subset of the test queries with country-type bridge entities where latent composability is notably high, we experiment with adding a Google Search filter to further exclude test cases where the head and answer entities appear together in *any* of approximately 400B documents indexed by the Google Search Engine. Note that such filtering is aggressive and greatly reduces the number of test cases usable for measuring latent composability. Since the latent composability of only five models is calculated with a denominator of greater or equal to 30, we calculate the average relative change of latent composability among these five models. Denoting $c$ as the original latent composability and $c'$ as the latent composability on the subset with an additional Google Search filter, we calculate the average relative drop after applying Google Search as $\mathbb{E}[\frac{c-c'}{c}]$, and the value is minimal as 0.03.

### C.4 Patchscopes Experiment

Using Patchscopes (Ghandeharioun et al., 2024) following the study of Biran et al. (2024), we check how often the latent representation of the bridge entity and the answer entity is constructed at the last token of the descriptive mention of the bridge entity (e.g., *“the year Scarlett Johansson was born”*) and the last token of the multi-hop query.

The experiment is done using the following procedure. First, we take a certain layer’s hidden state computed when an LLM processes a multi-hop query (the source prompt) at either the last token position of the descriptive mention of the bridge entity or the multi-hop query. Second, we feed the target prompt *“StarCraft: StarCraft is a science fiction real-time strategy game, Leonardo DiCaprio: Leonardo DiCaprio is an American actor, Samsung: Samsung is a South Korean multinational corporation, x”*[10] into the same LLM, with activation patching (Vig et al., 2020) of replacing a certain layer’s hidden state at the token *“x”* with the hidden state taken from the source prompt. Lastly, we let the model generate the output for the target prompt with the replaced hidden state, and check if the bridge entity or answer entity is included in the model’s output. Following Biran et al. (2024), we sample three generations for each patch with a temperature of 1.0 and count it a success if the entity is included in any of the generations.

Since the target prompt follows the format of *“entity: entity description”* where the entity description always starts by repeating the entity, if a latent representation of the entity is clearly constructed at the multi-hop query, it should be able to generate the entity from itself alone when it is patched to *“x”* in the target prompt. We perform activation patching from each layer of the computation of the source prompt to each layer of the computa-

[10] We slightly modify the original target prompt used in the work of Ghandeharioun et al. (2024) and Biran et al. (2024), to use the description of StarCraft instead of *“Syria: Syria is a country in the Middle East”*, in order to avoid any entity that falls into the types of the bridge entity for our dataset being used as the few-shot example and contaminating our analysis.

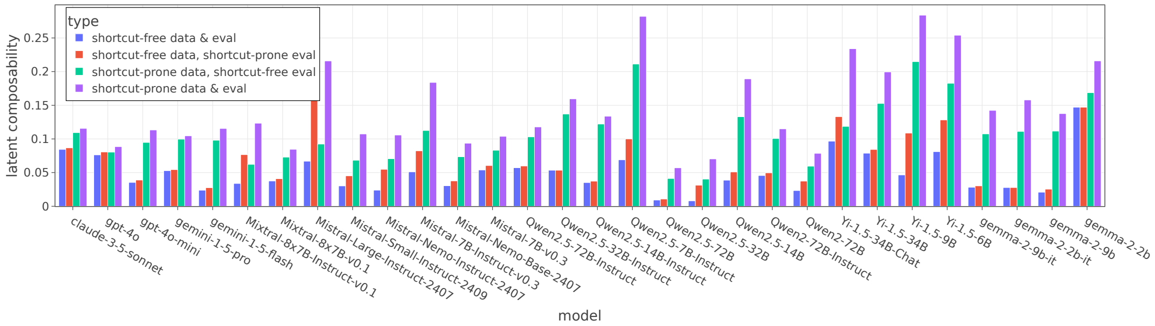


Figure 9: Latent composability measured with shortcut free/prone data and evaluation. The blue bars show the latent composability on SOCRATES evaluated with the proposed evaluation procedure, while the purple bars show the latent composability on shortcut-prone evaluation. The results are shown only when the denominator used to calculate the composability is greater or equal to 30. Latent composability measured with SOCRATES and the proposed evaluation procedure is consistently lower than that measured with shortcut-prone data and evaluation across all models, implying that overlooking shortcut exploitation can lead to an overestimation of the actual latent composability.

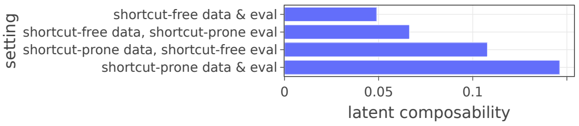


Figure 10: Latent composability measured with shortcut free/prone data and evaluation, averaged across models shown in Appendix Figure 9. Latent composability measured with SOCRATES and the proposed evaluation procedure is about three times lower than the shortcut-prone counterpart.

tion of the target layer and measure the *extraction rate* of the entity; whether the output with the activation patching contains the entity. Note that the extraction is successful only when the latent representation of the entity emerges sufficiently clearly to be able to decode the entity only from the representation itself, and thus the extraction rate can be thought as a lower bound of how often the latent representation of the entity is constructed by the model while processing the multi-hop query.

Figure 11 shows how often the hidden states taken from the layers at the end of the descriptive mention or the end of the source prompt (y-axis) generate the bridge or answer entity when patched into the layers of the target prompt (x-axis), for the queries with country and year-type bridge entities where both single hop facts are known by the model. The results for other types of queries are not shown due to an insufficient number of such cases. The bridge entity is generated more often, which suggests that the latent bridge entity representations are constructed more often, for the type of queries with higher latent composability (queries with country-type bridge entities).

## C.5 Emergence of Latent Multi-Hop Reasoning

OLMo (Groeneveld et al., 2024) provides intermediate training checkpoints (557 checkpoints from 1K to 557K pretraining steps) and the pretraining sequences that the model learns at each of the 557K steps. This allows us to: (1) definitively verify whether the head entity ($e_1$) and answer entity ($e_3$) of a multi-hop test query appear together in any single sequence during pretraining, without relying on any approximation, and (2) track the emergence of latent multi-hop reasoning ability by monitoring the model's accuracy as training progresses.

We build an ElasticSearch index using all of OLMo 7B's pretraining sequences to check whether ($e_1$, $e_2$), ($e_2$, $e_3$), and ($e_1$, $e_3$) co-appear in any single training sequence that the model learns at each pretraining step. Then, for the test queries where ($e_1$, $e_3$) never appears across all pretraining steps, we analyze the model's prediction accuracy for both single-hop and multi-hop queries. During the evaluation, we exclude any test case that is *guessable* or *unusable* at any of the 557 pretraining steps.

Through this evaluation procedure, we observe 13 (11.8%) cases where the model successfully performs multi-hop reasoning at some point during pretraining, out of 110 cases where the model is correct on both single-hop facts at some point and the model is not likely to be guessing the answer at any point during pretraining. In 12 of these 13 cases, the model begins to correctly answer the multi-hop query only after learning both constituent single-hop facts. While the number of such success cases

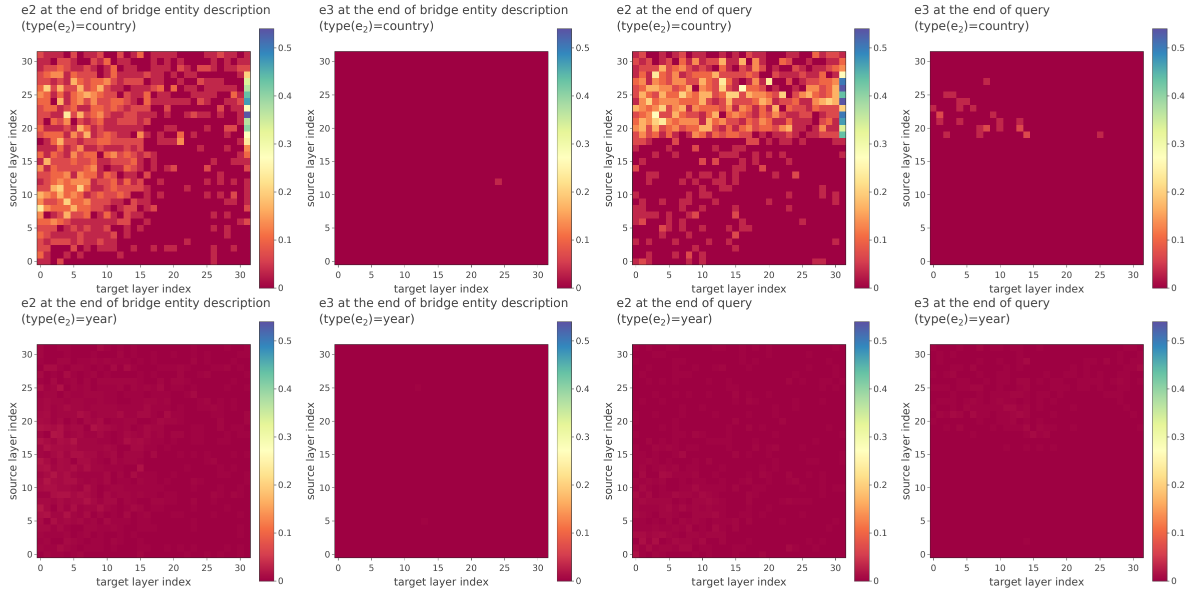


Figure 11: Experimental results with Mistral 7B v0.3 that apply Patchscopes (Ghandeharioun et al., 2024) to examine whether the model constructs latent representations of the bridge entity and answer entity at the last token of the descriptive mention of the bridge entity and the last token of the multi-hop query, for the queries with country-type bridge entities **(top)** and year-type bridge entities **(bottom)**. Latent representations of bridge entities are constructed more often for queries with country-type bridge entities (that have higher latent composability).

is currently limited by the model's capacity and is too small for quantitative analysis, these examples provide direct evidence that LLMs can develop latent multi-hop reasoning capabilities purely through pretraining.

Figure 4 demonstrates one successful case of OLMo 7B that demonstrates the emergence of latent multi-hop reasoning ability even when $e_1$ and $e_3$ have never co-appeared in any sequence throughout pretraining. It is also noticeable that the model starts to correctly predict the answer to the single-hop queries after consistently seeing ($e_1$, $e_2$) and ($e_2$, $e_3$) together across multiple pretraining steps. This aligns with the finding of Chang et al. (2024) that models learn simple facts by accumulating observations of the fact.

# D Discussion

## D.1 Importance of Considering Dataset Distribution

Due to our co-occurrence-based filtering process during dataset construction, year-type bridge entity cases comprise the majority of the dataset, while most test cases with other bridge entity types were filtered out. However, if the dataset had been constructed with mostly country-type bridge entity cases, the measured latent composability would have appeared much stronger. Therefore, when evaluating the latent multi-hop reasoning ability of LLMs, it is crucial to have diverse types of bridge entities, consider the distribution of the types of connected facts, and analyze performance separately for different types.

## D.2 Why Does Latent Composability Vary Across Bridge Entity Types?

We observe that latent composability varies significantly across bridge entity types, with notably high performance when facts are connected through country-type bridge entities.

Note that it is unlikely that such a success case of latent multi-hop reasoning is obtained due to a flaw in our dataset construction process. First, our careful dataset construction, which selects only cases where countries cannot be readily inferred (§A.3), accounts for easy guessing of countries from entity names (e.g., `university-locationcountry(University of Washington) = United States`). Therefore, the high latent composability of the country-type bridge entity cases does not come from the model simplifying the multi-hop query into a single-hop-like problem by easily guessing the first hop. Second, it's unlikely to stem from insufficient filtering of co-occurrences: adding a Google Search filter to exclude cases where entities appear together in search results does not drop latent composability, with an average relative drop of only 0.03 (details in §C.3).

Drawing from findings in finetuning studies, e.g., (Jiang et al., 2022; Wang et al., 2024), one speculative explanation of what has caused LLMs to develop strong composability for test queries with country-type bridge entity is that country-related facts might be more frequently learned in composition during pretraining. While these studies show that exposure to fact compositions during finetuning can improve multi-hop reasoning, we emphasize that extending these findings to pretraining remains an untested hypothesis that warrants future investigation.

### D.3 Why Is CoT Composability Much Stronger Than Latent Composability?

We conjecture that the explicit generation of the bridge entity is the main factor behind high CoT composability. Transformer-based LLMs go through a subject enrichment process that helps models recall the attributes of the subject (Geva et al., 2023; Gottesman and Geva, 2024). While LLMs can develop latent representations of bridge entities (e.g., *"1984"* from *"the year Scarlett Johansson was born"*) in early-middle layers (Yang et al., 2024b; Biran et al., 2024; Li et al., 2024; Wang et al., 2024), these representations may appear too late or not at all (Biran et al., 2024). In contrast, when CoT reasoning generates the correct bridge entity, it ensures a clear and early contextualized representation of the bridge entity to form, facilitating retrieval of the second single-hop fact.

Supporting this hypothesis, merely instructing models to *think step-by-step* (Kojima and Gu, 2022) but *only internally*, thus without generating the bridge entity, does not improve performance; Claude 3.5 Sonnet's latent composability remains low (6.1%) even with an explicit hint to identify and utilize the information of the bridge entity (instruction shown in §C.2). Moreover, 96.0% of CoT failures of Claude 3.5 Sonnet stem from incorrect bridge entity generation, highlighting its crucial role.

| evaluation result | success | success | failure | failure | *guessable* | *guessable* | *unusable* |
|---|---|---|---|---|---|---|---|
| relation composition type | person-birthyear-winner | university-locationcountry-anthem | university-inceptionyear-winner | person-birthyear-eventcity | person-birthyear-winner | person-birthyear-winner | university-inceptionyear-winner |
| relation composition subtype | person-birthyear-nobelchem | university-locationcountry-anthem | university-inceptionyear-nobellit | person-birthyear-eurovisioncity | person-birthyear-nobelphysics | person-birthyear-masterschampion | university-inceptionyear-nobelpsymed |
| $E_1$ | {'Ryan Cayabyab', 'Raymundo Cipriano Pujante Cayabyab'} | {'Royal University of Law and Economics'} | {'Nyenrode Business University', 'Nijenrode'} | {'Chalita Suansane'} | {'Turid Balke'} | {'Mario Cuenca Sandoval'} | {'Lagos State University', 'LASU'} |
| $E_2$ | {'1954'} | {'Cambodia', 'Kingdom of Cambodia', 'Kampuchea', 'Camboya', 'Campuchia', 'Kamboja', 'Kambodzha'} | {'1946'} | {'1995'} | {'1921'} | {'1975'} | {'1983'} |
| $E_3$ | {'Linus Pauling', 'Linus Carl Pauling', 'Linus C. Pauling'} | {'Nokor Reach'} | {'Hermann Hesse', 'Hermann Karl Hesse', 'H. Hesse'} | {'Dublin', 'Baile Átha Cliath', 'City of Dublin', 'Dublin city', 'Dublin, Ireland'} | {'Albert Einstein', 'Einstein', 'A. Einstein'} | {'Jack Nicklaus', 'Golden Bear', 'Jack William Nicklaus', 'William Nicklaus', 'Bear', 'Karnak'} | {'Barbara McClintock', 'Eleanor McClintock'} |
| $q(r_1(e_1))$ | Ryan Cayabyab's year of birth is | The country where Royal University of Law and Economics is based is | Nyenrode Business University's inception year is | Chalita Suansane's year of birth is | The birth year of Turid Balke is | Mario Cuenca Sandoval's birth year is | The year Lagos State University was founded is |
| completion | 1954. He is a Filipino composer, conductor, arranger, and pianist. He is also known as Mr. C. | Cambodia.<br>The country where Royal University of Law and Economics is based is Cambodia.<br>The country where Royal University of Law | 1946. The university was founded by business and industry as a post-war initiative to help rebuild the country. Nyenrode is | 1995. She is a Thai model and beauty pageant titleholder who was crowned Miss Universe Thailand 2016. She | 1921. She was born in Norway. She is a Norwegian actress. She is best known for her role as the mother of the main | 1975. He was born in Spain. He is a Spanish actor. He is best known for his role as the main character in the Spanish | 1983.<br>The year Lagos State University was founded is 1983.<br>The Lagos State University (L |
| $q(r_2(e_2))$ | In 1954, the laureate of the Nobel Prize in Chemistry was | The name of the national anthem of Cambodia is | In 1946, the Nobel Prize in Literature was awarded to | The city where the Eurovision Song Contest took place in 1995 is | In 1921, the Nobel Prize in Physics was awarded to | In 1975, the winner of the Masters Tournament was | The Nobel Prize in Physiology or Medicine in 1983 was awarded to |
| completion | Linus Pauling. He was awarded the prize for his research into the nature of the chemical bond and its application to the elucidation of the structure | "Nokor Reach" which means "Nation of Glory". It was adopted in 1941. The lyrics were written by Ch | Hermann Hesse, a German-born Swiss novelist and poet. Hesse is best known for his novels Siddhartha, Steppenw | the capital of the Republic of Ireland. Dublin is the largest city in Ireland and is located on the east coast of the island. The city is divided into two | Albert Einstein for his discovery of the law of the photoelectric effect. The photoelectric effect is the emission of electrons from a metal surface when light | Jack Nicklaus. He won the tournament for the fifth time, and he did it in style. Nicklaus shot a final round of 68 | Barbara McClintock for her discovery of mobile genetic elements.<br>The Nobel Prize in Physiology or Medicine in 1983 was |
| $q(r_2(r_1(e_1)))$ | In Ryan Cayabyab's year of birth, the laureate of the Nobel Prize in Chemistry was | The name of the national anthem of the country where Royal University of Law and Economics is based is | In Nyenrode Business University's inception year, the Nobel Prize in Literature was awarded to | The city where the Eurovision Song Contest took place in Chalita Suansane's year of birth is | In the birth year of Turid Balke, the Nobel Prize in Physics was awarded to | In Mario Cuenca Sandoval's birth year, the winner of the Masters Tournament was | The Nobel Prize in Physiology or Medicine in the year Lagos State University was founded was awarded to |
| completion | the American scientist Linus Pauling.<br>Ryan Cayabyab was born in 1954.<br>### When was Ryan C | "Nokor Reach" (Motherland).<br>The national anthem of Cambodia was adopted in 1941. | the French author Albert Camus. The jury praised Camus for his "clear-sightedness, which illuminates the problems of the human conscience | the host city of the 2018 edition.<br>Chalita Suansane represented Thailand at the 2016 Eurovision Song | Albert Einstein for his discovery of the law of the photoelectric effect.<br>In the birth year of Turid Balke, the Nobel Prize in | Jack Nicklaus.<br>The year was 1966.<br>Cuenca Sandoval, a senior at the University of North | the following scientists:<br>- 1983: Barbara McClintock, for her discovery of mobile genetic elements. - 19 |
| $q(r_2(\emptyset))$ | In the year, the laureate of the Nobel Prize in Chemistry was | The name of the national anthem of the country is | In the year, the Nobel Prize in Literature was awarded to | The city where the Eurovision Song Contest took place in the year is | In the year, the Nobel Prize in Physics was awarded to | In the year, the winner of the Masters Tournament was | The Nobel Prize in Physiology or Medicine in the year was awarded to |
| completion | awarded to the German chemist Richard Kuhn for his work on carotenoids and vitamins.<br>The Nobel Prize in Chemistry | "The Star-Spangled Banner". The lyrics of the national anthem of the United States of America were written by Francis Scott Key and the music | the French writer Jean-Marie Gustave Le Clézio.<br>The Nobel Prize in Literature 2008 was awarded to | the city of the Eurovision Song Contest. The Eurovision Song Contest is a song contest held annually among the member countries of the European Broadcasting | Albert Einstein for his work on the photoelectric effect. The photoelectric effect is the phenomenon in which electrons are emitted from a material when it | Tiger Woods. The Masters Tournament is one of the four major championships in professional golf. It is held annually at the Augusta National Golf Club in | three scientists for their work on the immune system.<br>The Nobel Prize in Physiology or Medicine in the year was awarded to three scientists for their |
| $q(r_2(r_1(\emptyset)))$ | In the person's year of birth, the laureate of the Nobel Prize in Chemistry was | The name of the national anthem of the country where the university is based is | In the university's inception year, the Nobel Prize in Literature was awarded to | The city where the Eurovision Song Contest took place in the person's year of birth is | In the birth year of the person, the Nobel Prize in Physics was awarded to | In the person's birth year, the winner of the Masters Tournament was | The Nobel Prize in Physiology or Medicine in the year the university was founded was awarded to |
| completion | the American chemist Theodore William Richards. He was awarded the prize for his accurate determination of the atomic weight of a number of chemical elements. | "God Save the Queen".<br>## What is the national anthem of the UK?<br>The national anthem of the United Kingdom is | the university's first professor of literature, Carl David af Wirsén.<br>The Nobel Prize in Literature is awarded by the Swedish Academy, | the most popular place to be born in the UK, according to new research.<br>The study, by the Office for National Statistics (ONS), found | the American physicist Arthur Ashkin for the invention of optical tweezers and their application to biological systems.<br>The Nobel Prize in Chemistry | Jack Nicklaus.<br>The winner of the U.S. Open was Tom Weiskopf.<br>The winner of the British Open was Lee | the German physiologist Emil von Behring for his work on serum therapy, particularly its application against diphtheria, but it was not |

Table 4: Example test cases, the corresponding completions of Qwen 2 72B, and the evaluation results. For all examples, the answers to both single-hop queries are correctly predicted. The green texts highlight correctly predicted answers. The red parts highlight the reasons for the negative results: failure, *guessable*, and *unusable*. The *guessable* and *unusable* cases are excluded from the evaluation of latent composability. These example test cases are not included in SOCRATES to prevent potential dataset leakage.