SOMETIMES I AM A TREE: DATA DRIVES UNSTABLE HIERARCHICAL GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks often favor shortcut heuristics based on surface-level patterns. Language models (LMs), for example, behave like n-gram models early in training. However, to correctly apply grammatical rules, LMs must instead rely on hierarchical syntactic representations rather than on surface-level heuristics derived from n-grams. In this work, we use cases studies of English grammar to explore how latent structures in training data drives models toward improved out-of-distribution (OOD) generalization. We then investigate how data composition can lead to inconsistent behavior across random seeds. Our results show that models stabilize in their OOD behavior only when they commit to either a surface-level linear rule or a hierarchical rule. The hierarchical rule, furthermore, is induced by grammatically complex sequences with deep embedding structures, whereas the linear rule is induced by simpler sequences. When the data contains a mix of simple and complex examples, potential rules compete; each independent training run either stabilizes by committing to a single rule or remains unstable in its OOD behavior. We also identify an exception to the relationship between stability and generalization: Models which memorize patterns from homogeneous training data can overfit stably, with different rules for memorized and unmemorized patterns. While existing works have attributed similar generalization behavior to training objective and model architecture, our findings emphasize the critical role of training data in shaping generalization patterns and how competition between data subsets contributes to inconsistent generalization outcomes.

1 INTRODUCTION

Neural networks often learn shortcut heuristics which reflect simple, surface-level patterns in data. In the case of language models (LMs) trained on next-token prediction objectives, this simplicity bias can lead models to behave like n-gram models, relying heavily on local dependencies without fully capturing the deeper, more complex structures of language (Choshen et al., 2022; Geirhos et al., 2020; Saphra & Lopez, 2018). However, LMs are also capable of breakthroughs in generalization, shifting from these simple heuristics to more sophisticated behaviors (Choshen et al., 2022; Chen et al., 2023; McCoy et al., 2020). Such transitions suggest that under certain training conditions, LMs can eventually overcome spurious shortcuts and use linguistic structures to generalize beyond surface-level patterns. Previous works often attribute this ability to model architecture and training objectives (Ahuja et al., 2024; McCoy et al., 2020). In this work, we investigate how *data* characteristics influence the generalization rules learned, especially when multiple solutions fit the training data equally well. We also examine the instabilities associated with generalization behaviors.

To understand when and why a model favors learning latent structures over surface-level heuristics, we use case studies in learning English grammar rules. Grammatically correct sentences in English must follow a set of rules that operate on a sequence's latent tree-like structure (Chomsky, 2015; Crain & Nakayama, 1987). When trained on next-token prediction, an LM may approximate these rules from surface-level statistics, acting as an n-gram model by applying a *linear rule*. However, such a model struggles to generalize to unseen grammatical patterns. Figure 1 (*bottom right*) shows that a LM can use a linear bigram model to capture the relationship between a subject noun and its verb by *inflecting* the verb with the same plurality as the subject. This LM would fail to generalize when a *distractor* noun, e.g., from a prepositional phrase, appears between subject and verb. In contrast,

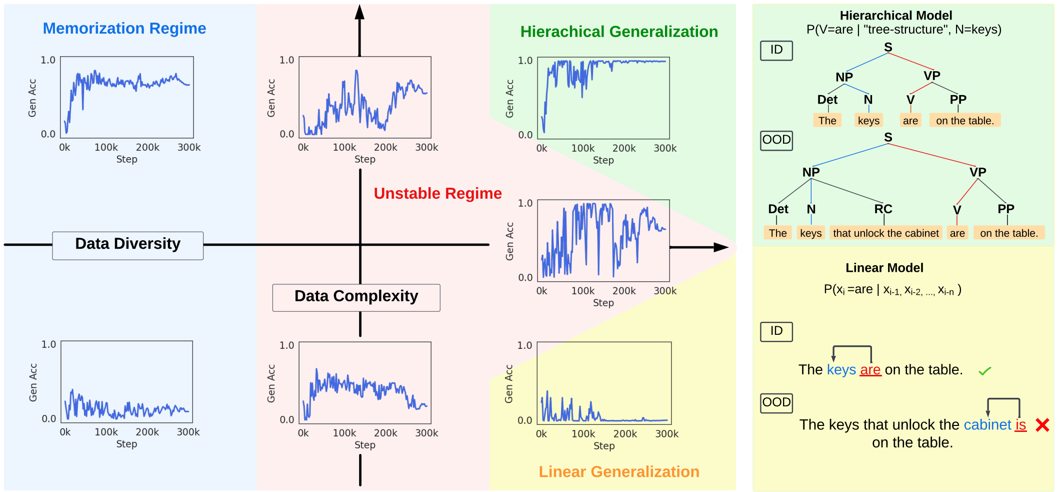

Figure 1: **Data plays a critical role in generalization behaviors and training stability.** *Left:* Along the data diversity x-axis, Low data diversity (as measured by variation in syntactic structure) leads the model to memorize unreliable sample-specific patterns, whereas high data diversity promotes commitment to a general rule. Along the data complexity y-axis, high data complexity (i.e., with more center embeddings) induces the hierarchical rule, while simpler data (right-branching sentences) induces the surface-level linear rule. Mixing these data types results in unstable OOD training behaviors. *Upper Right:* A model that captures hierarchical structure can generalize grammatical rules OOD by correctly identifying the subject as the noun closest to the root on the syntax tree graph. *Lower Right:* A model that uses the linear rule will treat the most recent noun as the target verb's subject and thereby fail to generalize to unseen sentence compositions.

Figure 1 (*upper right*) shows a model that instead uses a latent tree structure can learn the correct syntactic rule (i.e., the *hierarchical rule*), enabling it to generalize to novel sentence compositions.

Building on previous work (McCoy et al., 2018; 2020; Ahuja et al., 2024), we use two tasks—question formation and tense inflection—to investigate whether the model learns the hierarchical rule or defaults to the surface-level linear rule. We train models on ambiguous data, which is compatible with both rules, and evaluate them on OOD data which is compatible only with the hierarchical rule. We first find that a preference for OOD hierarchical generalization is induced by training on samples with center embeddings, where the subject is modified by an relative clause. This result mirrors a celebrated claim from linguistics (Wexler, 1980) that center embeddings are responsible for human syntax acquisition.

Models trained on the same data exhibit inconsistent OOD behaviors across random seeds. By examining training dynamics, we identify a connection between training stability and rule commitment: Only runs that commit to a rule can exhibit stable OOD performance during training. Connecting back to data, we show that training dynamics can be categorized into three distinct regimes that depend on data complexity and diversity, illustrated in Figure 1 *left*. Data diversity determines whether a model learns a general rule, while data complexity determines which rule is preferred. Models trained on a mix of hierarchical-inducing samples and linear-inducing samples are most unstable during training and exhibit the largest inconsistency across random seeds.

Taken together, our findings demonstrate that data composition plays a critical role in shaping model's OOD generalization behaviors. Our contributions are as follows:

- We show that sentences with complex grammatical structure—specifically center embeddings—drive LMs to favor hierarchical syntactic representations over surface-level n-gram heuristics, enabling correct out-of-distribution (OOD) generalization of grammatical rules (see Section 4).
- We demonstrate that models stabilize in OOD performance only when they commit to either a surface-level heuristic or a hierarchical rule (see Section 5).

- We show that when the training data mixes complex and simple grammatical structures, the resulting rules are inconsistent across random seeds and many models fail to stabilize OOD behavior by the end of training (see Section 5).
- We identify an exception to the relationship between stability and rule learning: Models trained on insufficiently diverse data stabilize in a memorization regime without learning either rule, highlighting another way that data can drive generalization failures (see Section 6).

2 RELATED WORK

We include the most relevant work in this section. For an extended discussion of related work, please refer to the Appendix A.

2.1 SYNTAX AND HIERARCHICAL GENERALIZATION

McCoy et al. (2018) first used the question formation task to study hierarchical generalization in neural networks, showing that RNNs trained with a seq-to-seq objective exhibit limited hierarchical generalization. However, adding attention mechanisms improved performance on the generalization set. Later, McCoy et al. (2020) found that tree-structured architectures consistently induce hierarchical generalization. Petty & Frank (2021) and Mueller et al. (2022) further investigated inductive biases and concluded that, like RNNs, transformers tend to generalize linearly. This view was challenged by Murty et al. (2023), who attributed the failure of prior attempts to insufficient training, demonstrating that decoder-only transformers can generalize hierarchically, but only after in-distribution performance has plateaued. Expanding on this, Ahuja et al. (2024) showed that hierarchical generalization is achieved only with models trained on a language modeling objective.

Previous work primarily attributed the source of inductive bias toward hierarchical rule to model architecture, whereas our study highlights the impact of data, and we further provide a precise measure of data complexity and data diversity. In addition, while McCoy et al. (2018) and Ahuja et al. (2024) observed inconsistencies across training runs, the underlying causes remain unexplored.

Similar to our work, Papadimitriou & Jurafsky (2023) and Papadimitriou & Jurafsky (2020) also studied how training data could introduce an inductive bias to affect language acquisition. specifically identified that by pretraining models on data with a recursive structure, finetuning them on natural language yields superior performances. This finding is closely related to our conclusions around center embeddings since the center embedding structure in language is recursive in nature.

2.2 TRAINING DYNAMICS AND GROKKING

Grokking refers to the surprising phenomenon where a neural network, after achieving seemingly poor performance for a long period, suddenly generalize on unseen data. Power et al. (2022) first observed this behavior in simple arithmetic tasks. Since then, the exact mechanism of grokking has been widely studied. In classic grokking, the model transitions from memorization to generalization, allowing it to achieve non-trivial performance on unseen data. Zhu et al. (2024) studies the role of data and finds that grokking only occurs when data set meets a the critical size. Berlot-Attwell et al. (2023) broadly studies how data diversity and complexity to generalization behaviors. Liu et al. (2022) studies how grokking can be induced with different weight norms, associating generalization with a specific goldilocks zone weight norm value. Different from existing grokking work, we studied a different types of grokking: “structural grokking" (Murty et al., 2023). In structural grokking, a model transitions from the simple linear rule to the hierarchical rule, leading to non-trivial performance on OOD data. Unlike existing works in classic grokking, we not only study the competition between memorization and generalization, but also on competition between different generalization rules. Importantly, we characterize the unstable regime in both data diversity and data complexity and we address connections between training stability and consistency under random variation.

2.3 RANDOM VARIATION

Although choices like hyperparameter settings, architecture, and optimizer all shape model outcomes, training remains inherently stochastic. Models are sensitive to random initialization and the order of training examples. Several studies (Zhou et al., 2020; D’Amour et al., 2022; Naik et al., 2018) have reported significant performance differences across model checkpoints and Zhou et al. (2020) noted that instability extends throughout the training curve. Dodge et al. (2020) found that both weight initialization and data order contribute equally to out-of-sample performance variation. Unlike prior

Table 1: **Examples from Two Grammar Case Studies.** *Left*: In the question formation task, the model moves the main auxiliary verb to the front to form a question. *Right*: In the tense inflection task, the model inflects the main verb from past to present tense, while respecting subject-verb agreement.

Dataset	Task Type	Examples
Question Formation	Quest (Ambiguous)	**Input:** My unicorn does move the dogs that do wait. **Output:** Does my unicorn move the dogs that do wait?
	Quest (Unambiguous)	**Input:** My unicorn who doesn't sing does move. **Linear Output:** Doesn't my unicorn who sing does move? **Hierarchical Output:** Does my unicorn who doesn't sing move?
Tense Inflection	Present (Ambiguous)	**Input:** My zebra behind the peacock smiled. **Output:** My zebra behind the peacock smiles.
	Present (Unambiguous)	**Input:** My zebra behind the peacocks smiled. **Linear output:** My zebra behind the peacocks smile. **Hierarchical output:** My zebra behind the peacocks smiles.

work, which focuses on the experimental implications of random variations, we investigate the source of these training inconsistencies and link them to characteristics of the training data.

3 EXPERIMENTAL SETUP

The question formation task and the tense inflection task are first proposed by Frank & Mathis (2007) and Linzen et al. (2016) as canonical tasks to assess a model's language modeling ability. In this study, we use the synthetic dataset constructed by McCoy et al. (2018) (for question formation) and McCoy et al. (2020) (for tense inflection).

3.1 QUESTION FORMATION TASK

In the **question formation (QF)** task, a declarative sentence is transformed into a question (see Table 1) by moving the main auxiliary verb (such as "*does*" in "*does move*") to the front. Our training data permits two strategies for choosing which verb to move: (1) *move first*: a linear rule that moves the first auxiliary, or (2) *move main*: a hierarchical rule—the correct rule in English grammar—based on the sentence's syntactic structure. This syntactic structure links each word into a tree-like structure in which edges specify syntactic dependencies (e.g., subject, preposition, object), as shown in Figure 2. The model leverages this tree representation to determine which auxiliary to move.

In Table 1, the first example is considered **ambiguous** because both the hierarchical and linear rules produce the correct outcome. In contrast, the second example is **unambiguous** because only the hierarchical rule produces the correct outcome. The training data contains only ambiguous samples, while the OOD generalization set includes only unambiguous samples. If a model uses a hierarchical representation of syntax, it should achieve 100% accuracy on both the in-distribution (ambiguous questions) and OOD generalization (unambiguous questions) sets. Conversely, if a model rise on linear rules, it will score 0% on the OOD generalization set, but still score 100% accuracy on the in-distribution set. We therefore use the model's accuracy on the OOD generalization set as a metric for hierarchical generalization.

3.2 TENSE INFLECTION TASK

In the **tense inflection (TI)** task, we provide the model with a sentence in the past tense, and the model transforms it into the present tense. Since past-tense verbs in English do not differentiate between singular and plural forms, the model must identify the subject to determine whether the present-tense verb should be inflected as singular or plural. The TI task tests whether the model follows the hierarchical or linear rule for subject-verb agreement. The linear rule inflects the verb based on the most recent noun, while the hierarchical rule correctly inflects the verb according to the subject. Like in the QF task, the training data contains ambiguous samples (example in Table 1), where the subject noun (i.e., "*zebra*") and the most recent noun (i.e., "*peacock*") always share the same plurality and therefore either rule produces the correct answer. The OOD generalization set includes unambiguous examples, where the subject and the most recent noun differ in plurality and therefore only the hierarchical rule produces the correct answer. Similar to the QF task, we use the model's main-verb prediction accuracy on the OOD set as a metric for hierarchical generalization.

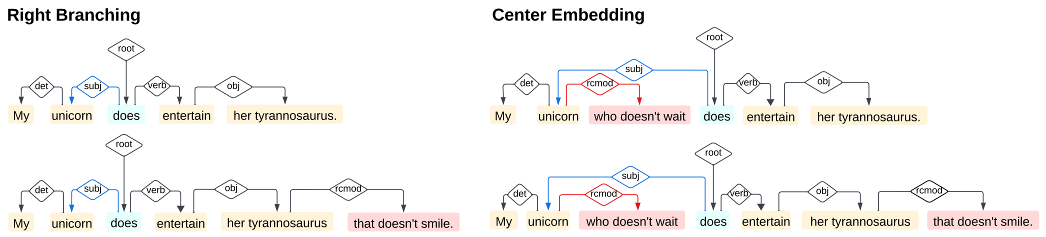

Figure 2: **Sentence Examples.** *Left:* Right branching sentence examples. The linear progression of the main phrase is not interrupted by the relative clause. *Right:* Center embedded sentence examples. When the relative clause modifies the subject, it interrupts the linear progression of the main clause.

3.3 MODEL, DATA AND TRAINING

We use a decoder-only Transformer architecture with 12M parameters: 6 layers of 8 heads with a 512-dimensional embedding for QF. For TI, we use the same transformer architecture but with 4 layers. All models are trained from scratch on a causal language modeling objective for 300K steps. We use the Adam optimizer (Kingma & Ba, 2014), a learning rate of 1e-4, and a linear decay schedule. We run all experiments on the same 50 random seeds. The training, validation and OOD generalization data are all generated with Context-Free Grammar (CFG) using a simplified set of grammatical rules. We use a word-level tokenizer with a vocabulary of size 72.

4 DATA COMPLEXITY DETERMINES RULE PREFERENCE

We begin by analyzing the QF task, exploring how different training data subsets influence a model's preference for the hierarchical rule over the simpler linear rule (Section 4.2). We then show that the same conclusion extends to the TI task (Section 4.3).

4.1 CENTER EMBEDDING

Center embedding occurs when a clause—often acting as a modifier—is placed within another clause or phrase. Figure 2 (*left*) illustrates two examples of center embedded sentences, where the embedded clause disrupts syntactic dependencies, such as the subject-verb-object relationship. Sentences without center embedding are exclusively right-branching. Right-branching structures may also include modifying clauses, but these are appended at the end of the main clause, maintaining its linear flow (see Figure 2, *right*). Center embedding has been central to linguistic studies on the types of data required to learn grammatical rules. According to Chomsky's generative grammar framework Chomsky (2015), center-embedded clauses give rise to hierarchical, tree-like syntactic structures. Additionally, Wexler (1980) posits that all English syntactic rules can be learned from "degree 2" sentences, which contain exactly one embedded clause.

We now investigate whether the same type of data can lead a LM to acquire the hierarchical grammar rule. To correctly predict the distribution of next tokens, LMs must track dependencies between sentence components. In right-branching sentences, LMs can rely on linear proximity to identify dependencies; for example, as shown in Figure 2, a simple bigram model suffices to capture the subject-verb relationship. However, center embeddings introduce relative clauses of various lengths, making linear n-gram models inefficient for capturing subject-verb dependencies. In these cases, modeling the subject-verb relationship with a tree structure is more compact and efficient.

4.2 QUESTION FORMATION

As specified in Section 3.1, the training data for QF must be ambiguous between the linear rule (i.e., moving the first auxiliary) and the hierarchical rule (i.e., moving the main auxiliary). Center embedded sentences do not meet this ambiguity requirement and, therefore, cannot appear in question formation training samples. To ensure the model is exposed to diverse sentence types, McCoy et al. (2018)] introduces a secondary task to the QF training dataset: declaration copying. Like question formation, the declaration-copying sample starts with a declarative sentence, but instead of transforming it, the model simply repeats it. Since the ambiguity requirement does not apply to the declaration-copying task, center-embedded sentences are included in this secondary task. Concrete examples of both tasks can be found in Appendix B.

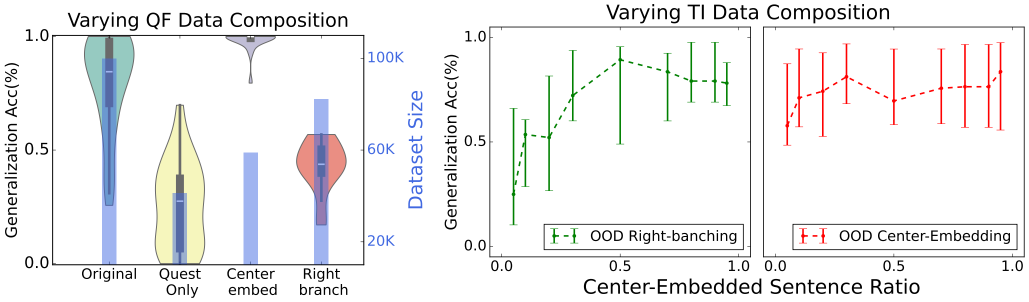

Figure 3: **Components of training data drive different generalization behaviors.** *Left:* Center embedded sentences, which in the QF training data only appear in declaration copying examples, induce hierarchical generalization. *Right:* Models are trained on different data mixes and evaluated on two OOD sets: unambiguous right branching sentences (*green*) and unambiguous center embedded sentences (*red*). For center embedded sentences, the hierarchical rule is preferred regardless of data mixes. For right branching sentences, the model's preference for the hierarchical rule is exclusively driven by having a large mix of center embedded sentences in the training data.

We train models on three subsets of the original training data, varying the composition of the declaration-copying examples. In *Quest Only*, we remove all declaration copying examples. In *Center embed*, we only keep center-embedded examples. In *Right branch*, we only keep right branching examples. For all three subsets, the question formation samples remain unchanged. Each setup reaches 100% in-distribution validation accuracy; however, the OOD generalization performance, shown in Figure 3 (*left*), differs significantly across the subsets. When the declaration-copying task is removed, none of the 50 runs achieve an OOD generalization accuracy above 75%, indicating that declaration copying examples are essential for inducing the hierarchical rule. When trained solely on center-embedded sentences in the declaration-copying task, models exhibit a strong preference for the hierarchical rule. In contrast, training only on right-branching sentences leads to poor hierarchical generalization. This evidence suggests that center-embedded sentences direct a model towards the hierarchical rule. While in this set of experiment, we did not control the number of modifying clauses present in the sentence. In Appendix C.1, we show that an alternative hypothesis based on the number of modifying clauses fails to parition the model's generalization behaviors.

4.3 TENSE INFLECTION

We now analyze hierarchical generalization in the tense inflection task, demonstrating the generality of our findings across grammatical rules. Linzen et al. (2016) first proposed the idea to use a verb inflections to assess the model's grammatical capabilities. McCoy et al. (2020) then adopted the question formation data to the tense inflection task by creating the TI dataset using a set of CFG rules and vocabularies similar to the ones used for QF.

In the TI training data, both right branching and center embedded sentences are made ambiguous by ensuring the distractor noun shares the same plurality as the subject. For right branching sentences, since there isn't a relative clause modifying the subject, a preposition phrase modifying the subject provides the distractor noun. In contrast, for center embedded sentences, since there is a relative clause modifying the subject, either the subject or the object of the modifying clause can act as the distractor noun. We list examples below:

1. Right branching: The main verb and the subject noun are separated by a prepositional phrase (e.g., “ *to the cabinet*"). The noun in the prepositional phrase acts as the distractor noun.
   Example: *The keys to the cabinet are on the table.*
2. Center Embedding: The main verb and subject noun are separated by a relative clause. Either the subject (Example 2) or the object (Example 1) of the relative clause acts as the distractor noun.
   Example 1: *The keys that unlock the cabinet are on the table.*
   Example 2: *The keys that the bear uses are on the table.*

Sentences with center embeddings exhibit a recursive structure: inside the relative clasue, one can find the same structure as the entire sentence. This recursive nature that drives hierarchical treatment. Our goal is to verify that center embeddings also drive OOD generalization in tense inflection. We create variations of the TI training data by adjusting the ratio of right branching to center embedded sentences in the training data, keeping the total training size constant. We train the model on 9 different training data mixes, and then test its generalization behaviors across two OOD sets, results shown in Figure 3 (*right*). The green line shows the model's generalization accuracy on *unambiguous* right branching sentences. When training data is dominated by ambiguous right branching sentences, the model fails to learn the hierarchical rule, as indicated by low OOD generalization accuracy. However, as we increase the proportion of center embedded sentences in the training data, these center embedded sentences—despite being ambiguous—bias the model towards applying hierarchical rule on OOD right branching sentences, reflected by improved generalization accuracy.

The red line in Figure 3 (*right*) shows the model's generalization accuracy on *unambiguous* center embedded sentences. Regardless of data mixes, the model treats center embeddings as hierarchical by default, and applies the hierarchical rule on unambiguous OOD data. By contrast, models only apply the same hierarchical rules to right branching sentences after exposure to sufficient amount of center embeddings during training. Based on those two observations, we conclude that center embeddings induce a general preference towards tree structure.

5 TRAINING STABILIZES IF A MODEL COMMITS TO A RULE

Why do some runs fail to generalize hierarchically even when trained on hierarchical-inducing data? In this section, we will show that these failures are a result of training instability; models only stabilize OOD if they commit to a general rule.

5.1 INSTABILITY DURING TRAINING

When training models on both QF and TI, some random seeds lead to highly unstable OOD behavior, with generalization accuracy often undergoing large swings during training. Figure 1 shows examples of generalization accuracy over training, with further details on training instability in Appendix D. Furthermore, the unstable behavior is not *consistent* across different seeds. We measure instability across training time using *total variation* (TV). Specifically, we checkpoint the model every 2K steps and measure the generalization accuracy at each checkpoint, denoting as Acc_i. The total variation is defined as:

$$\text{Total Variation (TV)} = \mathrm{Avg}_i\left(|\mathrm{Acc}_i - \mathrm{Acc}_{i-1}|\right)$$

In Figure 11, we visualize several examples runs and their total variation.

5.2 TRAINING STABILITY TIES TO RULE COMMITMENT

We next demonstrate the connection between stable OOD behavior (i.e., small TV) and rule commitment. We construct QF training datasets such that they contain different proportions of hierarchical-inducing and linear-inducing declarations, while keeping questions constant. Further details on the dataset can be found in Appendix C.2.

If the training data is dominated by either linear-inducing (linear=99%) or hierarchy-inducing (linear=0%) examples, more random seeds lead to stable OOD curves (Figure 5). When the training data is a heterogeneous mix instead, potential rules compete, leading to unstable training.

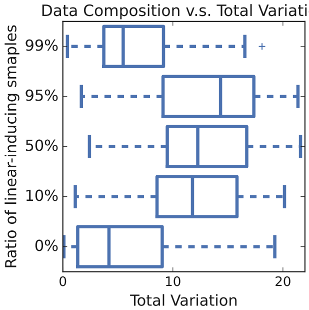

Figure 5: Competition between subsets of data drives training instability.

However, even a heterogeneous mix can lead to some stable training runs. Figure 4 shows that in these stable models, the final generalization accuracy is either 100% or 0%, indicating that the stable models have all committed to a general rule. While models can stabilize in either rule, data composition (i.e., proportions of hierarchical- and linear-inducing data) determines which rule is favored. In mixed data scenarios (e.g., linear=10% case), the final generalization accuracy is bimodally distributed for stable

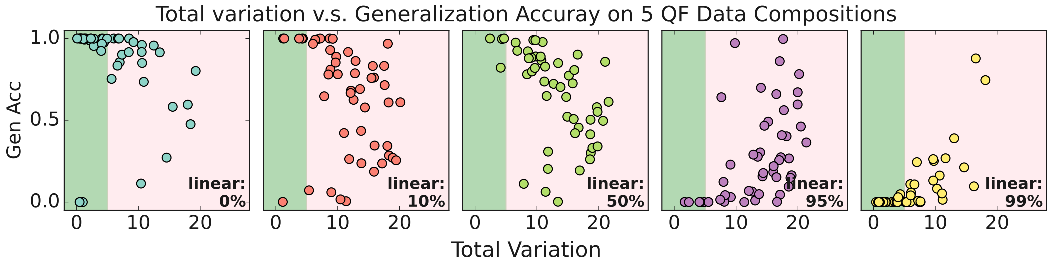

Figure 4: **Total Variation across training v.s. final generalization accuracy for QF task.** OOD behavior stabilizes during training if a model commits to a simple rule. By mixing data that induces the linear and hierarchical rules, we can create conditions that allow models to stabilize in either rule. “Linear" denotes the proportion of linear-inducing declarations in the data.

runs, clustering around 100% or 0%. This bimodality suggests that training stability is associated with a commitment to either the linear or the hierarchical rule, even when the data mix does not favor either rule.

In summary, with heterogeneous training data, competition between rules leads to more unstable training runs. Even with heterogeneous data mixes, however, some runs can still stabilize if they commit to one of the competing rules. In Appendix E.2, we replicate this analysis for the TI task, showing similar results.

6 DATA DIVERSITY LEADS TO GENERALIZATION

We have linked training stability to rule commitment. But why can’t networks stabilize without committing to a rule? In this section, we will explore the non-monotonic relationship between data diversity, training instability, and rule commitment. Specifically, we will show diverse data leads a model to extrapolate rules instead of memorizing individual examples.

6.1 MEASURING DATA DIVERSITY

In order to measure the diversity of our training data, we must compute the syntactic similarity between different example sentences. We describe a sentence pair’s similarity by the tree-edit distance (TED) of their latent tree representations (Chomsky, 2015). When two sentences share the same syntax tree, transforming one into the other requires only vocabulary changes. For example, “*My unicorn entertains her tyrannosaurus*,” and, “*Your zebra eats some apples*,” have different vocabulary but identical syntax trees. We define data diversity as the number of unique syntactic trees in the training data. We will show that when the model is exposed to a fewer unique syntax trees during training, it memorizes those patterns without extrapolating any rules to unseen sentence structures. Consequently, the model fails to commit to a general rule.

6.2 INVERSE U-SHAPED SCALING

In this section, we examine the relationship between data diversity and rule learning. In Appendix F, we further investigate the memorization patterns when trained on dataset of different diversity.

Commitment to hierarchical rule We first control data diversity on datasets that induce hierarchical generalization in QF. We construct variations of the QF training data, each with 50K question samples and 50K hierarchical-inducing declarations, while varying the diversity of the declaration examples. We train 20 random seeds for each training set variation. To measure intra-run instability, we use total variation, and to assess hierarchical rule commitment, we report the proportion of runs achieving generalization accuracy >95%. Figure 6 (left) shows the distribution of total variation across 20 seeds and the corresponding hierarchical generalization ratios.

We observe an inverse U-shaped relationship between data diversity and training instability, revealing three distinct regimes. In the low-diversity regime, training is stable but the model fails to commit to a rule. In the mid-diversity regime, training becomes unstable due to variation across batches. Overall, with insufficient diversity, relatively few runs can learn the hierarchical rule. Finally, in

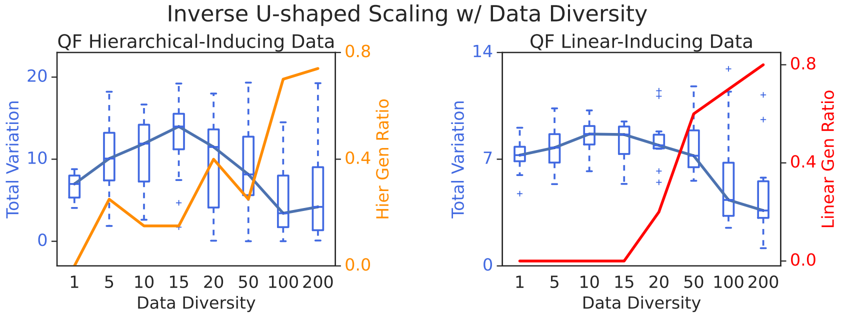

Figure 6: **Inverse U-shaped relationship between training stability and data diversity.** At low data diversity, training is stable but the model fails to commit to either the hierarchical or linear rule. With moderate data diversity, training becomes unstable. As diversity increases further, the model commits to a rule and training stabilizes again.

the high-diversity regime, training stabilizes again as the model commits to the hierarchical rule, indicated by a high hierarchical generalization ratio.

Commitment to linear rule In Figure 4 (linear 99% case), we observe a strong preference for the linear rule when the training data is *mostly* linear-inducing. However, Figure 3 shows that when the training data is *exclusively* linear-inducing, models fail to consistently learn the linear rule. This failure is explained by data diversity: right-branching sentences lack syntactic variation, as the main auxiliary always follows the subject noun. This lack of syntax diversity prevents rule extrapolation. By introducing as little as 1% of center-embedded sentences (Figure 4, linear=99%), we introduce the diversity necessary to learn the linear rule.

To confirm that data diversity is the key ingredient to rule learning, we create variations of QF training data with 50K questions and 50K declarations, including 99% right-branching and 1% center-embedded sentences. We control the diversity of center-embedded sentences as before and use the proportion of runs achieving generalization accuracy below 5% to quantify the likelihood of committing to the linear rule. As shown in Figure 6 (right), we observe a similar U-shaped scaling behavior, confirming that models only commit to a rule when trained on diverse data.

7 DISCUSSION AND CONCLUSIONS

By exploring the role of data structure in determining OOD generalization rules, we have also revealed which settings allow us to predict model behavior. We show that complex grammatical structures guide models toward hierarchical rules, while mixed data compositions lead to unstable dynamics and inconsistent rule commitment. These findings emphasize the importance of understanding how data diversity shapes both stability and generalization in neural networks. Our findings have a number of implications across machine learning and even formal linguistics.

Clusters of generalization behavior across seeds While errors are often treated as Gaussian noise in the theoretical literature, our findings suggest that errors may only be distributed unimodally for a given compositional solution. Our work joins the growing literature that suggests random variation not only has an effect, but can create clusters of OOD behaviors. Previously, clustered distributions have been documented in text classification heuristics (Juneja et al., 2022) and training dynamics (Hu et al., 2023). In our case, we note that generalization accuracy is only clearly multimodally distributed when specifically considering stable training runs. We suggest that research on compositional variation in training consider training stability in the future.

Implications for formal linguistics Our findings have potential implications for linguistics debates about the poverty of the stimulus (McCoy et al., 2018; Berwick et al., 2011). Linguists have extensively studied the question of what data is necessary and sufficient to learn grammatical rules. In particular, Wexler (1980) argue that all English syntactic rules are learnable given "degree 2" data: sentences with only one embedded clause nested within another clause. Our mixed scoping results show that without a stronger architectural inductive bias—the very subject of the poverty of the stimulus debate—degree 1 data alone cannot induce a preference for hierarchical structure. However, our work also supports the position of Lightfoot (1989) that lower degree data is adequate for a child to learn a specific rule, as the LM generalizes ID degree 1 QF rule examples to OOD degree 2 by using the hierarchical inductive bias induced by declaration examples.

Grokking, instability, and latent structure Murty et al. (2023), exploring the same data setting we do, call the transition from linear generalization to hierarchical generalization rules during training *structural grokking*. Classic grokking (Power et al., 2022), however, is different: Rather than a transition between generalization rules, it describes a transition from memorization to generalization.

Our findings clarify both scenarios. We link structural grokking to the instability formed by competition between linear- and hierarchical-inducing training subsets. Without competing subsets, the model immediately learns either the linear or the hierarchical rule without the gradual transition of structural grokking. This instability could represent the same phenomenon of circuit competition described by Ahuja et al. (2024). We find a similar pattern of instability in our study of data diversity, with implications for classic grokking. In this case, the competition is not between two rules, but instead between memorized heuristics—sufficient for modeling syntactically homogeneous training data—and simple OOD rules—required to efficiently model diverse training data. Yet again, while a strict memorization regime is relatively stable, the regime between memorization and generalization is unstable, leading to potential grokking.

Our findings suggest that memorization is just another rule that the model can adopt when it is the simplest way of capturing the training distribution. Such a framework unifies the grokking literature with other phenomena such as emergence (Schaeffer et al., 2023) and benign interpolation (Theunissen et al., 2020).

ETHICS STATEMENT

This research does not present any direct ethical concerns. The work involves empirical studies of machine learning models and their behavior in language tasks. No human subjects, sensitive data, or high-stakes applications were involved in this research. Therefore, no specific ethical considerations were necessary for this work.

REPRODUCIBILITY STATEMENT

All relevant details regarding the experimental setup including model architecture, hyperparameters, and data preprocessing, are included in the main text (Section 3.3) and appendices (Section C.2). Additionally, the code and scripts used to run the experiments are provided in the supplementary material and will be made publicly available upon acceptance.

REFERENCES

Kabir Ahuja, Vidhisha Balachandran, Madhur Panwar, Tianxing He, Noah A Smith, Navin Goyal, and Yulia Tsvetkov. Learning syntax without planting trees: Understanding when and why transformers generalize hierarchically. *arXiv [cs.CL]*, April 2024.

Boaz Barak, Benjamin L Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: SGD learns parities near the computational limit. *arXiv [cs.LG]*, July 2022.

Ian Berlot-Attwell, Kumar Krishna Agrawal, A Michael Carrell, Yash Sharma, and Naomi Saphra. Attribute diversity determines the systematicity gap in VQA. *arXiv [cs.LG]*, November 2023.

Robert C Berwick, Paul Pietroski, Beracah Yankama, and Noam Chomsky. Poverty of the stimulus revisited. *Cogn. Sci.*, 35(7):1207–1242, September 2011.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. *arXiv [cs.CL]*, September 2023.

Noam Chomsky. *Aspects of the theory of syntax*. The MIT Press. MIT Press, London, England, 50 edition, 2015.

Leshem Choshen, Guy Hacohen, Daphna Weinshall, and Omri Abend. The grammar-learning trajectories of neural language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8281–8297, Stroudsburg, PA, USA, January 2022. Association for Computational Linguistics.

Stephen Crain and Mineharu Nakayama. Structure dependence in grammar formation. *Language (Baltim.)*, 63(3):522, September 1987.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D Sculley. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61, 2022.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv [cs.CL]*, February 2020.

Robert Frank and Donald Mathis. Transformational networks. *Models of Human Language Acquisition*, 22, 2007.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2 (11):665–673, November 2020.

Katherine L Hermann and Andrew K Lampinen. What shapes feature representations? exploring datasets, architectures, and training. *arXiv [cs.LG]*, June 2020.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North*, pp. 4129–4138, Stroudsburg, PA, USA, 2019. Association for Computational Linguistics.

Michael Y Hu, Angelica Chen, Naomi Saphra, and Kyunghyun Cho. Latent state models of training dynamics. *arXiv [cs.LG]*, August 2023.

Jeevesh Juneja, Rachit Bansal, Kyunghyun Cho, João Sedoc, and Naomi Saphra. Linear connectivity reveals generalization strategies. *arXiv [cs.LG]*, May 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv [cs.LG]*, December 2014.

David Lightfoot. The child's trigger experience: Degree-0 learnability. *Behavioral and brain sciences*, 12(2):321–334, 1989.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Trans. Assoc. Comput. Linguist.*, 4:521–535, December 2016.

Ziming Liu, Eric J Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. *arXiv [cs.LG]*, October 2022.

Brian MacWhinney. *The childes project: Tools for analyzing talk, volume II: The database*. Psychology Press, London, England, 3 edition, January 2014.

R Thomas McCoy, Robert Frank, and Tal Linzen. Revisiting the poverty of the stimulus: hierarchical generalization without a hierarchical bias in recurrent neural networks. *arXiv [cs.CL]*, February 2018.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv [cs.CL]*, February 2019.

R Thomas McCoy, Robert Frank, and Tal Linzen. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Trans. Assoc. Comput. Linguist.*, 8: 125–140, December 2020.

William Merrill, Nikolaos Tsilivis, and Aman Shukla. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. *arXiv [cs.LG]*, March 2023.

Aaron Mueller and Tal Linzen. How to plant trees in language models: Data and architectural effects on the emergence of syntactic inductive biases. *arXiv [cs.CL]*, May 2023.

Aaron Mueller, Robert Frank, Tal Linzen, Luheng Wang, and Sebastian Schuster. Coloring the blank slate: Pre-training imparts a hierarchical inductive bias to sequence-to-sequence models. In *Findings of the Association for Computational Linguistics: ACL 2022*, Stroudsburg, PA, USA, 2022. Association for Computational Linguistics.

Aaron Mueller, Albert Webson, Jackson Petty, and Tal Linzen. In-context learning generalizes, but not always robustly: The case of syntax. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4761–4779, Stroudsburg, PA, USA, 2024. Association for Computational Linguistics.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D Manning. Characterizing intrinsic compositionality in transformers with tree projections. *arXiv [cs.CL]*, November 2022.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher Manning. Grokking of hierarchical structure in vanilla transformers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Stroudsburg, PA, USA, 2023. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2340–2353, 2018.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv [cs.LG]*, January 2023.

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

Isabel Papadimitriou and Dan Jurafsky. Learning music helps you read: Using transfer to study linguistic structure in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA, 2020. Association for Computational Linguistics.

Isabel Papadimitriou and Dan Jurafsky. Injecting structural hints: Using language models to study inductive biases in language learning. *arXiv [cs.CL]*, April 2023.

Jackson Petty and Robert Frank. Transformers generalize linearly. *arXiv [cs.CL]*, September 2021.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv [cs.LG]*, January 2022.

Naomi Saphra and Adam Lopez. Understanding learning dynamics of language models with SVCCA. *arXiv [cs.CL]*, November 2018.

Naomi Saphra and Adam Lopez. Understanding learning dynamics of language models with. In *Proceedings of the 2019 Conference of the North*, pp. 3257–3267, Stroudsburg, PA, USA, January 2019. Association for Computational Linguistics.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *arXiv [cs.AI]*, April 2023.

Thibault Sellam, Steve Yadlowsky, Ian Tenney, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Raluca Turc, Jacob Eisenstein, Dipanjan Das, and Ellie Pavlick. The MultiBERTs: BERT reproductions for robustness analysis. In *International Conference on Learning Representations*, October 2021.

Marthinus Wilhelmus Theunissen, Marelie Davel, and Etienne Barnard. Benign interpolation of noise in deep learning. *S. Afr. Comput. J.*, 32(2), December 2020.

Kenneth Wexler. Formal principles of language acquisition, 1980.

Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18(6):1245–1262, December 1989.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, November 2020.

Xuekai Zhu, Yao Fu, Bowen Zhou, and Zhouhan Lin. Critical data size of language models from a grokking perspective. *arXiv [cs.CL]*, January 2024.

A RELATED WORK EXTENDED

A.1 SYNTAX AND HIERARCHICAL GENERALIZATION

While works mentioned in Section 2.1 focused on models trained from scratch, another line of research examined the inductive bias of pretrained models. Mueller et al. (2024); Mueller & Linzen (2023) pretrained transformers on text corpora such as Wikipedia and CHILDES (MacWhinney, 2014) before fine-tuning them on the question formation task. They found that exposure to large amounts of natural language data enables transformers to generalize hierarchically.

Instead of using the question formation task as a probe, Hewitt & Manning (2019); Murty et al. (2022) directly interpreted model's internal representation to understand whether transformers constrain their computations to to follow tree-structure patterns. Hewitt & Manning (2019) demonstrated that the syntax tress are embedded in model's representation space. Similarly, Murty et al. (2022) projects transformers into a tree-structured network, and showed that transformers become more tree-like over the course of training on language data.

A.2 RANDOM VARIATION

Specific training choices, such as hyperparameters, are crucial to model outcomes. However, even when controlling for these factors, training machine learning models remains inherently stochastic—models can be sensitive to random initialization and the order of training examples. Zhou et al. (2020); D'Amour et al. (2022); Naik et al. (2018) reported significant performance differences across model checkpoints on various analysis and stress test sets. Zhou et al. (2020) further found that instability extends throughout the training curve, not just in final outcomes. To investigate the source of this inconsistency, Dodge et al. (2020) compared the effects of weight initialization and data order, concluding that both factors contribute equally to variations in out-of-sample performance.

Similarly, Sellam et al. (2021) found that repeating the pre-training process on BERT models can result in significantly different performances on downstream tasks. To promote more robust experimental testing, they introduced a set of 25 BERT-BASE checkpoints to ensure that experimental conclusions are not influenced by artifacts, such as specific instances of the model. In this work, we also observe training inconsistencies across runs on OOD data, both during training and at convergence. Unlike prior studies that focus on implications of random variations on experimental design, we study the source of training inconsistencies and link these inconsistencies to simplicity bias and the characteristics of the training data.

A.3 SIMPLICITY BIAS

Models often favor simpler functions early in training, a phenomenon known as simplicity bias (Hermann & Lampinen, 2020), which is also common in LMs. Choshen et al. (2022) found that early LMs behave like n-gram models, and Saphra & Lopez (2019) observed that early LMs learn simplified versions of the language modeling task. McCoy et al. (2019) showed that even fully trained models can rely on simple heuristics, like lexical overlap, to perform well on Natural Language Inference (NLI) tasks. Chen et al. (2023) further explored the connection between training dynamics and simplicity bias, showing that simpler functions learned early on can continue to influence fully trained models, and mitigating this bias can have long-term effects on training outcomes.

Phase transitions have been identified as markers of shifts from simplistic heuristics to more complex model behavior, often triggered by the amount of training data or model size. In language models, Olsson et al. (2022) showed that the emergence of induction heads in autoregressive models is linked to handling longer context sizes and in-context learning. Similar phase transitions have been studied in non-language domains, such as algorithmic tasks (Power et al., 2022; Merrill et al., 2023) and arithmetic tasks (Nanda et al., 2023; Barak et al., 2022).

In the context of hierarchical generalization, Ahuja et al. (2024) used a Bayesian approach to analyze the simplicity of hierarchical versus linear rules in modeling English syntax. They argued that transformers favor the hierarchical rule because it is simpler than the linear rule. However, their model fails to explain (1) why learning the hierarchical rule is delayed (i.e., after learning the linear rule) and (2) why hierarchical generalization is inconsistent across runs. In this work, we offer a different perspective, showing that a model's simplicity bias towards either rule is driven by the characteristics of the training data.

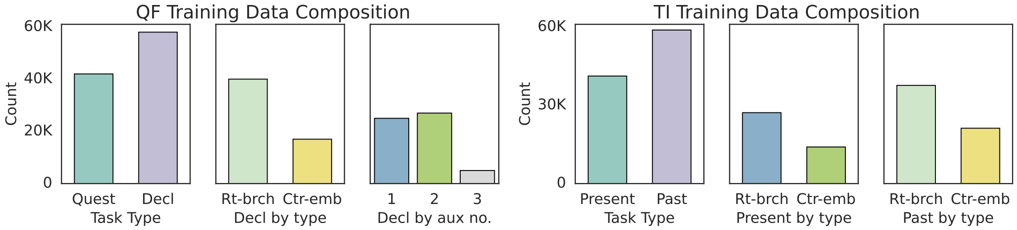

Figure 7: **Components of the original QF and TI training data.** *Left:* QF training data contains samples of two tasks types: question formation and declaration copying. We further break down samples in the declaration copying task by (1) branching type and (2) number of auxiliaries. *Right:* TI training data also contains samples of two task types: tense inflection and past tense copying. Both task contain center embedded and right branching sentences.

B TRAINING DATA SAMPLES

B.1 QUESTION FORMATION

When we mention "declarations," we are referring to the declaration copying task, and "questions" refer to the question formation task. Here are two examples randomly taken from the training data:

- Declaration Example: `our zebra doesn't applaud the unicorn . decl our zebra doesn't applaud the unicorn .`
- Question Example: `some unicorns do move . quest do some unicorns move ?`

Both tasks begin with an input declarative sentence, followed by a task indicator token (`decl` or `quest`), and end with the output. During training, the entire sequence is used in the causal language modeling objective. The in-distribution validation set and the OOD generalization set only contain question formation samples.

B.2 TENSE INFLECTION

- Past Example: `our peacocks above our walruses amused your zebras . PAST our peacocks above our walruses amused your zebras .`
- Present Example: `your unicorns that our xylophones comforted swam . PRESENT your unicorns that our xylophones comfort swim .`

The tense inflection task is indicate by the `PRESENT` token, and in Section 4.3, we only used tense inflection samples during training. In Appendix E.1, we further explore the use of a secondary copying task to achieve OOD generalization. Similar to the question formation training data, the secondary task only requires repeating the given sentence, which is always in the past tense, and the copying task is marked by the `PRESENT` token.

C ADDITIONAL RESULTS ON QUESTION FORMATION

C.1 RELATIVE CLAUSES ALONE CANNOT INDUCE HIERARCHICAL GENERALIZATION

Center embedding imposes two constraints on syntax: (1) the sentence must include a relative clause, and (2) this clause must modify the subject. In contrast, right-branching sentences may or may not include a relative clause. The experiments from Section 4.2 indicate that center-embedded sentences encourage the model to learn the hierarchical rule and generalize OOD. However, we have not ruled out the possibility that other data partitions could also induce hierarchical generalization. Specifically, the previous experiment did not control for the number of relative clauses.

In Figure 7 *left*, we show the breakdown of the question formation training data by different partitions. We now partition declarations based on the number of relative clauses they contain, creating three datasets, each with declarations limited to a specific number of relative clauses. Additionally, we conduct experiments on subsets where two of the three partitions are mixed, excluding the third.

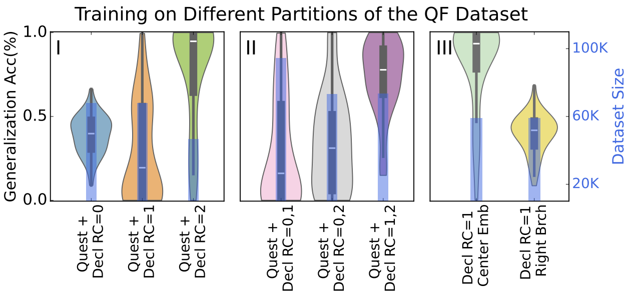

Figure 8: **Additional Experiment Control on QF Data** *Panel I&II:* Partitioning declaration sentences by number of relative clauses, we see higher random variation among models trained on sentences with only one added clause, and similarly high random variation among models trained on a mix different clause counts. *Panel III:* Sentences with one relative clause can be further partitioned based on whether the clause modifies the subject (center embedded) or the object (right branching). This additional evidence indicates that relative clause alone is not sufficient to induce hierarchical generalization.

Results, shown in Figure 8 (I and II), indicate that models trained on declarations without relative clauses fail to generalize hierarchically, while those trained on sentences with two relative clauses are more likely to succeed. This observation suggests an alternative hypothesis: that the number of relative clauses determines sentence complexity, with more complex sentences (i.e., those with additional relative clauses) encouraging the model to learn a tree-like syntactic representation.

However, experiments with declaration sentences containing one relative clause reveal that this alternative hypothesis is incorrect. With one relative clause, the clause either modifies the subject (center embedding) or the object (right branching). We retain samples with one relative clause in the declaration-copying task and further partition them by center embedding versus right branching. The distinct generalization behaviors, shown in Figure 8 (III), indicate that when controlling for relative clause count, only center-embedded sentences induce the hierarchical rule. For sentences with two relative clauses, both the subject and object have clause modifiers, resulting in consistent center embedding. This explains why the number of relative clauses may initially appear to affect generalization behaviors.

C.2 VARYING DATA RATIOS

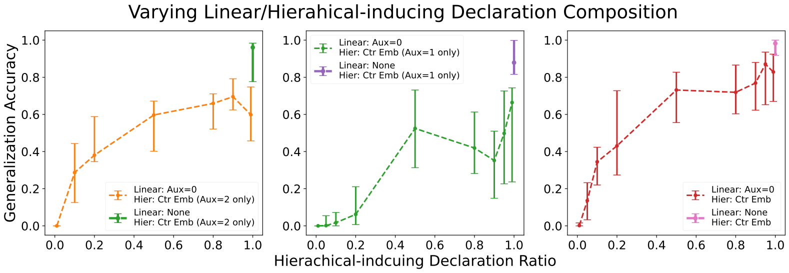

Figure 9: **Hierarchical generalization is sensitive to data compositions.** Datasets are standardized to contain 50K question formation samples and 50K declaration samples. For declaration samples, we experiment mixing different proportions of hierarchical-inducting declarations (center embedded sentences) and linear-inducing declarations (right branching sentences).

Data Composition Details We construct variations of the training data using the following procedure. Each new dataset contains 50K questions (reused from the original data) and 50K declarations, where we control the ratio between hierarchical-inducing and linear-inducing sentences. These datasets are used for the experiments in Section 5.2. To generate additional declarations, we keep the distribution of the unique syntax structures in original dataset. Specifically, for each sentence in the original data, we extract the syntax tree using the CGF rules and resample words from the vocabulary to create new sentence samples.

Additional Results We use the five datasets above to examine how different mix ratios affect a model's preference towards the hierarchical generalization. The median generalization accuracy, along with error bars representing the 35th and 65th percentiles, is shown in Figure 9. Including as little as 1% of linear-inducing declarations significantly reduces the model's likelihood of generalizing hierarchically. This effect is particularly pronounced when hierarchical-inducing sentences lack syntactic diversity, such as when only right branching sentences with no auxiliary (right panel) or right branching sentences with one auxiliary (middle panel) are present. Conversely, when the dataset is predominantly linear-inducing declarations, models consistently achieve 0% generalization accuracy, indicating a strong preference for the linear rule across all training runs.

D TRAINING INSTABILITY

In Figure 10, we visualize the training dynamics for 30 independent runs when trained on the original QF data. Each run differs in both model initialization and data order. Notice that the training dynamics for runs exhibit grokking behaviors: OOD generalization is delayed when compared to training loss convergence and validation performance convergence. These runs share a similar progression in training loss, validation accuracy, and generalization accuracy up until moment when the training loss converges. Interestingly, after convergence on training loss, all runs reach 0% on the generalization set, indicating that the model strictly prefers linear rules on OOD data. After that, models start to achieve non-trivial performance in generalization accuracy. However, for many runs the generalization accuracy does not increase monotonically. Instead, we observe massive swings in generalization accuracy during this training period as well as large inconsistency across different seeds. Overall, training is *always* stable for ID data while the performance for OOD data is inconsistent across seeds. We visualize runs with different of total variation values in Figure 11.

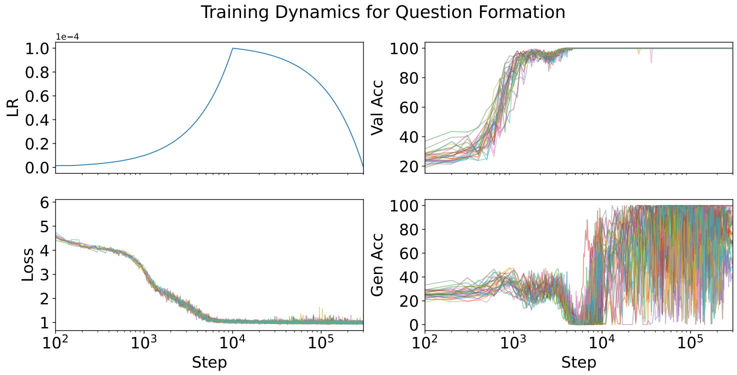

Figure 10: **Training Dynamics on original question formation data.** Training loss and in-distribution validation accuracy is stable during training and consistent across random seeds. In contrast, the model's performance on OOD data is both unstable during training and inconsistent across seeds. The instability and inconsistency is most prominent during grokking (i.e., when training loss has converged).

E ADDITIONAL RESULTS ON TENSE INFLECTION

E.1 LEVERAGING A SECONDARY TASK FOR OOD GENERALIZATION

In the original of TI training data (McCoy et al., 2020), a secondary task is also included to mimic the question formation training data. In this secondary task, instead of transforming a sentence from the past tense to the present tense, the model simply needs to repeat it. For concrete examples, see Appendix B. In experiments conducted in Section 3.2, we have eliminated the used of this secondary task because center embedded sentences can be included in tense inflection training samples without violating the ambiguity requirement. In this section, we will first confirm that for the training data originally proposed by McCoy et al. (2020), the use of secondary task is indeed not necessary. In addition, we will show an additional experiment where we can again leverage the secondary task to induce OOD generalization in TI.

Experiments on the original TI data Figure 7 (*right*) shows a breakdown of different components of the original TI training data. We first remove all the past-tense-copying samples from the data and

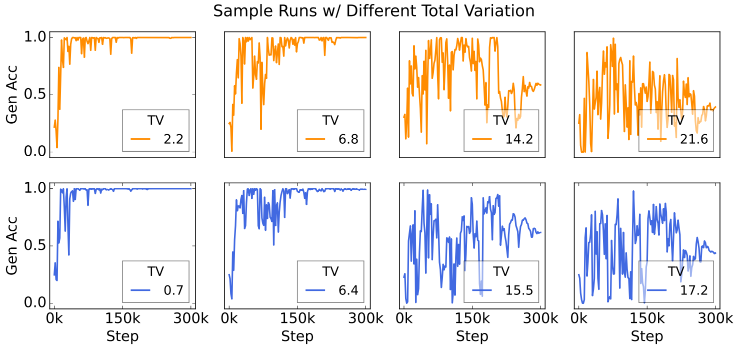

Figure 11: **Each training run either stabilizes in a simple OOD generalization rule or oscillates in its OOD accuracy.** The OOD generalization behaviors can be either stable or unstable when trained on different seeds. We use total variation to quantify the instability within one training run.

train models on the tense-inflection task only. Results of experiment (a) and (b), shown in Table 2, indicate that the past-tense-copying is not necessary to induce hierarchical generalization.

Leveraging a secondary task To further examine how different components in the original data impact the model's generalization behaviors, we partition each of the two tasks of the training data into right branching samples and center embedded samples, and train models on different combinations of those subsets. Result for experiment (c) in Table 2 indicates that only training on center embedded TI samples induces the hierarchical rule for center embedded sentences. However, the model is not able to correctly perform the tense inflection task on right branching sentences regardless of their ambiguity. However, in experiment (d) we include right branching sentences in the copy task, and the model achieves 100% accuracy on ambiguous right branching TI samples and achieves a non-trivial accuracy on unambiguous right branching TI samples. First, experiment (d) and (e) confirm again that center embedded sentences induce the preference for the hierarchical rule. Additionally, they show that by exposing the model to right branching sentence through a secondary task, the model is able to generalize the hierarchical rule to those sentences as well.

Results from experiment (e) and (f) forms a contrast to experiment (d) and (e). In experiment (e), we train models only on right branching TI samples, and the model prefers the linear rule on unambiguous right branching sentences. It also fails to generalize to center embedded sentences regardless of their ambiguity, achieving a generalization accuracy of 0%. If we include center embedded sentences in the copy task, the model still fails to learn the correct verb inflection even on ambiguous center embedded sentences.

Table 2: **Training on different partitions of the original TI data.** The past-tense-copying task exposes the model to both sentence types during training. However, the model's ability to learn the hierarchical rule is exclusively driven by center-embedded tense-inflection samples. We report the median accuracy and the interquartile range from 20 random seeds.

	Data Composition				**Model Performance (%)**			
	TI samples		COPY samples		ID(ambiguous)		OOD(unambiguous)	
Expt	Rt Brch	Ctr Emb	Rt Brch	Ctr Emb	Rt Brch	Ctr Emb	Rt Brch	Ctr Emb
(a)	✓	✓	✓	✓	100 (0.0)	100 (0.0)	64.2 (13.3)	74.2 (18.3)
(b)	✓	✓	✗	✗	100 (0.0)	100 (0.0)	62.2 (10.8)	79.6 (17.4)
(c)	✗	✓	✗	✗	0.0 (2.4)	100 (0.0)	0.0 (0.0)	77.5 (15.9)
(d)	✗	✓	✓	✗	100 (0.0)	100 (0.0)	60.0 (12.7)	85.6 (16.8)
(e)	✓	✗	✗	✗	100 (0.0)	0.0 (2.4)	0.0 (0.0)	0.0 (0.0)
(f)	✓	✗	✗	✓	100 (0.0)	41.6 (9.4)	0.5 (0.8)	0.1 (0.3)

E.2 TRAINING INSTABILITY AND RULE COMMITMENT FOR TENSE INFLECTION

We repeat the same total variation analysis in Section 5 for the tense inflection task. We use the data mixes from Section 4.3. Specifically, we include only tense inflection samples and vary the ratio between right branching and center embedded sentences. In Section 4.3, we have already concluded that the hierarchical rule is *always* preferred for center embedded sentences regardless of data mixes. For this reason, we are interested in examining the rule preference and training stability for unambiguous right branching sentences. In Figure 12 we visualize the relationship between total variation and the final generalization accuracy on unambiguous right branching sentences. The qualitative behavior is similar to what we have observed in QF (Section 5.2).

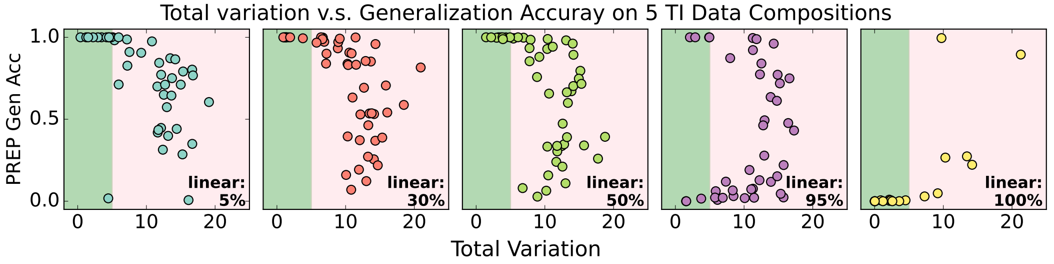

Figure 12: **Total Variation v.s. final generalization accuracy for TI task.** Similar to Figure 4, we observe the same horseshoe shaped behavior between training stability and final generalization accuracy for the TI task.

F DATA DIVERSITY AND MEMORIZATION PATTERNS

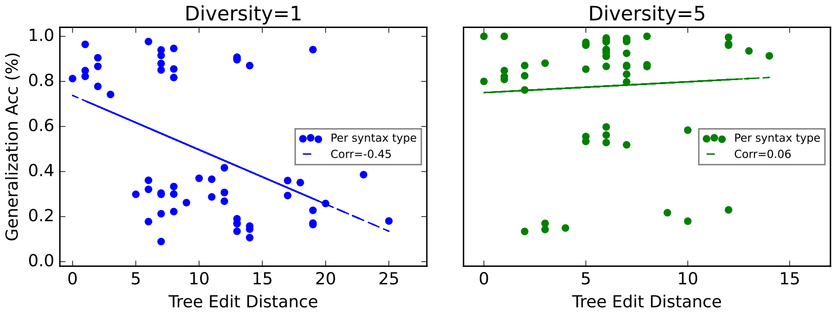

Figure 13: **OOD generalization v.s. syntax similarity to training data.** At low data diversity, model memorizes syntax patterns and applies the hierarchical rule only syntax structures similar to ones in the training data. With higher data diversity, model extrapolates rules and can apply the hierarchical rule even to unseen syntax structures that are dissimilar to training data.

We investigate model behavior when trained on data with limited diversity. By analyzing a model's generalization accuracy across different syntactic types, we aim to distinguish patterns indicative of either memorization or generalization.

Measuring data similarity In Section 6.1, we measured data diversity by counting unique syntax tree types. Building on this, we now use Tree-Edit Distance (TED) as a measure of sentence similarity. As before, we first construct syntax trees using CFG rules, then calculate TED using the Zhang-Shasha Tree-Edit Distance algorithm Zhang & Shasha (1989). We define TED=0 for sentences that share the same syntax structure but differ only in vocabulary. This similarity measure allows us to quantify, for each sample in the OOD generalization set, the closest matching sentence type in the training data. In the memorization regime, where the model encounters only a few syntax types, we suspect it cannot extrapolate rules to syntactically distinct OOD sentences. In contrast, with a more diverse syntax exposure, rule extrapolation may enable the model to apply rules even to OOD sentence types.

Experiment To verify our intuition about memorization and generalization, we train models on two variations of the QF data. In the first variation, the declaration-copying task has data diversity set to 1, meaning only one syntax type appears, and we specifically choose one with center embedding. In the second variation, the declaration-copying task has diversity set to 5, with all 5 types containing center embeddings. For both datasets, the question-formation task remains unchanged, consisting solely of right-branching sentences. For the diversity=1 dataset, we calculate TED for each unique syntax type in the OOD set against the single syntax type in the declaration-copying task. For the diversity=5 dataset, we compute TED between each OOD sample and the five syntax types in the declaration-copying task, taking the minimum. This TED score provides a measure of similarity between the OOD samples and those encountered during training. Our goal is to determine, based on training with these datasets, which OOD syntax types the model applies the hierarchical rule to.

Result In Figure 13, we visualize the final generalization accuracy for each OOD syntax type against its TED relative to the training data. When trained on low-diversity data (Figure 13, *left*), generalization accuracy is negatively correlated with TED. For syntax types seen in the declaration-copying task (TED=0) and those similar to it, the model applies the hierarchical rule. However, for syntax types with high TED, the model's behavior is random (25%), indicating failure to follow any rule. As data diversity increases slightly (Figure 13, *right*), generalization accuracy no longer correlates with TED, suggesting that once the model begins to extrapolate the hierarchical rule, it can apply this rule to a wider range of OOD syntax types.