How Do Language Models Compose Functions?

Apoorv Khandelwal **Ellie Pavlick**
Department of Computer Science
Brown University
{apoorvkh,ellie_pavlick}@brown.edu

Abstract

While large language models (LLMs) appear to be increasingly capable of solving compositional tasks, it is an open question whether they do so using compositional mechanisms. In this work, we investigate how feedforward LLMs solve two-hop factual recall tasks, which can be expressed compositionally as $g(f(x))$. We first confirm that modern LLMs continue to suffer from the "compositionality gap", i.e. their ability to compute both $z = f(x)$ and $y = g(z)$ does not entail their ability to compute the composition $y = g(f(x))$. We then decode residual stream representations and identify two processing mechanisms: one which solves tasks *compositionally*, computing $f(x)$ along the way to $g(f(x))$, and one which solves them *directly*, without any detectable signature of the intermediate variable $f(x)$. Finally, we find that embedding space geometry is strongly related to which mechanism is employed, where the idiomatic mechanism is dominant when tasks are represented by translations from x to $g(f(x))$ in the embedding spaces. We fully release our data and code at: https://github.com/apoorvkh/composing-functions.

1 Introduction

Compositional behavior (McCurdy et al., 2024) is widely considered essential for flexible and general intelligence (Szabó, Fall 2024). A long-running debate has asked whether compositional *behavior* necessarily entails compositional *representations* and *processes*. On the one hand, formal languages based on compositional syntax and semantics are guaranteed to support certain types of invariance and generalization, making them compelling models for how humans might achieve abstract cognitive abilities like language and logic (Fodor, 1975; Quilty-Dunn et al., 2023). On the other hand, critics are quick to point out that humans frequently deviate from ideal compositional and logical behavior, suggesting some other mechanism must underlie our advanced cognition (Kahneman & Tversky, 1972; Evans, 2002).

Large language models (LLMs) provide an opportunity to revisit this debate in a new light. LLMs exhibit behavior that is at least ostensibly compositional, and which is not easily explained away by trivially non-compositional mechanisms (McCoy et al., 2024; Griffiths et al., 2025). However, LLMs also lack the kinds of explicit symbolic architectural components that have long been assumed necessary for such compositionality. This provides an opportunity to ask: do LLMs produce compositional behavior by invoking compositional processes, or do they rely on something more idiomatic instead?

We offer an initial investigation into this question, focusing on a set of two-hop factual retrieval tasks, such as: given a book's title, output that book's author's birth year. All of the tasks we consider can be formally expressed as $y = g(f(x))$ and are thus defensibly "compositional" in the sense invoked in traditional symbolic models. We are interested in whether LLMs solve such tasks by approximating the mapping from x to y *compositionally*, by computing the intermediate variable $z = f(x)$, or *directly*, without a readily-detectable representation of any such z. We find that:

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1. Models' ability to compute both $x \rightarrow f(x)$ and $f(x) \rightarrow g(f(x))$ does not entail their ability to compute $x \rightarrow g(f(x))$. This extends earlier findings on the "compositionality gap" (Press et al., 2022), showing that the gap holds for modern models and on a larger set of tasks. This gap is not trivially reduced in larger models or necessarily by reasoning models (Sec. 3).

2. Models exhibit both *compositional* processing mechanisms and *direct* processing mechanisms, as defined above. The type of mechanism is only weakly associated with accuracy, suggesting that LLMs are able to use both effectively to compute correct answers (Sec. 4).

3. The choice of mechanism is predicted by the geometry of the embedding spaces. Specifically, when there exists a translation (for a given task) from x in the input embedding space to $g(f(x))$ in the output unembedding space, the LLM tends to favor direct computation over compositional processing (Sec. 5).

2 Task Setup

Our tasks involve predicting a composition $g(f(x))$ from an input x, using in-context learning (ICL) and where f and g are some pre-defined functions. See Table 1 for the full list of tasks we use. We choose common functions f and g that models might learn through their pre-training and for which the inputs and outputs are lexical units. This enables us to use well-established tools for analyzing the mechanisms and latent computations in Transformer models, focusing on a few token positions (i.e. residual streams) and a single autoregressive forward pass.

We design the set of tasks in our investigation to cover a qualitative variety of functions, such as arithmetic, factual recall, lexical functions, translation, rotation, and string manipulation. By construction, all of our tasks can be computed by applying f and then g, yielding the causal hops $x \rightarrow f(x) \rightarrow g(f(x))$. Some tasks (e.g. commutative tasks) can also be computed through the hops $x \rightarrow g(x) \rightarrow g(f(x))$ — in which case, the intermediate z may also equal $g(x)$. We differentiate these further, and also describe our dataset construction methodologies (including our sources and pre-processing), in Appendix A.

In our experiments, we randomly sample 10 in-context examples for a given task and query. Each in-context example is formatted with a "`Q: {input} \n A: {output} \n\n`" prompting structure and the test query is formatted with "`Q: {input} \n A:`".

Table 1: List of our tasks. The compositional function $(g \circ f)$ is constructed by f and g here. We list the number of examples (#) in each task's dataset, along with the variables x, $f(x)$, and $g(f(x))$ for one random example. We list $g(x)$ and $f(g(x))$ for tasks that define them in Appendix A.

f	g	#	$x \rightarrow f(x) \rightarrow g(f(x))$
Word → Antonym	English → Spanish	2398	bogus → authentic → auténtico
Word → Antonym	English → German	2398	philosophical → practical → praktisch
Word → Antonym	English → French	2398	excessive → insufficient → insuffisant
Book → Author	Author → Birth Year	2228	The Boy in the Striped Pyjamas → John Boyne → 1971
Song → Artist	Artist → Birth Year	958	Heartbreak Hotel → Elvis Presley → 1935
Landmark → Country	Country → Capital	1385	Taq-i Kisra → Iraq → Baghdad
Park → Country	Country → Capital	743	Mount Rainier National Park → United States → Washington, D.C.
Movie → Director	Director → Birth Year	2180	Cape Fear → Martin Scorsese → 1942
Person → University	University → Year	4992	Andi Gutmans → Technion – Israel Institute of Technology → 1924
Person → University	University → Founder	4996	Ezra Abbot → Bowdoin College → James Bowdoin
Product → Company	Company → CEO	1904	NES-101 → Nintendo → Shuntaro Furukawa
Product → Company	Company → HQ	2276	Toyota Alphard → Toyota → Toyota
x + 10	2x	1000	699 → 709 → 1418
x + 100	2x	1000	922 → 1022 → 2044
x mod 20	2x	1000	891 → 11 → 22
Word → Numeric	2x	1000	one hundred and forty-eight → 148 → 296
Word[:-1]	Word[::-1]	2946	responsible → responsibl → lbisnopser
Rotate(RGB, 120°)	RGB → Name	1000	8a735a → 598a73 → dimgray

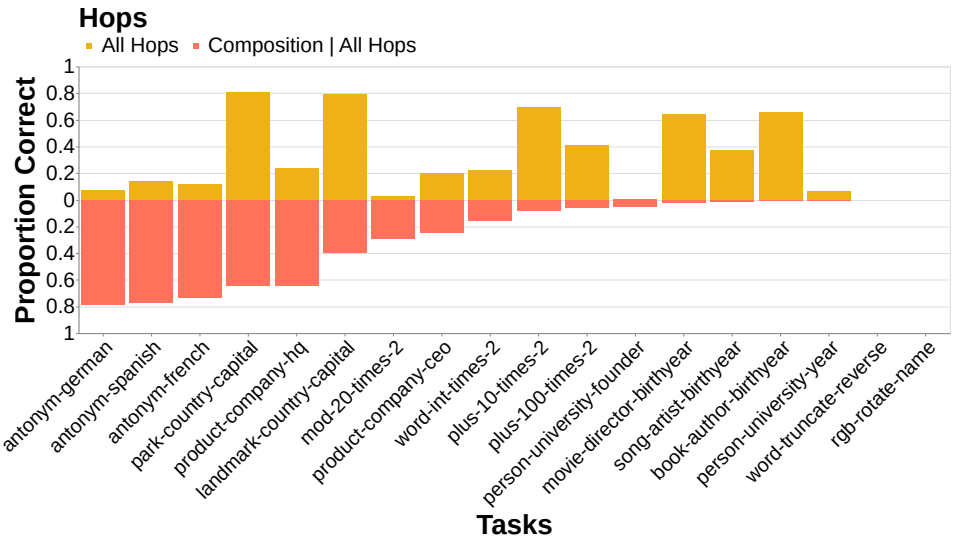


Figure 1: Compositionality gap for Llama 3 (3B) on our tasks. Yellow indicates the proportion of examples for which the model correctly solves all causal hops. Red indicates the (strict) subset of those examples for which the model additionally solves the composition. Correlation between red and yellow bars is $r^2 = 0.00$.

3 Compositionality Gap

Press et al. (2022) documented a "compositionality gap" in LLMs, showing that they consistently fail to solve compositions, despite solving the hops independently. Press et al. (2022) tested the GPT-3 family of models with natural language questions about celebrities and encyclopedic knowledge that required two-hops of factual recall. We confirm and extend this finding by testing modern LLMs on a larger set of compositional tasks.

3.1 Experimental Design

We prompt models with `input` $\rightarrow$ `output` mappings between lexical units.[1] We measure models' predictive accuracies using the ICL prompts from Sec. 2, greedy sampling, and the exact match evaluation metric. We test additional prompt templates in Appendix C and the effect of including task-specific instructions in the prompt in Appendix D. The *compositionality gap* is defined as the proportion of examples for which a model answers both $x \rightarrow f(x)$ and $f(x) \rightarrow g(f(x))$ correctly,[2] but $x \rightarrow g(f(x))$ incorrectly.

We test a few Llama 3 and OLMo 2 models on all of our tasks using all available examples. We emphasize results for Llama 3 (3B) in this section and include the other models in Appendix F. We also test an even wider set of models (also including DeepSeek and GPT models) on 4 tasks: `antonym-spanish`, `plus-100-times-2`, `park-country-capital`, and `book-author-birthyear` (which capture a representative set of processing signatures from Sec. 4). We aggregate metrics over these tasks and use 100 examples per task for testing.

3.2 Results

We show performance of the Llama 3 (3B) model on our tasks in Fig. 1. We clearly find a compositionality gap: the model is unable to solve the composition in 20–100% (varying by task) of examples for which it can solve all hops. We show the performances of our other models in Fig. 2. We find the compositionality gap does reduce with size from 72% $\rightarrow$ 39% (Llama 3, 1B $\rightarrow$ 405B)

[1] Note that this represents a methodological difference from Press et al. (2022), who prompted with long-form questions. Our format is chosen to fit with the interpretability methods we use in later sections.

[2] We extend this definition to further require success at $x \rightarrow g(x)$ and $g(x) \rightarrow g(f(x))$ in tasks where these are valid hops.

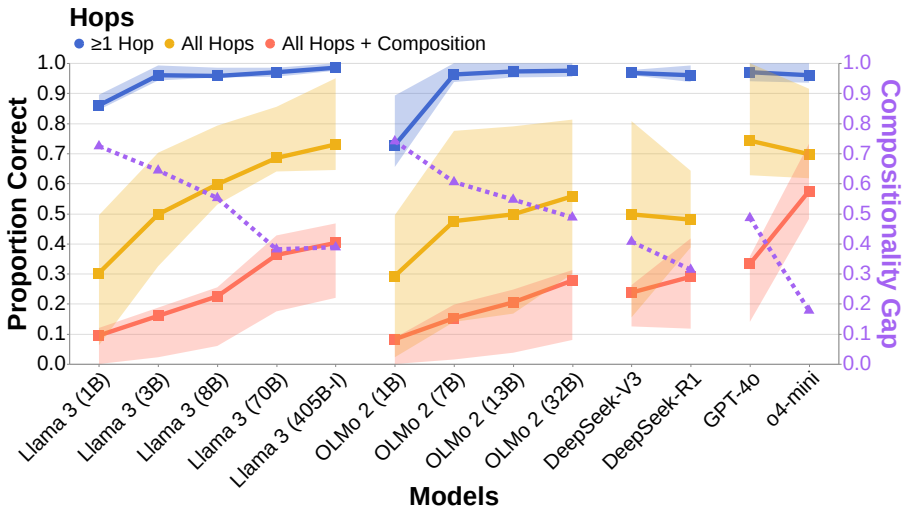


Figure 2: Compositionality gap (dashed purple line; lower is better) of various models aggregated over 4 tasks (100 examples each). Blue, yellow, and red lines show proportions of examples for which models correctly solve combinations of hops and the composition. Purple line shows the relative gap between yellow and red: the proportion of examples for which the model cannot solve the composition, out of those for which it can solve all hops. "-I" indicates the instruction-tuned variant of Llama 3 (405B). Error bands show interquartile range.

and 74% $\rightarrow$ 49% (OLMo 2, 1B $\rightarrow$ 32B). However, the gap clearly remains and we find monotonically diminishing improvements for both model families with respect to size. We plot the gap against model parameters and layers in Appendix E. In fact, the gap shows no improvement at all between the 70B and (instruction-tuned) 405B parameter Llama 3 models.

We also compare reasoning models (o4-mini and DeepSeek-R1; allotted a budget of 2000 reasoning tokens) against same-generation, non-reasoning models (GPT-4o and DeepSeek-V3) in Fig. 2. We find some reduction (41% $\rightarrow$ 31%) in the compositionality gap in the case of DeepSeek's reasoning model and significant reduction (49% $\rightarrow$ 18%) in the case of o4-mini. As o4-mini is proprietary (and both this and GPT-4o have additional "external tool-use" capabilities), it is difficult to speculate about the exact causes for these improvements. However, it is notable that even with advanced reasoning models, the gap does not necessarily disappear entirely.

Finally, we include a random sample of correct and incorrect predictions for every model in our supplemental materials.

4 Analyzing Processing Mechanisms

We next try to understand *how* the model correctly computes compositions in cases where it is successful. Our intuition is based on prior work from Merullo et al. (2024) which identifies a processing signature in models that solve one-hop relational tasks. That work shows that models predicting $y = f(x)$ iteratively surface vocabulary representations — first for x and then for y — in the residual stream. This "crossover" point was interpreted as evidence of the function f being applied to the argument x in order to yield the final answer $f(x)$ and was localized to specific computations in the MLPs.

In this section, we ask whether an analogous signature will emerge in the case of compositional functions, $g(f(x))$. That is, can we find distinct intermediate representations for x, followed by $f(x)$, and then $g(f(x))$ during the model's processing?

Here, we employ analyses most similar to Biran et al. (2024) and Yang et al. (2025) in the context of our evaluation (see Sec. 7 for further discussion on these works). We join other works in identifying stages of processing within language models (Tenney et al., 2019; Merullo et al., 2024; Lepori et al., 2025; Lu et al., 2025).

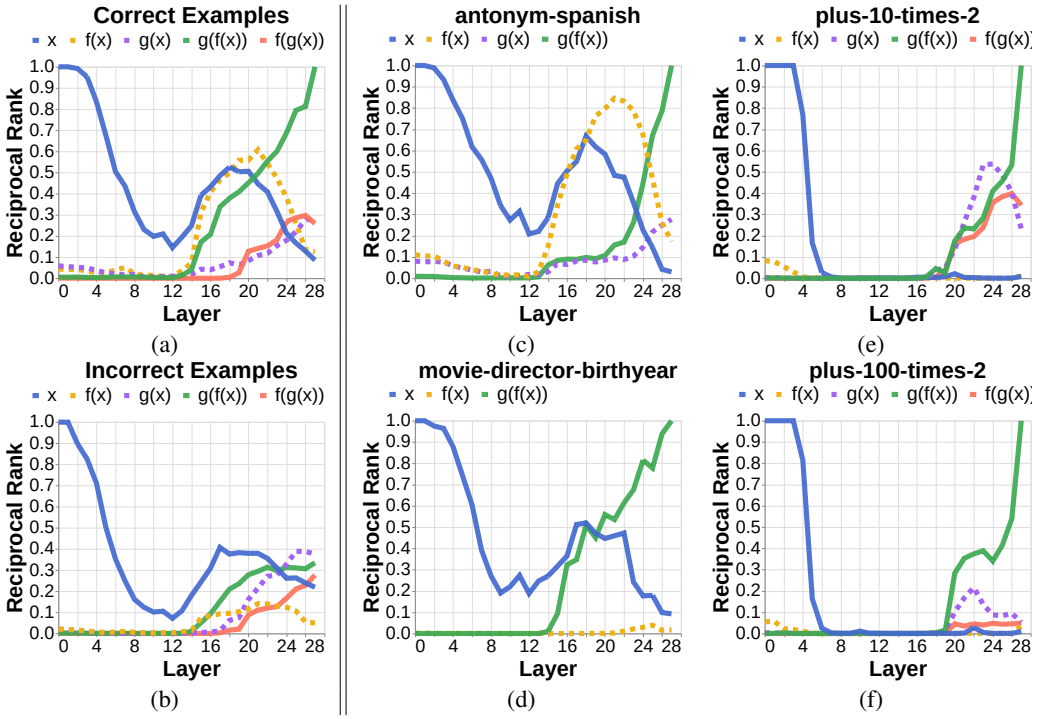


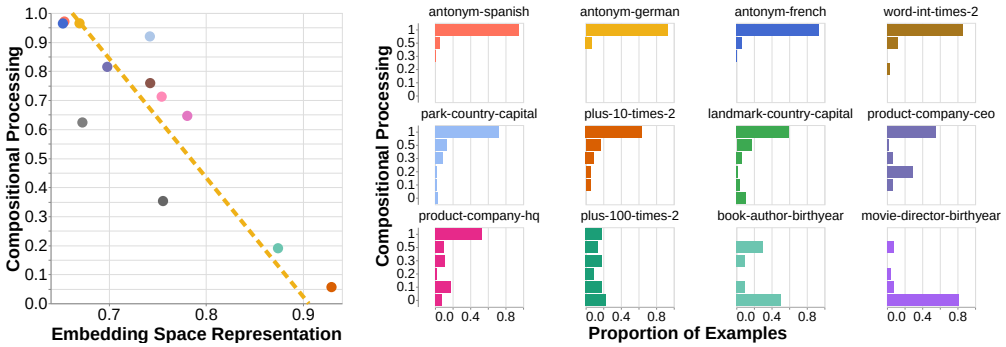
Figure 3: (a–b) Processing signatures aggregated over examples (across all tasks) in which Llama 3 (3B) solves all hops correctly, but the composition (a) correctly or (b) incorrectly. (c–f) Processing signatures for particular tasks — aggregated over examples where this model correctly solves all hops and the composition. (a–f) Lines show reciprocal ranks of relevant variables (decoded using logit lens) from residual streams corresponding to $x \rightarrow g(f(x))$. Intermediate variables are shown with dashed lines. The incorrect composition, $f(g(x))$, is shown by the red line when distinct from $g(f(x))$.

4.1 Experimental Design

We rely on existing methods which allow us to analyze processing signatures that are interpretable using the vocabulary space of the model (nostalgebraist, 2020; Geva et al., 2022). We specifically use *logit lens* (nostalgebraist, 2020), a method which projects intermediate representations into the vocabulary space using the language modeling head. Logit lens reveals information that models store in a linear representational subspace (i.e. the unembedding space) that is fundamental to the model. We include results using the token identity patchscope in Appendix J and using Jacobian lens in Appendix K. Both the patchscope and Jacobian lens strongly correlate with the measurements from logit lens (r^2 is 0.68 and 0.95, respectively).

We follow the approach from Merullo et al. (2024) to identify the processing signature of models that solve our compositional tasks and, in particular, representations of the intermediate variables, $f(x)$ and $g(x)$, prior to those for $g(f(x))$. We specifically use logit lens to analyze the residual streams corresponding to the computation $x \rightarrow g(f(x))$ and measure the reciprocal rank of our variables at each layer (see Appendix B for more details). We also use the maximum reciprocal rank of our intermediate variables across the layers as a heuristic for their overall presence in the computation. These decoded signatures are not guaranteed to be causal. Lack of decoding also does not imply absence of intermediate variable representations.

We focus this section's analysis on Llama 3 (3B) and summarize results for other models in Appendix I. We exclusively analyze examples where the model can solve all requisite hops. To ensure sufficient sample sizes, we exclude any task with fewer than 10 such examples where the model can also successfully solve the composition. In particular, the excluded tasks for Llama 3 (3B) are `song-artist-birthyear`, `person-university-year`, `person-university-founder`,



(a) (b)

Figure 4: Relationship between geometry and mechanisms of Llama 3 (3B). (a) Tasks correlate strongly between use of compositional mechanisms (heuristic from Sec. 4.1 based on reciprocal rank; on average across examples) and whether tasks are represented as translations in the embedding spaces ($r^2 = 0.77$). Conversely, accuracy is weakly correlated with these mechanism-related ($r^2 = 0.22$) and representation ($r^2 = 0.30$) metrics. (b) Distribution of examples for each task, shown as a histogram of intermediate variable reciprocal ranks. (a–b) Colors refer to corresponding tasks between points in (a) and histograms in (b).

`mod-20-times-2`, `word-truncate-reverse`, and `rgb-rotate-name`. We show results for these tasks in Appendices G and H.

4.2 Results

Fig. 3a shows the relative presence of each of the variables, across layers and aggregated over all instances in which the model ultimately produced the correct answer. In such cases, we see a very clear peak signal for the intermediate variable $f(x)$, as expected, between those for x and $g(f(x))$. Interestingly, this signal is much less clear for cases in which the model ultimately produces the incorrect answer (Fig. 3b). However, upon further inspection, there is little evidence of a causal relationship here, which we discuss further in Appendix H.

There are also plenty of individual examples in which the model produces a correct answer without showing any signature of the intermediate variables, and there is only a weak correlation by task ($r^2 = 0.22$) between predictive accuracy (measured as in Sec. 3.1) and the presence of intermediate variables as measured by our heuristic (Sec. 4.1).

Figs. 3c to 3f show model processing signatures for a few tasks, aggregated over cases in which the model produces correct answers. We see, for example, that there is a clear signature in the `antonym-spanish` task (Fig. 3c) for the intermediate computation of $f(x)$, the word's antonym, before it is translated into Spanish. In contrast, for the `movie-director-birthyear` task (Fig. 3d), there is no decodable signal for $f(x)$, the movie's director, before the model produces their birth year. This variation can be seen in qualitatively similar tasks as well: tasks with the same basic arithmetic structure (Figs. 3e and 3f) only sometimes carries detectable signatures of $f(x)$ or $g(x)$, depending on the task's operand (e.g. 10 or 100). We show processing signatures for the remaining tasks in Appendix G and for all tasks, aggregated over unsuccessful cases, in Appendix H.

5 Compositional Processing and Embedding Space Geometry

Given the variation in the strength of intermediate-variable signatures, we next ask whether embedding space geometry predicts this variation. It is well-known that some relations are represented by translations in classical word embedding spaces (Mikolov et al., 2013; Hewitt & Manning, 2019). Hernandez et al. (2024) also shows that, when LLMs compute certain `subject` $\rightarrow$ `object` relations, several layers of computation can be well approximated by simple, linear transformations. We extend these ideas together and posit that language models could directly represent some tasks as translations between the embedding and unembedding spaces (spanning language models' entire forward computation). We hypothesize that some compositional functions are represented more

fundamentally as these translations and that models compute such functions *directly*, rather than *compositionally* via the hops.

5.1 Experimental Design

To investigate our hypothesis, we fit translations for each task using least squares regression from x (average embedding across tokens) to $g(f(x))$ (mean unembedding across answer tokens, centered by the vocabulary-wide mean unembedding) on 100 examples.[3] We quantify how well the translation fits the task using its reconstruction accuracy (measured via cosine similarity) on the remaining examples. We quantify how "compositional" the processing is using our heuristic metric which captures the strength of the signal for the intermediate variables, $f(x)$ and $g(x)$ (see Sec. 4). We again restrict our analysis to examples where the model is successful on all hops and the composition, as well as tasks with at least 10 such examples.

5.2 Results

Fig. 4a shows that there is a strong inverse correlation ($r^2 = 0.77$) between embedding space task geometry and compositional processing for Llama 3 (3B). That is, the more a task is represented as a translation in the embedding spaces, the more likely the model is to display *idiomatic* (as opposed to *compositional*) processing.

This correlation is computed by averaging the indicator for compositional processing across instances for each task. Fig. 4b shows the de-aggregated distribution of our "compositionality" metric across the examples. For some tasks, it appears that nearly all individual examples behave the same way. For example, nearly every instance of `antonym-spanish` displays a compositional signature, while almost every instance of `movie-director-birthyear` displays an idiomatic one. On the other hand, this distribution is more uniform for other tasks, such as `plus-100-times-2`. This distribution appears to be bimodal across all examples: 82% have very low (< 0.1) or high (≥ 0.5) values for compositionality.

We include results for several other models in Appendix I, i.e. Llama 3 (3B) Instruct, Llama 3 (8B), OLMo 2 (7B), and OLMo 2 (13B). All the models we test continue to show strong correlations for the relationship presented here, with an average $r^2 = 0.71$. These models continue to show bimodal distributions of the choice of mechanism across examples. Linear representations yield weaker correlations than translations. We interpret embedding geometry as predictive of the observed processing signature, rather than as a cause of mechanism selection.

6 Discussion

Summary of Findings Our results suggest that tasks which appear to have the same computational structure may nonetheless be processed differently by LLMs. In particular, we consider functions which appear compositional in a formal sense — i.e. they can be represented as $y = g(f(x))$ for some reasonably defined f and g. We find evidence that LLMs only sometimes process such functions compositionally, showing evidence of representing or computing the value of $z = f(x)$ on the way to computing y. In other cases, LLMs appear to map x to y directly. Which of these processes is invoked appears to be related to how well the relationship between x and y is represented as a translation in the embedding space, e.g. as a result of pretraining (Merullo et al., 2025).

Implications for Theories of Compositionality There is a long-running debate about the degree to which compositional behavior (McCurdy et al., 2024) requires compositionality at the level of mechanisms. The two sides of this debate have often talked past each other, often using different types of computational architectures in order to model different aspects of behavior, for example, using explicitly compositional symbolic systems to model formal domains (Lake et al., 2017; Ellis et al., 2023) and using distributional or neural systems to model humans' more idiomatic performance (Erk, 2012; Lampinen et al., 2024).

Attempts to find compromises or "hybrid" systems often consist of neuro-symbolic systems which are designed top-down (Andreas et al., 2016; Ellis et al., 2018). Large language models offer an

[3] See Appendix L for additional analyses which consider correlations with the representations of the individual hops (rather than the compositional task).

alternative approach for advancing this debate. LLMs have proven capable of a range of behaviors that have traditionally required compositionality — e.g. generating language and writing formal computer code. However, LLMs lack the explicit symbolic mechanisms traditionally associated with such behaviors. Using methods from interpretability to understand how LLMs represent such functions internally enables us to approach the question in a "bottom up" manner, potentially offering novel hypotheses about the mechanisms that can generate behavior that is sometimes systematic and other times heuristic, as is the case in humans (Russin et al., 2025).

Our results suggest that LLMs employ a mix of compositional and idiomatic processing, and that the choice of mechanism is related to the representations of the functions that result from pretraining. This offers an interesting perspective on one question that is frequently at the heart of discussions of compositionality — i.e. what are the primitives and where do they come from (Carey, 2011)? The relationship between embedding space task geometry and compositionality of processing presented here suggests an attractive hypothesis that the primitives are those things which are well represented as a result of (pre-)training, and that compositional mechanisms are invoked to handle those things which are not sufficiently well represented. Future work in this direction would likely yield interesting new results and topics for debate.

Relationship to work on compositional generalization The work presented here concerns the (apparent) compositionality of the processing mechanism, but does not directly relate this mechanism to an LLM's capacity for compositional generalization. The majority of work on compositionality in neural networks (and LLMs) concerns compositional generalization, and the compositionality researchers surveyed by McCurdy et al. (2024) overwhelmingly agree that existing language models are insufficient in this regard. This belief is supported by evidence from many prior works (Sec. 7) and our investigation in Sec. 3.

Our work suggests that models employ both compositional and direct mechanisms to solve tasks. Intuitively, we would expect there to be a relationship between the use of the mechanism and the ability to generalize — i.e. the compositional mechanism should support generalization better than the idiomatic mechanism ("memorization"). However, we do not test this intuition directly in this paper. Future work could do so by employing causal interventions on the intermediate variables, for example (see Appendix M for some initial investigations). This would likely present new complexities and challenges that would enrich our understanding of compositionality, and of the relationships between mechanisms and behaviors in LLMs in general.

Practical Implications Our work offers geometry-based predictions of processing signatures, which may help formulate hypotheses about the functions LLMs treat as "primitive". If mechanisms are required for generalization, one may prefer models that fit truer primitives (e.g., f, g, h rather than $g \circ f$) and invoke the compositional mechanism, enabling unseen compositions such as $f \circ h$. Our analysis also provides processing signatures that reveal which functions a model attempts to invoke (successfully or not, e.g. Appendix H), and when individual data points are poorly fit by a function's embedding space representation, possibly indicating weak memorization during training.

7 Related Work

Latent multi-hop reasoning Our work is most closely related to recent or concurrent works which also study latent two-hop reasoning in large language models. Yang et al. (2024a) use causal interventions to identify the existence of the hops in the latent computation and whether they co-occur. Biran et al. (2024) employ the entity description patchscope (Ghandeharioun et al., 2024) to inspect intermediate representations and localize the hops, finding they are resolved in different layers and token positions. They propose a representational intervention ("backpatching") to correct failures based on this finding. Finally, Yang et al. (2025) use logit lens to analyze intermediate representations and consistently find a "compositional" processing signature across their tasks. Our work employs all of these interpretability methods (Sec. 4 and Appendices J and M) to analyze the hops, but specifically highlights and investigates variation between compositional and direct processing signatures. All

works (including our own) test different sets of tasks, make experimental design decisions according to their independent goals,[4]and make findings in context of their own experiments.

Among other works in this domain, Wang et al. (2024) trains a language model on synthetic compositional data and identifies a multi-hop reasoning circuit in this model. Shalev et al. (2024) conduct a distributional analysis (considering semantic category spaces, rather than individual tokens) using logit lens. Li et al. (2024); Yu et al. (2025) also propose interventions on intermediate representations and mechanisms to solve failure cases. Yang et al. (2024b) conduct an evaluation that is intentionally designed to omit opportunities for models to exploit shortcuts.

Compositionality Compositionality is long-studied (Fodor & Pylyshyn, 1988; Partee, 2004) but exact definitions evade general consensus. Russin et al. (2024) and McCurdy et al. (2024) offer recent overviews on the topic in the context of large language models. Russin et al. (2024) provide a historic account of compositionality and review studies of compositionality generalization in neural networks. McCurdy et al. (2024) survey compositionality researchers on how to define and evaluate compositional behavior in neural networks. These researchers agree that our current representational analyses are insufficient for evaluating models, but are divided about whether our behavioral analyses are sufficient.

In a partial effort towards defining compositionality, Hupkes et al. (2020) identify five particular aspects of compositionality and propose tests for each using a synthetic, fully compositional translation task. Systematicity is one such aspect and is prominently studied: see Vegner et al. (2025) for a survey of benchmarks for systematic generalization. Our work — in which we test whether $f(x)$ is evaluated before $g(f(x))$ — is closest to Hupkes et al. (2020)'s aspect of localism, in which "smaller constituents are evaluated before larger constituents".

Among many other works, Johnson et al. (2017), (Keysers et al., 2020), Lake & Baroni (2018), Hupkes et al. (2020), and Kim & Linzen (2020) offer prominent benchmarks that behaviorally test for compositional generalization in neural networks trained from scratch on compositional data. These works generally show that such models perform poorly on generalization, or at least poorly implement the compositional processes that underlie the data. Press et al. (2022) and Ma et al. (2023) continue to show significant failures in compositional generalization in pre-trained models. On the other hand, Furrer et al. (2020) points out that pre-training a masked language model rivals or outperforms architectures specifically designed for the SCAN (Lake & Baroni, 2018) and CFQ (Keysers et al., 2020) generalization benchmarks. Lepori et al. (2023) finds that neural networks learn to implement compositionality structurally in their weights, supporting this claim against the need for specialized symbolic mechanisms.

Compositionality of functions Several works consider how language models solve compositions of functions (rather than specifically multi-hop reasoning tasks). Dziri et al. (2023) studies how language models autoregressively solve such tasks, like multi-digit multiplication, by inspecting their scratchpads. Wattenberg & Viégas (2024) propose mechanisms which neural networks could use to implement relational compositions. Yu et al. (2023); Todd et al. (2024) propose zero-shot methods to invoke compositions of functions in language models that have learned the primitive functions. Zhou et al. (2024) find that language models can compose functions with meta-learning in a way that imitates human behavior.

Reproducibility

We make our data and code fully available so that all of our experiments can be replicated as closely as possible and all computational artifacts (datasets, plots, results) can be reconstructed. We do our best to include all experimental details in the main text and appendices of our paper.

[4]One notable example is that, while Yang et al. (2024a) and Biran et al. (2024) prompt their models with f and g (e.g. "`The mother of the singer of {x} is {y}`"), we typically omit this information from our prompts (i.e. more simply "`Q: {x} \n A: {y}`"), besides in our ablation, to avoid inducing bias towards the compositional mechanism.

Limitations

In this work, we primarily analyze the computation that occurs in a single forward pass of LLMs. It is also necessary to understand how other models (e.g. reasoning models or those with different inductive biases) implement compositional functions. Our findings reflect the tasks we happen to test (often, factual recall) under our specific experimental design. Further work should test other kinds of compositional functions, and try to more deeply understand the relationship between compositional mechanisms, behavior, and generalization.

We investigate a limited subset of mechanisms in language models and use current methods to conduct our analyses. Decoding can miss intermediate representations, and whole-stream patching can transfer subject or answer information as well as an intermediate value. Geometry is predictive, and both geometry and the signature may depend on other properties of the data or prompt. Some tasks (e.g. arithmetic) may also use algorithms we do not consider.

Acknowledgments and Disclosure of Funding

We are very grateful to the members of the Language Understanding and Representation (LUNAR) Lab at Brown University — especially Jacob Russin and Michael A. Lepori — and to Chen Sun, Najoung Kim, and Sohee Yang for their generous feedback. This research was conducted using computational resources and services at the Center for Computation and Visualization, Brown University. This work was supported in part by a Brown University Presidential Fellowship for Apoorv Khandelwal and in part by a Young Faculty Award from the Defense Advanced Research Projects Agency Grant #D24AP00261. Ellie Pavlick is a paid consultant for Google DeepMind. The content of this article does not necessarily reflect the views of the US Government or of Google, and no official endorsement of this work should be inferred.

References

Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. Neural Module Networks. *Computer Vision and Pattern Recognition (CVPR)*, 2016. URL https://arxiv.org/pdf/1511.02799.

Bast, H. and Buchhold, B. QLever: A Query Engine for Efficient SPARQL+Text Search. *Conference on Information and Knowledge Management (CIKM)*, 2017. URL https://dl.acm.org/doi/10.1145/3132847.3132921.

Biran, E., Gottesman, D., Yang, S., Geva, M., and Globerson, A. Hopping Too Late: Exploring the Limitations of Large Language Models on Multi-Hop Queries. *Empirical Methods in Natural Language Processing (EMNLP)*, 2024. URL https://arxiv.org/pdf/2406.12775.

Carey, S. Précis of The Origin of Concepts. *Behavioral and Brain Sciences (BBS)*, 2011. URL https://doi.org/10.1017/S0140525X10000919.

Dziri, N., Lu, X., Sclar, M., Li, X. L., Jiang, L., Lin, B. Y., Welleck, S., West, P., Bhagavatula, C., Le Bras, R., et al. Faith and Fate: Limits of Transformers on Compositionality. *Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/pdf/2305.18654.

Ellis, K., Morales, L., Sablé-Meyer, M., Solar-Lezama, A., and Tenenbaum, J. Library Learning for Neurally-Guided Bayesian Program Induction. *Neural Information Processing Systems (NeurIPS)*, 2018. URL https://dl.acm.org/doi/10.5555/3327757.3327878.

Ellis, K., Wong, L., Nye, M., Sable-Meyer, M., Cary, L., Anaya Pozo, L., Hewitt, L., Solar-Lezama, A., and Tenenbaum, J. B. DreamCoder: Growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning. *Philosophical Transactions of the Royal Society A*, 2023. URL https://arxiv.org/pdf/2006.08381.

Erk, K. Vector Space Models of Word Meaning and Phrase Meaning: A Survey. *Language and Linguistics Compass*, 2012. URL https://doi.org/10.1002/lnco.362.

Evans, J. S. B. Logic and Human Reasoning: An Assessment of the Deduction Paradigm. *Psychological Bulletin*, 2002. URL https://doi.org/10.1037/0033-2909.128.6.978.

Fodor, J. A. *The Language of Thought*. Harvard University Press, 1975.

Fodor, J. A. and Pylyshyn, Z. W. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 1988. URL https://doi.org/10.1016/0010-0277(88)90031-5.

Furrer, D., van Zee, M., Scales, N., and Schärli, N. Compositional Generalization in Semantic Parsing: Pre-training vs. Specialized Architectures, 2020. URL https://arxiv.org/pdf/2007.08970.

Geva, M., Caciularu, A., Wang, K. R., and Goldberg, Y. Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space. *Empirical Methods in Natural Language Processing (EMNLP)*, 2022. URL https://arxiv.org/pdf/2203.14680.

Ghandeharioun, A., Caciularu, A., Pearce, A., Dixon, L., and Geva, M. Patchscopes: A Unifying Framework for Inspecting Hidden Representations of Language Models. *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/pdf/2401.06102.

Griffiths, T. L., Lake, B. M., McCoy, R. T., Pavlick, E., and Webb, T. W. Whither symbols in the era of advanced neural networks?, 2025. URL https://arxiv.org/pdf/2508.05776.

Gurnee, W., Sofroniew, N., Pearce, A., Piotrowski, M., Kauvar, I., Chen, R., Soligo, A., Bogdan, P., Ong, E., Wang, R., Thompson, B., Abrahams, D., Kantamneni, S., Ameisen, E., Batson, J., and Lindsey, J. Verbalizable Representations Form a Global Workspace in Language Models. *Transformer Circuits Thread*, 2026. URL https://transformer-circuits.pub/2026/workspace/index.html.

Hernandez, E., Sharma, A. S., Haklay, T., Meng, K., Wattenberg, M., Andreas, J., Belinkov, Y., and Bau, D. Linearity of Relation Decoding in Transformer Language Models. *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/pdf/2308.09124.

Hewitt, J. and Manning, C. D. A Structural Probe for Finding Syntax in Word Representations. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. URL https://aclanthology.org/N19-1419.pdf.

Hupkes, D., Dankers, V., Mul, M., and Bruni, E. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research (JAIR)*, 2020. URL https://arxiv.org/pdf/1908.08351.

IMDb. IMDb Non-Commercial Datasets, 2024. URL https://developer.imdb.com/non-commercial-datasets/.

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. *Computer Vision and Pattern Recognition (CVPR)*, 2017. URL https://arxiv.org/pdf/1612.06890.

Kahneman, D. and Tversky, A. Subjective probability: A judgment of representativeness. *Cognitive Psychology*, 1972. URL https://doi.org/10.1016/0010-0285(72)90016-3.

Keysers, D., Schärli, N., Scales, N., Buisman, H., Furrer, D., Kashubin, S., Momchev, N., Sinopalnikov, D., Stafiniak, L., Tihon, T., Tsarkov, D., Wang, X., van Zee, M., and Bousquet, O. Measuring Compositional Generalization: A Comprehensive Method on Realistic Data. *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/pdf/1912.09713.

Kim, N. and Linzen, T. COGS: A Compositional Generalization Challenge Based on Semantic Interpretation. *Empirical Methods in Natural Language Processing (EMNLP)*, 2020. URL https://arxiv.org/pdf/2010.05465.

Lake, B. and Baroni, M. Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks. *International Conference on Machine Learning (ICML)*, 2018. URL https://arxiv.org/pdf/1711.00350.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building Machines That Learn and Think Like People. *Behavioral and Brain Sciences (BBS)*, 2017. URL https://arxiv.org/pdf/1604.00289.

Lampinen, A. K., Dasgupta, I., Chan, S. C. Y., Sheahan, H. R., Creswell, A., Kumaran, D., McClelland, J. L., and Hill, F. Language models, like humans, show content effects on reasoning tasks. *PNAS Nexus*, 2024. URL https://arxiv.org/pdf/2207.07051.

Lepori, M., Serre, T., and Pavlick, E. Break It Down: Evidence for Structural Compositionality in Neural Networks. *Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/pdf/2301.10884.

Lepori, M. A., Mozer, M. C., and Ghandeharioun, A. Racing Thoughts: Explaining Contextualization Errors in Large Language Models. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2025. URL https://arxiv.org/pdf/2410.02102.

Li, Z., Jiang, G., Xie, H., Song, L., Lian, D., and Wei, Y. Understanding and Patching Compositional Reasoning in LLMs. *Findings of the Association for Computational Linguistics (ACL)*, 2024. URL https://arxiv.org/pdf/2402.14328.

Lu, M., Zhang, R., Eickhoff, C., and Pavlick, E. Paths Not Taken: Understanding and Mending the Multilingual Factual Recall Pipeline, 2025. URL https://arxiv.org/pdf/2505.20546.

Ma, Z., Hong, J., Gul, M. O., Gandhi, M., Gao, I., and Krishna, R. CREPE: Can Vision-Language Foundation Models Reason Compositionally? *Computer Vision and Pattern Recognition (CVPR)*, 2023. URL https://arxiv.org/pdf/2212.07796.

McCoy, R. T., Yao, S., Friedman, D., Hardy, M. D., and Griffiths, T. L. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences (PNAS)*, 2024. URL https://arxiv.org/pdf/2309.13638.

McCurdy, K., Soulos, P., Smolensky, P., Fernandez, R., and Gao, J. Toward Compositional Behavior in Neural Models: A Survey of Current Views. *Empirical Methods in Natural Language Processing (EMNLP)*, 2024. URL https://aclanthology.org/2024.emnlp-main.524.pdf.

Merullo, J., Eickhoff, C., and Pavlick, E. Language Models Implement Simple Word2Vec-style Vector Arithmetic. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024. URL https://arxiv.org/pdf/2305.16130.

Merullo, J., Smith, N. A., Wiegreffe, S., and Elazar, Y. On Linear Representations and Pretraining Data Frequency in Language Models. *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/pdf/2504.12459.

Mikolov, T., Yih, W.-t., and Zweig, G. Linguistic Regularities in Continuous Space Word Representations. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2013. URL https://aclanthology.org/N13-1090.pdf.

Nguyen, K. A., Schulte im Walde, S., and Vu, N. T. Distinguishing Antonyms and Synonyms in a Pattern-based Neural Network. *European Chapter of the Association for Computational Linguistics (EACL)*, 2017. URL https://arxiv.org/pdf/1701.02962.

nostalgebraist. interpreting GPT: the logit lens, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Partee, B. H. *Compositionality in Formal Semantics*. Blackwell, 2004.

Press, O., Zhang, M., Min, S., Schmidt, L., Smith, N. A., and Lewis, M. Measuring and Narrowing the Compositionality Gap in Language Models. *Empirical Methods in Natural Language Processing (EMNLP)*, 2022. URL https://arxiv.org/pdf/2210.03350.

Quilty-Dunn, J., Porot, N., and Mandelbaum, E. The best game in town: The reemergence of the language-of-thought hypothesis across the cognitive sciences. *Behavioral and Brain Sciences (BBS)*, 2023. URL https://doi.org/10.1017/S0140525X22002849.

Russin, J., McGrath, S. W., Williams, D. J., and Elber-Dorozko, L. From Frege to chatGPT: Compositionality in language, cognition, and deep neural networks, 2024. URL https://arxiv.org/pdf/2405.15164.

Russin, J., Pavlick, E., and Frank, M. J. Parallel trade-offs in human cognition and neural networks: The dynamic interplay between in-context and in-weight learning. *Proceedings of the National Academy of Sciences (PNAS)*, 2025. URL https://www.pnas.org/doi/10.1073/pnas.2510270122.

Shalev, Y., Feder, A., and Goldstein, A. Distributional reasoning in LLMs: Parallel reasoning processes in multi-hop reasoning, 2024. URL https://arxiv.org/pdf/2406.13858.

Szabó, Z. G. Compositionality. In *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2024. URL https://plato.stanford.edu/archives/fall2024/entries/compositionality.

Tenney, I., Das, D., and Pavlick, E. BERT Rediscovers the Classical NLP Pipeline. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. URL https://arxiv.org/pdf/1905.05950.

Tiedemann, J. and Thottingal, S. OPUS-MT – Building open translation services for the World. *European Association for Machine Translation (EAMT)*, 2020. URL https://aclanthology.org/2020.eamt-1.61.pdf.

Todd, E., Li, M. L., Sharma, A. S., Mueller, A., Wallace, B. C., and Bau, D. Function Vectors in Large Language Models. *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/pdf/2310.15213.

Vegner, I., de Souza, S., Forch, V., Lewis, M., and Doumas, L. A. A. Behavioural vs. Representational Systematicity in End-to-End Models: An Opinionated Survey. *Association for Computational Linguistics (ACL)*, 2025. URL https://arxiv.org/pdf/2506.04461.

Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., and Shieber, S. Investigating Gender Bias in Language Models Using Causal Mediation Analysis. *Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/pdf/2004.12265.

Vrandečić, D. and Krötzsch, M. Wikidata: a free collaborative knowledgebase. *Communications of the ACM (CACM)*, 2014. URL https://dl.acm.org/doi/10.1145/2629489.

Wang, B., Yue, X., Su, Y., and Sun, H. Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization. *Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/pdf/2405.15071.

Wattenberg, M. and Viégas, F. B. Relational Composition in Neural Networks: A Survey and Call to Action. *Mechanistic Interpretability Workshop at ICML*, 2024. URL https://arxiv.org/pdf/2407.14662.

Yang, S., Gribovskaya, E., Kassner, N., Geva, M., and Riedel, S. Do Large Language Models Latently Perform Multi-Hop Reasoning? *Association for Computational Linguistics (ACL)*, 2024a. URL https://arxiv.org/pdf/2402.16837.

Yang, S., Kassner, N., Gribovskaya, E., Riedel, S., and Geva, M. Do Large Language Models Perform Latent Multi-Hop Reasoning without Exploiting Shortcuts? *Association for Computational Linguistics (ACL)*, 2024b. URL https://arxiv.org/pdf/2411.16679.

Yang, Z., Li, J., Xia, S., and Hu, X. Internal Chain-of-Thought: Empirical Evidence for Layer-wise Subtask Scheduling in LLMs. *Empirical Methods in Natural Language Processing (EMNLP)*, 2025. URL https://arxiv.org/pdf/2505.14530.

Yu, H., Jeong, M., Shin, J., Moon, H., Park, J., and Choi, S. Towards Zero-Shot Functional Compositionality of Language Models, 2023. URL https://arxiv.org/pdf/2303.03103.

Yu, Z., Belinkov, Y., and Ananiadou, S. Back Attention: Understanding and Enhancing Multi-Hop Reasoning in Large Language Models. *Empirical Methods in Natural Language Processing (EMNLP)*, 2025. URL https://arxiv.org/pdf/2502.10835.

Zhou, Y., Lake, B. M., and Williams, A. Compositional learning of functions in humans and machines. *Annual Meeting of the Cognitive Science Society (CogSci)*, 2024. URL https://arxiv.org/pdf/2403.12201.

A Data Creation

Table 2: List of our tasks, showing x, $g(x)$, and $f(g(x))$ for the random example in Table 1. Tasks with neither $g(x)$ nor $f(g(x))$ are omitted. $f(g(x))$ only shown if distinct from $g(f(x))$.

f	g	x	$g(x)$	$f(g(x))$
Word → Antonym	English → Spanish	bogus	false	—
Word → Antonym	English → German	philosophical	philosophisch	—
Word → Antonym	English → French	excessive	excessive	—
x + 10	2x	699	1398	1408
x + 100	2x	922	1844	1944
x mod 20	2x	891	1782	2
Word → Numeric	2x	one hundred and forty-eight	two hundred and ninety-six	—
Word[:-1]	Word[::-1]	responsible	elbisnopser	elbisnopse
Rotate(RGB, 120°)	RGB → Name	8a735a	dimgray	—

All tasks in Table 2 permit the additional computational pathway $x \to g(x) \to g(f(x))$. Those which don't list $f(g(x))$ are commutative and so $f(g(x)) = g(f(x))$ and applying f to $g(x)$ results in $g(f(x))$. The remaining tasks are not commutative, but their formal construction permits the hop $g(x) \to g(f(x))$ anyway. In particular, $g(f(x))$ equals $g(x)+20$ in `plus-10-times-2`, $g(x)+200$ in `plus-100-times-2`, $g(x) \bmod 40$ in `mod-20-times-2`, and $g(x)$[1:] in `word-truncate-reverse`.

A.1 Task Construction

Antonyms & Translations We obtain a list of antonyms from Todd et al. (2024) — further derived from Nguyen et al. (2017) — and obtain translations from Opus-MT (Tiedemann & Thottingal, 2020).

Factual Relations We obtain various factual relations from WikiData and IMDb Non-Commercial Datasets (Vrandečić & Krötzsch, 2014; IMDb, 2024; Bast & Buchhold, 2017). We apply a number of heuristics to obtain well-known and unambiguous mappings. For example, we filter entities by their "sitelinks" on WikiData or "votes" on IMDB (heuristics for popularity) to obtain well-known subjects. To avoid ambiguity, we identify subjects (songs, books, movies, people, etc.) with a single corresponding object (authors, attended universities, etc.). We omit parks and landmarks that exist in their country's capital. Our exact queries for generating each task can be found in our source code.

Arithmetic We use the range of numbers from 0 to 999 as x in our tasks. These numbers typically result in one token. We use the `num2words` library to obtain a mapping between words and numeric values. We use the list of antonyms as our list of words for the `word-truncate-reverse` task.

`Colors` In the `rgb-rotate-name` task, we randomly sample RGB colors, rotate them $120°$ by their hue, and map the resulting color value to that color's name (using the `webcolors` library and the common CSS 3 specification).

B Implementation Details

Examples & Prompts We prevent sampling of in-context examples that intersect in the variables $\{x, f(x), g(x), g(f(x)), f(g(x))\}$ with the query. And, as mentioned in Sec. 4.1, we exclude examples in Secs. 4 and 5 which overlap in the first token among their variables. So, although $x =$ "excessive" for the `antonym-french` task is listed in Table 2, this trivially shares the same first token as $g(x) =$ "excessive" and would be omitted from our analyses.

Our prompts are tokenized differently when predicting numbers or words, e.g. "`... \n A: 99`" results in `[ ][99]` whereas "`... \n A: modern`" results in `[ modern]`. We accordingly include the trailing space in our prompts when predicting numbers and omit it otherwise. We would then test for the single-token prediction of [99] and [modern] in this example.

Representational analysis In Sec. 4, we analyze the model's computation from $x \to g(f(x))$. Consider the query for "Heartbreak Hotel" $\to$ "1935": i.e. "`... Q: Heartbreak Hotel \n A: `". Here, multiple tokens (`[ Heart][break][ Hotel][ \][n][ A:][ ]`) are central to the computation. We therefore analyze all residual streams for these tokens. At each layer, we measure the signal for each variable by its maximum reciprocal rank across the streams. This procedure yields processing signatures, which quantify the presence of our variables at every layer.

We additionally represent each variable by its first token (since our decoding methods can only produce single-token probabilities). We exclude examples where different variables share the same first token and would be hard to differentiate. For example, $f(x) =$ " modern" and $g(f(x)) =$ " moderno" both share the first token `[ modern]`.

For models with a final RMSNorm, the readout direction in the residual stream is $\gamma \odot u_y$, where γ is the learned gain and u_y is the unembedding; comparisons with residual-stream directions should account for this distinction.

C Compositionality Gap by Prompt Template

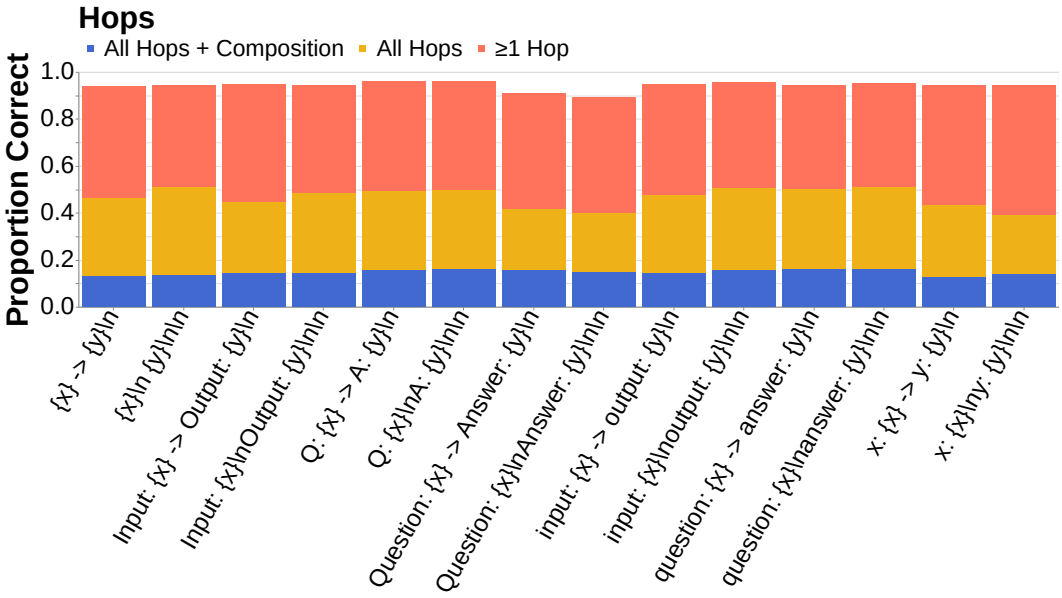


Figure 5: Ablation of Llama 3 (3B) model with different ICL prompt templates under the settings in Fig. 2. We find no significant difference in performance by choice of prompt.

D Compositionality Gap with Task Instructions

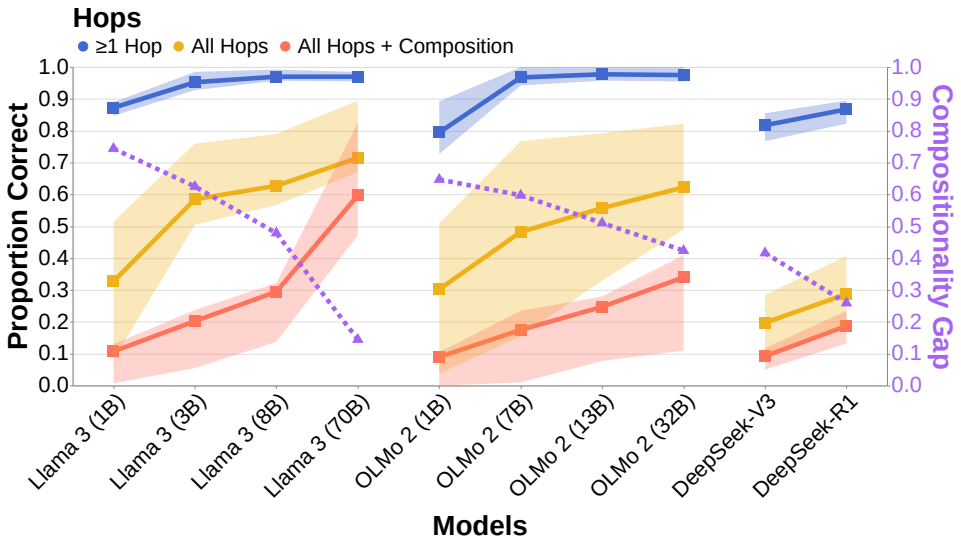


Figure 6: We repeat our experiments in Fig. 2, but always prefix the existing ICL prompt with a task instruction: e.g. "Add 100." for $x \rightarrow f(x)$ or "Add 100, multiply by 2." for $x \rightarrow g(f(x))$. We continue to find a clear compositionality gap in our results. We observe the gap diminishes more steeply w.r.t. model size or reasoning than in the original experiments. Llama 3 (70B) is also substantially closer to closing the gap. We omit the instruction-tuned Llama model and OpenAI models, whose outputs are harder to constrain (and thus evaluate) given the additional instruction prompt.

E Compositionality Gap by Size

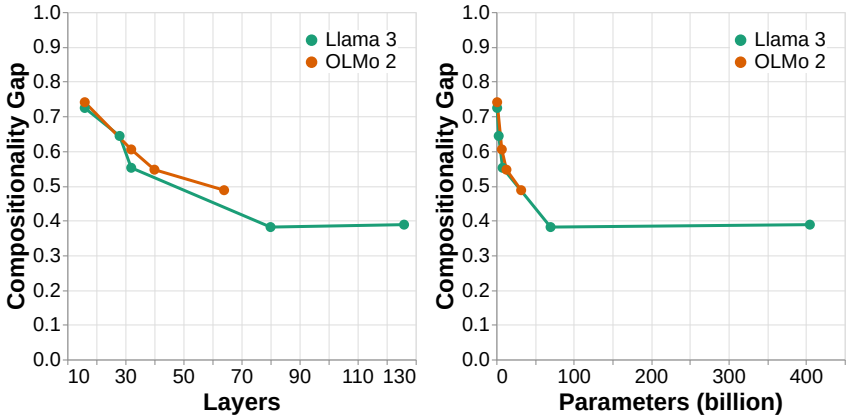


Figure 7: We illustrate the monotonically diminishing improvements to the compositionality gap resulting from increased model size (layers and parameters). We re-visualize results for the OLMo 2 and Llama 3 model families from Fig. 2.

F Compositionality Gap on Additional Models

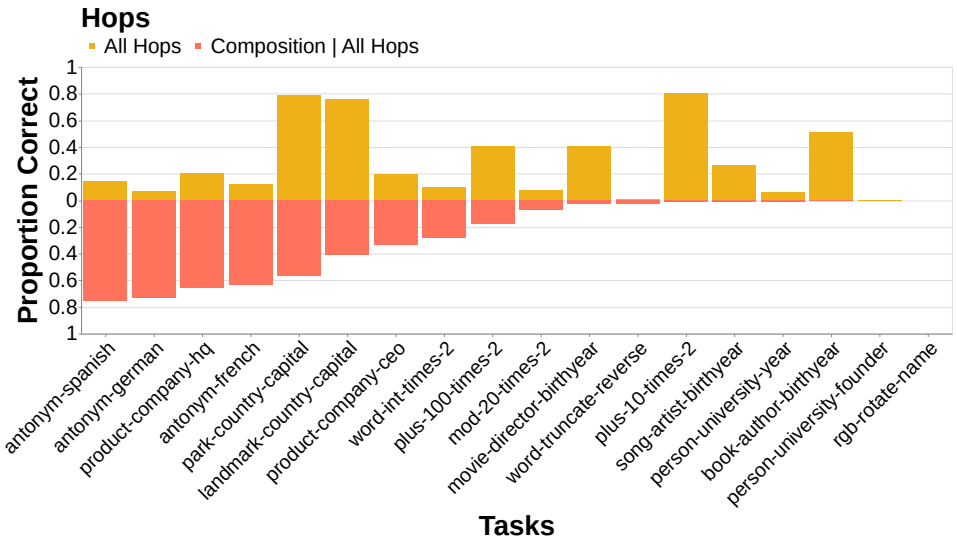


Figure 8: Compositionality gap for Llama 3 (3B) Instruct on all our tasks. Correlation between red and yellow bars is $r^2 = 0.00$.

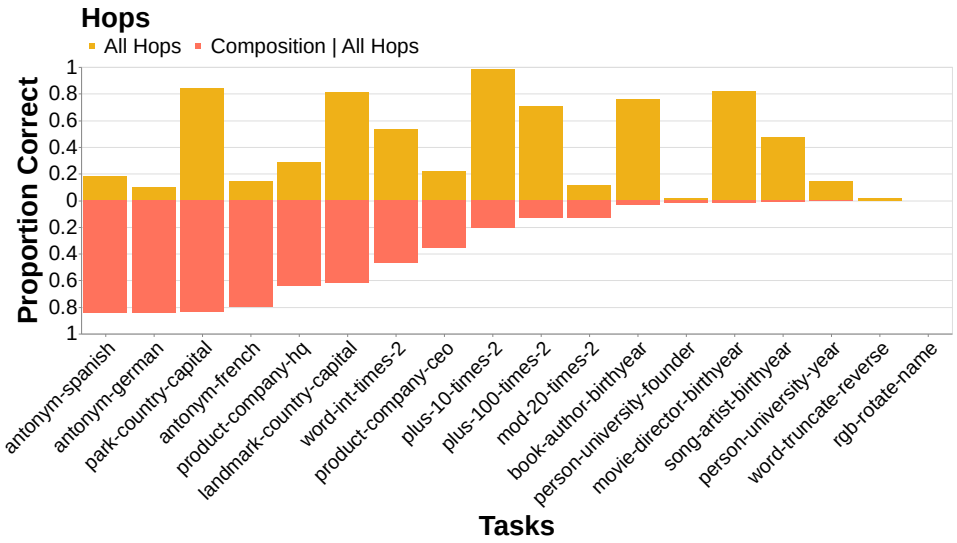


Figure 9: Compositionality gap for Llama 3 (8B) on all our tasks. Correlation between red and yellow bars is $r^2 = 0.00$.

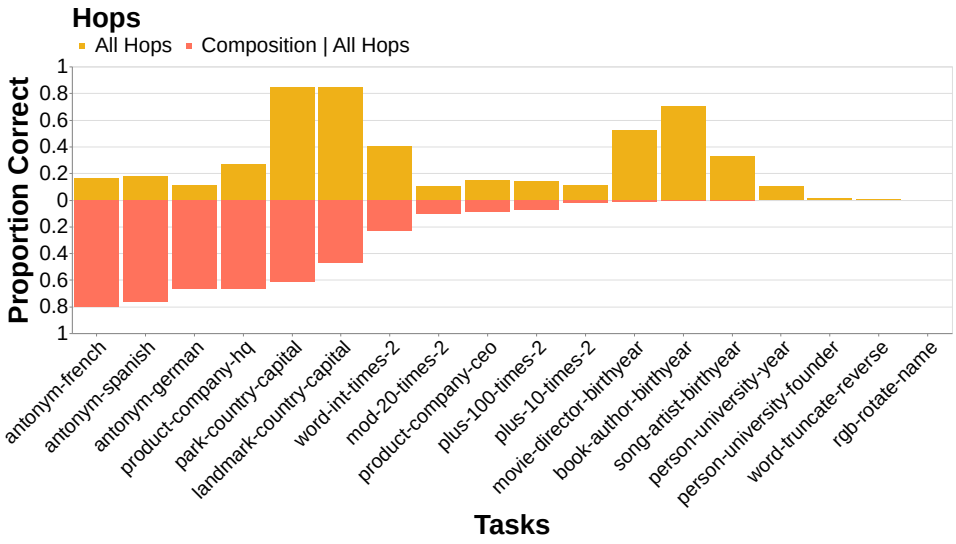


Figure 10: Compositionality gap for OLMo 2 (7B) on all our tasks. Correlation between red and yellow bars is $r^2 = 0.05$.



Figure 11: Compositionality gap for OLMo 2 (13B) on all our tasks. Correlation between red and yellow bars is $r^2 = 0.01$.

G Processing Signatures (Correct)

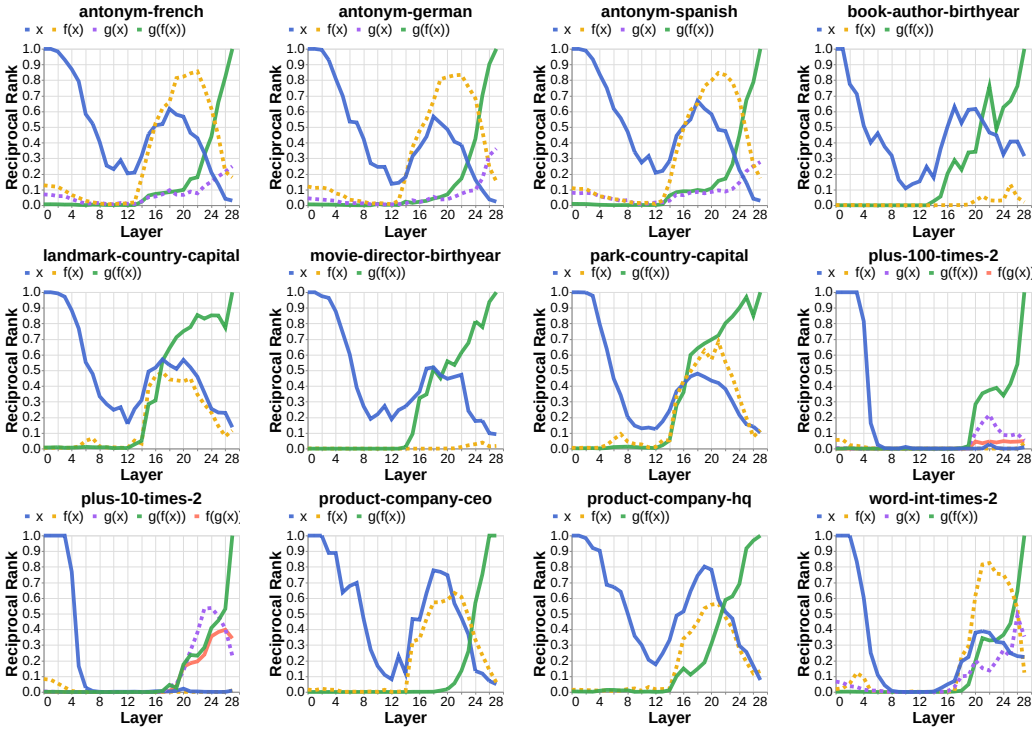


Figure 12: Aggregate processing signatures for each of our tasks, in which Llama 3 (3B) correctly solves all hops and the composition for at least 10 examples.

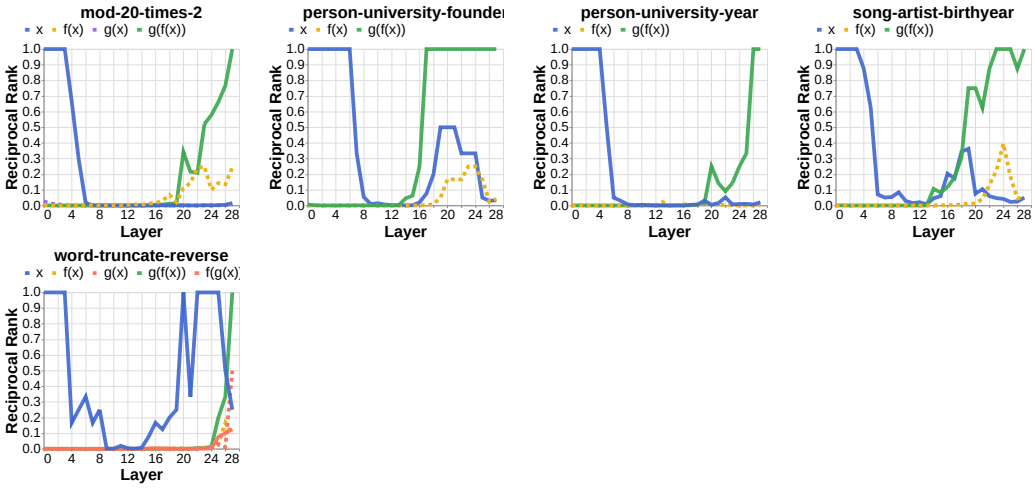


Figure 13: Aggregate processing signatures for each of our tasks, in which Llama 3 (3B) correctly solves all hops and the composition for less than 10 examples.

H Processing Signatures (Incorrect)

Although we see a difference in aggregate processing signatures (Figs. 3a and 3b), where the signal for the intermediate variables is clearer in the correct cases than the incorrect cases, this does not appear to be generally true (and is more likely due to data imbalances). We can see significant presence of the intermediate variables when considering incorrect examples, de-aggregated by task (Fig. 14).

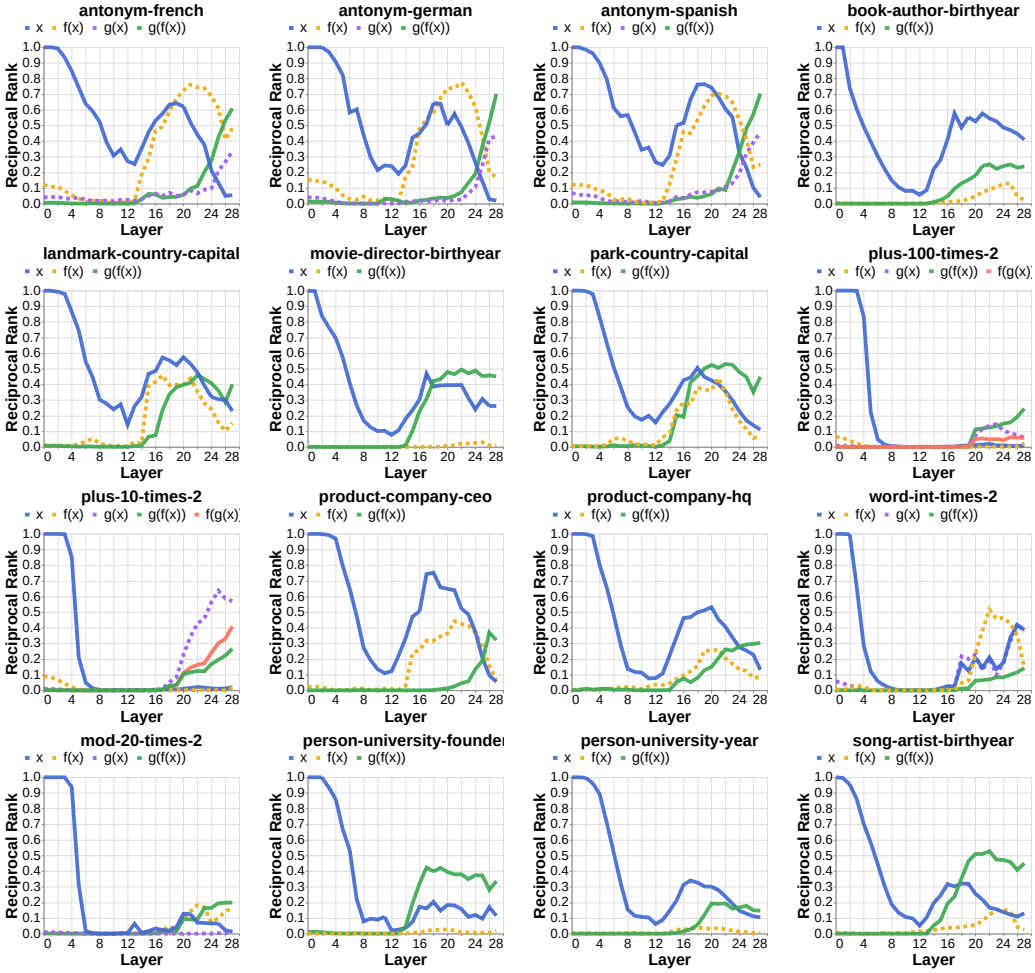


Figure 14: Aggregate processing signatures for each of our tasks, in which Llama 3 (3B) correctly solves all hops but not the composition for at least 10 examples.

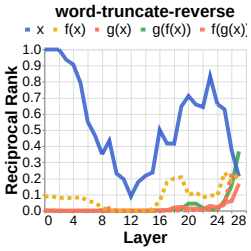
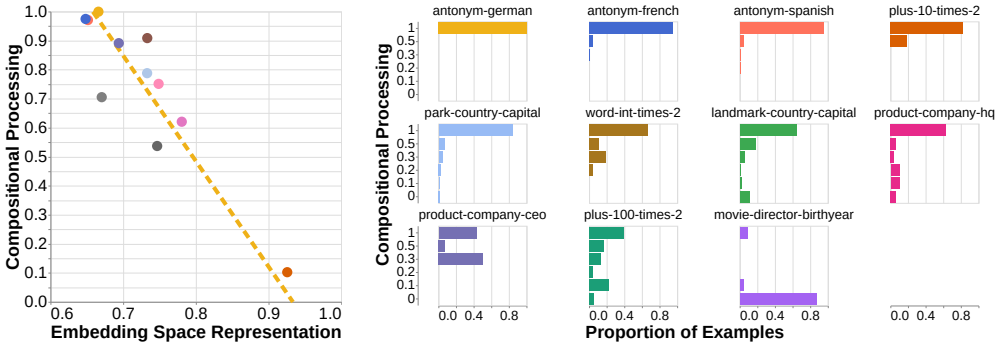


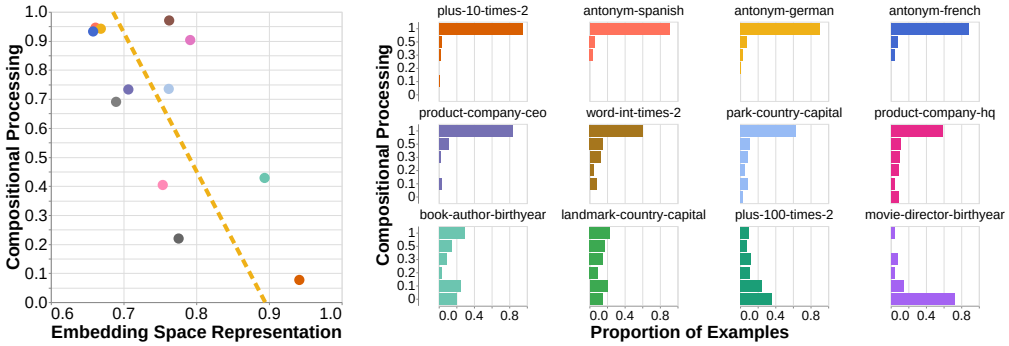
Figure 15: Aggregate processing signatures for each of our tasks, in which Llama 3 (3B) correctly solves all hops but not the composition for less than 10 examples.

I Relationship between Mechanisms & Geometry on Additional Models



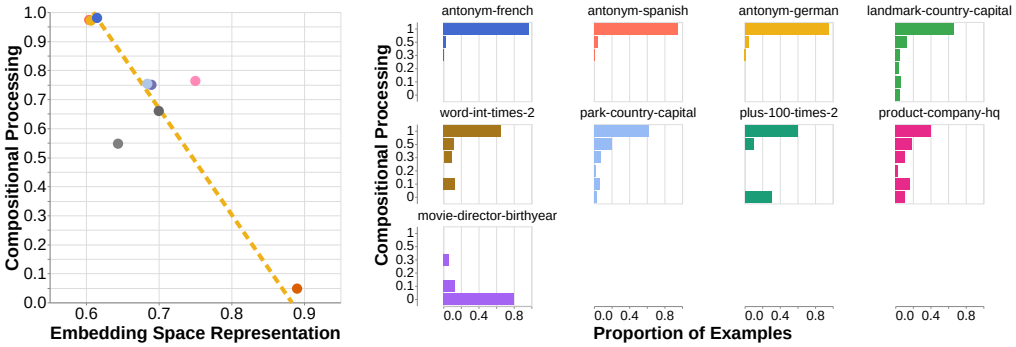
(a) (b)

Figure 16: For Llama 3 (3B) Instruct. Correlation has $r^2 = 0.83$. Accuracy is weakly correlated with these compositional processing ($r^2 = 0.14$) and representation ($r^2 = 0.21$) metrics. 81% of examples have very low or high indicators for the compositional mechanism.



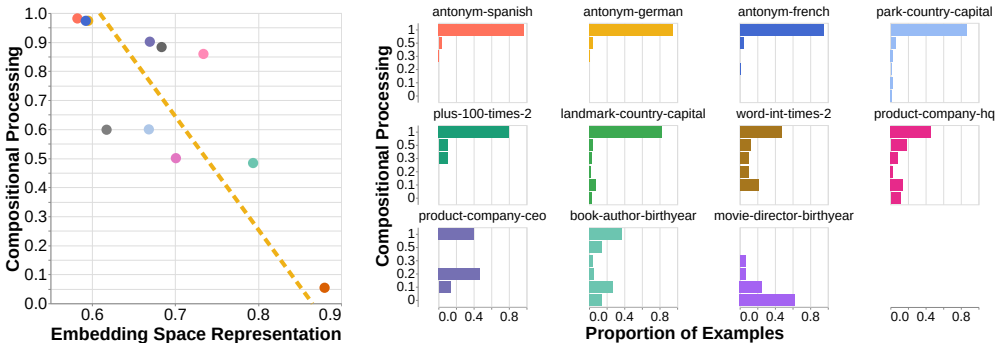
(a) (b)

Figure 17: For Llama 3 (8B). Correlation has $r^2 = 0.52$. Accuracy is weakly correlated with these compositional processing ($r^2 = 0.08$) and representation ($r^2 = 0.32$) metrics. 79% of examples have very low or high indicators for the compositional mechanism.



(a) (b)

Figure 18: For OLMo 2 (7B). Correlation has $r^2 = 0.78$. Accuracy is weakly correlated with these compositional processing ($r^2 = 0.14$) and representation ($r^2 = 0.09$) metrics. 85% of examples have very low or high indicators for the compositional mechanism.



(a) (b)

Figure 19: For OLMo 2 (13B). Correlation has $r^2 = 0.66$. Accuracy is weakly correlated with these compositional processing ($r^2 = 0.45$) and representation ($r^2 = 0.31$) metrics. 77% of examples have very low or high indicators for the compositional mechanism.

J Token Identity Patchscope

Here, we repeat the analyses in Secs. 4 and 5, but use the token identity patchscope (Ghandeharioun et al., 2024) instead of logit lens. This method is proposed as one that is more closely aligned with a language model's computation than other methods (such as logit lens).

We would specifically like to use this method to decode a representation into vocabulary-space logits. To do so, we prompt a model with the "token identity prompt", in which random tokens are repeated twice each, such as "`[A] [A] ; [B] [B] ; ... ; [?]`". We patch our representation of interest into the residual stream of this forward pass (at the corresponding layer and final token position). The language modeling logits resulting from our intervention then serve as the decoding for our representation.

We generally find similarities with our logit lens analyses: in tasks with "compositional" processing signatures, we continue to see growth of the signals for the intermediate variables with or before that for $g(f(x))$. Please zoom in to observe simultaneous growth, which may be difficult to see due to overlapping lines. And, although these plots may show growth of $f(x)$ and $g(f(x))$ in the same layers, recall that these computations can occur in different (e.g. earlier or later) residual streams (Appendix B).

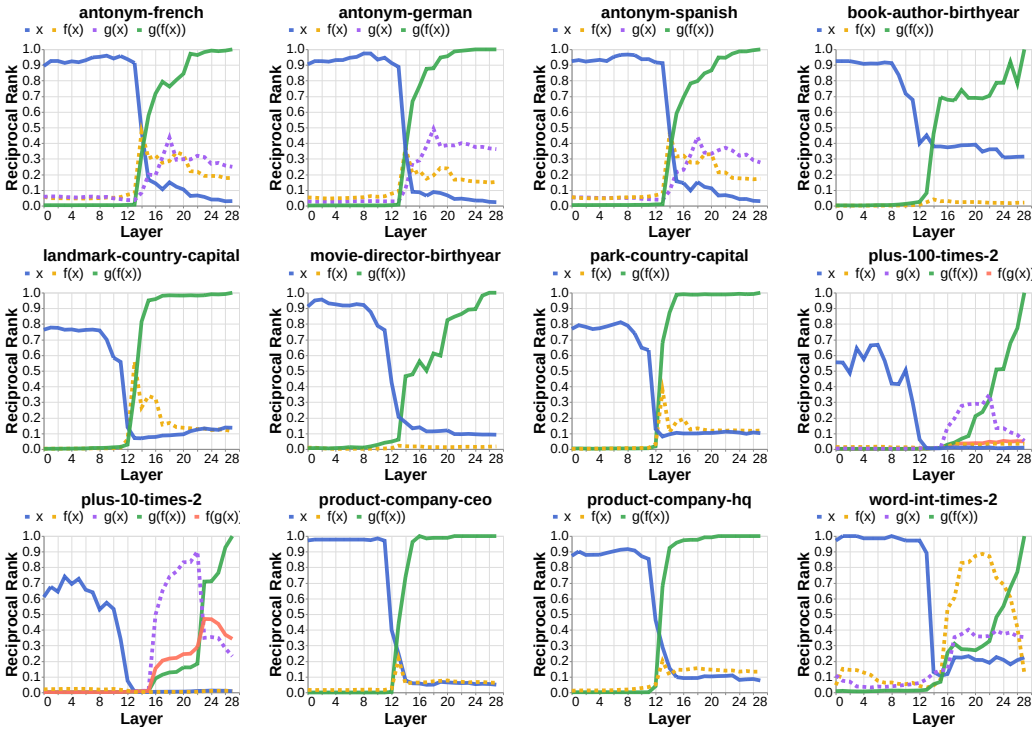


Figure 20: Aggregate processing signatures (using the token identity patchscope) for each of our tasks, in which Llama 3 (3B) correctly solves all hops and the composition for at least 10 examples.

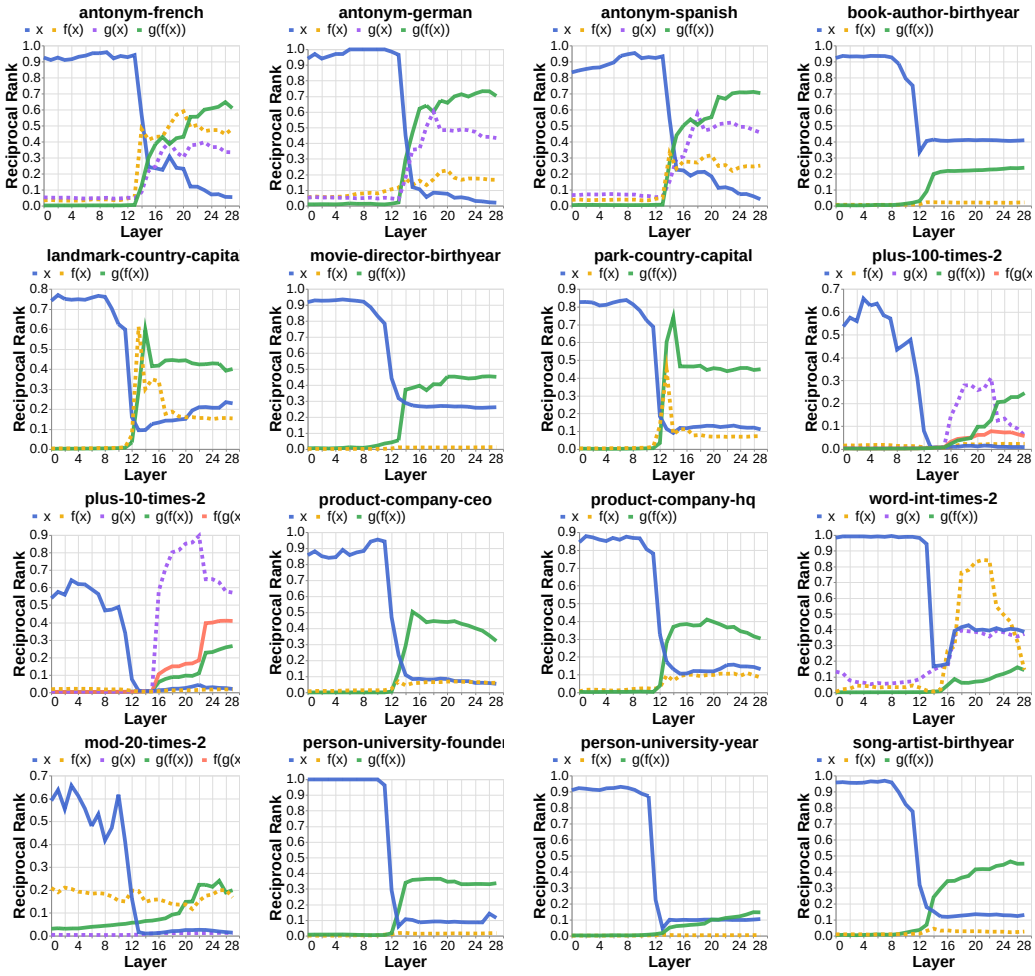


Figure 21: Aggregate processing signatures (using the token identity patchscope) for each of our tasks, in which Llama 3 (3B) correctly solves all hops but not the composition for at least 10 examples.

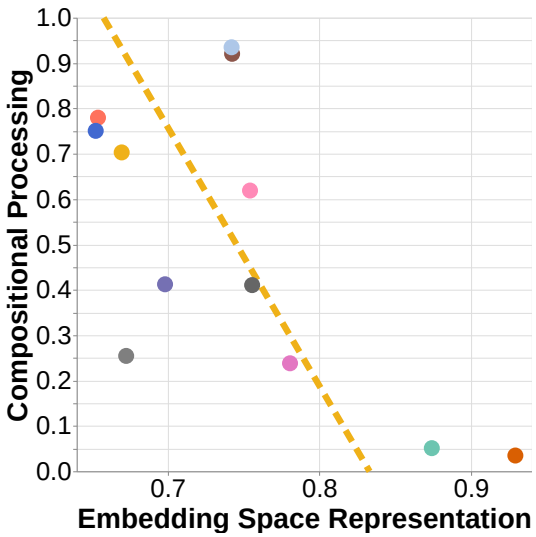


Figure 22: Correlation across tasks ($r^2 = 0.42$) for embedding space task representations and compositional processing. Analogous to Fig. 4a, using the intermediate variable metric from the token identity patchscope.

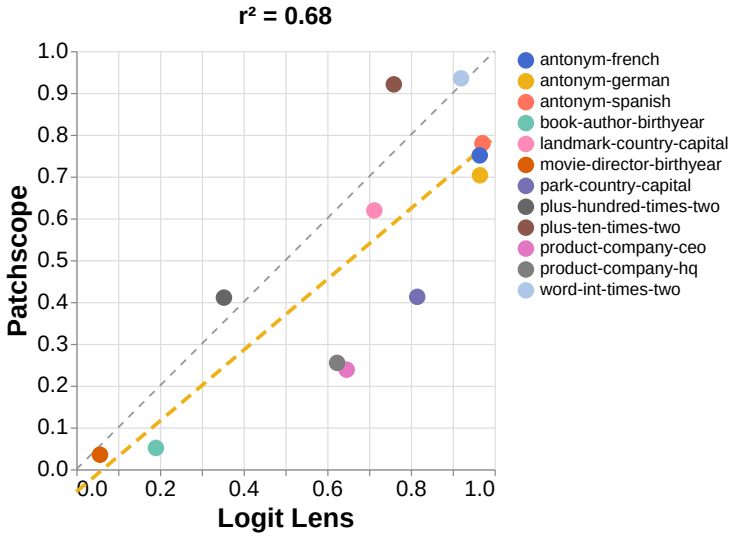


Figure 23: Task-level processing signatures from the token identity patchscope, compared with logit lens ($r^2 = 0.68$). Each point is a task for which Llama 3 (3B) correctly solves all hops and the composition for at least 10 examples.

K Jacobian Lens

Here, we repeat the analyses in Secs. 4 and 5, but use Jacobian lens (Gurnee et al., 2026) instead of logit lens. This method accounts for changes in representational coordinates between intermediate layers and the final layer.

We would specifically like to use this method to decode a representation into vocabulary-space logits. To do so, we estimate a Jacobian matrix for each layer, averaging derivatives from source activations to final-layer activations at the same or later token positions. We estimate these matrices using 1,000 chunks of 256 tokens from WikiText-103 and share them across tasks. For a representation h_ℓ, we apply the corresponding matrix J_ℓ, followed by the model's final normalization and language modeling head. The resulting logits then serve as the decoding for our representation.

We generally find similarities with our logit lens analyses: tasks with compositional signatures continue to show intermediate-variable signals, while tasks with direct signatures do not acquire a strong intermediate signal. Task-level scores correlate strongly with logit lens ($r^2 = 0.95$; Fig. 27). As in Appendix J, growth of intermediate and answer signals in the same layers may reflect computations at different token positions (Appendix B).

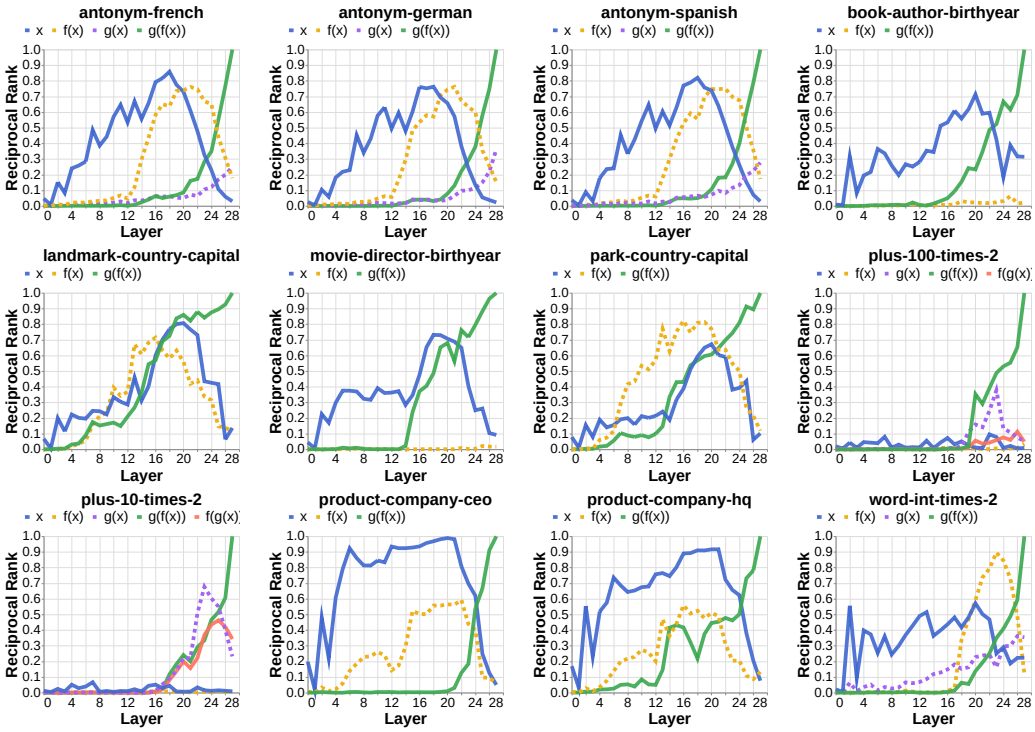


Figure 24: Aggregate processing signatures (using Jacobian lens) for each of our tasks, in which Llama 3 (3B) correctly solves all hops and the composition for at least 10 examples.

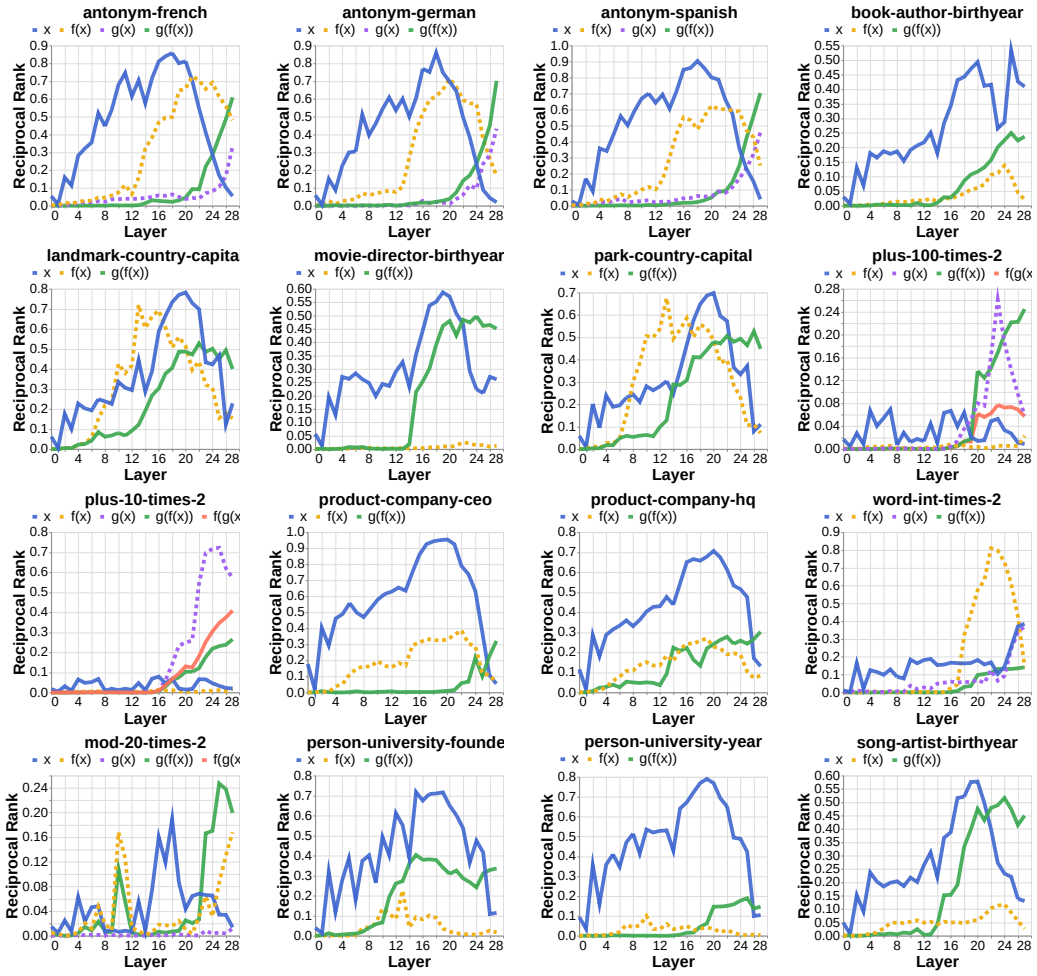


Figure 25: Aggregate processing signatures (using Jacobian lens) for each of our tasks, in which Llama 3 (3B) correctly solves all hops but not the composition for at least 10 examples.

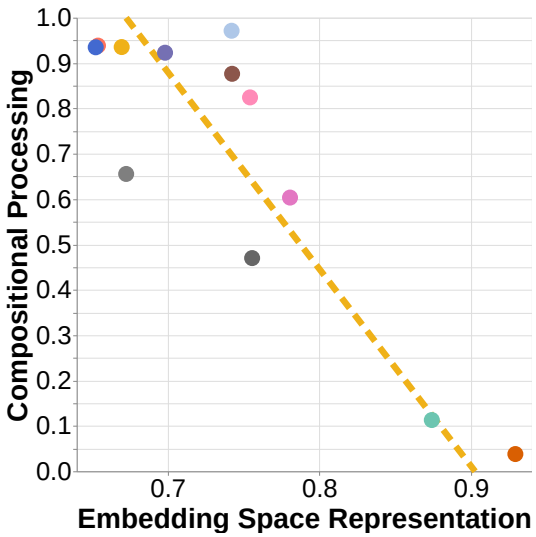


Figure 26: Correlation across tasks ($r^2 = 0.76$) for embedding space task representations and compositional processing. Analogous to Fig. 4a, using the intermediate variable metric from Jacobian lens.

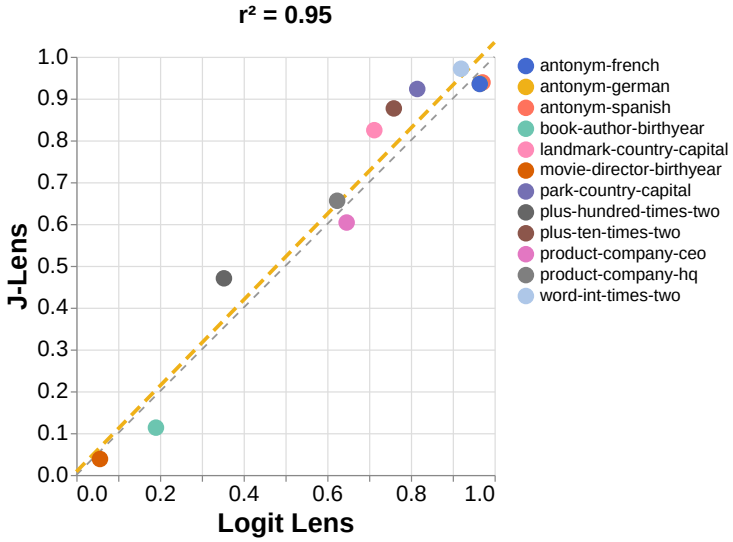


Figure 27: Task-level processing signatures from Jacobian lens, compared with logit lens ($r^2 = 0.95$). Each point is a task for which Llama 3 (3B) correctly solves all hops and the composition for at least 10 examples.

L Relationship between Mechanisms and Representations of Hops

Similarly to the experiment in Sec. 5 and Fig. 4a, we investigate the relationship between our compositionality heuristic and embedding space representations for variations of the hops (rather than of the compositional task).

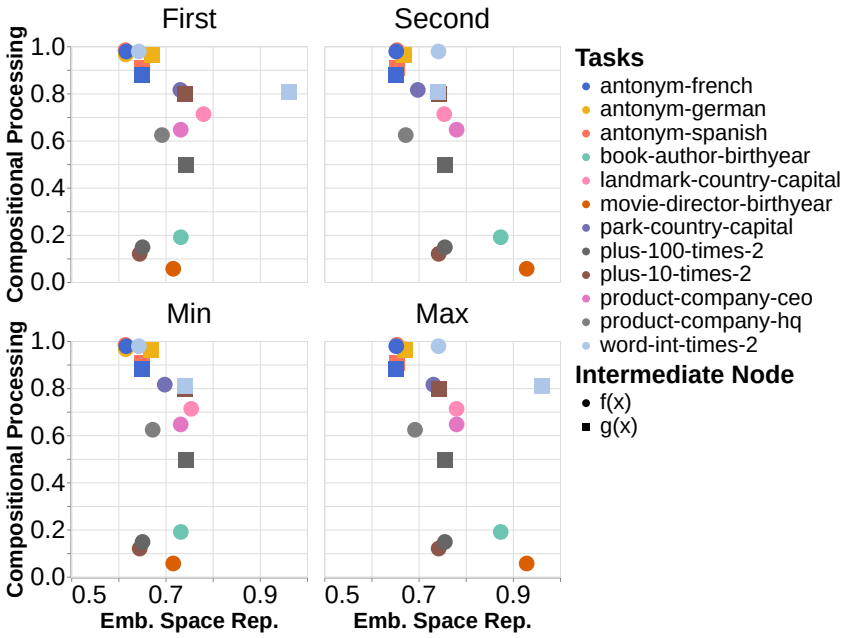


Figure 28: Relationships between compositional processing and embedding space representations for the hops. $r^2 = 0.01$ against the representation of the first hop; $r^2 = 0.58$ against the second hop; $r^2 = 0.10$ using the minimum representation between the hops; and $r^2 = 0.32$ using the maximum.

M Causality of Intermediate Variables

We would like to determine whether the variables, $f(x)$ and $g(x)$, we identify in models' intermediate representations have a causal effect on the outcome. To do so, we use activation patching (Vig et al., 2020), a common method for conducting causal interventions in interpretability, and patch representations across tasks.

We first identify tasks with the same f but different g, such as `antonym-spanish` ($g \circ f$) and `antonym-german` ($g' \circ f$). For some x and x', we extract a single intermediate representation from the forward pass of $g'(f(x'))$ and patch it into the forward pass of $g(f(x))$. On average (over many x and x'), we measure the causal effects on the predictions $g(f(x))$, $g(f(x'))$, $g'(f(x))$, and $g'(f(x'))$.

We extract the representation from $g'(f(x'))$ at the position and layer where $f(x')$ or $g(x')$ have the highest reciprocal rank (and only use instances where this value is at least 0.5). We patch this representation into the forward pass for $g(f(x))$ at the median location where intermediate values are highest (layer 18 and 71st percentile query token position; identified among variables that reach RR ≥ 0.5). We apply this intervention to two groups: instances with intermediate values that reach a peak RR ≤ 0.2 and ≥ 0.5. In other words, instances with direct or compositional processing signatures.

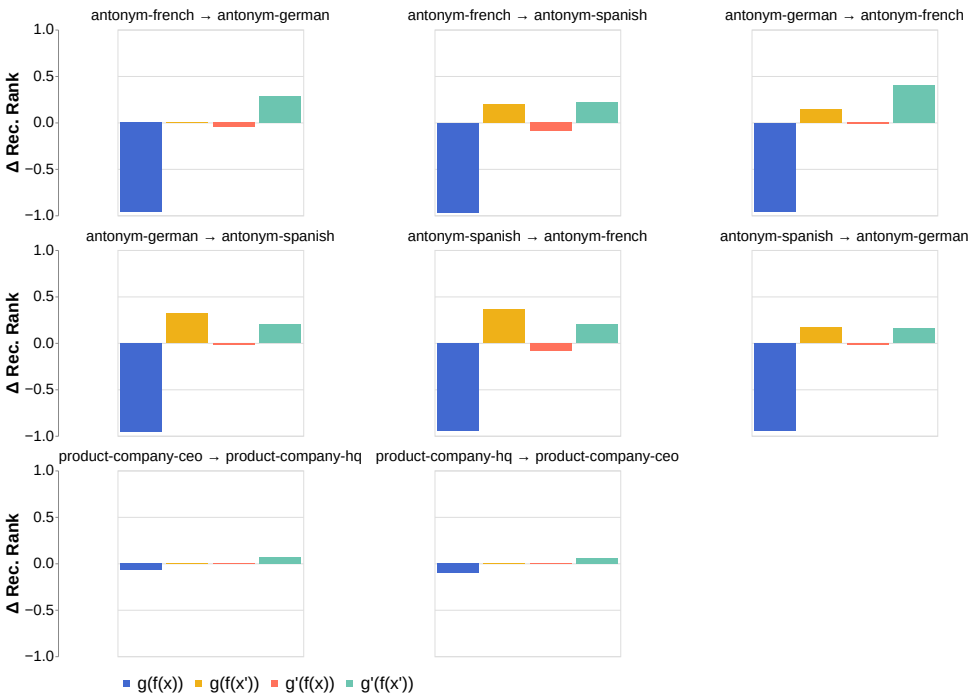


Figure 29: Causal effects on predicted values after patching from $g'(f(x'))$ to $g(f(x))$ for instances with compositional processing signatures.



Figure 30: Causal effects on predicted values after patching from $g'(f(x'))$ to $g(f(x))$ for instances with direct processing signatures.

The Antonym–Translation tasks (which tend to have compositional signatures) show the most significant causal effect: on average, $g(f(x))$ and $g'(f(x))$ decrease by -0.95 and -0.4, and $g(f(x'))$ and $g'(f(x'))$ increase by 0.20 and 0.24. The effect on $g(f(x'))$ clearly implicates the existence and causality of $f(x')$ in the patched activation; that on $g'(f(x'))$ indicates the additional existence of either itself or the function vector (Todd et al., 2024) for g' in that representation. The causal effects on compositional instances of `product-company-hq` and `product-company-ceo` are smaller.

But we can also see a clear difference between the causal effects on the compositional and direct instances. Indeed, the effects on `product-company-hq` and `product-company-ceo` are larger in their compositional instances. Patching activations from `plus-10-times-2` into `plus-100-times-2` primarily decreases $g(f(x))$ and increases $g'(f(x'))$, perhaps only implying the existence of the representation for $g'(f(x'))$ in the patched activation.

N Additional Controls and Prompt Dependence

We conduct the following controls with Llama 3 (3B). Unless otherwise stated, we use the examples for which the model solves all hops and the composition under the original ICL prompt. Compositional and direct receiver groups have maximum intermediate-variable RR ≥ 0.5 and ≤ 0.2,

respectively; examples between these thresholds are included only in the full population. We report reciprocal rank for the first token of each variable, following Appendix B.

N.1 Overview

Activation patching provides complementary evidence (Appendix M). Patching single-hop country representations into the two-hop park and landmark tasks increases the counterfactual capital's reciprocal rank by 0.126 ± 0.034 in compositional-signature examples (mean and SD across two task pairs). Within-task effects also persist when we exclude donor activations with a decodable answer. We patch the entire residual stream, so these interventions do not isolate a unique intermediate-variable subspace.

The signature also depends on the prompt. Replacing ICL with explicit zero-shot instructions increases the intermediate signal for `book-author-birthyear` and `plus-100-times-2` by 0.323 ± 0.068 RR (mean and SD across the two tasks), while `movie-director-birthyear` changes by only 0.038. In a separate in-context nonce-to-country lookup, changing the requested mapping reduces the original country's signal from 0.926 to 0.002, supporting task selectivity.

N.2 Decoding and Selectivity

The token identity patchscope and Jacobian lens yield task-level scores correlated with logit lens ($r^2 = 0.68$ and 0.95). Neither method reveals a strong intermediate signature where logit lens suggests absence, although they need not cover all representational subspaces.

We test whether $f(x)$ appears selectively in the computation $x \to g(f(x))$. For compositional tasks, its signal is generally weaker in $x \to g(x)$ (median difference: -0.68 RR across five tasks) and $x \to x$ (median: -0.54 RR, excluding `park-country-capital` and `product-company-hq`). In these two exceptions, the copy-context difference is more modest (-0.06 RR): countries and companies may be represented alongside parks and products even when they are not requested. Direct tasks generally have low signals in both control contexts.

We also score intermediate values from 50 other subjects in each query's residual streams. Mean scores are 0.070 ± 0.056 across eight compositional-signature tasks and 0.028 ± 0.032 across four direct-signature tasks (sample SD across task means). The excluded `word-int-times-2` task has a higher score for other subjects, 0.56, but the queried subject's intermediate is still represented most strongly (0.92).

Finally, we construct a nonce-country-capital task with a prompt-defined mapping from 40 nonce words to countries. We keep the queried nonce and prompt format fixed, but change the mapping from f to f'. The original country $f(x)$ is strongly decoded in $x \to g(f(x))$ (0.93 RR), but not in $x \to g(f'(x))$ (approximately 0.00 RR), even though it remains in the prompt. This demonstrates task selectivity for an in-context lookup; it need not generalize to every in-weights relation.

N.3 Patching Controls

We replace the entire residual-stream vector at one layer and query token position. Our original across-task design (Appendix M) extracts donors with an intermediate peak RR of at least 0.5 and applies them at layer 18 and the 71st percentile query token position, the median peak location in compositional-signature examples. This receiver location is fixed for both groups. The following experiments vary the donor computation or patch location.

Single-hop donors. We extract country representations from single-hop park-country or landmark-country runs and patch them into the other task's two-hop country-capital computation. The counterfactual answer is the capital of the donor country. Applying patches at early layers leaves subsequent layers available to compute the second hop. At layer 2, the mean effect for compositional receivers is $+0.126 \pm 0.034$ (sample SD across the two task pairs); direct receivers show much smaller effects (Table 3).

Within-task donors. We also patch between examples of the same two-hop task, selecting a donor location with a decodable intermediate but donor-answer RR ≤ 0.1. We apply the patch at the receiver's peak intermediate location. Positive effects persist in compositional-signature examples

Receiver task	Signature	n	ΔRR
park-country-capital	compositional	316	+0.150
park-country-capital	direct	20	+0.006
landmark-country-capital	compositional	308	+0.102
landmark-country-capital	direct	63	+0.012

Table 3: Single-hop donors applied at layer 2, at the receiver's peak query position. Entries are mean changes in the counterfactual capital's RR; n counts receiver examples.

after this filter (Table 4). Across the full populations of landmark, product-company-ceo, plus-10, and plus-100, the mean effect is $+0.139 \pm 0.045$ (sample SD across task means); the table separates compositional and direct receiver groups. Low decoded answer RR does not guarantee that all answer information is absent.

Receiver task	n_C	ΔRR_C	n_D	ΔRR_D	n_{all}	ΔRR_{all}
landmark-country-capital	226	+0.156	41	-0.032	285	+0.125
product-company-ceo	26	+0.212	10	-0.003	45	+0.122
book-author-birthyear	3	+0.000	6	-0.056	10	-0.033
movie-director-birthyear	2	+0.000	24	+0.000	26	+0.000
plus-10-times-two	29	+0.175	1	-0.000	32	+0.206
plus-100-times-two	7	+0.283	9	+0.005	22	+0.105

Table 4: Within-task patching with answer-filtered donors. C and D denote compositional and direct receiver groups; "all" also includes receivers between the thresholds. Entries are mean changes in donor-answer RR.

As a control for subject transfer, we patch from copy contexts ($x' \to x'$) into $x \to g(f(x))$. These patches produce approximately 0.00 change in the reciprocal rank of $g(f(x'))$ in the tested pairs. We also patch between book-author-birthyear and movie-director-birthyear examples whose authors and directors share the same set of people. Mean effects on the donor person's birthyear are approximately -0.01 RR in both directions. These controls do not guarantee that a whole-stream patch carries only the intermediate value.

Extraction-site sweeps. We repeat within-task patching using direct-signature donors and receivers, sweeping layers 16–21 and query-token percentiles 56–80, the interquartile region of intermediate peak locations. Each donor location is patched at the corresponding layer and percentile in the receiver. Taking the maximum effect over locations for each receiver gives mean changes of $+0.04$ RR for book-author-birthyear and $+0.09$ for movie-director-birthyear (Table 5). These are maxima over a search region, rather than fixed-location effects; they do not rule out other locations or representational subspaces.

Task	n	ΔRR (mean $\pm$ SD)
book-author-birthyear	6	$+0.039 \pm 0.034$
movie-director-birthyear	25	$+0.091 \pm 0.165$

Table 5: Direct-to-direct within-task location sweeps. Each entry is the mean $\pm$ sample SD across receivers of their maximum ΔRR over locations.

Donor depth and a different second hop. We extract nonce-country-capital donor representations at successive layers and patch them into park/landmark-country-currency/continent receivers. The receiver has no nonce-country legend, and the donor's capital answer differs from the receiver's counterfactual currency or continent. Thus, neither copying the nonce nor copying the donor answer suffices. Effects are negligible for donor layers 2–12 and emerge around layer 14, alongside the country decoding signal. This links decodability and causal transfer in the lookup task, while leaving open the possibility of other intermediate representations in the original in-weights tasks.

N.4 Zero-shot Instructions

We replace the ten demonstrations with one explicit, natural-language task instruction and retain the same examples as the ICL analysis. We do not re-filter for zero-shot success. Table 6 reports the paired change in the intermediate-variable score, including variation across examples. Book-author-birthyear and plus-100-times-2 gain substantial intermediate signal ($+0.323 \pm 0.068$ RR, mean and sample SD across the two tasks), while movie-director-birthyear changes little. Thus the processing signature depends on the task, example, and prompt together.

Task	n	ICL RR	Zero-shot RR	ΔRR (mean $\pm$ SD)
antonym-spanish	197	0.971	0.896	-0.075 ± 0.223
antonym-german	115	0.965	0.925	-0.040 ± 0.247
antonym-french	132	0.965	0.925	-0.040 ± 0.223
book-author-birthyear	10	0.190	0.562	$+0.371 \pm 0.382$
landmark-country-capital	401	0.713	0.436	-0.277 ± 0.381
park-country-capital	372	0.815	0.708	-0.107 ± 0.303
movie-director-birthyear	27	0.057	0.094	$+0.038 \pm 0.163$
product-company-ceo	45	0.646	0.314	-0.332 ± 0.257
product-company-hq	63	0.624	0.649	$+0.025 \pm 0.346$
plus-10-times-two	55	0.759	0.729	-0.030 ± 0.504
plus-100-times-two	22	0.353	0.627	$+0.274 \pm 0.352$
word-int-times-two	35	0.920	0.767	-0.153 ± 0.323

Table 6: Prompt dependence of maximum intermediate RR on tasks with at least ten analyzed examples. The final column gives the mean and sample SD of paired, per-example changes, rather than variation across tasks.

N.5 Exposure Proxies

We also examine input–answer co-occurrence in samples of OLMo's training data and prediction loss as proxies for exposure. Across ten tasks, median log co-occurrence has little association with Llama's intermediate-variable signature ($r^2 \approx 0.00$); these counts are a cross-corpus proxy for Llama. First-token prediction loss under task-specific questions also has a weak task-level correlation ($r^2 \approx 0.10$). These checks do not rule out other distributional confounders or establish that geometry causes a model to choose a processing mechanism.

O Additional Model Evaluation

We repeat the four-task evaluation from Fig. 2 with Claude Opus 5, GPT-5.6 Sol, and Gemini 3.6 Flash, using 100 queries per task and ten demonstrations. We use the same "Answer with only the answer, no explanation" prefix for all three models. The two conditions use no reasoning budget or medium reasoning effort through the providers' APIs. These are behavioral measurements; we do not apply the residual-stream analyses to these proprietary models.

These results are behavioral and operate on relatively simple tasks for frontier models: they don't necessarily imply greater engagement in latent compositional processing.

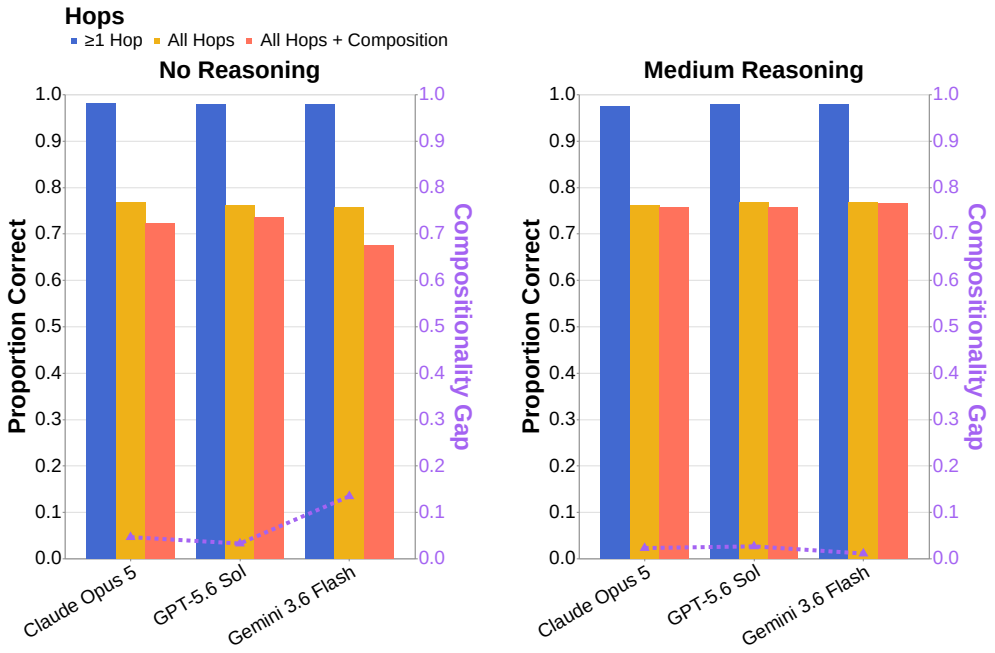


Figure 31: Additional model evaluation on the four tasks from Sec. 3. Bars show mean proportions of examples satisfying each correctness condition; the purple line gives the mean task-level compositionality gap across the four tasks.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Sec. 1 state the behavioral and representational findings. We distinguish decoded signatures and predictive geometry from identification of a unique causal mechanism.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The Limitations section and Appendix N discuss decoder coverage, intervention specificity, prompt dependence, and the scope of the geometric predictor.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper reports empirical analyses rather than theoretical results requiring proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Secs. 2, 3.1 and 4.1 and Appendices B and N describe the datasets, prompts, correctness filters, metrics, and interventions. The accompanying code provides experiment configurations and analysis scripts.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: We provide task-generation code, experiment configurations, and analysis scripts with the supplementary material. Model weights and source datasets must be obtained from their respective providers.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Sec. 2 and Appendices B, N and O specify the evaluation and intervention settings; the code provides the task queries, seeds, and model configurations.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Fig. 2 reports interquartile ranges across tasks. Appendix N reports sample sizes and explicitly distinguishes SD across examples from SD across task means; these are not estimates across independent model training runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: The experiment configurations specify resource requests and the run records include individual job information, but we do not provide a complete accounting of compute time for every experiment or the full project.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and the research conforms to it.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [No]

    Justification: The Discussion considers practical implications for compositional processing and generalization, but the paper does not provide a separate assessment of broader societal impacts.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We analyze existing language models and factual and synthetic tasks; we do not release a new generative model or a dataset requiring special access safeguards.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [No]

    Justification: Appendix A credits the source datasets and software, but the paper does not include a complete per-asset inventory of licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: Appendices A and B describes task construction and evaluation. The accompanying repository documents the experiment entry points, saved summaries, and requirements for reproducing the additional controls.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The study does not involve crowdsourcing or experiments with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The study does not involve human-subject experiments requiring IRB review.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: Pretrained LLMs are the objects of evaluation described in Secs. 3 and 4 and Appendix O, rather than a separate component used to develop the research methodology. Editing assistance does not change the experimental methods.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.