CHARACTERIZING PATTERN MATCHING AND ITS LIMITS ON COMPOSITIONAL TASK STRUCTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite impressive capabilities, LLMs often exhibit surface-level pattern-matching behaviors, evidenced by OOD generalization failures in compositional tasks. However, behavioral studies commonly employ task setups that allow multiple generalization sources (e.g., algebraic invariances, structural repetition), obscuring a precise and testable account of how well LLMs perform generalization through pattern matching and their limitations. To address this ambiguity, we first formalize pattern matching as functional equivalence, i.e., substituting input fragments observed to result in identical outputs in shared contexts. Then, we systematically study how decoder-only Transformer and Mamba behave in controlled tasks with compositional structures that isolate this mechanism. Our formalism yields predictive and quantitative insights: (1) Instance-wise success of pattern matching is tightly ordered by the number of contexts witnessing the relevant functional equivalence. (2) We derive and empirically confirm that the training data required for learning a two-hop structure grows at least quadratically with token-set size. The power-law scaling exponent agrees with predictions and remains stable across 20× parameter scaling and different architectures. (3) Path ambiguity is a structural barrier: when a variable influences the output via multiple paths, models fail to form unified intermediate state representations, impairing accuracy and interpretability. (4) Chain-of-Thought reduces data requirements yet does not resolve path ambiguity. Hence, we provide a predictive, falsifiable boundary for pattern matching and a foundational diagnostic for disentangling mixed generalization mechanisms.

1 INTRODUCTION

Despite the remarkable performance of Large Language Models (LLMs) (Brown et al., 2020; Touvron et al., 2023), compositional generalization studies report their "pattern-matching" behaviors, i.e., models exploiting local statistical regularities between input fragments and outputs in some cases (Loula et al., 2018; Johnson et al., 2017; Berglund et al., 2024; Wang et al., 2024; Mirzadeh et al., 2025; Keysers et al., 2020; Csordás et al., 2022). However, behavioral studies commonly employ task setups that allow multiple generalization sources (e.g., algebraic invariances, structural repetition), discussing pattern matching without a precise definition and diagnosing it post-hoc via benchmark failures. As a result, it remains unclear which behaviors should count as pattern matching and which should not, obscuring a constructive and testable account of its boundary.

To make this notion precise, we (1) introduce a model-agnostic, data-centric formalism for pattern matching and (2) systematically study how modern architectures, decoder-only Transformers (Vaswani et al., 2017) and Mamba (Gu & Dao, 2024) perform generalization through pattern matching. Specifically, first, we propose a model-agnostic and data-centric definition of pattern matching by formalizing the substitution of input patterns *observed* to result in identical outputs in shared contexts as **functional-equivalence** (Sec. 3; henceforth, we use *pattern matching* as equivalent to *functional-equivalence-based generalization*). This induces a **coverage boundary**: if learning relies only on such evidence, reliable prediction is expected only for test inputs reachable by these substitutions.

Second, to isolate and study pattern-matching behaviors, we use controlled setups that deliberately remove other generalization sources (e.g., algebraic invariances, structural repetition) and make

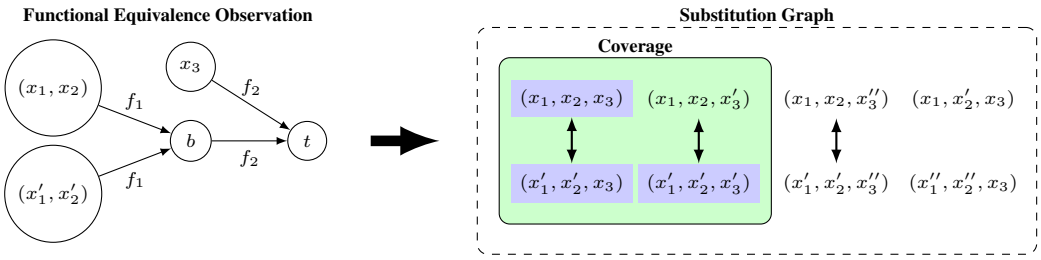


Figure 1: **Illustration of functional equivalence. Left:** In a two-hop task $(x_1, x_2, x_3) \mapsto t$ with $t = f_2(f_1(x_1, x_2), x_3)$, two fragments (x_1, x_2) and (x'_1, x'_2) satisfying $f_1(x_1, x_2) = f_1(x'_1, x'_2) = b$ consistently yield the same final output when combined with the same context x_3, supporting their **functional equivalence**. **Right:** Among all possible inputs (few shown), we draw an edge between any two inputs that differ only by functionally equivalent fragments to form a **substitution graph**. Then, **coverage** is the set of observed inputs (highlighted as blue) and all inputs connected to them. We define pattern matching as a type of generalization that occurs inside the coverage, harnessing functional equivalence.

functional equivalence the *primary available mechanism*. With this setting, our formalism yields *predictive and quantitative insights* about the limitations of pattern matching that, to our knowledge, have not been well characterized in prior works:

- **Generalization success is tightly ordered by the number of supporting contexts that witness the relevant functional equivalence.** Mechanistically, Transformers implement functional equivalence via clustered intermediate representations at specific layers/positions, with clustering strength aligning with evidence strength (Sec. 5).
- **We derive and empirically confirm that the training data required for pattern matching on a two-hop structure grows at least quadratically with token-set size.** Measured power-law exponent agrees with predictions and remains stable under roughly $20\times$ parameter increase from (68M to 1.5B) for GPT-2 (Radford et al., 2019), and also holds for Mamba (Gu & Dao, 2024) architecture (Sec. 6).
- **When the same variable influences the output along multiple computational paths, models fail to form unified intermediate state representations.** Analysis reveals that they instead develop context-dependent state representations, impairing both generalization and interpretability (Sec. 7).
- **Chain-of-Thought (CoT) supervision (Wei et al., 2022) reduces data requirements yet does *not* resolve path ambiguity without seeing nearly exhaustive in-domain combinations** (Sec. 8).

Finally, we situate this characterization of pattern matching within a mechanism-based taxonomy of generalization mechanisms, proposing two additional distinguishable mechanisms of generalization in compositional tasks: property-based and shared-operator generalization (Sec. 9 and App. H).

Our formalism opens several research directions with practical implications, e.g., targeted data augmentation to maximize coverage, and motivates expansion to broader tasks and architectures, as well as systematic studies of how pattern matching interacts with other generalization mechanisms. Overall, our study provides a predictive, falsifiable boundary for what can be achieved through pattern matching alone and a foundational diagnostic for disentangling mixed mechanisms in modern neural networks.

2 Related Work

Pattern matching behaviors of LLMs on compositional tasks. It is well perceived that pattern matching alone is inadequate for systematic generalization (Fodor & Pylyshyn, 1988), and modern LLMs display generalization abilities that seem to be far beyond what pattern matching alone can do, as measured by their remarkable performance on complex benchmarks (Achiam et al., 2023). However, a growing body of work has consistently reported that LLMs still fall short on benchmarks designed to test compositionality (Hupkes et al., 2020), including mathematical reasoning (Mirzadeh et al., 2025), multi-hop reasoning (Yang et al., 2024; Wang et al., 2024), and more (Lake & Baroni, 2018; Kim & Linzen, 2020; Csordás et al., 2022; Dziri et al., 2023). This gap between their capabilities

and pattern-matching behaviors on compositional tasks calls for a principled framework to define what pattern matching is and to what extent a model behavior can be attributed to pattern matching, but it is mostly discussed with behavioral studies under the context of a specifically designed benchmark. Our work addresses this gap by formally defining pattern matching, and systematically analyze models' behaviors with controlled tasks that are designed to isolate pattern-matching regime grounded on our framework.

Mechanistic interpretability. Mechanistic interpretability studies aim to understand how submechanisms implement models' behaviors (Elhage et al., 2021; Olsson et al., 2022; Nanda et al., 2023; Elhage et al., 2022). Recent work analyzes how Transformer components are causally related to certain behaviors (Meng et al., 2022; Hanna et al., 2023; Goldowsky-Dill et al., 2023). In particular, it is reported that in-domain compositional generalization can emerge through grokking, with identifiable intermediate state representations inside Transformers (Wang et al., 2024). Our framework complements these works by providing mechanistic insights on pattern matching. Our findings also explain why standard interpretability techniques like logit lens (nostalgebraist, 2020; Belrose et al., 2023) may fail to identify state representations in models trained on tasks with path ambiguities.

3 Formalizing pattern matching with functional equivalence

We now develop a formal framework for pattern matching. We first provide an intuitive illustration with a two-hop structure, then generalize to arbitrary fixed-length discrete-sequence tasks.

Imagine a learner observing data determined by $f : \mathcal{X}^3 \to \mathcal{X}$. The input $\boldsymbol{x} = (x_1, x_2, x_3) \in \mathcal{X}^3$ is a sequence of three discrete tokens and the output is a single token, where each token is chosen from a finite set $\mathcal{X}$.[1] Suppose (unknown to the learner) that f factorizes as the composition of two primitive functions, $f(\boldsymbol{x}) = f_2(f_1(x_1, x_2), x_3)$, where $f_1 : \mathcal{X}^2 \to \mathcal{X}$ and $f_2 : \mathcal{X}^2 \to \mathcal{X}$, as illustrated in Fig. 2a. How can the learner generalize by only seeing the input-output patterns?

Our key intuition is that **a learner exploits the underlying patterns only when two fragments of inputs are observed to behave identically.** For instance, assume that two fragments $(x_1, x_2), (x_1', x_2') \in \mathcal{X}^2$ give the same implicit intermediate state upon the application of f_1, i.e., $f_1(x_1, x_2) = f_1(x_1', x_2') = b$. These fragments behave identically regardless of context, i.e., they are **functionally equivalent**: for all $x_3 \in \mathcal{X}$, $f(x_1, x_2, x_3) = f(x_1', x_2', x_3)$. If observations consistently support their equivalence, i.e., $f(x_1, x_2, x_3) = f(x_1', x_2', x_3)$ for observed x_3 values, this equivalence can be supported (Fig. 1 Left). Intuitively, the learner would harness this equivalence pattern to predict $f(x_1', x_2', x_3'')$, provided the training set contains $f(x_1, x_2, x_3'')$.

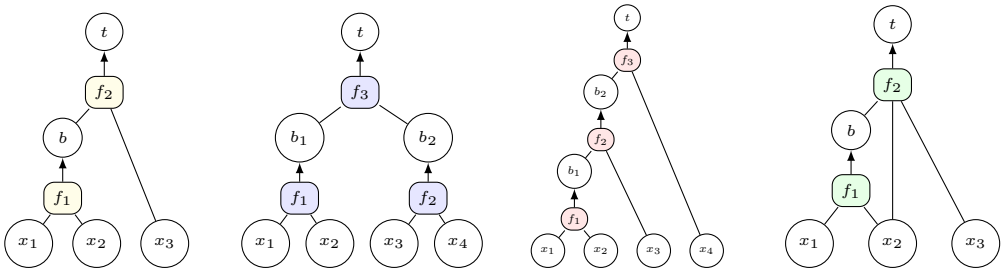
Equivalently, the learner can utilize the observed functional equivalence to correctly infer the output of an unseen input, if it can reach an observed input by 'safe substitutions' (edges in the substitution graph) supported by observations (Fig. 1 Right), which we define as a pattern matching. **Coverage** is a set of such inputs that are reachable from an observed input through chains of functionally equivalent substitutions. Then, coverage sets a boundary for what can be achieved by solely relying on substituting observed, equivalently behaving patterns. In other words, a learner can only generalize inside the coverage when it relies on functional equivalence, which we will define as pattern matching.

We now formalize these concepts for an arbitrary fixed-length task with an arbitrary set of discrete sequence observations. We restrict our attention to single-token prediction tasks defined as a deterministic mapping $f : \mathcal{X}^\ell \to \mathcal{X}$, where $\mathcal{X}$ is a finite set of tokens. We also consider a fixed observation set $D \subset \mathcal{X}^\ell$, a collection of inputs that are allowed to be observed by the learner. Write $\boldsymbol{x} = (x_1, \ldots, x_\ell) \in \mathcal{X}^\ell$ and, for a subset $I \subset [\ell] := \{1, \ldots, \ell\}$, let $\boldsymbol{x}_I := (x_i)_{i \in I}$ be a subsequence of $\boldsymbol{x}$. The first step is to formalize what it means for two subsequences to be **functionally equivalent**.

Definition 3.1 (Functional k-equivalence)**.** Fix a nonempty proper subset I of indices in $[\ell]$. Consider any set $S \subset \mathcal{X}^\ell$ of input sequences.[2] Given a pair of subsequences $\boldsymbol{a}, \boldsymbol{a}' \in \mathcal{X}^{|I|}$, we say a pair of inputs $\{\boldsymbol{x}, \boldsymbol{x}'\}$ to be an **I-co-occurrence of $\boldsymbol{a}$ and $\boldsymbol{a}'$ in** S if it satisfies $\{\boldsymbol{x}, \boldsymbol{x}'\} \subset S$ and $\boldsymbol{x}_I = \boldsymbol{a}$, $\boldsymbol{x}_I' = \boldsymbol{a}'$, and $\boldsymbol{x}_{[\ell]\setminus I} = \boldsymbol{x}'_{[\ell]\setminus I}$. Also, the subsequences $\boldsymbol{a}$ and $\boldsymbol{a}'$ are said to be **functionally k-equivalent at I in** S and denoted by $\boldsymbol{a} \equiv_S^I \boldsymbol{a}'$, if it satisfies:

[1] For brevity, we use a shared token set $\mathcal{X}$. Position-specific domains $\mathcal{X}_i$ can be embedded as subsets of an enlarged token set $\tilde{\mathcal{X}} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \mathcal{X}_3$ without loss of generality.

[2] The set S can be any subset of the whole domain, e.g., $\mathcal{X}^\ell$ itself, the train dataset D, or whatever else.



(a) 2-HOP (b) PARALLEL 2-HOP (c) 3-HOP (d) NON-TREE

Figure 2: Four synthetic task structures we study.

1. (**Sufficiency of co-occurrences.**) There are k or more distinct I-co-occurrences of $\boldsymbol{a}$ and $\boldsymbol{a}'$ in S;
2. (**Consistency.**) Every I-co-occurrence $\{\boldsymbol{x}, \boldsymbol{x}'\}$ of $\boldsymbol{a}$ and $\boldsymbol{a}'$ in S satisfies $f(\boldsymbol{x}) = f(\boldsymbol{x}')$.

In other words, two subsequences are functionally k-equivalent if they behave identically in the same contexts at least k times. The hyperparameter k represents the strength of evidence required to establish functional equivalence between two subsequences. The minimum value $k = 1$ corresponds to the weakest form of evidence, meaning a single shared context is sufficient to establish equivalence, whereas higher values of k demand more robust evidence.

Next, we ask: which inputs are reachable from observed data utilizing functional equivalence? To formalize this, we define **substitution graph**: Let $\mathcal{G}_{D,k} = (V, E)$ be an undirected graph with a vertex set $V = \mathcal{X}^\ell$ of all possible inputs. Two vertices $\boldsymbol{x}, \boldsymbol{x}' \in V$ are connected with an edge in E if and only if there exists an index set $I \subset [\ell]$ such that $\{\boldsymbol{x}, \boldsymbol{x}'\}$ is an I-co-occurrence (in V) of a pair of functionally k-equivalent sequences at I in D. This process is illustrated on the right side of Fig. 1, as a special case where $k = 1$. With this substitution graph $\mathcal{G}_{D,k}$, we formally define the **k-coverage** as a set of inputs which are connected[3] to at least one observed input as follows:

Definition 3.2 (k-coverage)**.** Consider a subset D of $\mathcal{X}^\ell$. *k-coverage*, denoted by $\mathrm{Cover}_k(D)$, is the set of all inputs in $\mathcal{X}^\ell$ that is connected to an input in D on the substitution graph $\mathcal{G}_{D,k}$.

Note that the notion of coverage is a stricter condition of the canonical definition of in-domain (ID), which is obtained by random train/test split (Wang et al., 2024) or taking combinations of observed internal computations (Dziri et al., 2023). In Sec. 5, we demonstrate that learners may not necessarily generalize on data that are classified as ID in a canonical sense, but coverage can precisely explain when and why this occurs. We also emphasize that coverage is a property of a dataset and is independent of model architectures and learning algorithms, and we demonstrate that the predictions made by our framework are invariant across model architecture and scale in Sec. 6 and 7. Finally, k-coverage can be algorithmically determined for any fixed-length discrete sequence tasks (Alg. 1), which we use for the analyses in the following sections.

Now, we formally define pattern matching as a kind of generalization that is done by substituting functionally k-equivalent fragments of inputs, whose boundary is precisely the k-coverage defined above. This formalization enables us to predict, before testing, which inputs will be reliably handled through pattern matching and which require additional mechanisms. In other words, we view pattern matching as possible only within k-coverage, and generalization outside the coverage requires generalization mechanisms other than pattern matching, which we discuss in Sec. 9 and App. H. In the following sections, we draw a systematic picture of how task structure, dataset, and model size interact to determine the success and failure of pattern matching through controlled setups, leading us to important and nontrivial insights.

4 EXPERIMENTAL SETUP

Dataset construction. We construct four synthetic tasks with different structures: 2-HOP, PARALLEL 2-HOP, 3-HOP, and NON-TREE (Fig. 2). To isolate functional-equivalence-based generalizations, we create random mappings from the product space of a token set to control generalization sources

[3]For an undirected graph $\mathcal{G}$, two vertices u, v are connected if $\mathcal{G}$ contains a path between u and v.

not attributable to compositional structures (i.e., commutativity). We explain the dataset construction process using 2-HOP task (Fig. 2a), $(x_1, x_2, x_3) \mapsto t$ with $t = f_2(f_1(x_1, x_2), x_3)$, as an example. We construct training datasets by defining a token set with size $|\mathcal{X}|$, and creating two random maps for the primitive functions $f_1 : \mathcal{X}^2 \to \mathcal{X}$ and $f_2 : \mathcal{X}^2 \to \mathcal{X}$. We mark a fraction $p_{\text{seen}} = 0.7$ of each function's domain as 'seen', gather all possible combinations where both functions are applied to inputs from their seen domains, and uniformly sample N examples to form a training dataset. See App. B.1 for more details of the dataset construction process.

Training & evaluation. Following (Wang et al., 2023), we train randomly initialized GPT-2 (Radford et al., 2019) models with 8 layers, 12 heads, and 768 dimensions (see App. B.2 for details). We construct two evaluation sets, each with 2,000 instances: **(1) ID Test Set** : all primitive function applications (e.g., $f_1(x_1, x_2)$ and $f_2(b, x_2)$ in 2-HOP task) are observed during training, but the specific combination was unseen. **(2) Out-of-coverage (canonical OOD) Test Set**: at least one primitive function application is never observed during training, which is used as a control group.

5 QUANTITATIVE ANALYSIS OF PATTERN MATCHING IN TRANSFORMERS

5.1 EVIDENCE STRENGTH IS TIGHTLY ALIGNED TO PATTERN MATCHING SUCCESS

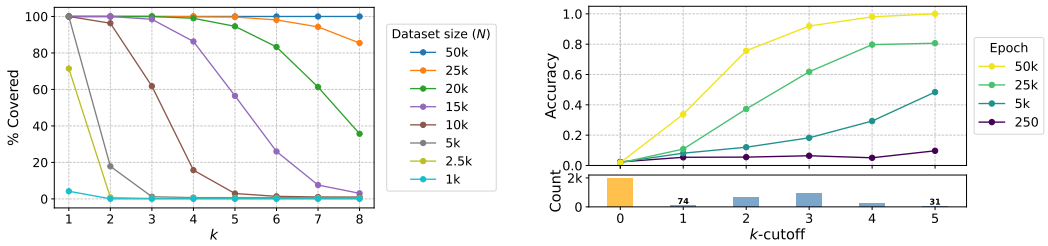


Figure 3: **Left**: Percentage of covered ID data depending on k values and dataset size (N), for 2-HOP task ($|\mathcal{X}|= 50$). **Right**: Test accuracy depending on k-cutoff values for 2-HOP task ($|\mathcal{X}|= 50$, N=10k). Each line represents a different training checkpoint. Note that out-of-coverage ($k = 0$) accuracy remains at chance level ($\approx 1/50$) regardless of training time. The bars below show the number of test data for each k-cutoff value.

We first analyze correlation between k-coverage and ID generalization performance of GPT-2 model. To this end, we implement and release a task-agnostic coverage determination algorithm (see App. C) that can be applied to diverse compositional structures. Then, we analyze what fraction of ID test data of 2-HOP task with $|\mathcal{X}|= 50$ lies inside k-coverage, depending on k and dataset size N. Fig. 3 (Left) shows that at $N = 5$k, every ID test example is already covered with minimal evidence ($k = 1$). Hence, in an ideal scenario where a single witness of functional equivalence suffices, training with the dataset as small as $N = 5$k will lead to perfect ID generalization.

However, experiments show that minimal coverage alone is insufficient for ID generalization in practice. To demonstrate this, we first define a sample's k-cutoff as the lowest k for which an input lies in k-coverage, measuring the strength of evidence for functional equivalence. For example, a k-cutoff of 3 means that an example is inside coverage with $k = 3$ but not with $k = 4$. For out-of-coverage data, we define k-cutoff as 0. Then, for the 2-HOP dataset with $N = 10$k, we classify each ID test instance according to its k-cutoff, and track the accuracy development of GPT-2 model for each group across 50k training epochs. As shown in Fig. 3 (Right), generalization success shows a strong relationship with k-cutoff values. Test data with low k-cutoff values show delayed improvement even after extensive training, while examples with stronger evidence generalize much faster.

These results yield two important insights. **First, successful generalization in practice requires a *robust* coverage so the model can confidently identify and utilize functional equivalence relationships.** The k parameter effectively quantifies this evidence strength, directly impacting generalization speed and reliability. Second, while our experiments use uniformly distributed data, the results can explain why models struggle with generalizing long-tail distributions in imbalanced real-world data (Mallen et al., 2023; Kandpal et al., 2023; Chang et al., 2024). Rare combinations naturally receive limited functional equivalence evidence (low k), placing them effectively outside practical coverage, despite technically being in-distribution. We note that our insights may guide targeted data augmentation strategies to maximize k-coverage in future work.

5.2 LATENT REPRESENTATION CLUSTERS DRIVE PATTERN MATCHING ON COVERAGE

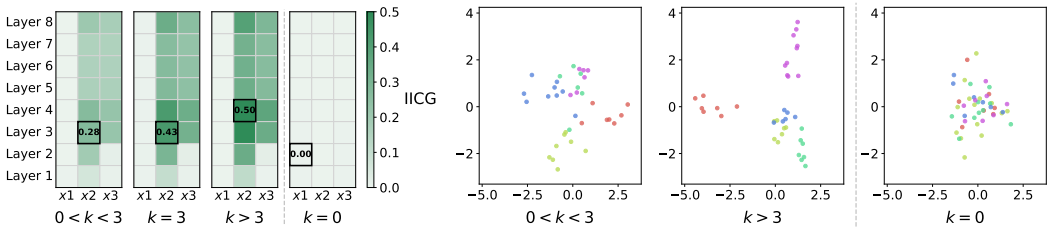


Figure 4: **Left**: Heatmap of Intra-Inter Cosine Gap (IICG) across layers and positions, sliced by k-cutoff. Higher IICG values indicate stronger clustering of representations that share the same intermediate state. The positions with the highest IICG values are marked with squares. **Right**: PCA visualization of latent representations at position x_2 and layer 3. Datapoints are classified by their intermediate states $b = f_1(x_1, x_2)$.

Next, we investigate how the model internally represents functional equivalence for k-covered inputs. Specifically, we inspect a GPT-2 trained on 2-HOP task ($|\mathcal{X}| = 50$, $N = 10$k) for 50k epochs (corresponding to the yellow line in Fig. 3 (Right)).[4] We observe that when a model successfully generalizes to ID test data, it maps functionally equivalent components into tight latent clusters, thereby encoding the equivalence relationships needed for compositional generalization.

To quantify this representation clustering phenomenon, we develop a metric that captures how distinctly the model separates functionally equivalent fragments from others. Specifically, we measure the difference between the average pairwise cosine similarity of latent vectors that share the same intermediate state $b = f_1(x_1, x_2)$ ($\overline{\cos}_{\text{intra}}$), and those that do not ($\overline{\cos}_{\text{inter}}$), for each position and layer of the model. We term this difference the **Intra–Inter Cosine Gap** $\text{IICG} = \overline{\cos}_{\text{intra}} - \overline{\cos}_{\text{inter}}$, where higher values indicate stronger within-group clustering relative to between-group separation. Fig. 4 (Left) reveals a clear relationship: higher k-cutoff values yield higher IICG scores at certain positions, **indicating that stronger functional equivalence evidence leads to more coherent internal representations.** In contrast, out-of-coverage ($k = 0$) examples exhibit no clustering pattern, as they lack evidence of functional equivalence in the training data. The PCA visualization at position x_2 and layer 3 (Right) shows this trend visually. We verify that the representation clusters play a causal role in pattern matching with causal tracing (Goldowsky-Dill et al., 2023; Hanna et al., 2023), a widely used technique to identify Transformer circuits (Fig. 8).

Our findings extend the previous insights from mechanistic interpretability studies (Wang et al., 2024) in several ways. First, we demonstrate that unified circuit formation is driven by functional equivalence evidence in the training data, not by explicit exposure to intermediate computation steps. Moreover, we find that these clustered representations are not necessarily aligned with vocabulary embeddings, implying that standard interpretability methods like logit lens nostalgebraist (2020) may fail to detect these functional equivalence representations despite their presence (see App. I).

6 DATA SCALING LAW OF PATTERN MATCHING BEHAVIORS

Our analysis in the previous section demonstrates that stronger functional equivalence evidence leads to better generalization. A natural follow-up question arises: How large should the training set be, to enable full generalization on all ID test data? Intuitively, this requires the training set to support (strongly enough) the functional equivalence of *every* pair of inputs that shares the same intermediate state b. Formally, for a 2-HOP task we need $(x_1, x_2) \equiv_D^{\{1,2\}} (x_1', x_2')$ whenever $f_1(x_1, x_2) = f_1(x_1', x_2')$. Assuming a learner requires at least k distinct pairs of evidence to establish functional equivalence of two fragments (i.e., generalizes only inside k-coverage), how does the required dataset size scale with the token set size $|\mathcal{X}|$? In practical terms, this question addresses how much data is required to cover all possible ID combinations with k-coverage, which is central to understanding data scaling requirements for pattern-matching generalization. For 2-HOP task, we derive the following scaling law (full statement and proof in App. E):

[4] Analyses for varying factors including task structures, entity set size ($|\mathcal{X}|$), dataset size (N), and training steps give consistent results; see App. D.

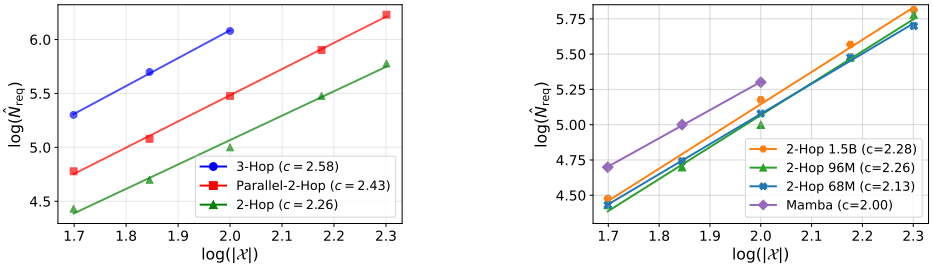


Figure 5: **Left:** Log-log plot of measured $\hat{N}_{\text{req}}$ vs. token set size ($|\mathcal{X}|$) across three compositional tasks. The slope c corresponds to the empirical power-law scaling exponent. Omitted points for 3-HOP are due to prohibitively large dataset requirements. **Right:** Power-law scaling behavior on 2-HOP task across varying GPT-2 model sizes (68M to 1.5B parameters) and Mamba model (For Mamba, we used 4 layers, a hidden dimension of 256, and a learning rate of 0.008, and $\hat{N}_{\text{req}}$ is measured for only $|\mathcal{X}|\leq 100$, since a larger token set size led to training instability). $R^2 > 0.99$ for all linear fitting.

Result 6.1 (Power-law lower bound)**.** Let $f : \mathcal{X}^3 \to \mathcal{X}$ be a 2-HOP composition. Assume a learner recognizes functional equivalence of two subsequences only after observing them in at least k distinct pairs of evidence (i.e., functionally k-equivalent).[5] Let $N_{\text{req}}(|\mathcal{X}|, k)$ be the smallest training dataset size that enables complete generalization to ID examples under this evidence threshold. Then, up to poly-logarithmic factors in $|\mathcal{X}|$, $N_{\text{req}}(|\mathcal{X}|, k) = \tilde{\Omega}(|\mathcal{X}|^{\alpha(k)})$, where $\alpha(k) = 2.5 - \frac{0.5}{k}$.

Result 6.1 predicts that a learner relying on pattern matching requires training dataset size scaling at least quadratically with respect to the token set size, to fully generalize on ID test data. To empirically confirm this, we define a practical threshold $\hat{N}_{\text{req}}$ to estimate $N_{\text{req}}(|\mathcal{X}|, k)$, as a minimal amount of training data required to exceed ID accuracy of 0.99 within 100 epochs after reaching the same level on training data (see App. F measurement details). Fig. 5 (Left) shows the measured power-law exponents for $\hat{N}_{\text{req}}$ vs. $|\mathcal{X}|$ across different task structures. The measured exponent for 2-HOP ($c = 2.26$) aligns well with our theoretical predictions of at least quadratic scaling. Although we derive the theoretical bound only for 2-HOP, we observe clear power-law relationships for more complex structures as well. The higher exponents for PARALLEL-2-HOP ($c = 2.43$) and 3-HOP ($c = 2.58$) tasks suggest that extra computational steps essentially add another dimension of relationships that require robust coverage, driving the steeper power-law scaling.

These exponents remain invariant across three different GPT-2 model sizes spanning a 20x range in parameters (from 68M to 1.5B) for all three tasks (Fig. 5 Right and Tab. 2 in App. F). We also show that the exponent measured with a Mamba model (4 layers and a hidden dimension of 256) falls inside the boundary predicted by the theory (same figure). Interestingly, the result in Fig. 6 (Middle) demonstrates that with increasing training dataset size N, there is a sharp phase transition from ID generalization failure to complete success near $N = 20$k.

Overall, the results support that the data scaling law is primarily determined by data properties rather than model capacity or architectures, and additional generalization mechanisms will be required to achieve milder scaling laws on such compositional tasks[6] We note that our result aligns with the practical observation that parameter scaling does not significantly improve the multi-hop reasoning capability of LLMs (Yang et al., 2024) and the data-hungry nature of compositional tasks (Lake & Baroni, 2018), suggesting that these could be partly attributed to pattern-matching behaviors. We leave further analysis of the connection between scaling behavior and pattern matching as an exciting future research direction.

7 PATH AMBIGUITY PROBLEM AS A FAILURE CASE OF PATTERN MATCHING

We identify a *path ambiguity problem* with our framework, a previously uncharacterized failure mode that pattern matching struggles with task structures where a single variable affects the output through multiple paths. In this section, we analyze NON-TREE task (Fig. 2d) as a case study, where x_2

[5]Note that this assumption is equivalent to modeling Fig. 3 (Right) as a step function.

[6]The observed scaling relationships are robust across different hyperparameters (weight decay and learning rate) and empirical decision criteria for $\hat{N}_{\text{req}}$ (see App. F).

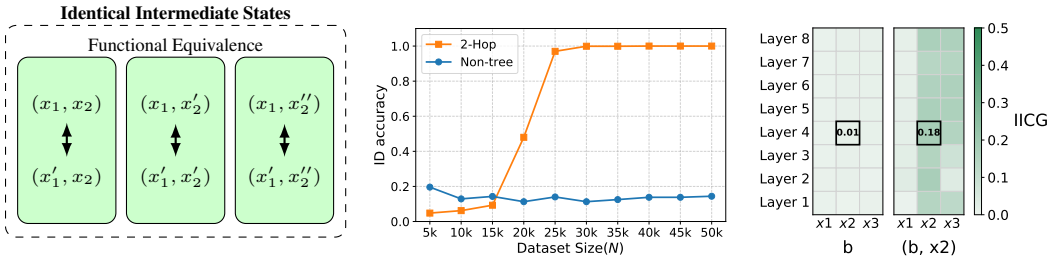


Figure 6: **Left**: In NON-TREE task, the representations of input subsequences with the same intermediate state $b = f_1(x_1, x_2)$ are split into multiple context-dependent state representations, conditioned on x_2 value. **Middle**: ID test accuracy after standard training with varying training dataset size ($|\mathcal{X}|= 50$, evaluated 100 epochs after training accuracy reaches 0.99). Observe a sharp transition from ID generalization failure to complete success near $N = 20$k for 2-HOP, which does not occur in NON-TREE task. **Right**: IICG heatmap from a model that achieved near-perfect ID accuracy (0.96) after extended training (36k epochs, $|\mathcal{X}|= 50$, $N = 50$k).

affects the output through two paths, as input to f_1 and directly to f_2. Unlike in the 2-HOP case, one cannot establish the functional equivalence of two subsequences (x_1, x_2) and (x'_1, x'_2) that produce the same intermediate state b, unless they also share the same x_2 value ($x_2 = x'_2$). It is because (x_1, x_2) and (x'_1, x'_2) are not guaranteed to behave identically (i.e., $f(x_1, x_2, x_3)$ is not necessarily equal to $f(x'_1, x'_2, x_3)$) when $x_1 \neq x_2$. Consequently, we can predict that Transformers trained on the NON-TREE will create context-dependent state representations that are conditioned on x_2 values, failing to unify them to represent the true intermediate state b (Fig. 6 Left).

Experiments show that the path ambiguity indeed hinders both generalization on the ID test set and the interpretability of intermediate state representations, as the model now establishes functional equivalence for each x_2-conditioned equivalent pair. Fig. 6 (Middle) shows that GPT-2 can fully generalize on the ID test set of 2-HOP task within a reasonable time with increasing data size, but fails with NON-TREE task, even provided with a near-exhaustive amount of possible ID combinations as training data.[7] Notably, scaling to 1.5B parameters does not show significant improvement in the performance (Fig. 17), and the Mamba model used in Sec. 6 shows the same trend of generalization failure (Fig. 18). In addition, extremely prolonged training (36k epochs) with near-exhaustive ID combinations eventually achieves ID accuracy of 0.96, however, IICG analysis reveals no evidence of a unified intermediate state representation formation, with near-zero IICG scores when grouping by the intermediate state value b (Fig. 6 Right). In contrast, grouping by x_2-conditioned intermediate state (b, x_2) leads to high IICG scores, showing the formation of context-dependent state representations. This context-dependence due to path ambiguity raises an interpretability concern, as standard linear probing-based techniques like logit lens (nostalgebraist, 2020; Belrose et al., 2023) would not reliably identify intermediate states when a model relies on pattern matching.

Hence, a generalization mechanism other than pattern matching will be required for a robust ID generalization on complex task structures that requires the access and update of intermediate states through multiple paths (e.g., planning tasks (Ruis et al., 2020; Kambhampati et al., 2024; Valmeekam et al., 2023)), where further characterization of this problem remains as an exciting future direction.

8 COT IMPROVES DATA EFFICIENCY, BUT PATH AMBIGUITY PERSISTS

CoT supervision (Wei et al., 2022; Kojima et al., 2022) dramatically improves performance on multi-step reasoning tasks. We investigate how CoT interacts with our framework and whether it can address the challenges observed in Sections 6 and 7. Specifically, we train models to sequentially generate intermediate states before final outputs, making 2-HOP a two-token prediction task: $(x_1, x_2, x_3) \mapsto (b, t)$, for example. This substantially improves data efficiency (Fig. 7 (Left)), with the power-law exponent dropping from 2.58 to 1.76 in 3-HOP task, aligning with previous studies on the sample efficiency of CoT (Srivastava et al., 2023; Kim & Suzuki, 2025; Wen et al., 2025). The scaling exponents measured for 2-HOP, 3-HOP, and even 5-HOP tasks become nearly identical with CoT supervision. We interpret this as CoT effectively 'flattening' multi-hop structures into sequences of single-hop tasks, reducing the compounding data requirements of deeper compositional structures.

[7] For $|\mathcal{X}|= 50$ and $p_{\text{seen}} = 0.7$, our largest run ($N = 50$k) includes virtually the entire domain ($\approx 0.7^2 \times |\mathcal{X}|^3 \approx$ 61k distinct ID triples).

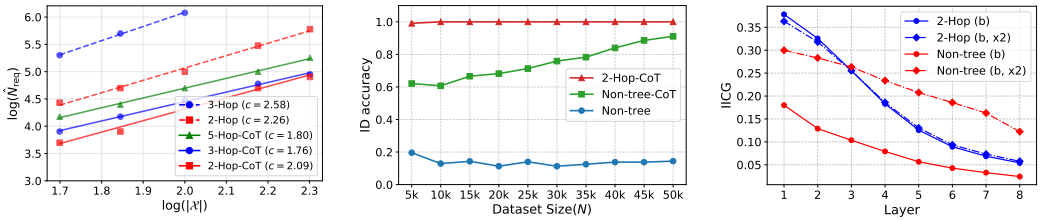


Figure 7: **Left:** Power-law scaling of required dataset size vs. token set size for tasks with CoT supervision. $R^2 > 0.98$ for all linear fits. **Middle:** Comparison of ID test Accuracy of NON-TREE task ($|\mathcal{X}|= 50$) with and without CoT supervision. **Right:** IICG score comparison for NON-TREE and 2-HOP task with CoT supervision ($|\mathcal{X}|= 50$, $N = 10$k). The scores are measured at each layer of intermediate state position b, based on two grouping strategies: b and (b, x_2). Models are trained for 100 epochs after reaching training accuracy>0.99.

However, we find the path ambiguity problem persists even with CoT supervision. Despite showing improvements, the models fail to achieve perfect ID generalization under the same training conditions that yield perfect performance in 2-HOP task (Fig. 7 (Middle). IICG analysis (Right) reveals that the model's representations remain partially context-dependent. For 2-HOP task, the representations cluster purely by intermediate states b, as indicated by the result that IICG measurement with x_2-conditioned states does not significantly shift the curve. In contrast, the IICG score for NON-TREE task is significantly elevated at every layer with the same conditioning, suggesting the absence of disentangled state representation inside the model. We hypothesize this arises since CoT supervision does not give enough evidence that different (x_1, x_2) pairs sharing the same b should yield identical second-step outputs, as functional equivalence holds only when $x_2 = x_2'$. Hence, while CoT supervision helps with sequential computation by breaking down multi-hop structures, it may partially inherit the limitations on handling tasks with path ambiguities we describe in Sec. 7. Our analysis may explain why LLMs struggle with complex planning tasks even when using CoT techniques and massive training data (Stechly et al., 2024), where we leave further analysis as future work.

9 DISCUSSION AND CONCLUSION

Our formalism of pattern matching and theoretical and experimental analyses yield quantitative and predictive insights into modern neural networks' pattern-matching behaviors, moving beyond post-hoc accounts of benchmark failures on compositional generalization tasks. Our theory and experiments show three fundamental limitations of pattern matching for learning compositional structures: (i) **evidence requirements:** instance-wise success aligns with the strength of functional-equivalence evidence (Sec. 5), (ii) **data scaling**: sample complexity is at least quadratic in token-set size for 2-HOP task (Sec. 6), and (iii) **path ambiguity**: a structural failure mode that impairs accuracy and interpretability even under high coverage and persists with CoT supervision (Sec. 7 and 8).

This naturally raises the question: **what generalization mechanisms enable generalization beyond the coverage boundary?** While a complete answer requires future work, we outline a mechanism-based taxonomy as a starting point for a constructive categorization of distinct generalization mechanisms beyond pattern matching (see App. H for complete discussion):

- **Functional equivalence-based generalization**, the main focus of this work.
- **Function property-based generalization** leverages algebraic invariances of individual primitive functions, e.g., commutativity or input irrelevance where certain arguments never affect the output. This distinguishes it from pattern matching, as it leverages a primitive function's global property that holds across all inputs, not only those observed.
- **Shared-operator generalization** leverages the reuse of the same computation across positions (e.g., when $f_1 = f_2$ in a two-hop task), which may be important in compositional generalization. For example, it is known that Transformers with inductive biases towards computation reuse can improve generalization on compositional tasks (Csordás et al., 2021).

We envision this taxonomy as a foundational diagnostic that quantifies when pattern matching suffices and when other mechanisms are required. We anticipate that future work will build on this foundation, towards a more complete and constructive understanding of compositional generalization and its failures.

REPRODUCIBILITY STATEMENT

All codes for dataset generation, training, and analysis are contained in the attached supplementary material, with proper instructions for reproducibility.

THE USE OF LARGE LANGUAGE MODELS (LLMS)

This work deployed LLMs to proofread for grammatical errors and improve the quality of writing.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774, 2023. URL https://arxiv.org/abs/2303.08774.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *ArXiv preprint*, abs/2303.08112, 2023. URL https://arxiv.org/abs/2303.08112.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: LLMs trained on "a is b" fail to learn "b is a". In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GPKTIktA0k.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. How do large language models acquire factual knowledge during pretraining? In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/6fdf57c71bc1f1ee29014b8dc52e723f-Abstract-Conference.html.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 619–634, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.49. URL https://aclanthology.org/2021.emnlp-main.49.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. CTL++: Evaluating generalization on never-seen compositional patterns of known functions, and compatibility of neural representations. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9758–9767, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.662. URL https://aclanthology.org/2022.emnlp-main.662.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HyzdRiR9Y7.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaïd Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/deb3c28192f979302c157cb653c15e90-Abstract-Conference.html.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *ArXiv preprint*, abs/2209.10652, 2022. URL https://arxiv.org/abs/2209.10652.

Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1960.

Paul Erdős and Alfréd Rényi. On random graphs i. *Publ. math. debrecen*, 6(290-297):18, 1959.

Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching. *ArXiv preprint*, abs/2304.05969, 2023. URL https://arxiv.org/abs/2304.05969.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Michael Hanna, Ollie Liu, and Alexandre Variengien. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/efbba7719cc5172d175240f24be11280-Abstract-Conference.html.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 1988–1997. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.215. URL https://doi.org/10.1109/CVPR.2017.215.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. Position: Llms can't plan, but can help planning in llm-modulo frameworks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Th8JPEmH4z.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15696–15707. PMLR, 2023. URL https://proceedings.mlr.press/v202/kandpal23a.html.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SygcCnNKwr.

Juno Kim and Taiji Suzuki. Transformers provably solve parity efficiently with chain of thought. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=n2NidsYDop.

Najoung Kim and Tal Linzen. COGS: A compositional generalization challenge based on semantic interpretation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9087–9105, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.731. URL https://aclanthology.org/2020.emnlp-main.731.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2879–2888. PMLR, 2018. URL http://proceedings.mlr.press/v80/lake18a.html.

Lucien Le Cam. An approximation theorem for the poisson binomial distribution. *Pacific Journal of Mathematics*, 10(4):1181–1197, 1960. doi: 10.2140/pjm.1960.10.1181.

João Loula, Marco Baroni, and Brenden Lake. Rearranging the familiar: Testing compositional generalization in recurrent networks. In Tal Linzen, Grzegorz Chrupała, and Afra Alishahi (eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 108–114, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5413. URL https://aclanthology.org/W18-5413.

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

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,

pp. 9802–9822, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AjXkRZIvjB.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=9XFSbDPmdW.

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

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *ArXiv preprint*, abs/2209.11895, 2022. URL https://arxiv.org/abs/2209.11895.

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

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M. Lake. A benchmark for systematic generalization in grounded language understanding. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e5a90182cc81e12ab5e72d66e0b46fe3-Abstract.html.

Arnab Sen Sharma, David Atkinson, and David Bau. Locating and editing factual associations in mamba. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=yoVRyrEgix.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Johan Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew Kyle Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, Cesar Ferri, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Christopher Waites, Christian Voigt, Christopher D Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, C. Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz

Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodolà, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germàn Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Xinyue Wang, Gonzalo Jaimovitch-Lopez, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Francis Anthony Shevlin, Hinrich Schuetze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B Simon, James Koppel, James Zheng, James Zou, Jan Kocon, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh Dhole, Kevin Gimpel, Kevin Omondi, Kory Wallace Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros-Colón, Luke Metz, Lütfi Kerem Senel, Maarten Bosma, Maarten Sap, Maartje Ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramirez-Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael Andrew Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan Andrew Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter W Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan Le Bras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Russ Salakhutdinov, Ryan Andrew Chi, Seungjae Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel Stern Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima Shammie Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven Piantadosi, Stuart Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsunori Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Venkatesh Ramasesh, vinay uday prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Sophie Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui

Wang, and Ziyi Wu. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=uyTL5Bvosj. Featured Certification.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. Chain of thoughtlessness? an analysis of cot in planning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/3365d974ce309623bd8151082d78206c-Abstract-Conference.html.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971, 2023. URL https://arxiv.org/abs/2302.13971.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo Hernandez, Sarath Sreedharan, and Subbarao Kambhampati. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/7a92bcdede88c7afd108072faf5485c8-Abstract-Datasets_and_Benchmarks.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokking of implicit reasoning in transformers: A mechanistic journey to the edge of generalization. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/ad217e0c7fecc71bdf48660ad6714b07-Abstract-Conference.html.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=NpsVSN6o4ul.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Kaiyue Wen, Huaqing Zhang, Hongzhou Lin, and Jingzhao Zhang. From sparse dependence to sparse attention: Unveiling how chain-of-thought enhances transformer sample efficiency. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AmEgWDhmTr.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10210–10229, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.550. URL https://aclanthology.org/2024.acl-long.550/.

Contents of the Appendix

A LIMITATIONS

We deliberately restrict to synthetic tasks to isolate structure-based limits without confounds from lexical or domain priors. We leave extending the coverage analysis to discrete sequence tasks with variable lengths and more natural data as future work. Additionally, our experiments focus on autoregressive architectures, and the applicability of the coverage principle to broader architectures remains to be validated.

B Detailed experimental setup

B.1 Dataset construction details

We now provide detailed information about our dataset construction process. While we primarily explain this process for the 2-Hop task, we follow similar procedures for the other compositional structures.

Vocabulary and Token Representation For a task with token set size $|\mathcal{X}|$, we create $|\mathcal{X}|$ special tokens of the form <t_0>, <t_1>, ..., <t_($|\mathcal{X}|-1$)>, which we append to the standard GPT-2 vocabulary. We also add special tokens to mark the end of sequences. For Chain-of-Thought (CoT) experiments, intermediate computations are represented in the target sequence as the actual intermediate token.

Function Construction For the 2-Hop task, we construct two primitive functions $f_1 : \mathcal{X}^2 \to \mathcal{X}$ and $f_2 : \mathcal{X}^2 \to \mathcal{X}$ by randomly mapping from their respective domains to the codomain $\mathcal{X}$. We create the domain by taking the Cartesian product of the token set with itself. For each function, we randomly designate a fraction $p_{\text{seen}} = 0.7$ of its domain as the "seen" portion, resulting in sets S_{f_1} and S_{f_2}.

Dataset Generation Algorithm To generate the training dataset, we first identify all possible combinations where both primitive operations come from their respective "seen" domains. Specifically, we find all valid tuples (x_1, x_2, x_3, t) such that:

$$(x_1, x_2) \in \text{domain}(S_{f_1}) \tag{1}$$

$$(f_1(x_1, x_2), x_3) \in \text{domain}(S_{f_2}) \tag{2}$$

$$t = f_2(f_1(x_1, x_2), x_3) \tag{3}$$

From this set of all possible in-domain combinations, we uniformly sample N examples to form our training dataset. When the number of possible combinations exceeds N, this sampling ensures the model sees only a subset of possible in-domain combinations.

Test Set Construction We carefully construct test sets to evaluate the model's generalization capabilities across different coverage conditions. Our test sets contain:

- **In-Domain (ID) Test Set**: Combinations not seen during training but where both primitive operations were observed in other contexts. These examples may lie within the coverage as defined by our framework.
- **Out-of-coverage (canonical OOD) Test Set**: Examples where at least one primitive operation was never observed in training. These fall outside the coverage.

Input-Output Format The dataset is formatted for auto-regressive token prediction. For the standard 2-Hop task, inputs comprise three tokens representing x_1, x_2, and x_3, while the target includes these input tokens followed by the prediction t and an end marker. Below are the examples of the dataset format for different settings.

- **Standard Format:**
  - Input: <t_5><t_12><t_3>
  - Target Completion: <t_17>
  - The model must predict the final output token followed by the end marker.
- **Chain-of-Thought Format:**
  - Input: <t_5><t_12><t_3>
  - Target Completion: <t_9><t_17>
  - The model must first predict the intermediate computation result <t_9> (where <t_9> = f_1(<t_5>, <t_12>)), followed by the final output.
- **Partial Computation Format (f_1):**
  - Input: <t_5><t_12>
  - Target Completion: <t_9>
  - These examples represent the primitive function applications used to construct the full compositional task.

For the other compositional tasks, we follow analogous construction procedures, adjusting the number of input tokens and the composition structure based on the specific task's requirements. For example, PARALLEL 2-HOP requires four input tokens, while 3-HOP follows a three-step composition chain requiring appropriate modifications to the function construction and sampling procedures.

B.2 TRAINING DETAILS

Table 1: Model configurations for different GPT-2 variants used in our experiments

Configuration	**GPT-2-Small**	**GPT-2**	**GPT-2-XL**
Number of Attention Heads	6	12	25
Number of Layers	4	8	48
Hidden Dimension	768	768	1600
Total Parameters	68M	96M	1.5B

For our experiments, we employ three GPT-2 model variants of increasing size: GPT-2-Small (68M parameters), GPT-2 (96M parameters), and GPT-2-XL (1.5B parameters). As shown in Tab. 1, GPT-2-Small consists of 4 layers with 6 attention heads and a hidden dimension of 768. The standard GPT-2 configuration used in most experiments features 8 layers with 12 attention heads while maintaining the same hidden dimension of 768. Our largest model, GPT-2-XL, significantly scales up the architecture with 48 layers, 25 attention heads, and an increased hidden dimension of 1600. The implementation follows the codebase from (Wang et al., 2024).

We train all models using the AdamW optimizer with beta values of (0.9, 0.999) and epsilon of 1e-8. We set the learning rate to 8e-4 with a weight decay of 0.1. A batch size of 16,384 is used, with full gradient descent applied for datasets smaller than the batch size. All training is conducted with mixed precision (fp16) on 4 NVIDIA A100 GPUs with 80GB memory each. We employ a constant learning rate schedule with a linear warmup period of 2,000 steps. This standardized training configuration is maintained across all experiments to ensure fair comparisons between different task structures and dataset sizes, unless explicitly varied in specific ablation studies.

C IMPLEMENTATION DETAILS FOR THE COVERAGE DETERMINATION ALGORITHM

Algorithm 1: k-Coverage Determination Algorithm

Input: Training examples $\mathcal{D} = \{(x_i, f(x_i))\}$, where $x_i \in \mathcal{X}^n$ and $f(x_i) \in \mathcal{X}$
Minimum evidence threshold $k \geq 1$
Output: Coverage set Cover($\mathcal{D}$)

```
/* STEP 1: Build behavior maps for each subsequence pattern */
```
foreach *subset* $I \subset [n]$, $I \neq \emptyset$, $I \neq [n]$ **do**
  Behavior$_I \leftarrow$ map from subsequence x_I to the mapping $\{x_{[n]\setminus I} \mapsto f(x) \mid x \in \mathcal{D}\}$
end
```
/* STEP 2: Identify functionally equivalent subsequences */
```
foreach *subset* $I \subset [n]$, $I \neq \emptyset$, $I \neq [n]$ **do**
  $UF_I \leftarrow$ new UnionFind()
  foreach *pair of subsequences* (α, β) *in Behavior*$_I$ **do**
    SharedComplements $\leftarrow$ complements observed with both α and β
    if *No contradictions in SharedComplements* ***and*** *matching evidence* $\geq k$ **then**
      UF_I.Union(α, β) ; // Mark as functionally equivalent
    end
  end
  EquivClasses$_I \leftarrow UF_I$
end
```
/* STEP 3: Build substitution graph */
```
$G \leftarrow$ empty graph with nodes for all $x \in \mathcal{X}^n$
foreach *pair of inputs* (x, y) *with* $f(x) = f(y)$ **do**
  foreach *subset* I *where* x *and* y *differ only on indices in* I **do**
    if *EquivClasses*$_I$*.Find*$(x_I) =$ *EquivClasses*$_I$*.Find*(y_I) **then**
      Add edge (x, y) to G
      break
    end
  end
end
```
/* STEP 4: Determine coverage */
```
Cover($\mathcal{D}$) $\leftarrow \bigcup_{x \in \mathcal{D}}$ ConnectedComponent(G, x)
return *Cover*($\mathcal{D}$)

Algorithm 1 presents our approach to computing the coverage set with a minimum evidence threshold k. The algorithm works in four main stages:

Stage 1: Behavior mapping We first analyze the training data to create a mapping of behaviors for each possible subsequence of the input. For each subset of indices I, we record how different subsequences x_I behave when paired with their complements $x_{[n]\setminus I}$, essentially mapping each subsequence to a function from complements to outputs.

Stage 2: Equivalence class construction For each subset of indices I, we build equivalence classes of subsequences that exhibit functionally identical behavior. Two subsequences are considered equivalent only if: (1) they share at least k distinct complements where they produce the same output, and (2) they never produce different outputs when given the same complement (no contradictions). We use a Union-Find data structure to efficiently track and merge these equivalence classes. The Union-Find (or Disjoint-Set) data structure efficiently maintains a collection of disjoint sets, supporting two key operations: (1) *Find* - determine which set an element belongs to, and (2) *Union* - merge two sets.

Stage 3: Substitution Graph Construction We construct a graph where nodes represent input sequences from our training and test sets, rather than the entire domain space (which would be computationally prohibitive for large token sets). We add an edge between two inputs x and y if and only if: (1) they produce the same output, (2) they differ only in one subsequence position set I, and (3) their differing subsequences belong to the same equivalence class. This graph represents the space of *safe substitutions* where one can replace a subsequence with a functionally equivalent alternative without changing the expected output. Our implementation uses parallel processing to efficiently construct this graph even for large datasets.

Stage 4: Coverage computation Finally, we compute the coverage set by taking the union of all connected components in the substitution graph that contain at least one training example. This set comprises all inputs that are reachable from the training data through chains of equivalent subsequence substitutions.

D Detailed analysis for representation unification experiments

D.1 Causal Tracing Methodology

To analyze the causal role of specific hidden representations in our Transformer model, we employ causal tracing, a technique that measures the effect of intervening on intermediate activations during inference (Goldowsky-Dill et al., 2023; Hanna et al., 2023). Specifically, we measure the causal effect using the *indirect effect* metric defined in (Sharma et al., 2024). This methodology allows us to identify which components and positions in the model most strongly contribute to compositional generalization. We illustrate the measurement with 2-Hop task.

Our analysis begins by collecting three types of computational traces:

1. **Clean run** (G): We run the model on a compositional task with input (x_1, x_2, x_3) where the corresponding output is $t = f_2(f_1(x_1, x_2), x_3)$.
2. **Corrupted run** (G^*): We replace the original input with a corrupted version by changing the first two tokens (x_1, x_2) to (x'_1, x'_2), where $f_1(x'_1, x'_2) \neq f_1(x_1, x_2)$. This ensures that the model produces a different final output $t^* \neq t$. During this run, we cache all hidden states ${h^*}_i^{(\ell)}$ for each token position i and layer ℓ.
3. **Patched run** ($G[\leftarrow {h^*}_i^{(\ell)}]$): We run the model on the input from the clean run, but at a specific token position i and layer ℓ, we replace the hidden state with the corresponding state from the corrupted run.

To quantify the causal effect of a specific hidden state $h_i^{(\ell)}$ on the model's prediction, we measure the *Indirect Effect* (IE):

$$\text{IE}_{h_i^{(\ell)}} = \frac{p[\leftarrow {h^*}_i^{(\ell)}](t^*) - p(t^*)}{p^*(t^*) - p(t^*)} \tag{4}$$

where:

- $p(t^*)$ is the probability assigned to the corrupted output t^* in the clean run G
- $p^*(t^*)$ is the probability assigned to the corrupted output t^* in the corrupted run G^*
- $p[\leftarrow {h^*}_i^{(\ell)}](t^*)$ is the probability assigned to the corrupted output t^* in the patched run $G[\leftarrow {h^*}_i^{(\ell)}]$

This metric quantifies how much corruption in a particular state affects the overall outcome. An IE value close to 1 indicates that the corruption of the state $h_i^{(\ell)}$ to ${h^*}_i^{(\ell)}$ alone almost completely changes the prediction to that of the corrupted run, suggesting that this state is causally important for the computation. Conversely, an IE value close to 0 indicates that the state has minimal causal impact on the prediction.

In our experiments, we apply causal tracing to analyze different subsets of test data categorized by their k-cutoff values, where k represents the minimum evidence threshold required for functional equivalence (as defined in Sec. 3 of the main text). This allows us to correlate the strength of functional equivalence evidence with the formation of unified internal representations.

D.2 Causal tracing results for each k-cutoff value in 2-Hop task

Figure 8 displays the causal tracing results for the 2-Hop task, broken down by different k-cutoff values. We observe that the causal patterns are similar across different k-cutoff values, with slight differences in where and how strongly the causal effects manifest in the model. This suggests that once an example falls within coverage (even with minimal evidence, $k = 1$), the model forms internal representations that play similar causal roles in prediction.

D.3 Token set size ablation

We show that the observed patterns of cosine similarity analysis and causal tracing in the 2-Hop task are consistent across different token set sizes $|\mathcal{X}|$. For $|\mathcal{X}|= 70, 100, 150, 200$, we analyze model checkpoints with training dataset size $N = \hat{N}_{\text{req}}(|\mathcal{X}|)$ that achieve training accuracy > 0.99. Figure

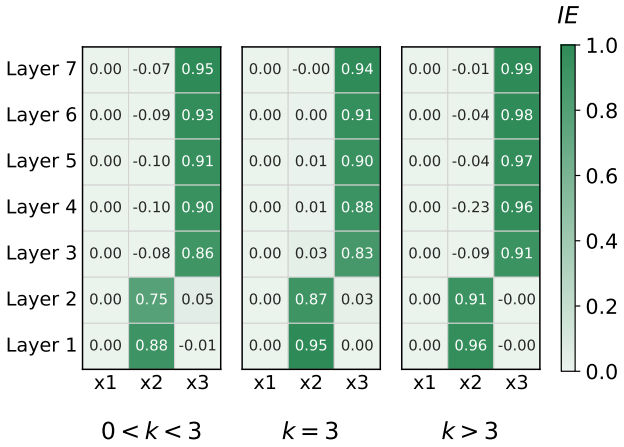


Figure 8: Causal tracing results for the 2-HOP task across different k-cutoff values, showing Indirect Effect (IE) scores at each layer and position.

Fig. 9 shows the results, indicating strong representation clustering at the lower layers of position x_2 for all cases. The causal tracing results in Fig. 10 show that the clustered functional equivalence representations at the lower layers of position x_2 play a causal role in determining the model's final prediction.

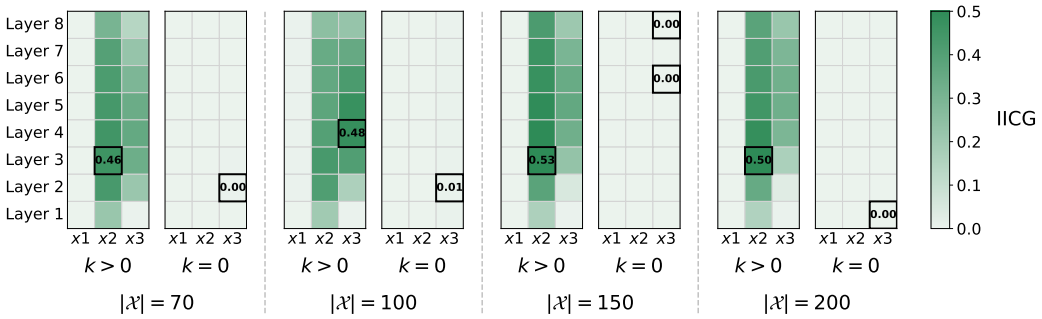


Figure 9: IICG heatmap across different token set sizes, showing consistent representation clustering patterns.

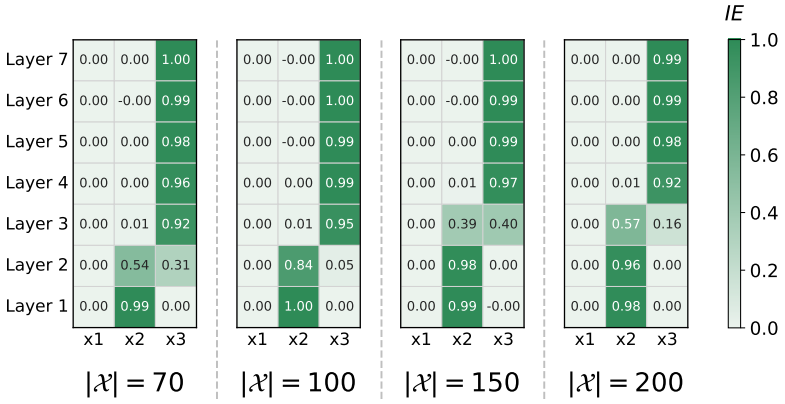


Figure 10: Causal tracing results showing indirect effect heatmaps for different token set sizes $|\mathcal{X}|$.

D.4 TASK ABLATION

We show that GPT-2 models trained on PARALLEL-2-HOP and 3-HOP tasks exhibit the same patterns: clustered functional equivalence representations of intermediate states at specific layers and positions, confirmed through cosine similarity analysis, with causal tracing analysis verifying their role in model predictions. For both tasks, we analyze with $|\mathcal{X}|= 50$ and examine model checkpoints with training dataset size $N = \hat{N}_{\text{req}}(|\mathcal{X}|)$ that achieve training accuracy > 0.99.

Figures 11 and 12 show the results for the PARALLEL-2-HOP task. The IICG patterns reveal strong representation clustering at mid-layers: at positions x_2 and x_3 when grouped by $b_1 = f_1(x_1, x_2)$, and at position x_4 when grouped by $b_2 = f_2(x_3, x_4)$. Causal tracing confirms the causal role of these clustered representations in the model's predictions.

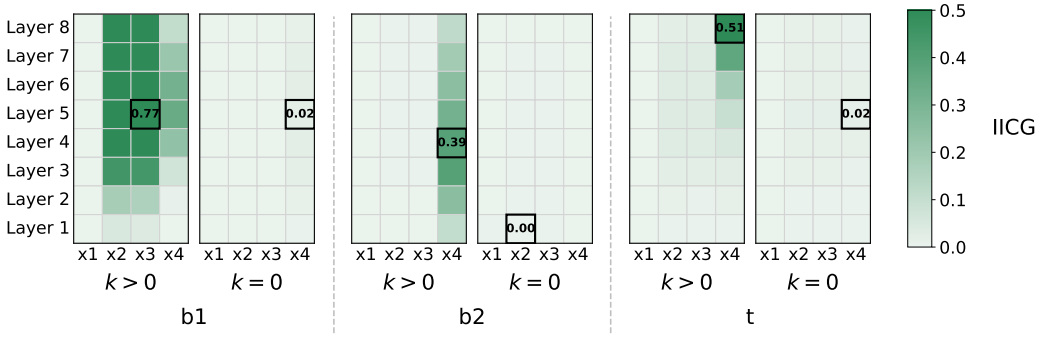


Figure 11: IICG heatmap for PARALLEL-2-HOP task with grouping strategies based on $b_1 = f_1(x_1, x_2)$ (**Left**), $b_2 = f_2(x_3, x_4)$ (**Middle**), and $t = f_3(b_1, b_2)$ (**Right**).

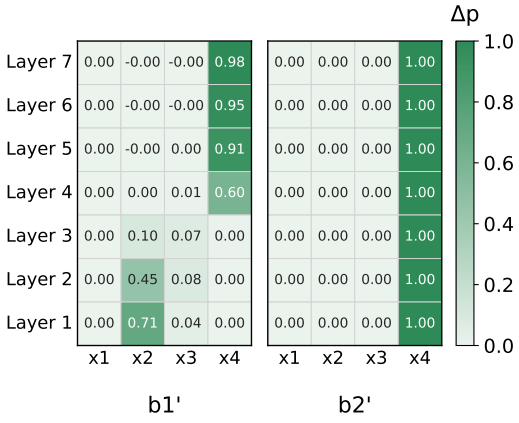


Figure 12: Causal tracing results showing indirect effect heatmap for PARALLEL-2-HOP task. **Left:** perturbation with different (x_1, x_2) pair leading to a different b_1 value. **Right:** perturbation with different (x_3, x_4) pair leading to a different b_2 value.

Similarly, Figures 13 and 14 show results for the 3-HOP task. The IICG patterns exhibit strong representation clustering at mid-layers: at position x_3 when grouped by $b_1 = f_1(x_1, x_2)$, and at position x_3 when grouped by $b_2 = f_2(b_1, x_3)$. Causal tracing again confirms the causal importance of these representations.

These results demonstrate that the formation of clustered intermediate state representations and their causal role in compositional generalization is a consistent phenomenon across different task structures, supporting the generality of our findings beyond the 2-HOP task.

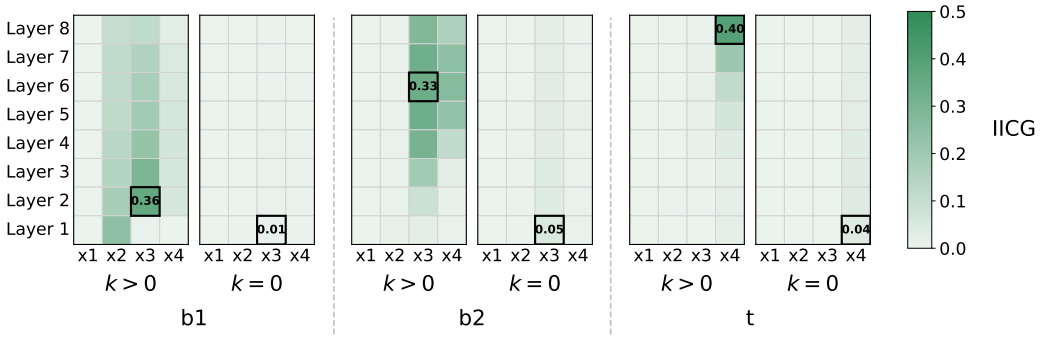


Figure 13: IICG heatmap for 3-HOP task with grouping strategies based on $b_1 = f_1(x_1, x_2)$ (**Left**), $b_2 = f_2(b_1, x_3)$ (**Middle**), and $t = f_3(b_2, x_4)$ (**Right**).

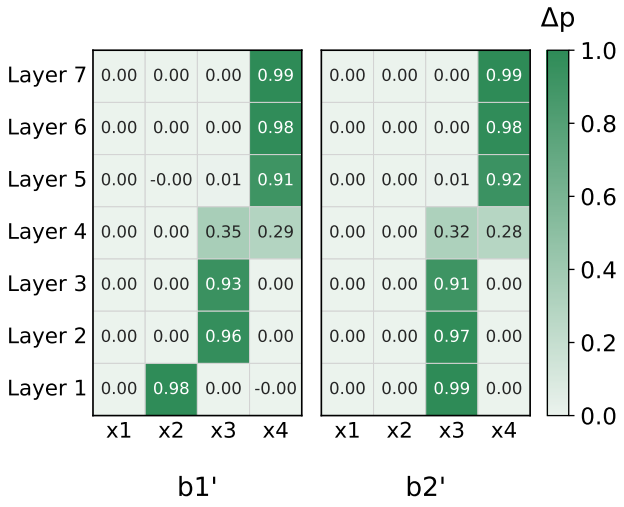


Figure 14: Causal tracing results showing indirect effect heatmap for 3-HOP task. **Left:** perturbation with different (x_1, x_2) pair leading to different b_1 value. **Right:** perturbation leading to different $b_2 = f_2(b_1, x_3)$ value.

E Derivation of Res. 6.1 (Necessary Power-Law Data Bound)

Problem setting Let $\mathcal{X}$ be a finite token set with cardinality $|\mathcal{X}|$. The target mapping

$$f : \mathcal{X}^3 \longrightarrow \mathcal{X}, \qquad f(x_1, x_2, x_3) = f_2(f_1(x_1, x_2),\, x_3)$$

is a *two-hop* composition of unknown primitives $f_1 : \mathcal{X}^2 \to \mathcal{X}$ and $f_2 : \mathcal{X}^2 \to \mathcal{X}$. Write $b = f_1(x_1, x_2) \in \mathcal{X}$ for the *intermediate state*. Throughout the derivation, we impose the following two assumptions.

(A1) **Balanced classes:** Each intermediate value $b \in \mathcal{X}$ is realized by exactly $|\mathcal{X}|$ first-hop pairs, *i.e.*, the sets $E_b := \{(x_1, x_2) \in \mathcal{X}^2 : f_1(x_1, x_2) = b\}$ all have size $|\mathcal{X}|$ and form a partition of $\mathcal{X}^2$.

(A2) **Uniform training sampler:** The training set D contains N triples drawn *uniformly with replacement* from the domain $\mathcal{X}^3$.

Functional k-equivalence and learner model For two first-hop fragments $a, a' \in \mathcal{X}^2$ define

$$a \sim a' \quad :\Longleftrightarrow \quad f_1(a) = f_1(a').$$

They are **functionally k-equivalent w.r.t.** D (denoted $a \equiv_D^k a'$) when there exist k *distinct contexts* $c_1, \ldots, c_k \in \mathcal{X}$ such that for every $r \le k$ both (a, c_r) and (a', c_r) appear in D and $f(a, c_r) = f(a', c_r)$. We call each $(a, c_r), (a', c_r)$ an *evidence pair*.

The coverage principle by itself is *only a necessity* statement: outside the k-coverage region, a purely pattern-matching learner's predictions are unconstrained. To convert this into a *sufficient* data condition, we adopt an explicit inductive bias, matching the premise of Result 6.1:

> **Learner assumption:** *Whenever two fragments become linked by k independent (i.e., pairwise from distinct contexts) evidence pairs, the learner treats them as functionally equivalent; the learner also propagates equivalence* transitively *along chains of such links.*

With this rule in place the relevant structure inside each class E_b is the *k-evidence graph*: vertices are the $|\mathcal{X}|$ first-hop pairs in the class and an edge connects two vertices whenever the pair is observed at least k times in shared contexts. If that graph is *connected* (every vertex reachable from every other), then every fragment in E_b is linked by a chain of k-evidence steps and is therefore recognized as equivalent by the learner. Hence

> **Data sufficiency criterion:** If the k-evidence graph of *each* class E_b is connected, the learner generalizes perfectly to all in-domain (ID) inputs.

This criterion requires *connectivity*, and we aim to derive the condition to yield the connectivity with high probability using Erdős–Rényi model (Erdős & Rényi, 1959; Erdős & Rényi, 1960). The minimal dataset size achieving this with high probability is denoted $N_{\mathrm{req}}(|\mathcal{X}|, k)$.

Note that the k-evidence graph on E_b is the restriction of the substitution graph $\mathcal{G}_{D,k}$ (Def. 3.2) to the vertex set $E_b \times \{x_3\}$ for any fixed x_3. Connectivity of every class therefore implies that *every* first-hop fragment lies in the same connected component as some training input, *i.e.*, the entire in-domain set is contained in k-coverage. Under the learner assumption, this is both necessary and sufficient for perfect ID generalization, yielding Res. 6.1.

E.1 Step 1: probability of a single evidence pair

Evidence pair probability Fix two distinct first–hop fragments $i, j \in E_b$ and a context $c \in \mathcal{X}$. We want to find p_1, the probability that context c provides an evidence pair for the functional equivalence of fragments i and j:

$$p_1 := \Pr[(i, c) \in D \text{ and } (j, c) \in D]$$

Let $q := 1/|\mathcal{X}|^3$ denote the probability of drawing any specific triple in a single draw. Using the inclusion–exclusion principle:

$$\begin{aligned} p_1 &= \Pr[(i, c) \in D \text{ and } (j, c) \in D] \\ &= 1 - \Pr[(i, c) \notin D] - \Pr[(j, c) \notin D] + \Pr[(i, c) \notin D \text{ and } (j, c) \notin D] \\ &= 1 - (1 - q)^N - (1 - q)^N + (1 - 2q)^N \end{aligned} \tag{S1.1}$$

For $q \ll 1$ (which holds when $|\mathcal{X}|$ is large), we can use Taylor expansion:

$$(1-q)^N = 1 - Nq + \frac{N(N-1)}{2}q^2 + O(q^3)$$
$$(1-2q)^N = 1 - 2Nq + \frac{N(N-1)}{2}(2q)^2 + O(q^3) \tag{S1.2}$$

Substituting these approximations:

$$\begin{aligned} p_1 &= 1 - 2\left(1 - Nq + \frac{N(N-1)}{2}q^2\right) + \left(1 - 2Nq + 2N(N-1)q^2\right) + O(q^3) \\ &= 1 - 2 + 2Nq - N(N-1)q^2 + 1 - 2Nq + 2N(N-1)q^2 + O(q^3) \\ &= N(N-1)q^2 + O(q^3) \\ &= \frac{N(N-1)}{|\mathcal{X}|^6} + O\left(\frac{1}{|\mathcal{X}|^9}\right) \\ &= \frac{N^2}{|\mathcal{X}|^6}(1 + o(1)) \end{aligned} \tag{S1.3}$$

Therefore, in the regime of interest ($N \gg |\mathcal{X}|$ but $N \ll |\mathcal{X}|^3$):

$$\boxed{p_1 = \frac{N^2}{|\mathcal{X}|^6}(1 + o(1))} \tag{S1.4}$$

Equation (S1.4) gives an exact expression for the probability (up to lower-order terms) that the *single* context c provides an *evidence pair* for the functional equivalence of fragments i and j.

Remark on "lucky coincidences" Because functional k-equivalence demands *consistency* across all k evidences, a single coincidental equality $f(i,c) = f(j,c)$ with $i \not\sim j$ can only masquerade as evidence when $k = 1$ *and* the dataset lacks any contradicting context c'. For $k \geq 2$ the joint probability that two *independent* contexts simultaneously produce such coincidences is $|\mathcal{X}|^{-k}$ per fragment pair and hence negligible relative to the isolate probability once $N = \Omega(|\mathcal{X}|^2)$. Consequently, the effects of such "lucky coincidences" on the lower bound in Res. 6.1 can be neglected.

E.2 Step 2: probability of observing k evidences for one fixed pair

Having established the probability p_1 for a single evidence pair in Step 1, we now derive p_k, the probability that a dataset provides at least k distinct contexts as evidence for the functional equivalence of two fragments i and j. This probability will determine the edge probability in the random graph model analyzed in Step 3.

Indicators for one fragment pair Fix two distinct first-hop fragments $i, j \in E_b$ and, for each context (third token) $c \in \mathcal{X}$, set

$$Z_{ij}(c) := \mathbf{1}[(i,c) \in D]\,\mathbf{1}[(j,c) \in D].$$

Thus $Z_{ij}(c) = 1$ exactly when the single context c supplies an *evidence pair* for the functional equivalence of i and j.

Single-context success probability From (S1.4),

$$p_1 := \Pr[Z_{ij}(c) = 1] = \frac{N^2}{|\mathcal{X}|^6}(1 + o(1)), \qquad (|\mathcal{X}| \to \infty). \tag{S2.1}$$

Negatively correlated counts and an i.i.d. surrogate Because all N draws come from a single multinomial $(N, 1/|\mathcal{X}|^3)$, the indicators $\{Z_{ij}(c)\}_{c\in\mathcal{X}}$ are *negatively correlated*: drawing many triples with one context leaves fewer draws for the others. Negative correlation *decreases* the probability that several $Z_{ij}(c)$ equal 1 simultaneously. Hence the tail probability for the true count

$$Y_{ij} := \sum_{c\in\mathcal{X}} Z_{ij}(c)$$

is *upper-bounded* by the tail of an i.i.d. binomial variable with the same single-trial success probability p_1. Concretely, define

$$Y_{ij}^{\star} \sim \mathrm{Binom}(|\mathcal{X}|, p_1), \qquad p_1 = \frac{N^2}{|\mathcal{X}|^6}(1 + o(1)).$$

Then for every real t

$$\Pr[Y_{ij} \geq t] \leq \Pr[Y_{ij}^{\star} \geq t].$$

Using $Y_{ij}^{\star}$ therefore *overestimates* the chance of obtaining k or more distinct evidence contexts, which is conservative for our goal of deriving a lower bound on the required dataset size N.

Since $Y_{ij}^{\star}$ is binomial with mean

$$\mu := \mathbb{E}[Y_{ij}^{\star}] = |\mathcal{X}|\, p_1 = \frac{N^2}{|\mathcal{X}|^5}(1 + o(1)), \tag{S2.2}$$

we may work henceforth with $Y_{ij}^{\star}$ alone; the resulting bounds apply verbatim to the original Y_{ij}.

Poisson tail via Le Cam Le Cam's theorem (Le Cam, 1960) states that the total-variation distance between $\mathrm{Binom}(n, p)$ and $\mathrm{Poisson}(\mu = np)$ is at most $2np^2$. At the scaling that will emerge in Step 3 ($N = |\mathcal{X}|^{2.5 - \frac{0.5}{k}}$), one has $2|\mathcal{X}|p_1^2 = 2|\mathcal{X}|^{-1-\frac{2}{k}} \to 0$ and $\mu = |\mathcal{X}|^{-\frac{1}{k}} \to 0$. Thus

$$\Pr[Y_{ij} \geq k] \leq \Pr[Y_{ij}^{\star} \geq k] = \sum_{r=k}^{\infty} \frac{e^{-\mu}\mu^r}{r!} \leq \frac{\mu^k}{k!}(1 + o(1)). \tag{S2.3}$$

This upper bound $p_k := \mu^k/k!\,(1 + o(1))$ will be the edge probability used in the connectivity threshold of Step 3.

E.3 Step 3: connectivity inside each equivalence class

With the edge probability p_k from Step 2, we now analyze when the k-evidence graphs become connected. Recall that under our learner assumption, perfect generalization requires every equivalence class to form a connected component in the k-evidence graph. We model this as a random graph connectivity problem.

Random graph construction Fix one balanced class E_b of size $n := |\mathcal{X}|$ (Assumption (A1)). Create a graph G_b whose vertices are the first-hop pairs in E_b, and place an undirected edge $\{i, j\}$ exactly when the pair (i, j) has been observed in *at least k distinct shared contexts*. By Step E.2, each potential edge appears with probability

$$p_k = \Pr[Y_{ij} \geq k] = \frac{\mu^k}{k!}, \qquad \mu = \frac{N^2}{|\mathcal{X}|^5}. \tag{S3.1}$$

Why G_b can be approximated as Erdős–Rényi As mentioned earlier, the dependence among distinct edges in G_b arises from the constraint that the total sample size is N, which induces a *negative correlation*. Such negative dependence *reduces* the likelihood of simultaneously creating many edges. Consequently, viewing G_b as an independent Erdős–Rényi graph $G(n, p_k)$ provides a conservative model, and any threshold we derive for connectivity under independence remains valid (or becomes easier to satisfy).

Classwise connectivity threshold For Erdős–Rényi graphs, the classical result of Erdős–Rényi (Erdős & Rényi, 1960) states

$$\Pr[G(n, p) \text{ is connected}] \xrightarrow[n\to\infty]{} 1 \quad \Longleftrightarrow \quad p \geq \frac{\log n + \omega(1)}{n}.$$

Setting $n = |\mathcal{X}|$ and $p = p_k$ yields the requirement

$$p_k = \frac{\mu^k}{k!} \geq \frac{\log|\mathcal{X}|}{|\mathcal{X}|}(1 + o(1)).$$

Substituting $\mu = N^2/|\mathcal{X}|^5$ and rearranging gives

$$\frac{N^{2k}}{|\mathcal{X}|^{5k}\, k!} \gtrsim \frac{\log|\mathcal{X}|}{|\mathcal{X}|}, \qquad \Longrightarrow \qquad N \gtrsim |\mathcal{X}|^{\frac{5k-1}{2k}} \left(k! \log|\mathcal{X}|\right)^{1/(2k)}.$$

Because $(k! \log|\mathcal{X}|)^{1/(2k)} = (\log|\mathcal{X}|)^{O(1/k)}$ grows only poly-logarithmically, we hide it inside a $\tilde{\Omega}(\,\cdot\,)$:

$$N = \tilde{\Omega}(|\mathcal{X}|^{\,2.5-\frac{0.5}{k}}).$$

Since every class must be connected to achieve the data–sufficiency criterion in the problem setting, we conclude

$$\boxed{N_{\mathrm{req}}(|\mathcal{X}|, k) = \tilde{\Omega}(|\mathcal{X}|^{\,2.5-\frac{0.5}{k}})} \quad \text{(with high probability).}$$

As a remark, we note that Erdős–Rényi theory also shows that if $N \leq |\mathcal{X}|^{\,2.5-\frac{0.5}{k}}/(\log|\mathcal{X}|)^{1/(2k)}$ then each G_b is *disconnected* with high probability, so the exponent $2.5 - \frac{0.5}{k}$ is in fact *sharp* (up to poly-logarithmic factors).

F ADDITIONAL RESULTS FOR POWER-LAW SCALING ANALYSIS

F.1 MEASUREMENT PROTOCOL FOR N_{req}

To empirically determine the minimum dataset size required for reliable compositional generalization (N_{req}), we develop a measurement protocol that accounts for practical computational constraints while ensuring robustness. For each token set size $|\mathcal{X}|$ and task structure, we test multiple dataset sizes until we identify the threshold point where the model successfully generalizes to the ID test set.

Specifically, our criterion for "reliable generalization" on ID is defined as:

- The model must reach ID test accuracy of 0.99 within 100 epochs after achieving training accuracy > 0.99.

This protocol balances several considerations:

1. **Training-to-generalization delay**: Larger datasets naturally require more iterations to fit training data. By measuring epochs after reaching training accuracy > 0.99, we focus on the generalization gap rather than conflating it with initial training difficulty.
2. **Epoch-based measurement**: Using epochs rather than raw training steps ensures that the model sees each functional equivalence evidence approximately the same number of times, regardless of dataset size. This provides a fairer comparison across different dataset sizes.
3. **Practical time constraints**: While indefinite training might eventually yield generalization with smaller datasets, we established a reasonable upper bound (100 epochs post-training convergence) to reflect practical limitations.
4. **Measurement precision**: For each identified N_{req}, we verified that 75% of this dataset size consistently failed to meet our generalization criterion. This establishes that our measurement error is at most $-\log(0.75) = 0.125$ in log scale, providing confidence in the derived power-law exponents.

F.2 MEASURED POWER-LAW SCALING CONSTANTS ACROSS TASK STRUCTURES AND MODEL SIZES

Using our measurement protocol, we measure the required dataset size N_{req} across three different compositional structures (2-HOP, PARALLEL-2-HOP, and 3-HOP) and three model scales (68M, 96M, and 1.5B parameters). For each task structure, we vary the token set size $|\mathcal{X}|$ from 50 to 200, allowing us to observe the scaling relationship.

Table 2 presents the power-law exponents obtained by linear fitting $\log(|\mathcal{X}|)$ vs. $\log(N_{\text{req}})$ plots, all with $R^2 > 0.99$. The consistency of exponents across model sizes suggests that the observed power-law scaling relates to properties of the compositional tasks themselves, rather than model capacity. This observation aligns with our theoretical derivation in Section 5.1, which predicts that the required dataset size scales at least quadratically with token set size.

Table 2: Power law exponents for different tasks and GPT-2 sizes, obtained by linear fitting $\log(|\mathcal{X}|)$ vs. $\log(N_{\text{req}})$ plots. $R^2 > 0.99$ for all linear fitting.

Model Size	2-HOP	PARALLEL-2-HOP	3-HOP
68M	2.13	2.47	2.61
96M	2.26	2.35	2.50
1.5B	2.28	2.17	2.60

F.3 ROBUSTNESS TO HYPERPARAMETER VARIATIONS

To verify that our observed power-law scaling relationship is not an artifact of specific hyperparameter choices, we conduct ablation studies with modified training configurations. Figure 15 demonstrates that for the 2-HOP task with $|\mathcal{X}|= 50$, the following changes did not significantly affect the measured N_{req} or the derived power-law exponent:

1. **Learning rate reduction**: Halving the learning rate from 8e-4 to 4e-4
2. **Weight decay reduction**: Decreasing weight decay by a factor of 10 (from 0.1 to 0.01)

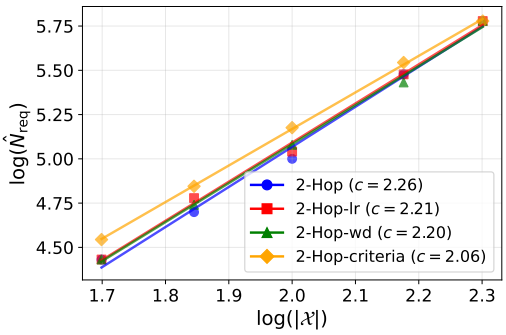


Figure 15: Robustness of power-law scaling relationship to hyperparameter variations in the 2-HOP task with $|\mathcal{X}|= 50$. Each line shows the training and test accuracy curves for a different configuration: (1) baseline, (2) reduced learning rate (4e-4, half of baseline), (3) reduced weight decay (0.01, one-tenth of baseline), and (4) changed generalization criteria (test accuracy > 0.95 within 10 epochs after training accuracy > 0.95). $R^2 > 0.99$ for all linear fitting.

3. **Generalization criteria modification**: Requiring test accuracy > 0.95 within 10 epochs after training accuracy > 0.95

This robustness to hyperparameter variations suggests that the power-law relationship between token set size and required dataset size is primarily a property of the compositional generalization process, rather than an artifact of specific optimization settings.

G Detailed analysis for Non-tree task

This section provides additional analyses that support our findings in Sec. 7 regarding the challenges of path ambiguity in the Non-tree task.

G.1 Coverage analysis

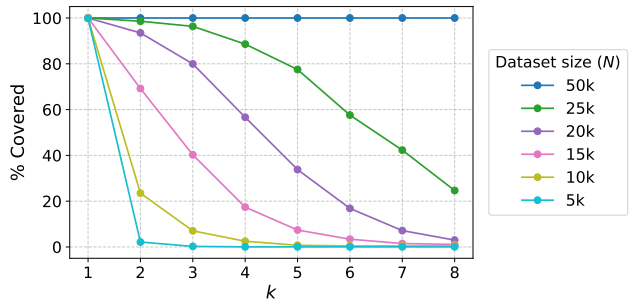


Figure 16: Coverage analysis for Non-tree task with $|\mathcal{X}|= 50$. The graph shows the percentage of ID test data covered at different k values across various dataset sizes (N). Compared to the 2-Hop task (Fig. 3, left), Non-tree has significantly lower coverage at equivalent dataset sizes, indicating that path ambiguity impedes the formation of functional equivalence relationships.

Fig. 16 demonstrates that with equivalent training dataset sizes, a smaller percentage of ID test examples fall inside k-coverage for the Non-tree task compared to the 2-Hop task shown in Fig. 3 (Left). This aligns with our theoretical analysis in Sec. 7, which predicts that path ambiguity limits the establishment of functional equivalence relationships between input subsequences, as the model cannot generalize across different x_2 values in the Non-tree structure even when they produce the same intermediate state $b = f_1(x_1, x_2)$.

G.2 Effect of model scaling

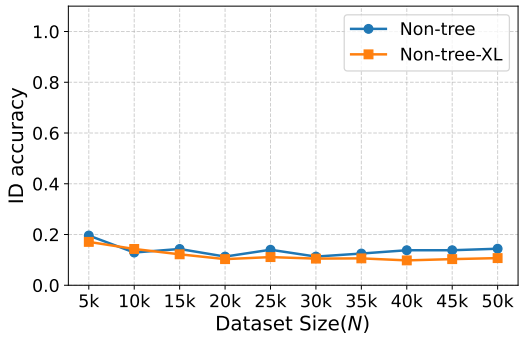


Figure 17: ID test accuracy comparison between GPT-2 (96M parameters) and GPT-2-XL (1.5B parameters) on the Non-tree task with $|\mathcal{X}|= 50$, measured 100 epochs after training accuracy exceeds 0.99. Despite the 15x increase in parameter count, the accuracy does not increase.

Fig. 17 shows that scaling up the model size to GPT-2-XL (1.5B parameters) does not significantly improve generalization performance on the Non-tree task, even when measured 100 epochs after reaching training accuracy > 0.99. This suggests that the challenges posed by path ambiguity cannot be overcome simply by increasing model capacity, supporting our claim that the limitation is structural rather than related to model capacity.

G.3 Comparison of Mamba and GPT-2 on Non-tree task

Fig. 18 shows that Mamba model (4 layers, hidden dimension of 256, trained with learning rate of 0.008) shows a similar trend of ID test accuracy on Non-tree task compared to GPT-2, suggesting that the generalization failure is more likely due to the task structure itself, rather than a specific model architecture.

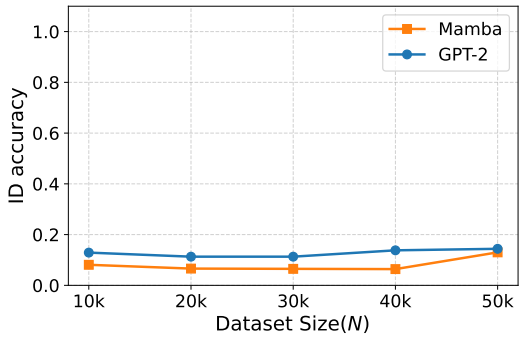


Figure 18: ID test accuracy comparison between GPT-2 and Mamba on the NON-TREE task with $|\mathcal{X}|= 50$, measured 100 epochs after training accuracy exceeds 0.99.

G.4 REPRESENTATION ANALYSIS IN SUCCESSFUL GENERALIZATION

For a model that eventually achieved near-perfect ID accuracy (0.96) after extended training (36k epochs, $|\mathcal{X}|= 50$, $N = 50$k), we conduct causal tracing analysis to understand how it achieves generalization despite path ambiguity.

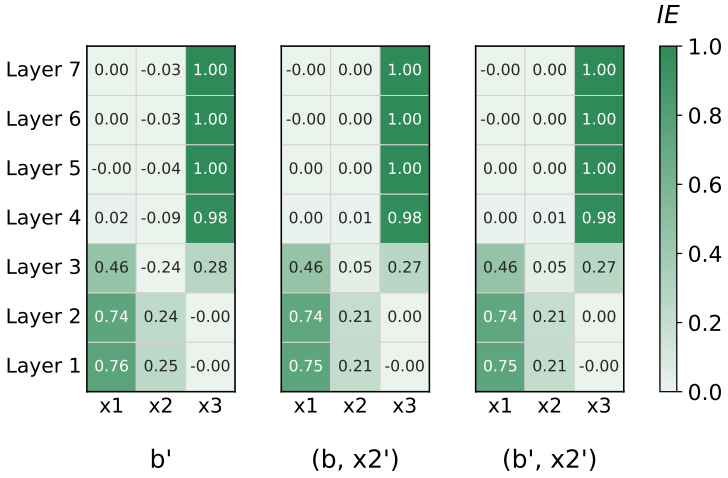


Figure 19: Causal tracing analysis for the NON-TREE model after extended training. The heatmap shows indirect effect values across different layer-token positions. **Left**: perturbation leading to different intermediate state $b = f_1(x_1, x_2)$. **Middle**: same b but different x_2. **Right**: different b and x_2.

The causal tracing results in Fig. 19 reveal how the model achieves generalization in the presence of path ambiguity. Across all perturbation strategies, the model's predictions show strong causal dependence on representations at both the x_1 and x_2 positions, indicating reliance on direct access to both input tokens rather than an abstracted intermediate computation. This pattern contrasts sharply with the 2-HOP task, where causal effects concentrate primarily at positions corresponding to clustered functional equivalence representations.

This analysis demonstrates that even models achieving high accuracy on NON-TREE tasks do so by developing context-dependent representations rather than unified abstractions of intermediate states. The model forms separate computational pathways conditioned on the x_2 value, rather than learning a single unified representation of the intermediate state $b = f_1(x_1, x_2)$. This represents a fundamentally different solution strategy compared to the 2-HOP task, with implications for both generalization capability and interpretability.

H DETAILED DISCUSSION ON THE TAXONOMY FUR UNDERSTANDING GENERALIZATION MECHANISMS

In this section, we initiate a discussion to disambiguate the mixed mechanisms of generalization into isolated testable parts by sketching a preliminary taxonomy that distinguishes three complementary mechanisms of generalization. We note that we do not view our categorization as a complete one.

Type-I: Functional equivalence-based generalization (pattern matching). This is precisely what we formalized through this work: models learn that different input fragments yield identical results in shared contexts, enabling generalization to new fragment combinations. Crucially, this generalization remains bounded by coverage, and reliable generalization fails without sufficient functional equivalence evidence. In other words, it describes the ceiling of pattern matching.

Type-II: Function property-based generalization. This mechanism exploits intrinsic properties of individual primitive functions, e.g., algebraic invariances such as commutativity or *input irrelevance*, where certain arguments never affect the output (e.g., $f(x_1, x_2) = f(x_1)$ even when distractor x_2 is present (Wen et al., 2025)). Unlike the previous type, this mechanism explains the generalization beyond the coverage by leveraging 'global' properties that hold across all possible inputs of a primitive, beyond what is actually observed. We interpret the Reversal Curse phenomenon (Berglund et al., 2024) as an example of the layered nature of challenges across multiple generalization types. Our framework predicts the failure of pattern matching on this problem, since training on "A is B" provides no functional equivalence evidence for "B is A^{-1}". An architectural modifications to learn inverse mappings from the same training data to handle this problem (Lv et al., 2024) can be interpreted as a utilization of Type-II generalization to enable generalization beyond coverage.

Type-III: Shared-operator generalization. This mechanism emerges through the reuse of identical primitive functions across computational positions (e.g., when $f_1 = f_2$). Recurrent architectures (Hochreiter & Schmidhuber, 1997) exemplify the utilization of this through weight sharing across time steps, enabling processing of variable-length sequences (Graves et al., 2014). Similarly, it has been reported in Transformers with inductive biases towards reuse of the same computation through parameter sharing (Dehghani et al., 2019; Csordás et al., 2021; Wang et al., 2024) can improve generalization on complex compositional tasks where the same primitive function can be reused in various contexts. We interpret this mechanism as exploiting structural repetition.

Distinguishing mechanisms from phenomena. Compared to prior categorizations of generalization, which focus on observed phenomena (Lake & Baroni, 2018; Hupkes et al., 2020), we categorize the underlying mechanisms. As noted in Sec. 1, many behavioral studies have examined tasks mixing functional equivalence, primitives' intrinsic properties, and operator reuse within the same benchmark, making it difficult to pinpoint the true source of success or failure. We therefore advocate clearer experimental control and community discussion around this mechanistic distinction to sharpen future analyses of neural generalization.

Implications and future directions. Real compositional tasks typically involve combinations of all three types (and possibly more). While preliminary, we believe this taxonomy guides future research design on constructive characterization of neural networks' generalization behaviors on discrete sequence tasks. In this broader context, this work can be understood as a characterization and formalization of pattern-matching generalization to clarify its specific boundaries. When models succeed beyond our coverage predictions, we view these as exploiting other generalization mechanisms, i.e., beyond pattern matching. Our focused study suggests that challenges to reliable generalization remain as long as models rely primarily on pattern matching, requiring methodological innovations that harness non-pattern-matching mechanisms, e.g., variable binding. We hope this preliminary taxonomy serves as a research program towards our better understanding of generalization, and confirming or refuting its utility is an empirical matter that we invite the community to explore.

I PARTIAL COMPUTATION OBSERVATION DRIVES THE ALIGNMENT OF FUNCTIONAL EQUIVALENCE REPRESENTATION AND VOCABULARY SPACE

In this section, we investigate how exposure to partial computations affects the interpretability of intermediate state representations through vocabulary space alignment. We compare two training conditions on a modified 2-HOP task with $|\mathcal{X}|= 50$ and $N = 10$k, after 40k epochs of training:

1. **Standard Training**: $f_1 \neq f_2$, model only sees complete two-hop examples $(x_1, x_2, x_3) \mapsto t$.
2. **With Partial Computation**: $f_1 = f_2$, model additionally sees all possible partial computations $(x_1, x_2) \mapsto b$ where $b = f_1(x_1, x_2)$ (2,500 partial examples, not counted toward the $N = 10$k two-hop training data).

To assess interpretability, we measure the Mean Reciprocal Rank (MRR) of intermediate state representations when projected to vocabulary space using the unembedding matrix. Low MRR indicates that the model's internal representation of intermediate state b aligns with the corresponding vocabulary token.

Fig. 20 shows a striking contrast between the two conditions. Under standard training, the MRR score remains very high throughout training, indicating that intermediate representations are not aligned with vocabulary space despite the model successfully learning the compositional task. However, when partial computations are included, the MRR score becomes very high, demonstrating clear vocabulary alignment.

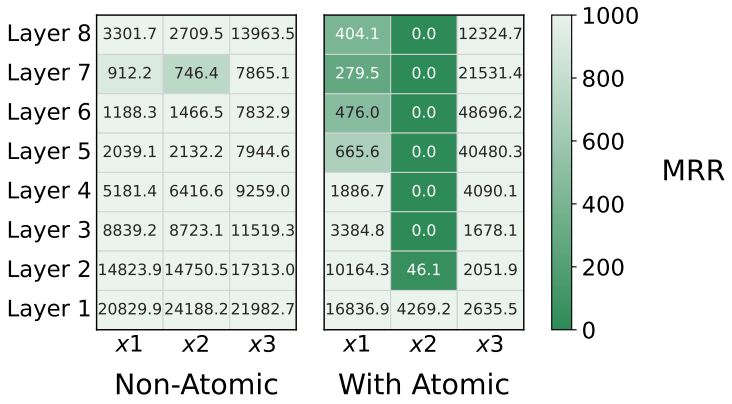


Figure 20: MRR scores for intermediate state representations projected to vocabulary space. **Left:** Standard training ($f_1 \neq f_2$, no partial computation) shows very high MRR regardless of position and layer. **Right:** Training with partial computation ($f_1 = f_2$, with partial examples) shows MRR of 0 in layers 3 to 8 at position x_2, indicating strong vocabulary alignment.

This experiment suggests that **logit lens interpretability is orthogonal to functional equivalence representation formation**. A model can develop functionally correct intermediate representations that enable compositional generalization while remaining completely uninterpretable through standard vocabulary projection techniques. Interpretability via logit lens requires explicit vocabulary anchoring through exposure to partial computations that map intermediate states to vocabulary tokens.

This finding has important implications for mechanistic interpretability research: the absence of interpretable representations through logit lens does not indicate the absence of structured internal computation. Furthermore, it suggests that interpretability techniques may need to account for how training data shapes the alignment between internal representations and vocabulary space, rather than assuming such alignment emerges naturally from task performance.