CHARACTERIZING PATTERN MATCHING AND ITS LIMITS ON COMPOSITIONAL TASK STRUCTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite impressive capabilities, LLMs' successes often rely on pattern-matching behaviors, yet these are also linked to 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., identifying pairs of subsequences of inputs that consistently lead to identical results when the rest of the input is held constant. 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 well predicted by the number of contexts witnessing the relevant functional equivalence. (2) We prove a tight sample complexity bound of learning a two-hop structure by identifying the exponent of the data scaling law for perfect in-domain generalization. Our empirical results align with the theoretical prediction, under 20× parameter scaling and across 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 suggest that the core mechanism of generalization might be "pattern matching", i.e., models learning 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., 2024a; 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 rather than characterizing it predictively. 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 like decoder-only Transformers (Vaswani et al., 2017) and Mamba (Gu & Dao, 2024) perform generalization through pattern matching. Specifically, we first 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.

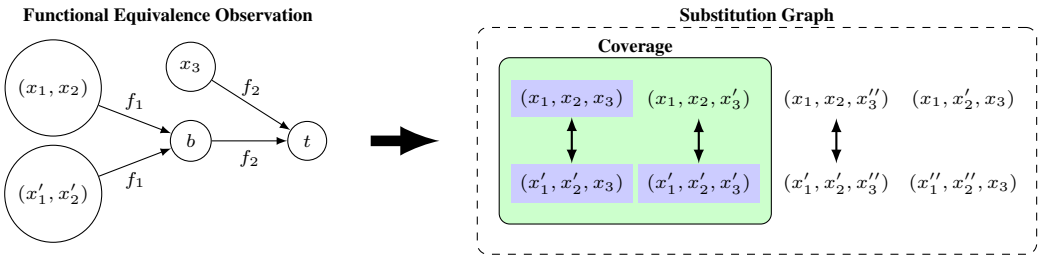


Figure 1: **Illustration of functional equivalence. Left:** In a two-hop task $(x_1, x_2, x_3) \mapsto t$ with $t = f_2(f_1(x_1, x_2), x_3)$, two fragments (x_1, x_2) and (x'_1, x'_2) satisfying $f_1(x_1, x_2) = f_1(x'_1, x'_2) = b$ consistently yield the same final output when combined with the same 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.

Moreover, to isolate and study pattern-matching behaviors, we use controlled setups that deliberately remove other generalization sources and make 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 well predicted 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). FIXED
- **We prove a tight sample complexity bound of pattern matching on a two-hop structure by identifying the exponent of the data scaling law (in terms of the token set size) which is necessary and sufficient for perfect in-domain generalization (Theorem 6.1).** The measured power-law exponent agrees with our theoretical bounds and remains stable under roughly 20× 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). FIXED
- **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. 9.2 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., 2024a), 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's 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 we systematically analyze models' behaviors with controlled tasks that are designed to isolate pattern-matching regimes grounded on our framework.

Mechanistic interpretability. Mechanistic interpretability studies aim to understand how sub-mechanisms 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., 2024a). Our framework complements these works by providing mechanistic insights about 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

[1] For brevity, we use a shared token set $\mathcal{X}$ in the main text. A more general notion using position-specific domains (e.g., '$\mathcal{X}_i$') can be used; see App. E.1.

[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.

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. (**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 FIXED
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)**.** The *k-coverage* of $D \subset \mathcal{X}^\ell$, denoted by $\mathrm{Cover}_k(D)$, is the set of all FIXED
inputs in $\mathcal{X}^\ell$ connected to an $\boldsymbol{x} \in D$ with a path in the substitution graph $\mathcal{G}^{(D,k)}$. FIXED

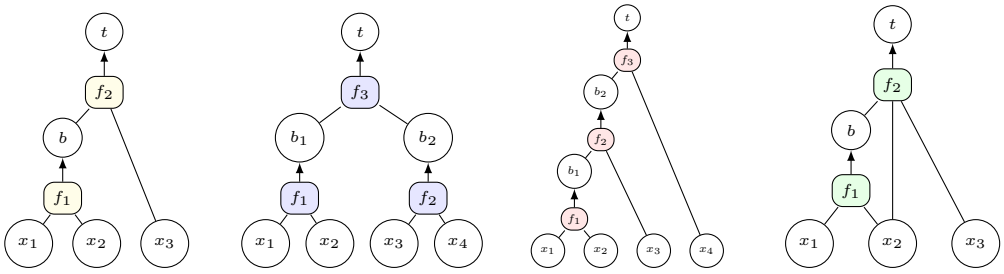
Note that the notion of coverage is a stricter condition of the canonical definition of in-domain (ID), which is obtained by random train/test split (Wang et al., 2024a) or taking combinations of observed internal computations (Dziri et al., 2023). In Sec. 5, we demonstrate that learners may not necessarily 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 a product space of token sets to control the generalization sources 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.

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



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

Figure 2: Four synthetic task structures we study.

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 as a base model (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 $f_2(b, x_3)$ in 2-HOP task) are observed during training, but their 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

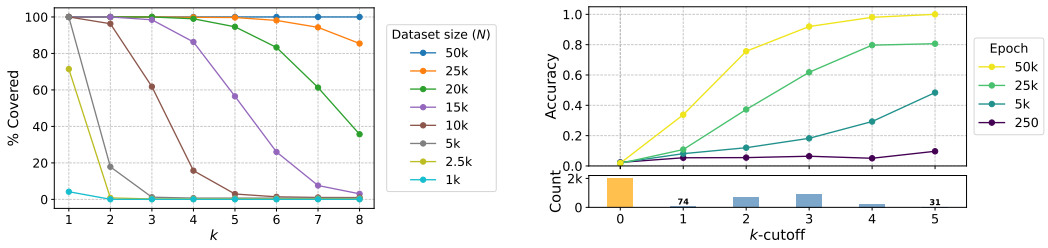


Figure 3: **Left**: Percentage of covered ID data depending on k values and dataset size (N), for 2-HOP task ($|\mathcal{X}| = 50$). **Right**: Test accuracy depending on k-cutoff values for 2-HOP task ($|\mathcal{X}| = 50$, N=10k). Each line represents a different training checkpoint. Note that out-of-coverage ($k = 0$) 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 the correlation between k-coverage and ID generalization performance of the 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, we demonstrate in our experiments that minimal coverage (i.e., $k = 1$) alone is practically insufficient for ID generalization. To demonstrate this, let us fix a training dataset D and define the k-cutoff of each input sequence as the lowest value of 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 the GPT-2 model for each group across 50k training epochs. As shown in Fig. 3 (Right), ID test accuracy shows a strongly positive correlation 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 ID generalization in practice requires a *robust* coverage so the model can confidently identify and utilize functional equivalence**

relationships. The parameter k effectively quantifies this evidence strength, directly impacting generalization speed and reliability. Second, while our experiments use uniformly sampled datasets, 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). Despite technically being in-distribution, rare combinations naturally receive limited evidence of functional equivalence (low k), effectively and practically placing them outside the coverage. We believe that our insights will motivate future research on targeted data augmentation strategies to maximize k-coverage.

5.2 Latent Representation Clusters Drive Pattern Matching on Coverage

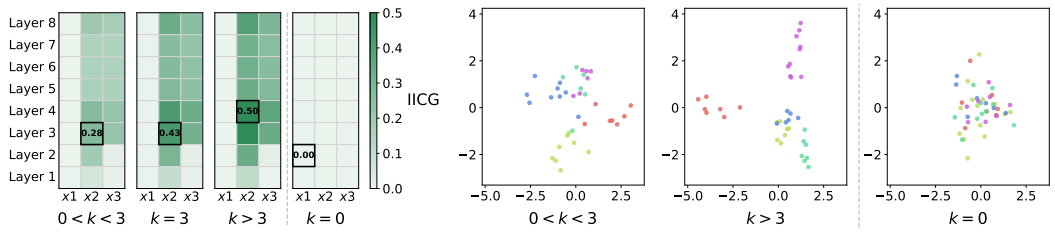


Figure 4: **Left**: Heatmap of Intra-Inter Cosine Gap (IICG) across layers and positions, sliced by k-cutoff. Higher IICG values indicate stronger clustering of representations that share the same intermediate state. The positions with the highest IICG values are marked with squares. **Right**: PCA visualization 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 positive correlation: 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 literature (Wang et al., 2024a) 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

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

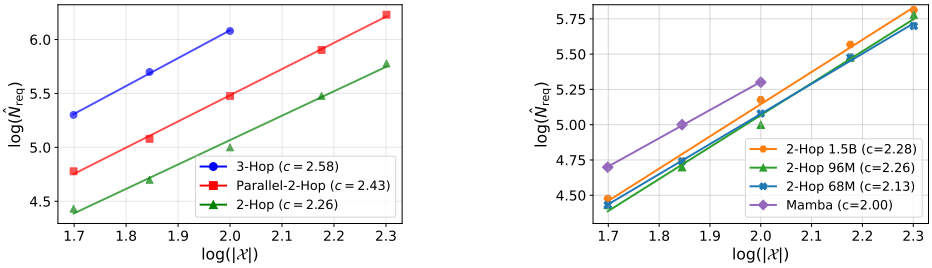


Figure 5: **Left:** Log-log plot of measured $\hat{N}_{\text{req}}$ vs. token set size ($|\mathcal{X}|$) across three compositional tasks. The slope c corresponds to the empirical power-law scaling exponent. Omitted points for 3-HOP are due to prohibitively large dataset requirements. **Right:** Power-law 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.

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 that generalization is constrained by the k-coverage, how should the train data size scale in the token set size $|\mathcal{X}|$ to achieve it completely? In practical terms, this question seeks to determine the amount of data necessary and/or sufficient to cover all ID combinations with k-coverage, which is crucial for understanding the data cost for pattern-matching generalization. To quantify the data cost, we establish and prove the following sample-complexity upper bound for learning a 2-HOP task (the complete statement and proof are provided in App. E): FIXED

Theorem 6.1 (Informal; Corollaries E.9 and E.17)**.** *Consider a* 2-HOP *task with a token set of size n. For a uniformly randomly sampled train dataset D of size N, consider a learner that generalizes within the k-coverage of D. Then, for large enough n, the learner achieves perfect ID generalization with high probability if $N \gtrsim n^c$ with $c = 2.5 - \frac{0.5}{k}$. In contrast, the learner (with $k \geq 2$) does not achive perfect ID generalization with high probability for some* 2-HOP *task if $n^2 \lesssim N \lesssim n^c$. Here, we ignore the polylogarithmic factors in n.* FIXED

Theorem 6.1 presents a tight sample complexity bound $\tilde{\Theta}(n^c)$ for pattern-matching-based generalization, showing that the training dataset required to ensure full ID generalization grows polynomially in the token set size n, with an exponent $c \in [2, 2.5)$. 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 for the 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. 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. FIXED

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.[5] We note that our result aligns with the practical observation that parameter scaling does not significantly improve the multi-hop reasoning

[5]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).

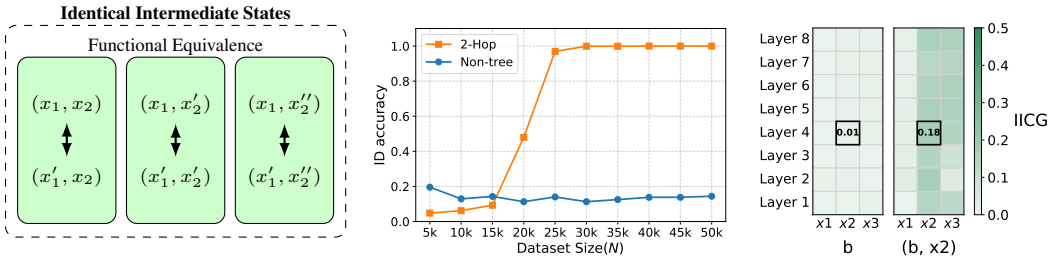


Figure 6: **Left**: In NON-TREE task, the representations of input subsequences with the same intermediate state $b = f_1(x_1, x_2)$ are split into multiple context-dependent state representations, conditioned on x_2 value. **Middle**: ID test accuracy after standard training with varying training dataset 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).

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 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.[6] 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 require 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.

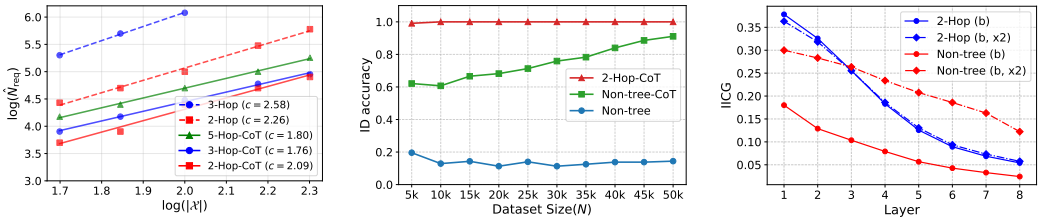


Figure 7: **Left:** Power-law scaling of required dataset size vs. token set size for tasks with CoT supervision. $R^2 > 0.98$ for all linear fits. **Middle:** Comparison of ID test Accuracy of NON-TREE task ($|\mathcal{X}| = 50$) with and without CoT supervision. **Right:** IICG score comparison for 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.

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 the 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.

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 the 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

9.1 Practical Implications

Although the experiments in this work have focused on synthetic setups to dissect purely pattern-matching behaviors, our results provide valuable practical implications for understanding and improving LLMs on natural language tasks. FIXED

First, it accounts for the data-hungry nature of compositional tasks by demonstrating that robust coverage is required for reliable generalization (Lake et al., 2017). For instance, our scaling-law analysis (Theorem 6.1) provides a quantitative explanation for the observed behavior that the data demand for generalization on multi-hop natural language data dramatically increases with the number of hops (Yao et al., 2025). Such a data-hungry nature of compositional tasks is also well observed in semantic parsing (Dong & Lapata, 2016), where it has been shown that the diversity of data (i.e., the same component is shown in diverse contexts) is more important than the sheer size

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

of the dataset for better generalization (Keysers et al., 2020). Based on these insights, we believe our results will motivate strategic data augmentation methods that seek to maximize coverage by ensuring diverse shared contexts for functionally equivalent components (Andreas, 2020).

Second, our framework helps interpret various real-world failure cases observed in LLMs as originating from their reliance on pattern matching. For example, our framework aligns with the reversal curse phenomenon, which is the failure to automatically infer that 'A is a child of B' by observing that 'B is a parent of A' (Berglund et al., 2024). This is because pattern matching is fundamentally incapable of generalizing to the reversing relation without explicit functional equivalence evidence for that in the training dataset. Similarly, our framework can explain the notoriously hard problem of logical negation in LLMs (Truong et al., 2023). A purely pattern-matching learner cannot deduce that the negation rule (i.e., if a statement 'p' is true, then 'not p' is false and vice versa) should apply to any well-formed statement, including the ones that aren't observed, through finite observations. Finally, difficulties in solving complex planning tasks with LLMs (Valmeekam et al., 2023; Stechly et al., 2024; Kambhampati et al., 2024; Wang et al., 2024b) might be partially attributed to path ambiguities, since such tasks likely require the correct tracking of intermediate states, which can be affected by various computational paths.

Hence, we believe our framework and analyses can provide important practical insights, which future work can extend to investigate LLMs' pattern-matching behaviors under more realistic setups.

9.2 Towards a Taxonomy of Generalization Mechanisms

Natural language tasks possess algebraic and structural properties that differ from the random mappings we studied. For example, the same knowledge can be used in any hop of the multi-hop reasoning in practice, unlike our 2-Hop task that used $f_1 \neq f_2$. In practice, a learner may reasonably harness such properties of a given task to generalize beyond the boundary of pattern-matching defined by coverage. Therefore, it is natural to ask: **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: FIXED

- **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. See App. H for a complete discussion on the categorization of generalization mechanisms. FIXED

10 Conclusion

In this work, we formalized a framework for characterizing pattern matching. Our theoretical and experimental analyses yielded quantitative and predictive insights into modern neural networks' pattern-matching behaviors, moving beyond post-hoc accounts of a model's behavior on compositional tasks: (i) the alignment of instance-wise success with the strength of functional-equivalence evidence (Sec. 5), (ii) the theoretical identification and empirical verification of a sharp sample complexity bound for complete ID generalization on the 2-Hop task through pattern matching (Sec. 6), and (iii) the identification of the path ambiguity problem that impairs accuracy and interpretability even under high coverage and CoT supervision (Sec. 7 and 8). We anticipate that future work will build on this foundation, towards a more complete and constructive understanding of compositional generalization and its failures. FIXED

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.

David. author. Aldous. *Probability Approximations via the Poisson Clumping Heuristic*. New York, NY : Springer New York : Imprint: Springer, 1989. URL http://dx.doi.org/10.1007/978-1-4757-6283-9.

Jacob Andreas. Good-enough compositional data augmentation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7556–7566, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.676. URL https://aclanthology.org/2020.acl-main.676.

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.

Alexei Borodin, Andrei Okounkov, and Grigori Olshanski. Asymptotics of plancherel measures for symmetric groups. *Journal of the American Mathematical Society*, 13(3):481–515, 2000.

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.

Li Dong and Mirella Lapata. Language to logical form with neural attention. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 33–43, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1004. URL https://aclanthology.org/P16-1004/.

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.

Pál Erdős and Alfréd Rényi. On a classical problem of probability theory. *A Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei*, 6(1-2):215–220, 1961.

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

E. Godehardt and J. Jaworski. Two models of random intersection graphs for classification. In Manfred Schwaiger and Otto Opitz (eds.), *Exploratory Data Analysis in Empirical Research*, pp. 67–81, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-642-55721-7.

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.

Lars Holst. On birthday, collectors', occupancy and other classical urn problems. *International Statistical Review/Revue Internationale de Statistique*, pp. 15–27, 1986.

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.

Philippe Jacquet and Wojciech Szpankowski. Analytical depoissonization and its applications. *Theoretical Computer Science*, 201(1-2):1–62, 1998.

Kurt Johansson. The longest increasing subsequence in a random permutation and a unitary random matrix model. *Mathematical Research Letters*, 5(1):68–82, 1998.

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.

Marc Kac. On deviations between theoretical and empirical distributions. *Proceedings of the National Academy of Sciences*, 35(5):252–257, 1949.

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.

Michał Karoński, Edward R Scheinerman, and Karen B Singer-Cohen. On random intersection graphs: The subgraph problem. *Combinatorics, Probability and Computing*, 8(1-2):131–159, 1999.

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.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017.

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.

Donald J Newman. The double dixie cup problem. *The American Mathematical Monthly*, 67(1): 58–61, 1960.

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.

Katarzyna Rybarczyk. Sharp threshold functions for random intersection graphs via a coupling method. *The Electronic Journal of Combinatorics*, 18, 2011. doi: 10.37236/523.

Katarzyna Rybarczyk. The coupling method for inhomogeneous random intersection graphs. *The Electronic Journal of Combinatorics*, 24:P2.10, 2017.

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.

Karen B. Singer. *Random intersection graphs*. PhD thesis, The Johns Hopkins University, 1995.

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.

Thinh Hung Truong, Timothy Baldwin, Karin Verspoor, and Trevor Cohn. Language models are not naysayers: an analysis of language models on negation benchmarks. In Alexis Palmer and Jose Camacho-collados (eds.), *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pp. 101–114, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.starsem-1.10. URL https://aclanthology.org/2023.starsem-1.10/.

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*, 2024a. URL http://papers.nips.cc/paper_files/paper/2024/hash/ad217e0c7fecc71bdf48660ad6714b07-Abstract-Conference.html.

Kevin Wang, Junbo Li, Neel P. Bhatt, Yihan Xi, qiang liu, ufuk topcu, and Zhangyang Wang. Ontheplanning abilities of openAI's o1 models: Feasibility, optimality, and generalizability. In *Language Gamification - NeurIPS 2024 Workshop*, 2024b. URL https://openreview.net/forum?id=qgvQx30Z0R.

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/.

Yuekun Yao, Yupei Du, Dawei Zhu, Michael Hahn, and Alexander Koller. Language models can learn implicit multi-hop reasoning, but only if they have lots of training data. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9695–9713, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.490. URL https://aclanthology.org/2025.emnlp-main.490/.

Jun Zhao, Osman Yağan, and Virgil Gligor. On k-connectivity and minimum vertex degree in random s-intersection graphs. In *2015 Proceedings of the Twelfth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pp. 1–15. SIAM, 2014.

Jun Zhao, Osman Yağan, and Virgil Gligor. On connectivity and robustness in random intersection graphs. *IEEE Transactions on Automatic Control*, 62(5):2121–2136, 2017. doi: 10.1109/TAC.2016.2601564.

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

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 unseen 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., 2024a).

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

```
/* STEP 1: Build behavior maps for each subsequence pattern */
```
foreach *nonempty subset* $I \subsetneq [n]$ **do**
  Behavior$_I \leftarrow$ map from subsequence x_I to the mapping $\{x_{[n]\setminus I} \mapsto f(x) \mid x \in D\}$
end

```
/* STEP 2: Identify functionally equivalent subsequences */
```
foreach *nonempty subset* $I \subsetneq [n]$ **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$(D) \leftarrow \bigcup_{x \in D}$ ConnectedComponent(G, x)
return *Cover*(D)

Alg. 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

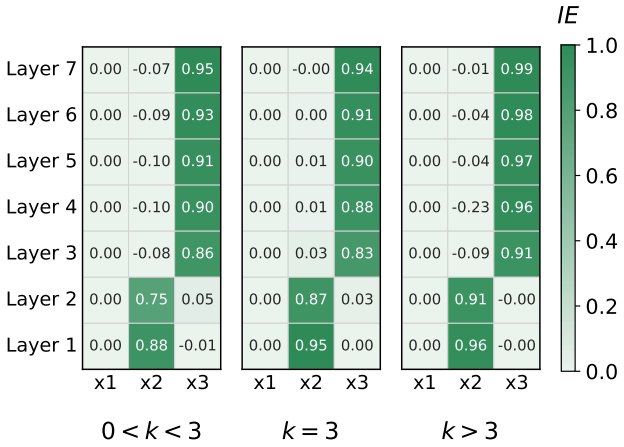


Figure 8: Causal tracing results for the 2-HOP task across different k-cutoff values, showing Indirect Effect (IE) scores at each layer and position.

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.

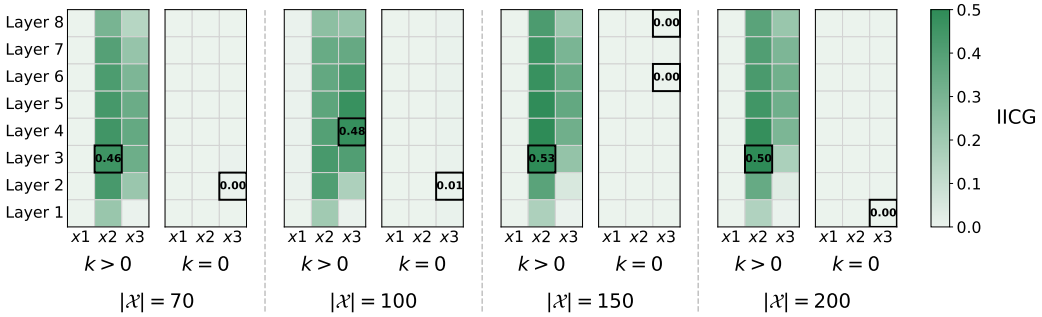


Figure 9: IICG heatmap across different token set sizes, showing consistent representation clustering patterns.

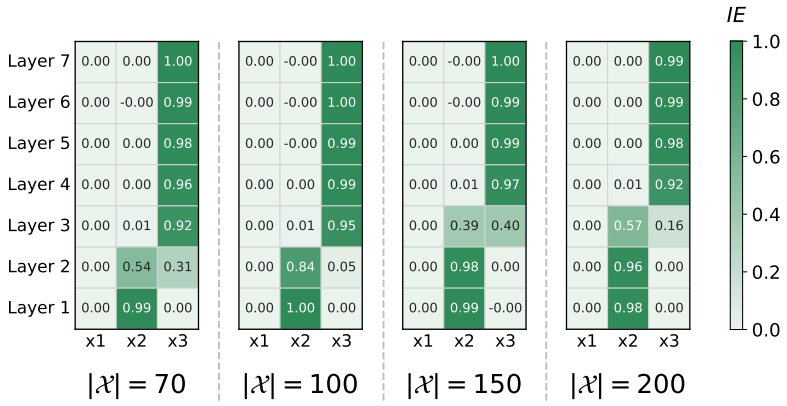


Figure 10: Causal tracing results showing indirect effect heatmaps for different token set sizes $|\mathcal{X}|$.

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.

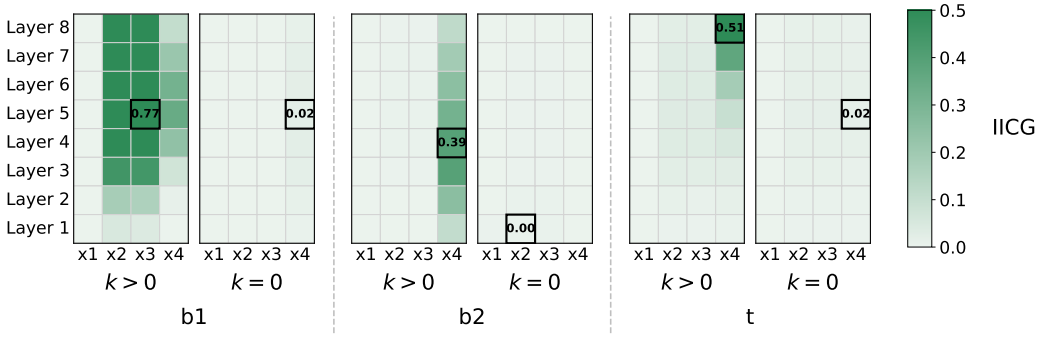


Figure 11: IICG heatmap for PARALLEL-2-HOP task with grouping strategies based on $b_1 = f_1(x_1, x_2)$ (**Left**), $b_2 = f_2(x_3, x_4)$ (**Middle**), and $t = f_3(b_1, b_2)$ (**Right**).

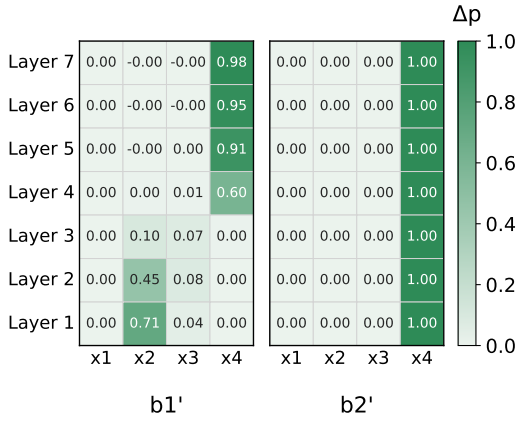


Figure 12: Causal tracing results showing indirect effect heatmap for PARALLEL-2-HOP task. **Left:** perturbation with different (x_1, x_2) pair leading to a different b_1 value. **Right:** perturbation with different (x_3, x_4) pair leading to a different b_2 value.

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.

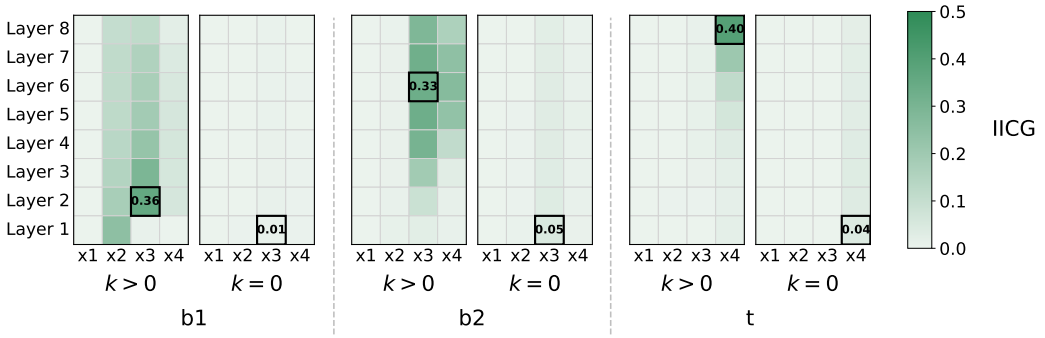


Figure 13: IICG heatmap for 3-HOP task with grouping strategies based on $b_1 = f_1(x_1, x_2)$ (**Left**), $b_2 = f_2(b_1, x_3)$ (**Middle**), and $t = f_3(b_2, x_4)$ (**Right**).

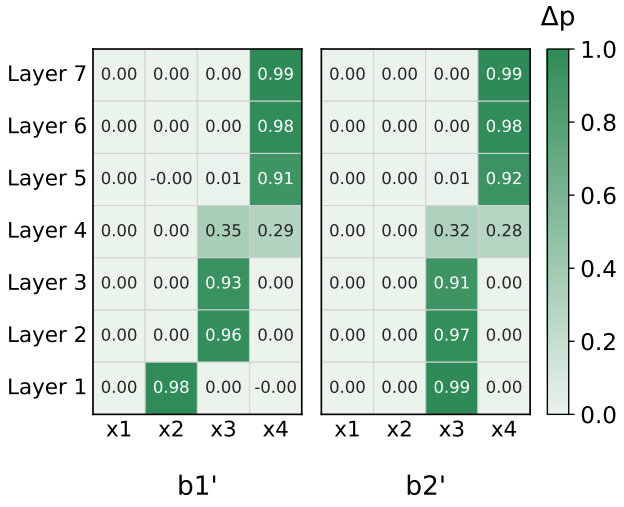


Figure 14: Causal tracing results showing indirect effect heatmap for 3-HOP task. **Left:** perturbation with different (x_1, x_2) pair leading to different b_1 value. **Right:** perturbation leading to different $b_2 = f_2(b_1, x_3)$ value.

E PROOFS: ASYMPTOTICALLY PERFECT COVERAGE IN 2-HOP TASK

We establish a scaling-law analysis for the 2-HOP task (as illustrated in Fig. 2a) within our formal framework for understanding pattern matching (Sec. 3). While a simplified statement (Theorem 6.1) is already presented in the main text, we provide the complete results of our analyses here in great detail, along with their rigorous and self-contained proofs. This appendix is long; we have organized it into the following structure. FIXED

- In App. E.1, we first review and (re)define the problem setting for 2-HOP task. Since the functional (k-)equivalence at indices 1 and 2 matters for the 2-HOP task, we argue that it is enough for us to focus on $I = \{1, 2\}$, thereby simplifying some key definitions (e.g., substitution graph, k-coverage) accordingly.
- In App. E.2, we state and prove our main theorems that characterize the number of training data sufficient for a learner to achieve a perfect in-domain generalization with high probability. There are two main theorems: one is for $k \geq 2$ (Theorem E.7) and another is for $k = 1$ (Theorem E.8). As a consequence, we further assume a specific regime of set cardinalities—namely, where all token sets have $\Theta(n)$ elements—and provide Corollary E.9, combining both cases for $k \geq 2$ and $k = 1$.
  - We also claim that our results are nearly *tight*, especially for $k \geq 2$. It is done by constructing a worst-case subclass of 2-HOP problems in which, given a dataset slightly smaller than our upper bound, the learner fails in achieving a perfect ID generalization with high probability (Theorem E.16 and Corollary E.17). FIXED

  Theorem 6.1 in the main text is indeed a simplified combination of Corollaries E.9 and E.17.
- In App. E.3, we provide (almost) all postponed proofs of lemmas used in App. E.2. We first prove that the *connectivity* of all b-evidence graphs (Def. E.5), which are the intermediate-state-specific induced subgraphs of the substitution graph, is a sufficient condition for the perfect in-domain generalization we want to show: see Lemma E.10.
  - App. E.3.1 is dedicated to explaining the renowned *Poissonization* technique, rigorously proving one of our main lemmas, Lemma E.12. Under a usual fixed-sized dataset sampling scheme, the edge connections of the substitution graph are not guaranteed to be independent. Nevertheless, the Poissonization technique enables us to disentangle these dependencies, thereby simplifying our analysis.
  - App. E.3.2 reviews the binomial random k-intersection graphs, a topic in the random graph theory literature. We bring some results on the threshold functions for their connectivity (without bringing their involved proofs). It is crucial because we show that the b-evidence graphs can be regarded as binomial random k-intersection graphs under the Poissonization.

E.1 PRELIMINARIES: DEFINITIONS, ASSUMPTIONS, AND FACTS

Ground-truth mapping. Consider a ground-truth mapping $f : \mathbb{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \to Y$ defined as $f(x_1, x_2, x_3) = f_2(f_1(x_1, x_2), x_3)$, a *two-hop* composition of primitive functions $f_1 : \mathcal{X}_1 \times \mathcal{X}_2 \to B$ and $f_2 : B \times \mathcal{X}_3 \to Y$. Here, the set $B := f_1(\mathcal{X}_1 \times \mathcal{X}_2)$ is a collection of *intermediate states*, implicitly assuming surjectivity of f_1 without loss of generality.

Train dataset. We take a train dataset D by independently sampling input sequences N times from the uniform distribution over $\mathbb{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$. For the sake of simplicity, we allow duplicates/replacements of train samples in D, thereby considering D as a multiset.

We find it helpful in our analysis to define a random vector $(Z_{\boldsymbol{x}})_{\boldsymbol{x} \in \mathbb{X}}$ that has a one-to-one correspondence with a dataset constructed by sampling with replacement as described below.

Proposition E.1. *Given a train dataset D and for each $\boldsymbol{x} \in \mathbb{X}$, define $Z_{\boldsymbol{x}}$ as the sampled count of $\boldsymbol{x}$ (i.e., the number of identical duplicates equal to $\boldsymbol{x}$) in D. Then, if D is generated by with-replacement sampling N times from $\mathbb{X}$, the random vector $(Z_{\boldsymbol{x}})_{\boldsymbol{x} \in \mathbb{X}}$ follows a multinomial distribution with parameters $n = N$ and $p_{\boldsymbol{x}} = \frac{1}{|\mathbb{X}|}$ ($\forall \boldsymbol{x} \in \mathbb{X}$). That is, for all collections of integers $z_{\boldsymbol{x}} \geq 0$ ($\forall \boldsymbol{x} \in \mathbb{X}$) such that $\sum_{\boldsymbol{x} \in \mathbb{X}} z_{\boldsymbol{x}} = N$,*

$$\mathbb{P}(Z_{\boldsymbol{x}} = z_{\boldsymbol{x}} \ \forall \boldsymbol{x} \in \mathbb{X}) = \frac{N!}{|\mathbb{X}|^N \cdot \prod_{\boldsymbol{x} \in \mathbb{X}} (z_{\boldsymbol{x}}!)}.$$

The one-to-one correspondence between D and $(Z_{\boldsymbol{x}})_{\boldsymbol{x} \in \mathbb{X}}$ is straightforward: their realizations uniquely determine each other (up to the order of sampling).

A learner making predictions using functional k-equivalence. Let us consider an artificial learner that is perfectly trained on D (i.e., always predicts correctly on D). We suppose that the learner confidently predicts the output for the elements in k-coverage of D (defined with functional k-equivalence), while producing a random (or unconfident) prediction outside of it. However, the confident prediction is not necessarily correct in general. Luckily, since we suppose a clear two-hop compositional structure of the task, the confident predictions based on functional equivalences at indices $I = \{1, 2\}$ are guaranteed to be correct. Hence, to study the scope of pattern-matching-based in-domain (ID) generalization as confident and correct predictions on unseen inputs, we restrict the scope of our analysis to the index set $\{1, 2\}$ for our analysis in this appendix.

To formalize how far the learner can generalize (Def. E.4) and how far we hope the learner to generalize (Def. E.2), let us (re)define the key terms for our analysis of the two-hop task, based on the restriction with respect to the index set $I = \{1, 2\}$. For brevity, let us write $x_{12} := (x_1, x_2) \in \mathcal{X}_{12} := \mathcal{X}_1 \times \mathcal{X}_2$.

Definition E.2 (In-domain closure, in terms of $I = \{1, 2\}$)**.** For a train dataset D, its **in-domain closure** $\overline{D}$ is defined as

$$\overline{D} = \{(x_{12}, x_3) \in \mathbb{X} : \exists (x'_{12}, x'_3) \in \mathbb{X} \text{ such that } f_1(x_{12}) = f_1(x'_{12}), (x'_{12}, x_3) \in D, \text{and } (x_{12}, x'_3) \in D\}.$$

Also, its element $(x_{12}, x_3) \in \overline{D}$ is said to be an **in-domain data**.

According to the definition above, the in-domain closure $\overline{D}$ contains all input sequences (x_{12}, x_3) such that (i) its subsequences $x_{12} (= \boldsymbol{x}_I)$ and $x_3 (= \boldsymbol{x}_{I^c})$ has already been observed in the train dataset D, and (ii) its output can be inferred using functional equivalence at indices $I = \{1, 2\}$, namely, $f(x_{12}, x_3) = f(x'_{12}, x_3)$ for a training data $(x'_{12}, x_3) \in D$. Hence, the set $\overline{D}$ is the largest set of input sequences whose output can be correctly predicted using the exact task structure (e.g., the functional equivalence at $\{1, 2\}$). We also remark that it is obvious to see that $D \subset \overline{D}$. We wish the learner would acquire such prediction capabilities from a training dataset. Nevertheless, the task structure is not directly accessible to our learner since it can only identify the necessary functional relationships through functional k-equivalences (Def. 3.1).

On the other hand, the k-coverage quantifies how far the assumed learner can actually make correct predictions using the observed functional k-equivalences from the training dataset D. The definition (Def. E.4) resembles that of the connectivity of a graph with a vertex set $\mathcal{X}_{12}$.

Recall that in our main text, we defined a *substitution graph* with the entire input space $\mathbb{X}$ as its vertex set since we aim to check all functional equivalences across all index sets. However, since we are particularly interested in the functional equivalences at $I = \{1, 2\}$, we can simplify its definition as below.

Definition E.3 (Substitution graph, in terms of $I = \{1, 2\}$)**.** For a train dataset D and a positive integer k, we define a **substitution graph** (in terms of $I = \{1, 2\}$) as $\mathcal{G}_\bullet^{(D,k)} = (V_\bullet, E_\bullet^{(D,k)})$. Here, the vertex set is $V_\bullet := \mathcal{X}_{12}$ and the edge set is

$$E_\bullet^{(D,k)} := \{\{x_{12}, x'_{12}\} \subset V_\bullet : x_{12} \neq x'_{12} \text{ and } |S_f(x_{12}, x'_{12} \mid D)| \geq k\},$$

where the set $S_f(x_{12}, x'_{12} \mid D)$ is defined as

$$S_f(x_{12}, x'_{12} \mid D) := \{\bar{x}_3 \in \mathcal{X}_3 : f(x_{12}, \bar{x}_3) = f(x'_{12}, \bar{x}_3), (x_{12}, \bar{x}_3) \in D, (x'_{12}, \bar{x}_3) \in D\}.$$

In other words, in the substitution graph $\mathcal{G}_\bullet^{(D,k)}$, two vertices x_{12} and x'_{12} are adjacent if they have at least k distinct co-occerences in D. With this definition, we define the k-coverage as follows.

Definition E.4 (k-coverage, in terms of $I = \{1, 2\}$)**.** For a train dataset D and a positive integer k, the k**-coverage** of D (in terms of $I = \{1, 2\}$) is a subset of $\mathbb{X}$ defined as

$$\begin{aligned}
&\mathrm{Cover}_k(D) \\
&\overset{\text{def}}{=} \left\{(x_{12}, x_3) \in \mathbb{X} : \exists x'_{12} \in \mathcal{X}_{12} \text{ connected to } x_{12} \text{ with a path in } \mathcal{G}_\bullet^{(D,k)}, \text{ and } (x'_{12}, x_3) \in D\right\} \\
&= \left\{(u_0, x_3) \in \mathbb{X} : \exists \ell \geq 0,\ \exists u_0, u_1, \cdots, u_\ell \in \mathcal{X}_{12} \text{ such that } \{u_{i-1}, u_i\} \in E_\bullet^{(D,k)}\ (\forall i \in [\ell])\ \&\ (u_\ell, x_3) \in D\right\},
\end{aligned}$$

where $[\ell] = \{1, 2, 3, \cdots, \ell\}$. Also, an element $(x_{12}, x_3) \in \mathrm{Cover}_k(D)$ is said to be **covered**.

Again, we can easily verify that $D \subset \mathrm{Cover}_k(D)$ by choosing $\ell = 0$ in the definition above. However, the inclusion relation between the in-domain coverage ($\overline{D}$) and the k-coverage ($\mathrm{Cover}_k(D)$) cannot

be explicitly determined in general. From the next sub-section, we aim to analyze when (or, with how large D) we do have a relation "$\overline{D} \subset \mathrm{Cover}_k(D)$", which we call **perfect coverage** of in-domain data, with high probability (with respect to the randomness of D).

We additionally introduce some useful symbols for our later proofs. We denote by B_D the set of all intermediate states observed in D: i.e.,

$$B_D := \{b \in B : \exists (x_{12}, x_3) \in D \text{ such that } f_1(x_{12}) = b\}\,.$$

Moreover, we introduce the *b-evidence graph* $\mathcal{G}_b^{(D,k)}$, defined as a subgraph of $\mathcal{G}_\bullet^{(D,k)}$ induced by the set $V_b = f_1^{-1}(\{b\}) := \{x_{12} \in \mathcal{X}_{12} : f_1(x_{12}) = b\} \subset V_\bullet$ of vertices sharing the same intermediate state $b \in B$.

Definition E.5 (Evidence graphs)**.** For a train dataset D, a positive integer k, and an intermediate state $b \in B$, we define a b**-evidence graph** (in terms of $I = \{1, 2\}$) as $\mathcal{G}_b^{(D,k)} = (V_b, E_b^{(D,k)})$. Here, the vertex set is $V_b := f_1^{-1}(\{b\})$ and the edge set is

$$E_b^{(D,k)} := \{\{x_{12}, x'_{12}\} \subset V_b : x_{12} \neq x'_{12} \text{ and } |S(x_{12}, x'_{12} \mid D)| \geq k\}\,, \tag{5}$$

where the set $S(x_{12}, x'_{12} \mid D)$ is defined as

$$S(x_{12}, x'_{12} \mid D) := \{\bar{x}_3 \in \mathcal{X}_3 : (x_{12}, \bar{x}_3) \in D, (x'_{12}, \bar{x}_3) \in D\}\,. \tag{6}$$

Lastly, we introduce an assumption that will be assumed only when we claim the tightness of our sample complexity upper bound results.

Assumption E.6. The learner uses the parameter $k \geq k_\star + 1$, where $k_\star \geq 0$ is defined as

$$k_\star = \max_{(b,b') \in B^2} |H(b, b')| \qquad \text{subject to} \quad b \neq b',$$

and $H(b, b') = \{x_3 \in \mathcal{X}_3 : f_2(b, x_3) = f_2(b', x_3)\}$.

According to the assumption above, $f_2(b, x_3) = f_2(b', x_3)$ can hold for more than $k_\star$ elements in $\mathcal{X}_3$ only when $b = b'$. It is quite a natural assumption for general 2-HOP tasks since it naturally eliminates pathological examples that can barely be regarded as 2-HOP tasks (e.g., a constant function f on $\mathbb{X}$).

The assumption above additionally assumes that our learner *pessimistically* recognizes the functional equivalences between $\{1, 2\}$-subsequences, by using $k \geq k_\star + 1$. If $k = k_\star + 1$, the learner is using the precise value of k to recognize functionally equivalent subsequences sharing the same intermediate state; a higher value of k indicates a more pessimistic learner. Because of this pessimism, once the learner recognizes that x_{12} and x'_{12} are functionally k-equivalent (i.e., $\{x_{12}, x'_{12}\} \in E_\bullet^{(D,k)}$), it is guaranteed that x_{12} and x'_{12} share the same intermediate state (i.e., $f_1(x_{12}) = f_1(x'_{12})$). More importantly, it implies that there must be no edges (in $E_\bullet^{(D,k)}$) connecting two graphs $\mathcal{G}_b^{(D,k)}$ and $\mathcal{G}_{b'}^{(D,k)}$ for distinct $b, b' \in B$; otherwise, there must be a pair of subsequences $x_{12} \in V_b$ and $x'_{12} \in V_{b'}$ that are functionally k-equivalent, contradicting the assumption.

Even though Assumption E.6 appears to pertain to the learning rule of the learner, it can also be interpreted as an assumption about the ground-truth mapping, which necessitates $k_\star \leq k - 1$ for a given $k \geq 1$.

E.2 Main Theorem: Sample Complexity Bound for Perfect Coverage With High Probability

The ultimate goal of this appendix is to characterize a sufficient number of training data $|D|$ (i.e., sample complexity upper bound) so that the k-coverage includes all in-domain data, i.e., $\overline{D} \subset \mathrm{Cover}_k(D)$, with high probability. Here, we will state and prove our main theorems, Theorem E.7 (for $k \geq 2$) and Theorem E.8 (for $k = 1$), which are followed by a simplified combination of them (Corollary E.9). Soon after, we will also claim the near-tightness of these theorems by showing that, for $k \geq 2$, the FIXED training datasets slightly smaller than the obtained sample complexity are sufficient to guarantee $\overline{D} \not\subset \mathrm{Cover}_k(D)$ with high probability (Theorem E.16 and Corollary E.17). Although we present the proofs of the theorems here, the complete proofs of most lemmas used in the main proofs are postponed to the later part of this appendix (App. E.3).

Here are the statements of our main theorems for the sample complexity upper bounds, which will be proved soon after.

Theorem E.7 (Sample Complexity Upper Bound, $k \geq 2$)**.** *Under the two-hop task setup described in App. E.1, suppose that $k \geq 2$ is a universal constant. Let $\hat{n} := \min_{b \in B_D} |V_b|$ and $\check{n} := \max_{b \in B_D} |V_b|$, and assume that $|\mathcal{X}_3| = \Omega(\check{n})$. Fix any $\delta > 0$. Then, there exists $N_\delta > 0$ satisfying*

$$N_\delta \leq O\left(\max\left\{\left(\frac{k!\ (\ln \hat{n})}{\hat{n}}\right)^{\frac{1}{2k}} \cdot |\mathcal{X}_1 \times \mathcal{X}_2|\sqrt{|\mathcal{X}_3|},\ \ln\left(\frac{1}{\delta}\right)\right\}\right)$$

such that, for any large enough $\hat{n}$ and any integer $N > N_\delta$, we have

$$\mathbb{P}_N\left(\overline{D} \subset \mathrm{Cover}_k(D)\right) \geq 1 - \delta.$$

Here, $\mathbb{P}_N$ is the probability measure for the train dataset D constructed by uniformly randomly sampling its elements N times, allowing duplicates, independently from $\mathbb{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$.

Theorem E.8 (Sample Complexity Upper Bound, $k = 1$)**.** *Under the two-hop task setup described in App. E.1, suppose that $k = 1$. Let $\hat{n} := \min_{b \in B_D} |V_b|$ and $\check{n} := \max_{b \in B_D} |V_b|$, and assume that $|\mathcal{X}_3| = O(\hat{n})$. Fix any $\delta > 0$. Then, there exists $N_\delta > 0$ satisfying*

$$N_\delta \leq O\left(\max\left\{(\ln \check{n}) \cdot |\mathcal{X}_1 \times \mathcal{X}_2|\,,\ \ln\left(\frac{1}{\delta}\right)\right\}\right)$$

such that, for any large enough $\hat{n}$ and any $N > N_\delta$, we have

$$\mathbb{P}_N\left(\overline{D} \subset \mathrm{Cover}_k(D)\right) \geq 1 - \delta.$$

We use the same definition of $\mathbb{P}_N$ as in Theorem E.7.

In particular, these two theorems can be easily combined (hence the detailed proof is skipped) in the regime of almost balanced cardinalities as $\Theta(n)$, which is described in the following corollary.

Corollary E.9 (Power-Law Sample Complexity Upper Bound, Simple)**.** *Under the two-hop task setup described in App. E.1, Assume that $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$, and V_b ($\forall b \in B$) are all the sets of size $\Theta(n)$. Fix any $k \geq 1$ and any $\delta > 0$ Then, there exists $N_\delta(n) > 0$ satisfying*

$$N_\delta(n) \leq O\left(\max\left\{n^{2.5-\frac{0.5}{k}} \cdot (k! \cdot \ln n)^\gamma,\ \ln\frac{2}{\delta}\right\}\right), \qquad \text{with } \gamma = \begin{cases} 1, & \text{for } k = 1; \\ \dfrac{1}{2k}, & \text{for } k \geq 2, \end{cases}$$

such that, for any large enough n and any $N > N_\delta(n)$, the learner with a uniformly randomly sampled training dataset D (with replacements) achieves a perfect coverage of in-domain data, i.e., $\overline{D} \subset \mathrm{Cover}_k(D)$, with probability at least $1 - \delta$.

Now, we prove our main theorem for $k \geq 2$ (Theorem E.7). After that, we will present the proof of the other case ($k = 1$, Theorem E.8), which follows a similar plot except for a few different steps at the end.

Proof of Theorem E.7. We begin the proof with the following observation: a sufficient condition for a perfect coverage is that every b-evidence graph (Def. E.5) is a connected graph.

Lemma E.10. *If all $\mathcal{G}_b^{(D,k)}$ are connected graphs ($\forall b \in B_D$), then we have $\overline{D} \subset \mathrm{Cover}_k(D)$.*

Refer to App. E.3 for the proof. Thanks to the lemma above, it suffices to show the following with sufficiently large $N = |D|$:

$$\mathbb{P}_N\left(\overline{D} \not\subset \mathrm{Cover}_k(D)\right) \overset{\text{Lemma E.10}}{\leq} \mathbb{P}_N\left(\exists b \in B_D \text{ such that } \mathcal{G}_b^{(D,k)} \text{ is disconnected}\right) \leq \delta.$$

To this end, consider the multinomial random vector $(Z_{\boldsymbol{x}})_{\boldsymbol{x} \in \mathbb{X}}$ corresponding to a randomly sampled training dataset D, as defined in Proposition E.1. Define a property $\mathcal{C}$ as

$$(Z_{\boldsymbol{x}})_{\boldsymbol{x} \in \mathbb{X}} \in \mathcal{C} \iff \mathcal{G}_b^{(D,k)} \text{ is connected } \forall b \in B_D. \tag{7}$$

Then, we want to compute an upper bound of a probability $\mathbb{P}_N\left((Z_{\boldsymbol{x}})_{\boldsymbol{x}\in\mathbb{X}} \notin \mathcal{C}\right)$. Observe that the property $\mathcal{C}$ is a *non-decreasing* property (Def. E.11): if a vector $(Z_{\boldsymbol{x}})_{\boldsymbol{x}\in\mathbb{X}}$ satisfies $\mathcal{C}$, then it still satisfies $\mathcal{C}$ after increasing in its entries by non-negative amounts. In other words, the negation of $\mathcal{C}$ ("not satisfying $\mathcal{C}$") is a *non-increasing* property. This is because adding a data point to the training dataset D does not remove any edge from all evidence graphs, thereby preserving the connectivity. In general, we define the *monotone* property of vectors as below.

Definition E.11 (Monotone property)**.** We refer to a property $\mathcal{A}$ of m-dimensional vectors as a **non-decreasing property** when, for any vector $(\Delta_1, \cdots, \Delta_m)$ with non-negative entries $\Delta_i \geq 0$ $(\forall i = 1, \cdots, m)$,

$$(z_1, \cdots, z_m) \in \mathcal{A} \implies (z_1 + \Delta_1, \cdots, z_m + \Delta_m) \in \mathcal{A}. \qquad \text{(Non-Decreasing Property)}$$

On the other hand, we define **non-increasing property** $\mathcal{A}$ as what satisfies

$$(z_1, \cdots, z_m) \in \mathcal{A} \implies (z_1 - \Delta_1, \cdots, z_m - \Delta_m) \in \mathcal{A}. \qquad \text{(Non-Increasing Property)}$$

Taking advantage of the monotonicity, we can apply the Poissonization technique (App. E.3.1) to obtain an upper bound on the probability $\mathbb{P}_N\left((Z_{\boldsymbol{x}})_{\boldsymbol{x}\in\mathbb{X}} \notin \mathcal{C}\right)$. The Poissonization technique we use is summarized as the lemma below, which will be proved in App. E.3.1:

Lemma E.12 (De-Poissonizaiton Lemma for Monotone Multinomial Events)**.** *Fix any $n \geq 1$. Define*

$$P_n := \mathbb{P}_{(Z_1,\cdots,Z_m)\sim\text{Multinomial}(n;\, p_1,\cdots,p_m)}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right).$$

Let $\mathcal{A}$ be a non-decreasing property (Def. E.11) of m-dimensional vector. Then, we have an upper bound and a lower bound for P_n as follows:

$$P_n \leq \mathbb{P}_{\text{Po}(n+\varepsilon)}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right) + \exp\left(-\tfrac{(3-c)\varepsilon^2}{6n}\right), \qquad (\forall c \in (0,3),\ \forall \varepsilon \in (0, cn))$$

$$P_n \geq \mathbb{P}_{\text{Po}(n-\varepsilon)}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right) - \exp\left(-\tfrac{\varepsilon^2}{2n}\right). \qquad (\forall \varepsilon \in (0, n))$$

If $\mathcal{A}$ is a non-increasing property (Def. E.11), then we have similar upper and lower bounds for P_n:

$$P_n \leq \mathbb{P}_{\text{Po}(n-\varepsilon)}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right) + \exp\left(-\tfrac{\varepsilon^2}{2n}\right), \qquad (\forall \varepsilon \in (0, n))$$

$$P_n \geq \mathbb{P}_{\text{Po}(n+\varepsilon)}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right) - \exp\left(-\tfrac{(3-c)\varepsilon^2}{6n}\right). \qquad (\forall c \in (0,3),\ \forall \varepsilon \in (0, cn))$$

Here, we denote by $\mathbb{P}_{\text{Po}(\lambda)}$ the probability measure under $Z_i \overset{\text{indep.}}{\sim} \text{Poisson}(\lambda p_i)$ $(\forall i = 1, \cdots, m)$.

In particular, considering the disconnectedness $(\exists b \in B_D)$ as a non-increasing property, we have

$$\begin{aligned}
&\mathbb{P}_N\left(\exists b \in B_D \text{ such that } \mathcal{G}_b^{(D,k)} \text{ is disconnected}\right) \\
&\leq \mathbb{P}_{\text{Po}(N-\varepsilon)}\left(\exists b \in B_D \text{ such that } \mathcal{G}_b^{(D,k)} \text{ is disconnected}\right) + \exp\left(-\frac{\varepsilon^2}{2N}\right) \\
&\leq \sum_{b\in B_D} \mathbb{P}_{\text{Po}(N-\varepsilon)}\left(\mathcal{G}_b^{(D,k)} \text{ is disconnected}\right) + \exp\left(-\frac{\varepsilon^2}{2N}\right) \qquad (\because \text{union bound})
\end{aligned}$$

for any positive number $\varepsilon < N$, where $\mathbb{P}_{\text{Po}(N-\varepsilon)}$ is the probability measure under i.i.d. Poisson random variables $Z_{\boldsymbol{x}} \sim \text{Poisson}\left(\frac{N-\varepsilon}{|\mathbb{X}|}\right)$ $(\forall \boldsymbol{x} \in \mathbb{X})$. Let us take $\varepsilon = \sqrt{2N \ln\left(\frac{2}{\delta}\right)}$, which is smaller than N by our choice of N_δ and ensures that

$$\exp\left(-\frac{\varepsilon^2}{2N}\right) = \exp\left(-\ln\left(\frac{2}{\delta}\right)\right) = \frac{\delta}{2}.$$

Now, we claim that each b-evidence graph $\mathcal{G}_b^{(D,k)}$ is a binomial random k-intersection graph (App. E.3.2) for every $b \in B_D$, under the Poissonization governed by $\mathbb{P}_{\text{Po}(N-\varepsilon)}$.

Lemma E.13 (Evidence Graphs are Binomial k-Intersection Graphs under Poissonization)**.** *Let any $b \in B_D$. Consider a vector $(Z_{\boldsymbol{x}})_{\boldsymbol{x}\in\mathbb{X}}$ associated with a dataset D (i.e., an input sequence $\boldsymbol{x}$ is sampled $Z_{\boldsymbol{x}}$ times in D). Let $\mathbb{P}_{\text{Po}(\lambda)}$ be a probability measure for $Z_{\boldsymbol{x}} \overset{\text{i.i.d.}}{\sim} \text{Poisson}\left(\lambda/|\mathbb{X}|\right)$. Then,*

under $\mathbb{P}_{\mathrm{Po}(\lambda)}$, the b-evidence graph $\mathcal{G}_b^{(D,k)}$ (Def. E.5) is an instance of binomial random k-intersection graph $\mathcal{G}^{(k)}(n_b, m, p)$ with parameters

$$n_b = |V_b|\,, \quad m = |\mathcal{X}_3|\,, \quad p = 1 - \exp\left(-\frac{\lambda}{|\mathbb{X}|}\right).$$

Based on such an observation, we can use the following seminal result about the connectivity of binomial random k-intersection graphs with $p = 1 - \exp(-(N-\varepsilon)/|\mathbb{X}|)$. In particular, we will use only the "$k \geq 2$" part (which tightly matches our assumption $m = \Omega(n_b)$ $(\forall b \in B_D)$) of the lemma below:

Lemma E.14 (Zhao et al., 2014, Theorem 2; Zhao et al., 2017, Theorem 1 & Remark 1)**.** *Fix any $k \geq 1$. Suppose that*

$$m = \begin{cases} \Omega\left(\min\left\{n\left(\ln n\right)^5, n^\rho\right\}\right), & \text{if } k = 1, \text{ for any } \rho > 1; \\ \Omega\left(n\right), & \text{if } k \geq 2, \end{cases}$$

and

$$p = \left(\frac{k!\,\left(\ln n + \alpha_n\right)}{n}\right)^{\frac{1}{2k}} \cdot \frac{1}{\sqrt{m}} \tag{8}$$

for any sequence $\{\alpha_n\}$ which attains a limit $\alpha_\infty \in [-\infty, +\infty]$ as $n \to \infty$. Then,

$$\lim_{n\to\infty} \mathbb{P}\left(\mathcal{G}^{(k)}(n,m,p) \text{ is connected}\right) = \lim_{n\to\infty} \mathbb{P}\left(\min \deg \mathcal{G}^{(k)}(n,m,p) \geq 1\right) = \exp\left(-e^{-\alpha_\infty}\right),$$

where we compute $\exp\left(-e^{-(-\infty)}\right) = 0$ and $\exp\left(-e^{-(+\infty)}\right) = 1$.

Now, we aim to find a sufficient condition for N to have

$$\sum_{b\in B_D} \mathbb{P}_{\mathrm{Po}(N-\varepsilon)}\left(\mathcal{G}_b^{(D,k)} \text{ is disconnected}\right) \leq \frac{\delta}{2}. \tag{9}$$

Let us choose the sequence $\alpha_n = \ln n$ in Lemma E.14, which guarantees that

$$\mathbb{P}(\mathcal{G}^{(k)}(n,m,p) \text{ is connected}) \to 1 \quad \text{as } n \to \infty.$$

From the definition of the limit of a sequence, for any choice of $\delta > 0$, let us define $n_0(\delta) > e > 0$ such that

$$\mathbb{P}(\mathcal{G}^{(k)}(n,m,p) \text{ is disconnected}) \leq \delta, \qquad \forall n > n_0(\delta).$$

Then, choosing all $n_b = |V_b|$ to be greater than $n_0\left(\frac{\delta}{2|B_D|}\right)$, we yield the bound (9). Note that a choice of p larger than the threshold in Eq. (8) will never change that $\mathbb{P}(\mathcal{G}^{(k)}(n,m,p)$ is connected$) \to 1$ as $n \to \infty$. Thus, since we choose $\alpha_n = \ln n$, it suffices to have

$$p = 1 - \exp\left(-\frac{N-\varepsilon}{|\mathbb{X}|}\right) \geq \left(\frac{k!\,\left(2\ln n_b\right)}{n_b}\right)^{\frac{1}{2k}} \cdot \frac{1}{\sqrt{|\mathcal{X}_3|}}$$

for each n_b $(\forall b \in B_D)$. Since $1 - e^{-u} \geq (1 - \frac{1}{e})u$ for $0 < u < 1$ and $\frac{\ln u}{u}$ is a decreasing function for large enough $u > e$, it suffices to have

$$\left(1 - \frac{1}{e}\right)\frac{N-\varepsilon}{|\mathbb{X}|} \geq \left(\frac{k!\,\left(2\ln \hat{n}\right)}{\hat{n}}\right)^{\frac{1}{2k}} \cdot \frac{1}{\sqrt{|\mathcal{X}_3|}}$$

for $\hat{n} := \min_{b\in B_D} n_b$. Plugging in $\varepsilon = \sqrt{2N \ln\left(\frac{2}{\delta}\right)}$ and $|\mathbb{X}| = |\mathcal{X}_1 \times \mathcal{X}_2|\,|\mathcal{X}_3|$, we have

$$N - \sqrt{2N \ln\left(\frac{2}{\delta}\right)} \geq \left(\frac{e}{e-1}\right) \cdot \left(\frac{k!\,\left(2\ln \hat{n}\right)}{\hat{n}}\right)^{\frac{1}{2k}} \cdot |\mathcal{X}_1 \times \mathcal{X}_2|\,\sqrt{|\mathcal{X}_3|},$$

which is satisfied by

$$N \geq 4 \cdot \max\left\{ \left(\frac{e}{e-1}\right) \cdot \left(\frac{k!\ (2\ln \hat{n})}{\hat{n}}\right)^{\frac{1}{2k}} \cdot |\mathcal{X}_1 \times \mathcal{X}_2| \sqrt{|\mathcal{X}_3|},\ \ln\left(\frac{2}{\delta}\right) \right\}.$$

In conclusion, for large enough $\hat{n} > \max\left\{ n_0\left(\frac{\delta}{2|B_D|}\right), e \right\}$, provided that the training dataset size N satisfies the inequality above, we finally have

$$\begin{aligned} &\mathbb{P}_N\left(\exists b \in B_D \text{ such that } \mathcal{G}_b^{(D,k)} \text{ is disconnected}\right) \\ &\leq \sum_{b \in B_D} \mathbb{P}_{\mathrm{Po}(N-\varepsilon)}\left(\mathcal{G}_b^{(D,k)} \text{ is disconnected}\right) + \exp\left(-\frac{\varepsilon^2}{2N}\right) \\ &\leq \frac{\delta}{2} + \frac{\delta}{2} = \delta. \end{aligned}$$

This concludes the proof of the theorem. ■

Proof of Theorem E.8. Even for $k = 1$, we follow almost the same proof as that of Theorem E.7, except for the last few steps. Namely, we again observe that every $\mathcal{G}_b^{(D,1)}$ is a binomial random 1-intersection graph $\mathcal{G}^{(1)}(n_b, m, p)$ with parameters $n_b = |V_b|$, $m = |\mathcal{X}_3|$, and $p = 1 - \exp(-(N - \varepsilon)/|\mathbb{X}|)$. To guarantee the union bound in Eq. (9), however, we use a different lemma: Lemma E.15 (instead of Lemma E.14). In particular, we will use only the "$\rho \leq 1$" part (which tightly matches our assumption $m = O(n_b)$ ($b \in B_D$)) of the lemma below:

Lemma E.15 (Singer, 1995, Propositions 3.1–2, Theorem 3.3)**.** *Let $k = 1$. Suppose that $m = n^\rho$ for $\rho > 0$ and*

$$p = \begin{cases} \dfrac{\ln n + \alpha_n}{m} & \text{for } \rho \leq 1; \\ \sqrt{\dfrac{\ln n + \alpha_n}{mn}} & \text{for } \rho > 1, \end{cases}$$

for any sequence $\{\alpha_n\}$ which attains a limit $\alpha_\infty \in \{-\infty, +\infty\}$ as $n \to \infty$. Then,

$$\lim_{n\to\infty} \mathbb{P}(\mathcal{G}^{(1)}(n,m,p) \text{ is connected}) = \lim_{n\to\infty} \mathbb{P}\left(\min \deg \mathcal{G}^{(1)}(n,m,p) \geq 1\right) = \begin{cases} 0, & \text{if } \alpha_\infty = -\infty; \\ 1, & \text{if } \alpha_\infty = +\infty. \end{cases}$$

Again, let us choose the sequence $\alpha_n = \ln n$ in Lemma E.15, which guarantees that $\mathbb{P}(\mathcal{G}^{(1)}(n,m,p) \text{ is connected}) \to 1$ as $n \to \infty$. From the definition of the limit of a sequence, for any choice of $\delta > 0$, let us define $n_0(\delta) > 1$ such that

$$\mathbb{P}(\mathcal{G}^{(1)}(n,m,p) \text{ is disconnected}) \leq \delta, \qquad \forall n > n_0(\delta).$$

Then, choosing all $n_b = |V_b|$ (and thus $\hat{n} = \min_{b \in B_D} n_b$) to be greater than $n_0\left(\frac{\delta}{2|B_D|}\right)$, we yield the same bound (9). Now, by our choice $\alpha_n = \ln n$, it suffices to have

$$p = 1 - \exp\left(-\frac{N-\varepsilon}{|\mathbb{X}|}\right) \geq \frac{2\ln n_b}{|\mathcal{X}_3|}$$

for each n_b ($\forall b \in B_D$). Since $1 - e^{-u} \geq (1 - \frac{1}{e})u$ for $0 < u < 1$ and $\ln u$ is an increasing function for $u > 1$, it suffices to have

$$\left(1 - \frac{1}{e}\right)\frac{N-\varepsilon}{|\mathbb{X}|} \geq \frac{2\ln \check{n}}{|\mathcal{X}_3|}$$

for $\check{n} := \max_{b \in B_D} n_b$. Plugging in $\varepsilon = \sqrt{2N \ln\left(\frac{2}{\delta}\right)}$ and $|\mathbb{X}| = |\mathcal{X}_1 \times \mathcal{X}_2|\,|\mathcal{X}_3|$, we have

$$N - \sqrt{2N \ln\left(\frac{2}{\delta}\right)} \cdot \geq \left(\frac{e}{e-1}\right)(2\ln \check{n}) \cdot |\mathcal{X}_1 \times \mathcal{X}_2|\,,$$

which is satisfied by

$$N \geq 4 \cdot \max\left\{\left(\frac{e}{e-1}\right) \cdot (2\ln \check{n}) \cdot |\mathcal{X}_1 \times \mathcal{X}_2|\,,\ \ln\left(\frac{2}{\delta}\right)\right\}.$$

In conclusion, for large enough $\check{n} \geq \hat{n} > \max\left\{n_0\left(\frac{\delta}{2|B_D|}\right), 1\right\}$, provided that the training dataset size N satisfies the inequality above, we finally have

$$\begin{aligned}
&\mathbb{P}_N\left(\exists b \in B_D \text{ such that } \mathcal{G}_b^{(D,1)} \text{ is disconnected}\right) \\
&\leq \sum_{b \in B_D} \mathbb{P}_{\mathrm{Po}(N-\varepsilon)}\left(\mathcal{G}_b^{(D,1)} \text{ is disconnected}\right) + \exp\left(-\frac{\varepsilon^2}{2N}\right) \\
&\leq \frac{\delta}{2} + \frac{\delta}{2} = \delta.
\end{aligned}$$

This concludes the proof. ■

Subsequently, we argue that the sample complexity upper bound obtained in Theorem E.7 (for $k \geq 2$) is *nearly tight*.[7] To this end, we show that the learner (we assumed in App. E.1) cannot avoid an *incomplete* coverage (i.e., $\overline{D} \not\subset \mathrm{Cover}_k(D)$) with high probability, with a dataset slightly smaller than our upper bound for certain instances of the 2-HOP task. In particular, it is enough to consider a subclass of the 2-HOP task satisfying a mild condition (Assumption E.6) that any two distinct intermediate states in B never share the same output for more than $k_\star (< k)$ tokens in $\mathcal{X}_3$. The formal statement of the tightness result is shown below: FIXED

Theorem E.16 (Near-Tightness of the Sample Complexity Upper Bound for $k \geq 2$)**.** *Under the two-hop task setup described in App. E.1, suppose that $k \geq 2$ is a universal constant. Let $\hat{n} := \min_{b \in B} |V_b|$ and $\check{n} := \max_{b \in B} |V_b|$. Assume that $|\mathcal{X}_3| = \Omega(\hat{n})$ and $|\mathcal{X}_1 \times \mathcal{X}_2| \geq 6$. Let us further assume that the ground-truth mapping f satisfies Assumption E.6. Fix any $\delta > 0$. Then, there exists $\widetilde{N}_\delta > 0$ satisfying*

$$\widetilde{N}_\delta \geq \left(\frac{(k-1)!\,(\ln \hat{n})}{\hat{n}}\right)^{\frac{1}{2k}} \cdot |\mathcal{X}_1 \times \mathcal{X}_2| \sqrt{|\mathcal{X}_3|}$$

such that, for any large enough $\hat{n}$ and any integer N satisfying the range

$$\max\left\{|\mathcal{X}_1 \times \mathcal{X}_2| \ln \frac{4\,|\mathcal{X}_1 \times \mathcal{X}_2|}{\delta},\ \frac{|\mathbb{X}|}{\hat{n}} \ln \frac{4\,|B \times \mathcal{X}_3|}{\delta}\right\} \leq N < \widetilde{N}_\delta$$

we have

$$\mathbb{P}_N\left(\overline{D} \subset \mathrm{Cover}_k(D)\right) \leq \delta.$$

We use the same definition of $\mathbb{P}_N$ as in Theorem E.7.

A caveat of Theorem E.16 is that it makes sense only when the range of N is nonempty. Luckily, it is easy to verify its validity in the regime of balanced cardinalities, similar to what we assumed in Corollary E.9. As a result, we establish the *tightness* (for $k \geq 2$, up to a constant factor) of our sample complexity upper bound in Corollary E.9, among the data sizes $N \gtrsim n^2 \ln n$ (Corollary E.17). Note that, as shown in the middle of the proof of Theorem E.16, $O(n^2 \ln n)$ is indeed the data size which is sufficient to guarantee that the learner observes all possible pairs in $\mathcal{X}_1 \times \mathcal{X}_2$ and $B \times \mathcal{X}_3$, ensuring that all possible input sequences in $\mathbb{X}$ are in-domain, with high probability.

[7] We often say a complexity upper bound is tight if we can find a worst-case example whose complexity lower bound is almost identical to the upper bound.

Corollary E.17 (Tightness of Sample Complexity Upper Bound for $k \geq 2$, Simple)**.** *Under the two-hop task setup described in App. E.1, suppose that $|\mathcal{X}_1| = |\mathcal{X}_2| = |\mathcal{X}_3| = |B| = |V_b| = n$ $(\forall b \in B)$. Also, assume that the ground-truth mapping f satisfies Assumption E.6. Fix any $k \geq 2$ and any $\delta > 0$. Then, there exists $\widetilde{N}_\delta(n) > 0$ satisfying*

$$\widetilde{N}_\delta(n) \geq n^{2.5-\frac{0.5}{k}} \cdot ((k-1)! \cdot \ln n)^{\frac{1}{2k}}$$

such that, for any large enough n (e.g., such that $kn > 256e(\ln n)^3$ and $\sqrt{n}(\ln n)^{\frac{1}{k}} > 4(\ln \frac{4}{\delta})^2$) and any integer N satisfying that

$$n^2 \left(2 \ln n + \ln \frac{4}{\delta}\right) \leq N < \widetilde{N}_\delta(n),$$

*the learner with a uniformly randomly sampled training dataset D (with replacements) **cannot** achieve a perfect coverage of in-domain data, i.e., $\overline{D} \not\subset \mathrm{Cover}_k(D)$, with probability at least $1 - \delta$.*

From now on, we prove Theorem E.16, the tightness result of the sample complexity upper bound obtained in Theorem E.7.

Proof of Theorem E.16. We aim to prove that, even when the dataset D is large enough to ensure that $\overline{D} = \mathbb{X}$, the incomplete coverage may happen with high probability for a certain range of data size N. To this end, let us first define two sets D_{12} and D_{B3} as follows:

$$D_{12} := \{x_{12} \in \mathcal{X}_{12} : \exists x_3 \in \mathcal{X}_3 \text{ such that } (x_{12}, x_3) \in D\};$$
$$D_{B3} := \{(b, x_3) \in B \times \mathcal{X}_3 : \exists x_{12} \in V_b \text{ such that } (x_{12}, x_3) \in D\}.$$

Also, we denote by $\min \deg \mathcal{G} := \min_{v \in V} \deg_{\mathcal{G}}(v)$ the minimum degree among all vertices of a graph $\mathcal{G} = (V, E)$. With these definitions in place, we now apply the following lemma, which describes a necessary condition for perfect coverage.

Lemma E.18. *Assume that $D_{12} = \mathcal{X}_{12}$ and $D_{B3} = B \times \mathcal{X}_3$ hold. Then, these imply that $\overline{D} = \mathbb{X}$. Furthermore, suppose that Assumption E.6 holds for a given $1 \leq k \leq |\mathcal{X}_3|$. Then, $\overline{D} \subset \mathrm{Cover}_k(D)$ implies that $\min \deg \mathcal{G}_b^{(D,k)} \geq 1$ for all $b \in B$.*

Refer to App. E.3 for its proof. Now, it suffices to show the inequality below:

$$\begin{aligned}
&\mathbb{P}_N\left(\overline{D} \subset \mathrm{Cover}_k(D)\right) \\
&\leq \mathbb{P}_N\left(\overline{D} \subset \mathrm{Cover}_k(D) \text{ or } D_{12} \neq \mathcal{X}_{12} \text{ or } D_{B3} \neq B \times \mathcal{X}_3\right) \\
&\leq \mathbb{P}_N\left(\min \deg \mathcal{G}_b^{(D,k)} \geq 1 \ (\forall b \in B) \text{ or } D_{12} \neq \mathcal{X}_{12} \text{ or } D_{B3} \neq B \times \mathcal{X}_3\right) \qquad (\because \text{Lemma E.18}) \\
&\leq \mathbb{P}_N\left(\min \deg \mathcal{G}_b^{(D,k)} \geq 1 \ (\forall b \in B)\right) + \mathbb{P}_N\left(D_{12} \neq \mathcal{X}_{12}\right) + \mathbb{P}_N\left(D_{B3} \neq B \times \mathcal{X}_3\right) \leq \delta. \qquad (10)
\end{aligned}$$

To this end, consider the multinomial random vector $(Z_{\boldsymbol{x}})_{\boldsymbol{x} \in \mathbb{X}} \sim \mathrm{Multinomial}(N; (p_{\boldsymbol{x}})_{\boldsymbol{x} \in \mathbb{X}})$ corresponding to a randomly sampled training dataset D, as defined in Proposition E.1. Define a property $\mathcal{M}_1$ as

$$(Z_{\boldsymbol{x}})_{\boldsymbol{x} \in \mathbb{X}} \in \mathcal{M}_1 \iff \min \deg \mathcal{G}_b^{(D,k)} \geq 1 \ \ (\forall b \in B). \qquad (11)$$

Also, observe that

$$(T_{x_{12}})_{x_{12} \in \mathcal{X}_{12}} \sim \mathrm{Multinomial}(N; (q_{x_{12}})_{x_{12} \in \mathcal{X}_{12}}), \qquad \text{if } \begin{cases} T_{x_{12}} = \sum_{x_3 \in X_3} Z_{(x_{12}, x_3)}, \\ q_{x_{12}} = \sum_{x_3 \in X_3} p_{(x_{12}, x_3)}, \end{cases}$$

$$(U_w)_{w \in B \times \mathcal{X}_3} \sim \mathrm{Multinomial}(N; (r_w)_{w \in B \times \mathcal{X}_{12}}), \qquad \text{if } \begin{cases} U_{(b,x_3)} = \sum_{x_{12} \in V_b} Z_{(x_{12}, x_3)}, \\ r_{(b,x_3)} = \sum_{x_{12} \in V_b} p_{(x_{12}, x_3)}, \end{cases}$$

where we can actually put $p_{(x_{12},x_3)} = \frac{1}{|\mathbb{X}|}$, $q_{x_{12}} = \frac{|\mathcal{X}_3|}{|\mathbb{X}|} = \frac{1}{|\mathcal{X}_{12}|}$, and $r_{(b,x_3)} = \frac{|V_b|}{|\mathbb{X}|}$ above. Using this notation, we know that

$$D_{12} = \mathcal{X}_{12} \iff T_{x_{12}} \geq 1 \ \ (\forall x_{12} \in \mathcal{X}_{12});$$
$$D_{B3} = B \times \mathcal{X}_3 \iff U_w \geq 1 \ \ (\forall w \in B \times \mathcal{X}_3).$$

Hence, to show Eq. (10), it suffices to prove the following three inequalities:

$$\mathbb{P}_{(Z_{\boldsymbol{x}})_{\boldsymbol{x}\in\mathbb{X}} \sim \mathrm{Multinomial}(N;(p_{\boldsymbol{x}})_{\boldsymbol{x}\in\mathbb{X}})} \left((Z_{\boldsymbol{x}})_{\boldsymbol{x}\in\mathbb{X}} \in \mathcal{M}_1\right) \leq \frac{\delta}{2}; \tag{12}$$

$$\mathbb{P}_{(T_{x_{12}})_{x_{12}\in\mathcal{X}_{12}} \sim \mathrm{Multinomial}(N;(q_{x_{12}})_{x_{12}\in\mathcal{X}_{12}})} \left(\exists x_{12} \in \mathcal{X}_{12} \text{ such that } T_{x_{12}} = 0\right) \leq \frac{\delta}{4}; \tag{13}$$

$$\mathbb{P}_{(U_w)_{w\in B\times\mathcal{X}_3} \sim \mathrm{Multinomial}(N;(r_w)_{w\in B\times\mathcal{X}_{12}})} \left(\exists w \in B \times \mathcal{X}_3 \text{ such that } U_w = 0\right) \leq \frac{\delta}{4}. \tag{14}$$

A size N of training dataset D that is sufficient to ensure Eqs. (13) and (14) can be characterized with the following lemma.

Lemma E.19 (A Tail Bound for Coupon Collector's Problem)**.** *Consider a multinomial random vector* $(Z_1, \cdots, Z_m) \sim \mathrm{Multinomial}(n; p_1, \cdots, p_m)$*. Then,*

$$\mathbb{P}(\exists i \text{ such that } Z_i = 0) \leq \sum_{i=1}^{m} (1-p_i)^n.$$

In particular, if $\hat{p} = \min_{1\leq i\leq m} p_i$*, then for any* $\delta > 0$*,*

$$n \geq \frac{1}{\hat{p}} \ln \frac{m}{\delta} \implies \mathbb{P}(\exists i \text{ such that } Z_i = 0) \leq \delta.$$

See App. E.3 for the proof. Applying the lemma and $\hat{n} = \min_{b\in B} |V_b|$, we can figure out that the data size

$$N \geq \max\left\{ |\mathcal{X}_{12}| \ln\left(\frac{4\,|\mathcal{X}_{12}|}{\delta}\right), \ \frac{|\mathbb{X}|}{\hat{n}} \ln\left(\frac{4\,|B\times\mathcal{X}_3|}{\delta}\right)\right\} \tag{15}$$

is enough to ensure both Eqs. (13) and (14). The remaining task now is to determine a condition for N that guarantees Eq. (12).

Observe that $\mathcal{M}_1$ (defined as Eq. (11)) is a *non-decreasing* property (Def. E.11) because of a similar reason for the monotonicity of the property $\mathcal{C}$ (defined as Eq. (7); see the proof of Theorem E.7). Thanks to the monotonicity of $\mathcal{M}_1$, we can apply the Poissonization technique (Lemma E.12) once again. That is, taking any fixed $c \in (0, 2)$, we have

$$\begin{aligned}
&\mathbb{P}_N\left(\min \deg \mathcal{G}_b^{(D,k)} \geq 1 \ \ (\forall b \in B)\right) \\
&\leq \mathbb{P}_{\mathrm{Po}(N+\varepsilon)}\left(\min \deg \mathcal{G}_b^{(D,k)} \geq 1 \ \ (\forall b \in B)\right) + \exp\left(-\frac{(3-c)\varepsilon^2}{6N}\right) \\
&\leq \mathbb{P}_{\mathrm{Po}(N+\varepsilon)}\left(\min \deg \mathcal{G}_b^{(D,k)} \geq 1\right) + \exp\left(-\frac{(3-c)\varepsilon^2}{6N}\right) \qquad (\forall b \in B),
\end{aligned}$$

for any $\varepsilon \in (0, cN)$, where $\mathbb{P}_{\mathrm{Po}(N+\varepsilon)}$ is the probability measure under i.i.d. Poisson random variables $Z_{\boldsymbol{x}} \sim \mathrm{Poisson}\left(\frac{N+\varepsilon}{|\mathbb{X}|}\right)$ $(\forall \boldsymbol{x} \in \mathbb{X})$. Let us take $\varepsilon = \sqrt{\frac{6N}{3-c} \ln\left(\frac{4}{\delta}\right)}$, which is smaller than N by Eq. (15) (since $|\mathcal{X}_{12}| \geq 6 > \frac{6}{3-c}$) and ensures that

$$\exp\left(-\frac{(3-c)\varepsilon^2}{6N}\right) = \exp\left(-\ln\left(\frac{4}{\delta}\right)\right) = \frac{\delta}{4}.$$

Moreover, we again use the fact that $\mathcal{G}_b^{(D,k)}$ is an instance of binomial random k-intersection graph $\mathcal{G}^{(k)}(n_b, m, p)$ with parameters $n_b = |V_b|$, $m = |\mathcal{X}_3|$, and $p = 1 - \exp\left(-(N+\varepsilon)/|\mathbb{X}|\right)$, under the Poissonization governed by $\mathbb{P}_{\mathrm{Po}(N+\varepsilon)}$. Since we assume $k \geq 2$, we can use the "$k \geq 2$" part of Lemma E.14. In particular, we are to apply Lemma E.14 for an evidence graph $\mathcal{G}_b^{(D,k)}$ with

$b \in \arg\min_{b' \in B} |V_b|$ (thus, $n_b = \hat{n}$), which is possible since we assume $m = |\mathcal{X}_3| = \Omega(\hat{n})$. We aim to find a sufficient condition for N to have

$$\mathbb{P}_{\mathrm{Po}(N+\varepsilon)}\left(\min \deg \mathcal{G}_b^{(D,k)} \geq 1\right) \leq \frac{\delta}{4}. \tag{16}$$

Let us choose the sequence $\alpha_n = -(1-\frac{1}{k})\ln n$ in Lemma E.14, which guarantees that

$$\mathbb{P}(\min \deg \mathcal{G}^{(k)}(n,m,p) \geq 1) \to 0 \quad \text{as } n \to \infty.$$

From the definition of the limit of a sequence, for any choice of $\delta > 0$, let us consider $n_0(\delta) > 0$ satisfying that

$$\mathbb{P}\left(\min \deg \mathcal{G}^{(k)}(n,m,p) \geq 1\right) \leq \delta, \qquad n > n_0(\delta).$$

Then, choosing $\hat{n}$ greater than $n_0\left(\frac{\delta}{4}\right)$, we yield the bound (16). Observe that a choice of p smaller than the threshold in Eq. (8) will never change the limit $\mathbb{P}(\min \deg \mathcal{G}^{(k)}(n,m,p) \geq 1) \to 0$. Thus, by our choice $\alpha_n = -(1-\frac{1}{k})\ln n$, it is enough to have

$$p = 1 - \exp\left(-\frac{N+\varepsilon}{|\mathbb{X}|}\right) \leq \left(\frac{(k-1)!\,(\ln \hat{n})}{\hat{n}}\right)^{\frac{1}{2k}} \cdot \frac{1}{\sqrt{|\mathcal{X}_3|}}.$$

Applying $1 - e^{-u} \leq u$ for $u \in \mathbb{R}$, it suffices to have

$$\frac{N+\varepsilon}{|\mathbb{X}|} \leq \left(\frac{(k-1)!\,(\ln \hat{n})}{\hat{n}}\right)^{\frac{1}{2k}} \cdot \frac{1}{\sqrt{|\mathcal{X}_3|}}.$$

Plugging $\varepsilon < cN$ and $|\mathbb{X}| = |\mathcal{X}_1 \times \mathcal{X}_2|\,|\mathcal{X}_3|$ in, we eventually have a sufficient condition

$$N \leq \frac{1}{1+c} \cdot \left(\frac{(k-1)!\,(\ln \hat{n})}{\hat{n}}\right)^{\frac{1}{2k}} \cdot |\mathcal{X}_1 \times \mathcal{X}_2| \sqrt{|\mathcal{X}_3|}.$$

To summarize, for large enough $\hat{n} > n_0\left(\frac{\delta}{4}\right)$, provided that the data size N satisfies

$$\max\left\{|\mathcal{X}_{12}| \ln \frac{4\,|\mathcal{X}_{12}|}{\delta}, \frac{|\mathbb{X}|}{\hat{n}} \ln \frac{4\,|B \times \mathcal{X}_3|}{\delta}\right\} \leq N \leq \frac{1}{1+c} \cdot \left(\frac{(k-1)!\,(\ln \hat{n})}{\hat{n}}\right)^{\frac{1}{2k}} \cdot |\mathcal{X}_1 \times \mathcal{X}_2| \sqrt{|\mathcal{X}_3|}$$

and we take $\varepsilon = \sqrt{\frac{6N}{3-c} \ln\left(\frac{4}{\delta}\right)}$, we finally have

$$\begin{aligned}
&\mathbb{P}_N\left(\overline{D} \subset \mathrm{Cover}_k(D)\right) \\
&\leq \mathbb{P}_{\mathrm{Po}(N+\varepsilon)}\left(\min \deg \mathcal{G}_b^{(D,k)} \geq 1\right) + \exp\left(-\frac{(3-c)\varepsilon^2}{6N}\right) + \mathbb{P}_N\left(D_{12} \neq \mathcal{X}_{12}\right) + \mathbb{P}_N\left(D_{B3} \neq B \times \mathcal{X}_3\right) \\
&\leq \frac{\delta}{4} + \frac{\delta}{4} + \frac{\delta}{4} + \frac{\delta}{4} = \delta.
\end{aligned}$$

Since the choice of $c \in (0,2)$ is arbitrary, we obtain the same result ($\mathbb{P}_N(\cdots) \leq \delta$) by letting $c \searrow 0$ and choosing N which satisfies that

$$\max\left\{|\mathcal{X}_{12}| \ln \frac{4\,|\mathcal{X}_{12}|}{\delta}, \frac{|\mathbb{X}|}{\hat{n}} \ln \frac{4\,|B \times \mathcal{X}_3|}{\delta}\right\} \leq N < \left(\frac{(k-1)!\,(\ln \hat{n})}{\hat{n}}\right)^{\frac{1}{2k}} \cdot |\mathcal{X}_1 \times \mathcal{X}_2| \sqrt{|\mathcal{X}_3|}.$$

■

Before moving on to the postponed proofs of lemmas, we lastly remark that the same proof (of Theorem E.16) can hardly apply to the case of $k = 1$ in general. This is because, for the sake of simplicity in applying the necessary condition for a perfect coverage (Lemma E.18), we first characterize a minimal data size to guarantee $\overline{D} = \mathbb{X}$ with high probability as Eq. (15), using the tail bound of the coupon collector's problem (Lemma E.19). Unfortunately, it already exceeds the sample complexity upper bound to ensure $\overline{D} \subset \mathrm{Cover}_1(D)$ (obtained as Theorem E.8) for $k = 1$ with high probability, especially in the regime of balanced cardinalities assumed in Corollary E.17.

E.3 Backgrounds, Useful Facts, and Lemmas

Now, we delve into the detailed backgrounds to provide a comprehensive understanding of the main theorems' proof, along with the deferred proofs of the lemmas used earlier.

We begin with the proof of a sufficient condition for a perfect coverage (Lemma E.10). For readability, we restate the lemma here.

Lemma E.20. *If all $\mathcal{G}_b^{(D,k)}$ are connected graphs ($\forall b \in B_D$), then we have $\overline{D} \subset \mathrm{Cover}_k(D)$.*

Proof of Lemma E.10. Take any $(x_{12}, x_3) \in \overline{D}$. By definition (Def. E.2), there exists $x'_{12} \in \mathcal{X}_{12}$ such that $f_1(x_{12}) = f_1(x'_{12})$ and $(x'_{12}, x_3) \in D$. Let $b = f_1(x'_{12}) \in B_D$. By assumption, there exists a path in $\mathcal{G}_b^{(D,k)}$ connecting x_{12} and x'_{12}, which we write as $u_0(= x_{12}), u_1, \ldots, u_\ell(= x'_{12})$ for some integer $\ell \geq 1$. Then, for each $i \in [\ell]$, we have a set $S(u_{i-1}, u_i \mid D)$ (defined in Def. E.5) of size at least k. For each $\bar{x}_3 \in S(u_{i-1}, u_i \mid D)$, both $(u_{i-1}, \bar{x}_3)$ and $(u_i, \bar{x}_3)$ are both in D. Also,

$$f(u_{i-1}, \bar{x}_3) = f_2(b, \bar{x}_3) = f(u_i, \bar{x}_3),$$

meaning that the set $S_f(u_{i-1}, u_i \mid D)$ (defined in Def. E.3) is the same as $S(u_{i-1}, u_i \mid D)$, a set of size at least k, for each $i \in [\ell]$. Hence, (x_{12}, x_3) satisfies the definition of k-coverage with a path $u_0, \ldots, u_\ell$, i.e., $(x_{12}, x_3) \in \mathrm{Cover}_k(D)$. □

Next, we show a necessary condition for a perfect coverage (Lemma E.18). Before proving the lemma, we recall that we have defined two sets: FIXED

$$D_{12} := \{x_{12} \in \mathcal{X}_{12} : \exists x_3 \in \mathcal{X}_3 \text{ such that } (x_{12}, x_3) \in D\};$$
$$D_{B3} := \{(b, x_3) \in B \times \mathcal{X}_3 : \exists x_{12} \in V_b \text{ such that } (x_{12}, x_3) \in D\}.$$

Also, for a graph $\mathcal{G} = (V, E)$, we define $\min\deg \mathcal{G} := \min_{v \in V} \deg_{\mathcal{G}}(v)$ as the minimum degree among all vertices of $\mathcal{G}$.

Lemma E.21. *Assume that $D_{12} = \mathcal{X}_{12}$ and $D_{B3} = B \times \mathcal{X}_3$ hold. Then, these imply that $\overline{D} = \mathbb{X}$. Furthermore, suppose that Assumption E.6 holds for a given $1 \leq k \leq |\mathcal{X}_3|$. Then, $\overline{D} \subset \mathrm{Cover}_k(D)$ implies that $\min\deg \mathcal{G}_b^{(D,k)} \geq 1$ for all $b \in B$.*

Proof of Lemma E.18. We first show that $D_{12} = \mathcal{X}_{12}$ and $D_{B3} = B \times \mathcal{X}_3$ imply that $\overline{D} = \mathbb{X}$. Since we already have $\overline{D} \subset \mathbb{X}$ by definition, it suffices to prove $\mathbb{X} \subset \overline{D}$. Let any $x_{12} \in \mathcal{X}_{12}$ and $x_3 \in \mathcal{X}_3$, and take $b = f_1(x_{12}) \in B$. Let us define:

$$W_{x_{12}} = \{x'_3 \in \mathcal{X}_3 : (x_{12}, x'_3) \in D\};$$
$$\widetilde{W}_{x_3}^{(b)} = \{x'_{12} \in V_b : (x'_{12}, x_3) \in D\}.$$

The conditions $D_{12} = \mathcal{X}_{12}$ and $D_{B3} = B \times \mathcal{X}_3$ imply that $W_{x_{12}} \neq \varnothing$ and $\widetilde{W}_{x_3}^{(b)} \neq \varnothing$, respectively. If we take any $x'_3 \in W_{x_{12}}$ and $x'_{12} \in \widetilde{W}_{x_3}^{(b)}$, they hold that $(x_{12}, x'_3) \in D$, $(x'_{12}, x_3) \in D$, and $f_1(x_{12}) = b = f_1(x'_{12})$. This set of conditions is equivalent to $(x_{12}, x_3) \in \overline{D}$; hence, we have just shown that

$$D_{12} = \mathcal{X}_{12} \text{ and } D_{B3} = B \times \mathcal{X}_3 \implies \overline{D} = \mathbb{X}.$$

Now, we prove the contrapositive of the lemma. Suppose that $\min\deg \mathcal{G}_b^{(D,k)} = 0$ for some $b \in B$; we aim to show that $\mathrm{Cover}_k(D) \neq \mathbb{X} = \overline{D}$, or, $\mathbb{X} \setminus \mathrm{Cover}_k(D) \neq \varnothing$.

Take a $b \in B$ such that $\min\deg \mathcal{G}_b^{(D,k)} = 0$, which implies that such a graph $\mathcal{G}_b^{(D,k)}$ has an isolated vertex $x_{12} \in V_b$ (i.e., $\deg_{\mathcal{G}_b^{(D,k)}}(x_{12}) = 0$). Let us fix such an x_{12}. Since we assume Assumption E.6, there must not be any edge connection between $\mathcal{G}_b^{(D,k)}$ and $\mathcal{G}_{b'}^{(D,k)}$ for distinct $b, b' \in B$. Thus, x_{12} must be an isolated vertex in the whole substitution graph $\mathcal{G}_\bullet^{(D,k)}$. The isolation implies that, for any $x_3 \in \mathcal{X}_3$, the input sequence (x_{12}, x_3) cannot be in the k-coverage of D unless it is already in D.

Observe that $W_{x'_{12}} \neq \varnothing$ for any $x'_{12} \in \mathcal{X}_{12}$; otherwise, it inevitably holds that $(x'_{12}, \tilde{x}_3) \neq \overline{D}$ for every $\tilde{x}_3 \in \mathcal{X}_3$, which contradicts to the fact $\overline{D} = \mathbb{X}$ (under the assumption of the lemma). Thus, we can think of the following two cases:

- **Case 1** ($\varnothing \neq W_{x_{12}} \subsetneq \mathcal{X}_3$)**.** Take any $x_3 \notin W_{x_{12}}$, which implies $(x_{12}, x_3) \notin D$. Thus, we have an element $(x_{12}, x_3) \in \mathbb{X} \setminus \mathrm{Cover}_k(D)$ because x_{12} is an isolated vertex.
- **Case 2** ($W_{x_{12}} = \mathcal{X}_3$)**.** Since x_{12} is an isolated vertex, it is not adjacent to any other vertex x'_{12}. Fix any such an $x'_{12} \in \mathcal{X}_{12} \setminus \{x_{12}\}$. By definition of the edges in $E_\bullet^{(D,k)}$, we know that $\left|W_{x_{12}} \cap W_{x'_{12}}\right| < k$. Since $W_{x_{12}} = \mathcal{X}_3$, we also know that $\left|W_{x'_{12}}\right| < k \leq |\mathcal{X}_3|$. This implies that x'_{12} cannot be adjacent to any other vertices in $\mathcal{X}_{12}$, meaning that x'_{12} is an isolated vertex. Moreover, since $W_{x'_{12}} \subsetneq \mathcal{X}_3$ ($\because \left|W_{x'_{12}}\right| < |\mathcal{X}_3|$), we take any $x_3 \notin W_{x'_{12}}$. Then, since $(x'_{12}, x_3) \notin D$ and x'_{12} is isolated, we have an element $(x'_{12}, x_3) \in \mathbb{X} \setminus \mathrm{Cover}_k(D)$.

In both cases above, we obtain the same result that $\mathbb{X} \setminus \mathrm{Cover}_k(D) \neq \varnothing$. It concludes the proof of the lemma. □

We also prove the tail probability bound for the coupon collector's problem (Newman, 1960; Erdős & Rényi, 1961) here. We again restate the lemma here for the sake of readability. FIXED

Lemma E.22 (A Tail Bound for Coupon Collector's Problem)**.** *Consider a multinomial random vector* $(Z_1, \cdots, Z_m) \sim \mathrm{Multinomial}(n; p_1, \cdots, p_m)$*. Then,*

$$\mathbb{P}(\exists i \text{ such that } Z_i = 0) \leq \sum_{i=1}^{m} (1 - p_i)^n.$$

In particular, if $\hat{p} = \min_{1 \leq i \leq m} p_i$*, then for any* $\delta > 0$*,*

$$n \geq \frac{1}{\hat{p}} \ln \frac{m}{\delta} \implies \mathbb{P}(\exists i \text{ such that } Z_i = 0) \leq \delta.$$

Proof of Lemma E.19. Observe that $Z_i \sim \mathrm{Bin}(n, p_i)$ for each $i \in [m]$. It implies that $\mathbb{P}(Z_i = 0) = (1 - p_i)^n$. One can yield the same result by directly summing up the multinomial probability masses and applying the multinomial theorem: for instance,

$$\mathbb{P}(Z_1 = 0) = \sum_{\substack{z_2 + \cdots + z_m = n \\ z_2, \cdots, z_m \geq 0}} \frac{n!}{z_2! \cdots z_m!} p_2^{z_2} \cdots p_m^{z_m} = (p_2 + \cdots + p_m)^n = (1 - p_1)^n.$$

Thus, by applying a union bound,

$$\mathbb{P}(\exists i \text{ such that } Z_i = 0) \leq \sum_{i=1}^{m} \mathbb{P}(Z_i = 0) = \sum_{i=1}^{m} (1 - p_i)^n \leq m(1 - \hat{p})^n \leq m\, e^{-n\hat{p}},$$

where we apply $\hat{p} = \min_{1 \leq i \leq m} p_i$ and $1 + u \leq e^u$ ($\forall u \in \mathbb{R}$) in the last two inequalities above. Solving $m\, e^{-n\hat{p}} \leq \delta$, we conclude that $n \geq \frac{1}{\hat{p}} \ln \frac{m}{\delta}$ implies $\mathbb{P}(\exists i \text{ such that } Z_i = 0) \leq \delta$. □

E.3.1 (De-)Poissonization of Monotone Multinomial Events

Randomization is a probabilistic technique that provides a convenient way to analyze a sequence by treating the sequence index as a random variable/process. When the sequence $\{a_n\}_{n \geq 0}$ is *monotone* and *bounded*, in particular, it provides us upper/lower bounds of the difference between a_n and the expectation $\mathbb{E}[a_N]$ in terms of a non-negative integral random variable N. The following lemma provides a simple conversion between a deterministic monotone bounded sequence and the expectation of the randomized sequence, which we will prove later.

Lemma E.23 (De-Randomization Lemma, General)**.** *Consider a non-decreasing real-valued sequence* $\{a_j\}_{j \geq 0}$ *which lies in a closed interval* $[m, M]$*, i.e.,* $m \leq a_0 \leq a_1 \leq a_2 \leq \cdots \leq M$*. Let* N *be a non-negative integer-valued random variable with probability mass* $\mathbb{P}(N = j) = p_j$ *(*$j = 0, 1, 2, \cdots$*). Then, for any* $n \geq 0$*, it holds that*

$$\mathbb{E}[a_N] - (M - m) \cdot \mathbb{P}(N > n) \leq a_n \leq \mathbb{E}[a_N] + (M - m) \cdot \mathbb{P}(N < n). \quad \text{(Non-Decreasing Case)}$$

On the other hand, if $\{a_j\}_{j \geq 0}$ *is non-increasing, i.e.,* $M \geq a_0 \geq a_1 \geq a_2 \geq \cdots \geq m$*, a similar result holds that*

$$\mathbb{E}[a_N] - (M - m) \cdot \mathbb{P}(N < n) \leq a_n \leq \mathbb{E}[a_N] + (M - m) \cdot \mathbb{P}(N > n). \quad \text{(Non-Increasing Case)}$$

Here, all expectations are taken with respect to N*.*

Remark E.24 (On the choice of random index N). When obtaining an estimate of a sequence value using the lemma above, it is unnecessary to utilize the same random variable N for both upper and lower bounds. Namely, when the sequence is non-decreasing, a choice of N with $\mathbb{E}[N] > n$ gives a small lower tail $\mathbb{P}(N < n)$ for the upper bound; whereas an N with $0 < \mathbb{E}[N] < n$ gives a small upper tail $\mathbb{P}(N > n)$ for the lower bound. We can think of the opposite relationship for a non-increasing sequence as well. Moreover, to guarantee the tail bounds to be small, we often choose N having a lot of mass concentrated around its mean (e.g., sub-Gaussian); then, we can take advantage of concentration inequalities associated with N to have a better estimate of $a_n \approx \mathbb{E}[a_N]$.

Poissonization (Kac, 1949; Holst, 1986; Aldous, 1989; Jacquet & Szpankowski, 1998; Johansson, 1998; Borodin et al., 2000) can be thought of as a special case of randomization using Poisson random variables. A Poisson random variable $N \sim \mathrm{Poisson}(\lambda)$ with a parameter $\lambda > 0$ is equipped with a probability mass function

$$\mathbb{P}(N = n) = \frac{e^{-\lambda}\lambda^n}{n!}. \qquad (n = 0, 1, 2, \cdots) \tag{Poisson}$$

In this section, our primary goal is to complete the proof of Lemma E.12, asserting that the Poissonization technique is particularly advantageous for analyzing a monotone property (Def. E.11) of multinomial random vectors (e.g., the connectivity of a random graph whose edge connections are sampled with replacements). To see why, let us first review an elementary property of Poisson random variables: the sum of independent Poisson random variables is again a Poisson random variable, whose parameter is the sum of the parameters of individual Poisson variables. We defer the proof for brevity.

Lemma E.25. *Consider mutually independent Poisson random variables $Z_i \sim \mathrm{Poisson}(\lambda_i)$ $(i = 1, \cdots, m)$. Then, $\sum_{i=1}^m Z_i \sim \mathrm{Poisson}\left(\sum_{i=1}^m \lambda_i\right)$.*

Next, we explore the relationship between multinomial and Poisson distributions by establishing an equivalence between the probability associated with multinomial random vectors and the conditional probability of mutually independent Poisson random variables whose sum is fixed. As a result, we can construct a Poissonization of a sequence of probabilities by regarding the multinomial random vector's parameter n as a Poisson random variable. The detailed formal statement is below, although we again defer its proof.

Lemma E.26. *Let $\mathcal{A}$ be any property of a (random) m-dimensional vector: we write $(z_1, \cdots, z_m) \in \mathcal{A}$ if $(z_1, \cdots, z_m)$ satisfies the property $\mathcal{A}$. Then, for any $\lambda > 0$, and $p_1, \cdots, p_m > 0$ that sums to 1 (i.e., $\sum_{i=1}^m p_i = 1$),*

$$\mathbb{P}_{\mathrm{Multi}(n)}\Big((Z_1, \cdots, Z_m) \in \mathcal{A}\Big) = \mathbb{P}_{\mathrm{Po}(\lambda)}\left((Z_1, \cdots, Z_m) \in \mathcal{A} \,\middle|\, \sum_{i=1}^m Z_i = n\right).$$

Here, $\mathbb{P}_{\mathrm{Multi}(n)}$ is the probability measure under $(Z_1, \ldots, Z_m) \sim \mathrm{Multinomial}\,(n; p_1, \ldots, p_m)$, whereas $\mathbb{P}_{\mathrm{Po}(\lambda)}$ is the probability measure under $Z_i \sim \mathrm{Poisson}(\lambda p_i)$ $(i = 1, \cdots, m)$ which are mutually independent. As a result, it holds that

$$\mathbb{E}_{N \sim \mathrm{Poisson}(\lambda)}\left[\mathbb{P}_{\mathrm{Multi}(N)}\Big((Z_1, \cdots, Z_m) \in \mathcal{A}\Big)\right] = \mathbb{P}_{\mathrm{Po}(\lambda)}\Big((Z_1, \cdots, Z_m) \in \mathcal{A}\Big).$$

We now move our attention to a monotone property of (finite-dimensional) vectors, of which we recall the definition again.

Definition E.27 (Monotone property)**.** We refer to a property $\mathcal{A}$ of m-dimensional vectors as a **non-decreasing property** when, for any vector $(\Delta_1, \cdots, \Delta_m)$ with non-negative entries $\Delta_i \geq 0$ $(\forall i = 1, \cdots, m)$,

$$(z_1, \cdots, z_m) \in \mathcal{A} \implies (z_1 + \Delta_1, \cdots, z_m + \Delta_m) \in \mathcal{A}. \tag{Non-Decreasing Property}$$

On the other hand, we define **non-increasing property** $\mathcal{A}$ as what satisfies

$$(z_1, \cdots, z_m) \in \mathcal{A} \implies (z_1 - \Delta_1, \cdots, z_m - \Delta_m) \in \mathcal{A}. \tag{Non-Increasing Property}$$

A monotone property of multinomial random vectors has an interesting feature: the probability of satisfying the property is also monotone in the parameter n of the multinomial distribution (Lemma E.29). This is roughly because the multinomial distribution is inspired by multiple independent trials of with-replacement sampling: an additional trial corresponds to the increase of the parameter n by 1. This intuitive explanation can be formalized into the following lemma.

Lemma E.28. *Consider a multinomial random vector* $(Z_1, \cdots, Z_m) \sim \text{Multinomial}(n; p_1, \cdots, p_m)$ *for* $n \geq 1$ *and* $p_1, \cdots, p_m > 0$ *such that* $\sum_{i=1}^m p_i = 1$. *That is, for any non-negative integers* $z_1, \cdots, z_m$ *whose sum is* n,

$$\mathbb{P}(Z_1 = z_1, \cdots, Z_m = z_m) = \frac{n!}{z_1! \cdots z_m!} \cdot p_1^{z_1} \cdots p_m^{z_m}. \tag{Multinomial}$$

Also, consider a categorical (so-called multinoulli*) random variable* $j \sim \text{Categorical}(p_1, ..., p_m)$ *which can have a value among the integers* $\{1, 2, \cdots, m\}$, *i.e., for each* $i = 1, \cdots, m$,

$$\mathbb{P}(j = i) = p_i. \tag{Categorical}$$

Then, if we let $\widetilde{Z}_i = Z_i + \mathbb{1}_{\{j=i\}}$ *for* $1 \leq i \leq m$, *we have a new multinomial random vector as follows:*

$$(\widetilde{Z}_1, \cdots, \widetilde{Z}_m) \sim \text{Multinomial}(n+1; p_1, \cdots, p_m).$$

Using the above as a key lemma, we can prove the following lemma about the monotone property of a multinomial random vector.

Lemma E.29. *Recall the definition of* $\mathbb{P}_{\text{Multi}(n)}$ *from Lemma E.26. Fix any integers* $1 \leq n \leq n'$. *If a property* $\mathcal{A}$ *is non-decreasing, then*

$$\mathbb{P}_{\text{Multi}(n)}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right) \leq \mathbb{P}_{\text{Multi}(n')}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right).$$

The direction of the inequality should be opposite ("$\geq$*") if the property is non-increasing.*

Most importantly, thanks to Lemmas E.26 and E.29, we can apply the de-randomization lemma (Lemma E.23) for the sequence $\{a_n\}$ of probabilities defined with a monotone property $\mathcal{A}$ as

$$a_n := \mathbb{P}_{\text{Multi}(n)}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right).$$

The only things left to obtain a complete (de)-Poissonization lemma for upper/lower-bounding the sequence a_n are the tail probability bounds for a Poisson distribution. The concentration inequalities for the Poisson distribution are already well-known,[8] although we derive slightly different forms of them for our own purposes.

Lemma E.30. *If* $N \sim \text{Poisson}(n - \varepsilon)$ *for any* $n > 0$ *and* $0 < \varepsilon < n$, *it holds that*

$$\mathbb{P}(N \geq n) \leq \exp\left(-\frac{\varepsilon^2}{2n}\right). \tag{Upper Tail Bound}$$

On the other hand, if $N \sim \text{Poisson}(n + \varepsilon)$ *for any* $n > 0$ *and* $0 < \varepsilon < cn$ *for some* $c \in (0, 3)$, *it holds that*

$$\mathbb{P}(N \leq n) \leq \exp\left(-\frac{(3-c)\varepsilon^2}{6n}\right). \tag{Lower Tail Bound}$$

As a result, we finally summarize the arguments in this section as the following de-Poissonization lemma for a monotone property of multinomial random vectors:

Lemma E.31 (De-Poissonizaiton Lemma for Monotone Multinomial Events)**.** *Fix any* $n \geq 1$. *Define*

$$P_n := \mathbb{P}_{(Z_1, \cdots, Z_m) \sim \text{Multinomial}(n; p_1, \cdots, p_m)}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right).$$

Let $\mathcal{A}$ *be a non-decreasing property (Def. E.11) of* m*-dimensional vector. Then, we have an upper bound and a lower bound for* P_n *as follows:*

$$P_n \leq \mathbb{P}_{\text{Po}(n+\varepsilon)}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right) + \exp\left(-\tfrac{(3-c)\varepsilon^2}{6n}\right), \qquad (\forall c \in (0,3),\ \forall \varepsilon \in (0, cn))$$

$$P_n \geq \mathbb{P}_{\text{Po}(n-\varepsilon)}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right) - \exp\left(-\tfrac{\varepsilon^2}{2n}\right). \qquad (\forall \varepsilon \in (0, n))$$

If $\mathcal{A}$ *is a non-increasing property (Def. E.11), then we have similar upper and lower bounds for* P_n*:*

$$P_n \leq \mathbb{P}_{\text{Po}(n-\varepsilon)}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right) + \exp\left(-\tfrac{\varepsilon^2}{2n}\right), \qquad (\forall \varepsilon \in (0, n))$$

$$P_n \geq \mathbb{P}_{\text{Po}(n+\varepsilon)}\left((Z_1, \cdots, Z_m) \in \mathcal{A}\right) - \exp\left(-\tfrac{(3-c)\varepsilon^2}{6n}\right). \qquad (\forall c \in (0,3),\ \forall \varepsilon \in (0, cn))$$

[8] For example, see a document online: "A short note on Poisson tail bounds."

Here, we denote by $\mathbb{P}_{\mathrm{Po}(\lambda)}$ the probability measure under $Z_i \overset{\text{indep.}}{\sim} \mathrm{Poisson}(\lambda p_i)$ ($\forall i = 1, \cdots, m$).

Proof of Lemma E.12. This is the combination of Lemmas E.23, E.26, E.29 and E.30. □

We use this de-Poissonization lemma in the proof of our main theorem (Theorems E.7 and E.8).

Lastly, we conclude this section by presenting all the deferred proofs.

Proof of Lemma E.23. In the first part of the proof, we deal with the non-decreasing case. Observe that, since a_N has a finite expectation ($m \le \mathbb{E}[a_N] \le M$), the infinite sum $\sum_{j\ge 0} p_j a_j$ converges. Now, let us first obtain the upper bound. Since $\sum_{j\ge 0} p_j = 1$,

$$\begin{aligned}
a_n &= \sum_{0\le j<n} p_j a_n + \sum_{j\ge n} p_j a_n \\
&\le \sum_{0\le j<n} p_j a_n + \sum_{j\ge n} p_j a_j && (\because a_j \ge a_n \text{ if } j \ge n) \\
&= \sum_{0\le j<n} p_j (a_n - a_j) + \sum_{j\ge 0} p_j a_j \\
&\le (M-m)\cdot \sum_{0\le j<n} p_j + \sum_{j>0} p_j a_j && (\because a_n - a_j \le M - m) \\
&= (M-m)\cdot \mathbb{P}(N<n) + \mathbb{E}[a_N].
\end{aligned}$$

Likewise, we obtain the lower bound.

$$\begin{aligned}
a_n &= \sum_{0\le j\le n} p_j a_n + \sum_{j>n} p_j a_n \\
&\ge \sum_{0\le j\le n} p_j a_j + \sum_{j>n} p_j a_n && (\because a_j \le a_n \text{ if } j \le n) \\
&= \sum_{j\ge 0} p_j a_j - \sum_{j>n} p_j (a_j - a_n) \\
&\ge \sum_{j\ge 0} p_j a_j - (M-m)\cdot \sum_{j>n} p_j && (\because a_n - a_j \ge m - M) \\
&= \mathbb{E}[a_N] - (M-m)\cdot \mathbb{P}(N>n).
\end{aligned}$$

These imply the inequality (Non-Decreasing Case) and prove the first part of the lemma.

The second part for the non-increasing sequence $\{a_n\}$ directly follows by applying the first part to the non-decreasing sequence $\{b_n\}$ defined as $b_n := M + m - a_n$. Indeed, we have

$$\mathbb{E}[M+m-a_N] - (M-m)\cdot\mathbb{P}(N>n) \le M+m-a_n \le \mathbb{E}[M+m-a_N] + (M-m)\cdot\mathbb{P}(N<n),$$

which implies the inequality (Non-Increasing Case) and concludes the proof of the lemma. □

Proof of Lemma E.25. Denote the sum of random variables by $\bar{Z}_m := \sum_{i=1}^m Z_i$. Likewise, let us write $\bar{\lambda}_m := \sum_{i=1}^m \lambda_i$. To prove the lemma, we proceed with the induction on $m \ge 1$.

Since the base case is obvious ($Z_1 \sim \mathrm{Poisson}(\lambda_1)$), let us assume $m \ge 2$ and compute the probability mass. The inductive assumption says $\bar{Z}_{m-1} \sim \mathrm{Poisson}\left(\bar{\lambda}_{m-1}\right)$, which is independent of Z_m due to

the mutual independence of $Z_1, \cdots, Z_m$. Then, for any non-negative integer $\bar{z}$,

$$\begin{aligned}
\mathbb{P}\left(\bar{Z}_m = \bar{z}\right) &= \sum_{z_m=0}^{\bar{z}} \mathbb{P}(\bar{Z}_{m-1} = \bar{z} - z_m, Z_m = z_m) \\
&= 1 \cdot \sum_{z_m=0}^{\bar{z}} \mathbb{P}(\bar{Z}_{m-1} = \bar{z} - z_m) \cdot \mathbb{P}(Z_m = z_m) \qquad (\because \bar{Z}_{m-1} \perp Z_m) \\
&= \frac{\bar{z}!}{\bar{z}!} \cdot \sum_{z_m=0}^{\bar{z}} \frac{\exp\left(-\bar{\lambda}_{m-1}\right) \cdot (\bar{\lambda}_{m-1})^{\bar{z}-z_m}}{(\bar{z}-z_m)!} \cdot \frac{\exp(-\lambda_m) \cdot (\lambda_m)^{z_m}}{z_m!} \\
&= \frac{\exp\left(-\sum_{i=1}^m \lambda_i\right)}{\bar{z}!} \cdot \sum_{z_m=0}^{\bar{z}} \frac{\bar{z}!}{(\bar{z}-z_m)! \cdot z_m!} (\bar{\lambda}_{m-1})^{\bar{z}-z_m} \lambda_m^{z_m} \\
&= \frac{\exp\left(-\sum_{i=1}^m \lambda_i\right)}{\bar{z}!} \cdot \left(\bar{\lambda}_{m-1} + \lambda_m\right)^{\bar{z}} \qquad (\because \text{ binomial theorem}) \\
&= \frac{\exp\left(-\sum_{i=1}^m \lambda_i\right)}{\bar{z}!} \cdot \left(\bar{\lambda}_m\right)^{\bar{z}}.
\end{aligned}$$

Since the choice of $\bar{z}$ is arbitrary, by induction, this ends up proving that $\bar{Z}_m \sim \text{Poisson}\left(\bar{\lambda}_m\right)$.

Lastly, we remark that one may prove the same result without induction, by directly applying the multinomial theorem (which is essentially a recurrent application of the binomial theorem). □

Proof of Lemma E.26. By the law of total probability, for any probability measure $\mathbb{P}$ under any distribution of a random vector $(Z_1, \cdots, Z_m)$,

$$\mathbb{P}((Z_1, \cdots, Z_m) \in \mathcal{A}) = \sum_{(z_1, \cdots, z_m) \in \mathcal{A}} \mathbb{P}(Z_1 = z_1, \cdots, Z_m = z_m).$$

Thus, it suffices to compare the multinomial distribution and the conditional distribution of a Poisson random vector given a fixed sum: namely, we aim to show here that

$$\mathbb{P}_{\text{Multi}(n)}(Z_1 = z_1, \cdots, Z_m = z_m) = \mathbb{P}_{\text{Po}(\lambda)}\left(Z_1 = z_1, \cdots, Z_m = z_m \,\middle|\, \sum_{i=1}^m Z_i = n\right). \tag{17}$$

Moreover, it is sufficient for proving the first equation to study the case with non-negative integers $z_1, \cdots, z_m$ such that $\sum_{i=1}^m z_i = n$; otherwise, both sides of Eq. (17) are zero. In this case, observe an inclusion between events

$$\{Z_1 = z_1, \cdots, Z_m = z_m\} \subseteq \left\{\sum_{i=1}^m Z_i = n\right\}. \tag{18}$$

Because of this, we can compute a conditional probability mass as:

$$\begin{aligned}
&\mathbb{P}_{\text{Po}(\lambda)}\left(Z_1 = z_1, \cdots, Z_m = z_m \,\middle|\, \sum_{i=1}^m Z_i = n\right) \\
&= \frac{\mathbb{P}_{\text{Po}(\lambda)}\left(Z_1 = z_1, \cdots, Z_m = z_m\right)}{\mathbb{P}_{\text{Po}(\lambda)}\left(\sum_{i=1}^m Z_i = n\right)} \qquad (\because \text{ Eq. (18)}) \\
&= \left(\prod_{i=1}^m \frac{e^{-\lambda p_i}(\lambda p_i)^{z_i}}{z_i!}\right) \cdot \left(\frac{e^{-\lambda}\lambda^n}{n!}\right)^{-1} \qquad (\because \textstyle\sum_{i=1}^m Z_i \sim \text{Poisson}(\lambda), \text{ due to Lemma E.25}) \\
&= \frac{n!}{z_1! \cdots z_m!} \cdot p_1^{z_1} \cdots p_m^{z_m} \qquad (\because \textstyle\sum_{i=1}^m z_i = n) \\
&= \mathbb{P}_{\text{Multi}(n)}(Z_1 = z_1, \cdots, Z_m = z_m).
\end{aligned}$$

Therefore, we have just proved Eq. (17).

Lastly, we conclude the proof by computing the expectation in terms of $N \sim \mathrm{Poisson}(\lambda)$: since $\mathbb{P}(N=j) = \mathbb{P}_{\mathrm{Po}(\lambda)}\left(\sum_{i=1}^m Z_i = j\right)$ ($\because$ Lemma E.25), by law of total probability,

$$\begin{aligned}
&\mathbb{E}_{N\sim\mathrm{Poisson}(\lambda)}\left[\mathbb{P}_{\mathrm{Multi}(N)}\Big((Z_1,\cdots,Z_m)\in\mathcal{A}\Big)\right] \\
&= \sum_{j\geq 0}\mathbb{P}_{\mathrm{Multi}(j)}\Big((Z_1,\cdots,Z_m)\in\mathcal{A}\Big)\cdot\mathbb{P}(N=j) \\
&= \sum_{j\geq 0}\mathbb{P}_{\mathrm{Po}(\lambda)}\left((Z_1,\cdots,Z_m)\in\mathcal{A}\;\middle|\;\sum_{i=1}^m Z_i = j\right)\cdot\mathbb{P}(N=j) \\
&= \sum_{j\geq 0}\mathbb{P}_{\mathrm{Po}(\lambda)}\left((Z_1,\cdots,Z_m)\in\mathcal{A}\text{ and }\sum_{i=1}^m Z_i = j\right) \\
&= \mathbb{P}_{\mathrm{Po}(\lambda)}\left((Z_1,\cdots,Z_m)\in\mathcal{A}\right).
\end{aligned}$$

□

Proof of Lemma E.28. Take any integers $z_1,\cdots,z_m$ such that $z_1+\cdots+z_m = n+1$. Let us calculate the probability mass. Applying the law of total probability,

$$\begin{aligned}
&\mathbb{P}\left(\widetilde{Z}_1 = z_1,\cdots,\widetilde{Z}_m = z_m\right) \\
&= \sum_{i=1}^m \mathbb{P}\left(Z_1+\mathbb{1}_{\{j=1\}} = z_1,\cdots,Z_m+\mathbb{1}_{\{j=m\}} = z_m \mid j=i\right)\cdot\mathbb{P}(j=i) \\
&= \sum_{i=1}^m \mathbb{P}\left(Z_i = z_i-1, Z_r = z_r\;\;(\forall r\neq i)\right)\cdot\mathbb{P}(j=i) \qquad (\because j\perp(Z_1,\cdots,Z_m)) \\
&= \sum_{i=1}^m \left(\frac{n!\cdot z_i}{z_1!\cdots z_m!}\cdot\frac{p_1^{z_1}\cdots p_m^{z_m}}{p_i}\right)\cdot p_i \\
&= \frac{n!}{z_1!\cdots z_m!}\cdot p_1^{z_1}\cdots p_m^{z_m}\cdot\sum_{i=1}^m z_i \\
&= \frac{(n+1)!}{z_1!\cdots z_m!}\cdot p_1^{z_1}\cdots p_m^{z_m}.
\end{aligned}$$

Since the choice of $z_1,\cdots,z_m$ is arbitrary (given their fixed sum), it proves that $(\widetilde{Z}_1,\cdots,\widetilde{Z}_m)$ follows the distribution $\mathrm{Multinomial}(n+1;p_1,\cdots,p_m)$, as desired. □

Proof of Lemma E.29. We only consider the non-decreasing property $\mathcal{A}$ since the proof of the non-increasing case can be done symmetrically. Then, by induction, it suffices to prove

$$\mathbb{P}_{\mathrm{Multi}(n)}\left((Z_1,\cdots,Z_m)\in\mathcal{A}\right) \leq \mathbb{P}_{\mathrm{Multi}(n+1)}\left((Z_1,\cdots,Z_m)\in\mathcal{A}\right). \tag{19}$$

Fix any $n\geq 1$ and consider $(Z_1,\cdots,Z_m)\sim\mathrm{Multinomial}(n;p_1,\cdots,p_m)$. Then, by a property of multinomial random variables (Lemma E.28), we have another multinomial vector $(\widetilde{Z}_1,\cdots,\widetilde{Z}_m)\sim\mathrm{Multinomial}(n+1;p_1,\cdots,p_m)$ by $\widetilde{Z}_i = Z_i+\mathbb{1}_{\{j=i\}}$, where $j\sim\mathrm{Categorical}(p_1,\cdots,p_m)$. Observe that

$$\mathbb{P}\left((\widetilde{Z}_1,\cdots,\widetilde{Z}_m)\in\mathcal{A}\;\middle|\;(Z_1,\cdots,Z_m)\in\mathcal{A}\right) = 1,$$

since $Z_i\leq\widetilde{Z}_i$ $(\forall i)$ and $\mathcal{A}$ is a non-decreasing property.

Hence, we can derive the following by applying the law of total probability:

$$\begin{aligned}
&\mathbb{P}\left((\widetilde{Z}_1,\cdots,\widetilde{Z}_m)\in\mathcal{A}\right)\\
&=\mathbb{P}\left((\widetilde{Z}_1,\cdots,\widetilde{Z}_m)\in\mathcal{A}\;\middle|\;(Z_1,\cdots,Z_m)\notin\mathcal{A}\right)\cdot\mathbb{P}\left((Z_1,\cdots,Z_m)\notin\mathcal{A}\right)\\
&\quad+\mathbb{P}\left((\widetilde{Z}_1,\cdots,\widetilde{Z}_m)\in\mathcal{A}\;\middle|\;(Z_1,\cdots,Z_m)\in\mathcal{A}\right)\cdot\mathbb{P}\left((Z_1,\cdots,Z_m)\in\mathcal{A}\right)\\
&\geq 0+1\cdot\mathbb{P}\left((Z_1,\cdots,Z_m)\in\mathcal{A}\right)\\
&=\mathbb{P}_{\mathrm{Multi}(n)}\left((Z_1,\cdots,Z_m)\in\mathcal{A}\right).
\end{aligned}$$

This proves Eq. (19), concluding the proof. □

Proof of Lemma E.30. Let us recall the moment generating function (MGF) of a Poisson random variable $N\sim\mathrm{Poisson}(\lambda)$ is $\mathbb{E}[e^{tN}]=\exp\left(\lambda(e^t-1)\right)$ $(\forall t\in\mathbb{R})$:

$$\mathbb{E}[e^{tN}]=\sum_{j\geq 0}e^{tj}\cdot\frac{e^{-\lambda}\lambda^j}{j!}=e^{-\lambda}\cdot\sum_{j\geq 0}\frac{(\lambda e^t)^j}{j!}=\exp\left(-\lambda\right)\exp\left(\lambda e^t\right)=\exp\left(\lambda(e^t-1)\right).$$

For the upper tail (Chernoff) bound, we use $\lambda=n-\varepsilon$ and apply Markov inequality on e^{tN}: for any $n>\varepsilon>0$ and $t>0$,

$$\mathbb{P}(N\geq n)=\mathbb{P}\left(e^{tN}\geq e^{tn}\right)\leq\mathbb{E}\left[e^{tN}\right]\cdot e^{-tn}\leq\exp\left((n-\varepsilon)(e^t-1)-tn\right).$$

Since t is arbitrary, we take the infimum of both sides over $t>0$, obtaining

$$\begin{aligned}
\mathbb{P}(N>n)&\leq\inf_{t>0}\exp\left((n-\varepsilon)(e^t-1)-tn\right)\\
&=\exp\left(n\left(\ln\left(1-\frac{\varepsilon}{n}\right)+\frac{\varepsilon}{n}\right)\right)\\
&\leq\exp\left(-\frac{\varepsilon^2}{2n}\right). \qquad (\because\ \ln(1-u)+u\leq-\frac{u^2}{2}\quad\text{if } u>0)
\end{aligned}$$

This proves Eq. (Upper Tail Bound). On the other hand, for the lower tail (Chernoff) bound, we use $\lambda=n+\varepsilon$ and apply Markov inequality on e^{-tN}: for any $n>0$, $\varepsilon>0$, and $t>0$,

$$\mathbb{P}(N\leq n)=\mathbb{P}\left(e^{-tN}\geq e^{-tn}\right)\leq\mathbb{E}\left[e^{-tN}\right]\cdot e^{tn}\leq\exp\left((n+\varepsilon)(e^{-t}-1)+tn\right).$$

Similarly as before, taking the infimum over $t>0$,

$$\begin{aligned}
\mathbb{P}(N<n)&\leq\inf_{t>0}\exp\left((n+\varepsilon)(e^{-t}-1)+tn\right)\\
&=\exp\left(n\left(\ln\left(1+\frac{\varepsilon}{n}\right)-\frac{\varepsilon}{n}\right)\right)\\
&\leq\exp\left(-\frac{\varepsilon^2}{2n}+\frac{\varepsilon^3}{6n^2}\right). \qquad (\because\ \ln(1+u)-u\leq-\frac{u^2}{2}+\frac{u^3}{6}\quad\text{if } u>0)
\end{aligned}$$

Observe that, if $0<\varepsilon<cn$ for some $c\in(0,3)$,

$$-\frac{\varepsilon^2}{2n}+\frac{\varepsilon^3}{6n^2}\leq-\frac{\varepsilon^2}{2n}+\frac{\varepsilon^2}{6n}\cdot c=-\frac{(3-c)\varepsilon^2}{6n}.$$

This proves Eq. (Lower Tail Bound). □

E.3.2 Binomial Random Intersection Graphs

Let us explain the random k-intersection graph (Singer, 1995; Karoński et al., 1999; Godehardt & Jaworski, 2003). Consider a set V of vertices and another set W of items. Each vertex $v\in V$ is randomly assigned a subset of items $W_v\subset W$. Then, a pair of vertices u and v are adjacent (i.e., connected with an edge) in a random k-intersection graph if and only if at least k items are shared between u and v, i.e., $|W_u\cap W_v|\geq k$. We remark that k is a universal constant throughout our paper.

One of the most well-studied models of random intersection graphs is *binomial random k-intersection graphs* $\mathcal{G}^{(k)}(n,m,p)$, with $n=|V|$ and $m=|W|$. In the $\mathcal{G}^{(k)}(n,m,p)$ model, for every vertex $v\in V$, each item $w\in W$ is assigned to v independently with the same probability $p\in(0,1)$. That is,

the indicator variables $\mathbb{1}_{\{w \text{ is assigned to } v\}}$ are i.i.d. Bernoulli random variables with the same parameter p, for all vertices $v \in V$ and items $w \in W$. The word *binomial* is attached to the name because the number of items assigned to each vertex is a random variable following $\mathrm{Bin}(m, p)$.

We are particularly interested in the $\mathcal{G}^{(k)}(n, m, p)$ model because of the following observation.

Lemma E.32 (Evidence Graphs are Binomial k-Intersection Graphs under Poissonization)**.** *Let any $b \in B_D$. Consider a vector $(Z_{\boldsymbol{x}})_{\boldsymbol{x} \in \mathbb{X}}$ associated with a dataset D (i.e., an input sequence $\boldsymbol{x}$ is sampled $Z_{\boldsymbol{x}}$ times in D). Let $\mathbb{P}_{\mathrm{Po}(\lambda)}$ be a probability measure for $Z_{\boldsymbol{x}} \overset{\text{i.i.d.}}{\sim} \mathrm{Poisson}\,(\lambda/|\mathbb{X}|)$. Then, under $\mathbb{P}_{\mathrm{Po}(\lambda)}$, the b-evidence graph $\mathcal{G}_b^{(D,k)}$ (Def. E.5) is an instance of binomial random k-intersection graph $\mathcal{G}^{(k)}(n_b, m, p)$ with parameters*

$$n_b = |V_b|\,, \quad m = |\mathcal{X}_3|\,, \quad p = 1 - \exp\left(-\frac{\lambda}{|\mathbb{X}|}\right).$$

Proof of Lemma E.13. Recall that the vertex set of $\mathcal{G}_b^{(D,k)}$ is $V_b = \{x_{12} \in \mathcal{X}_{12} : f_1(x_{12}) = b\} \subset \mathcal{X}_{12}$. Also, consider $W := \mathcal{X}_3$ as the set of all 'items.' Define a set of items assigned to each vertex $x_{12} \in V_b$ as

$$W_{x_{12}} = \{x_3 \in W : (x_{12}, x_3) \in D\} = \left\{x_3 \in W : Z_{(x_{12},x_3)} \geq 1\right\}.$$

Observe that all vertex-item assignments are i.i.d. Bernoulli random variables with the same probability parameter p: for any $x_{12} \in V_b$ and $x_3 \in \mathcal{X}_3$,

$$p = \mathbb{P}_{\mathrm{Po}(\lambda)}(Z_{(x_{12},x_3)} \geq 1) = 1 - \mathbb{P}_{\mathrm{Po}(\lambda)}(Z_{(x_{12},x_3)} = 0) = 1 - \exp\left(-\frac{\lambda}{|\mathbb{X}|}\right).$$

In addition, observe that the set $S(x_{12}, x'_{12} \mid D)$ defined in Eq. (6) is identical to the intersection $W_{x_{12}} \cap W_{x'_{12}}$. Hence, by the definition of the edge set $E_b^{(D,k)}$ in Eq. (5), two distinct vertices x_{12} and x'_{12} are adjacent if and only if $\left|W_{x_{12}} \cap W_{x'_{12}}\right| \geq k$. In summary, every b-evidence graph $\mathcal{G}_b^{(D,k)}$ is a binomial random k-intersection graph $\mathcal{G}^{(k)}(n_b, m, p)$ with parameters $n_b = |V_b|$, $m = |\mathcal{X}_3|$, and $p = 1 - \exp(-\lambda/|\mathbb{X}|)$. □

Note that n, m, and p are not necessarily independent of each other; the parameters $m = m_n$ and $p = p_{n,m}$ are often regarded as functions of n. Indeed, it is often studied in the literature on random graphs that a sufficient condition for the parameters (in terms of $n \to \infty$) to guarantee certain graph properties, at least asymptotically. Here, we review a few of the seminal results on the connectivity (as well as the disappearance of isolated vertices) of binomial random k-intersection graphs (Singer, 1995; Zhao et al., 2014; 2017; Rybarczyk, 2011; 2017).[9] The proofs are involved; thus, we omit them. FIXED

Lemma E.33 (Zhao et al., 2014, Theorem 2; Zhao et al., 2017, Theorem 1 & Remark 1)**.** *Fix any $k \geq 1$. Suppose that*

$$m = \begin{cases} \Omega\left(\min\left\{n\,(\ln n)^5\,, n^\rho\right\}\right), & \text{if } k = 1, \text{ for any } \rho > 1; \\ \Omega\,(n)\,, & \text{if } k \geq 2, \end{cases}$$

and

$$p = \left(\frac{k!\,(\ln n + \alpha_n)}{n}\right)^{\frac{1}{2k}} \cdot \frac{1}{\sqrt{m}} \tag{8}$$

for any sequence $\{\alpha_n\}$ which attains a limit $\alpha_\infty \in [-\infty, +\infty]$ as $n \to \infty$. Then,

$$\lim_{n\to\infty} \mathbb{P}\left(\mathcal{G}^{(k)}(n, m, p) \text{ is connected}\right) = \lim_{n\to\infty} \mathbb{P}\left(\min \deg \mathcal{G}^{(k)}(n, m, p) \geq 1\right) = \exp\left(-e^{-\alpha_\infty}\right),$$

where we compute $\exp\left(-e^{-(-\infty)}\right) = 0$ and $\exp\left(-e^{-(+\infty)}\right) = 1$.

[9] Caveat: in the literature of (random) graph theory, the letter 'k' is often used for k-connectivity (i.e., being connected after removing less than k vertices/edges). On the other hand, they study 'random s-intersection graphs,' using s as the intersection constraint parameter.

Lemma E.34 (Singer, 1995, Propositions 3.1–2, Theorem 3.3)**.** *Let $k = 1$. Suppose that $m = n^\rho$ for $\rho > 0$ and*

$$p = \begin{cases} \dfrac{\ln n + \alpha_n}{m} & \text{for } \rho \le 1; \\ \sqrt{\dfrac{\ln n + \alpha_n}{mn}} & \text{for } \rho > 1, \end{cases}$$

for any sequence $\{\alpha_n\}$ which attains a limit $\alpha_\infty \in \{-\infty, +\infty\}$ as $n \to \infty$. Then,

$$\lim_{n\to\infty} \mathbb{P}(\mathcal{G}^{(1)}(n,m,p) \text{ is connected}) = \lim_{n\to\infty} \mathbb{P}\left(\min \deg \mathcal{G}^{(1)}(n,m,p) \ge 1\right) = \begin{cases} 0, & \text{if } \alpha_\infty = -\infty; \\ 1, & \text{if } \alpha_\infty = +\infty. \end{cases}$$

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 Exponents 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: The 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)

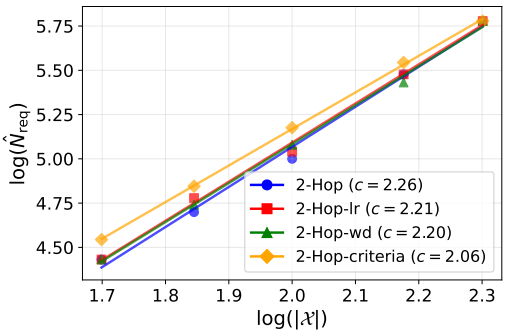


Figure 15: Robustness of power-law scaling relationship to hyperparameter variations in the 2-HOP task with $|\mathcal{X}| = 50$. Each line shows the training and test accuracy curves for a different configuration: (1) baseline, (2) reduced learning rate (4e-4, half of baseline), (3) reduced weight decay (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

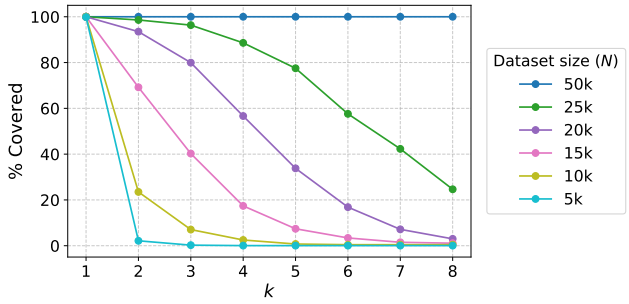


Figure 16: Coverage analysis for Non-tree task with $|\mathcal{X}| = 50$. The graph shows the percentage of ID test data covered at different k values across various dataset sizes (N). Compared to the 2-Hop task (Fig. 3, left), Non-tree has significantly lower coverage at equivalent dataset sizes, indicating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

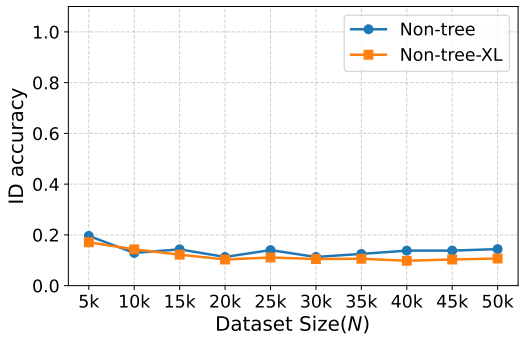


Figure 17: ID test accuracy comparison between GPT-2 (96M parameters) and GPT-2-XL (1.5B parameters) on the Non-tree task with $|\mathcal{X}| = 50$, measured 100 epochs after training accuracy exceeds 0.99. Despite the 15x increase in parameter count, the accuracy does not increase.

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 between Mamba and GPT-2 on Non-tree Task

Fig. 18 shows that the 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

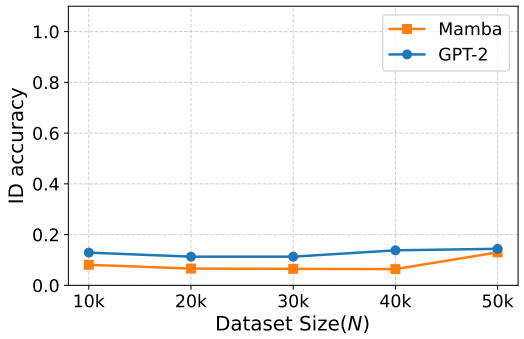


Figure 18: ID test accuracy comparison between GPT-2 and Mamba on the NON-TREE task with $|\mathcal{X}| = 50$, measured 100 epochs after training accuracy exceeds 0.99.

that the generalization failure is more likely due to the task structure itself, rather than a specific model architecture.

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.

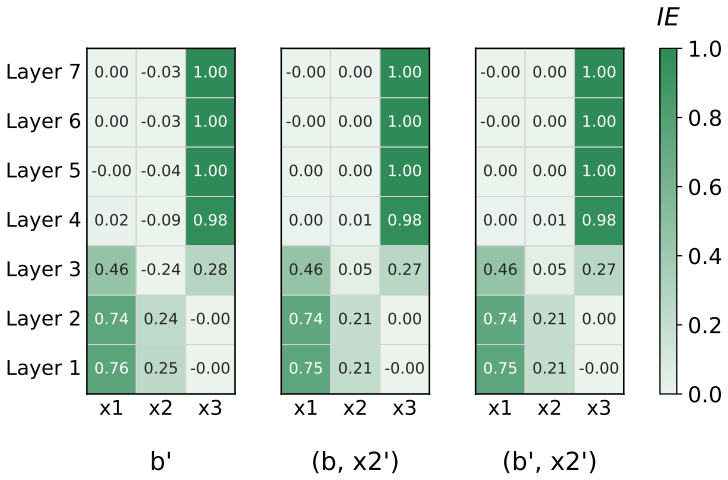


Figure 19: Causal tracing analysis for the NON-TREE model after extended training. The heatmap shows indirect effect values across different layer-token positions. **Left**: perturbation leading to different intermediate state $b = f_1(x_1, x_2)$. **Middle**: same b but different x_2. **Right**: different 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 for 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 modification 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., 2024a) 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. The 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.

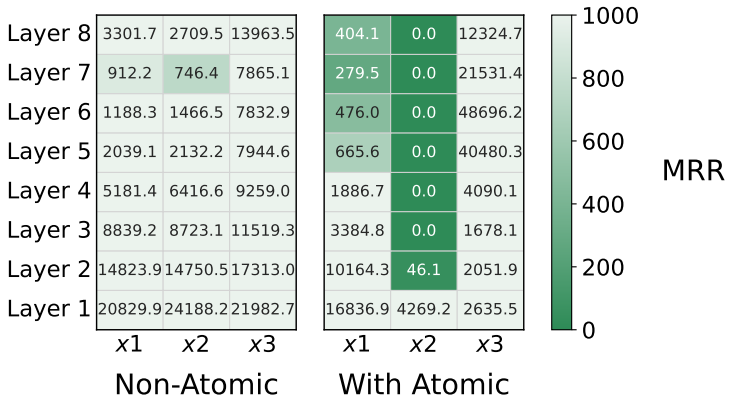


Figure 20: MRR scores for intermediate state representations projected to vocabulary space. **Left:** Standard training ($f_1 \neq f_2$, no partial computation) shows very high MRR regardless of position and layer. **Right:** Training with partial computation ($f_1 = f_2$, with partial examples) shows 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 the 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.