Sometimes I am a Tree: Data Drives Unstable Hierarchical Generalization in LMs

Tian Qin
Harvard University
tqin@g.harvard.edu

Naomi Saphra
Harvard University
nsaphra@fas.harvard.edu

David Alvarez-Melis
Harvard University & MSR
dam@seas.harvard.edu

Abstract

Early in training, LMs can behave like n-gram models, but eventually, they often learn tree-based syntactic rules and generalize hierarchically out of distribution (OOD). We study this shift using controlled grammar-learning tasks: question formation and tense inflection. We find a model learns to generalize hierarchically if its training data is *complex*—in particular, if it includes center-embedded clauses, a special syntactic structure. Under this definition, complex data drives hierarchical rules, while less complex data encourages shortcut learning in the form of n-gram-like linear rules. Furthermore, we find that a model uses rules to generalize, whether hierarchical or linear, if its training data is *diverse*—in particular, if it includes many distinct syntax trees in the training set. Under this definition, diverse data promotes stable rule learning, whereas less diverse data promotes memorization of individual syntactic sequences. Finally, intermediate diversity and intermediate complexity form an *unstable regime*, which is characterized by oscillatory learning dynamics and inconsistent behaviors across random seeds. These results highlight the central role of training data in shaping generalization and explain why competing strategies can lead to unstable outcomes.

1 Introduction

Early in training, LMs can behave like n-gram models, relying on local heuristics without capturing the deeper structure of language (Choshen et al., 2022; Geirhos et al., 2020; Saphra and Lopez, 2018). However, they also exhibit breakthroughs in generalization by suddenly shifting into more sophisticated behaviors (Choshen et al., 2022; Chen et al., 2023; McCoy et al., 2020a). While previous works attribute these advanced capabilities to architecture and training objectives (Ahuja et al., 2024; McCoy et al., 2020a), we investigate how two key *data* characteristics—complexity and diversity—influence training outcomes.[1]

To study when models favor latent structures over surface heuristics, we focus on two controlled grammar-learning tasks: question formation and tense inflection (McCoy et al., 2020b). Each task can be solved either by a **linear rule**, which corresponds to an n-gram–like heuristic that operates on the most recent noun or auxiliary, or by the **hierarchical rule**, which reflects the correct syntactic structure used to generate the data. In the training and in-distribution (ID) data, sentences are constructed to be **ambiguous**, so both rules give the correct answer. In the out-of-distribution (OOD) data, this *ambiguity is removed*, and only the hierarchical rule succeeds.

We illustrate the differences between rules in Fig. 1, where the model inflects the main verb of a sentence to agree with a subject's noun. In the n-gram-like linear rule (*bottom right*), the model uses the nearest noun as the subject and fails when a distractor noun appears in a prepositional phrase. In the hierarchical case *(top right*), the model uses a latent tree structure to correctly identify the subject and apply, enabling generalization to any grammatical sentence. Prior work shows that transformer-based LMs can shift from the linear heuristic to the hierarchical rule when trained long enough, a transition called **structural grokking** (Murty et al., 2023), in parallel to the well-known shift from memorization to generalization in classic grokking (Power et al., 2022).

Building on this line of work (McCoy et al., 2018, 2020a; Ahuja et al., 2024; Murty et al., 2023), we study how training *data* properties shape whether models adopt the hierarchical rule or default to the linear rule. We define **data complexity** as the presence of center-embedded clauses, a syntactic structure where the subject is interrupted by

[1]Code can be found at https://github.com/sunnytqin/hier_gen.git.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 11734–11752
November 4-9, 2025

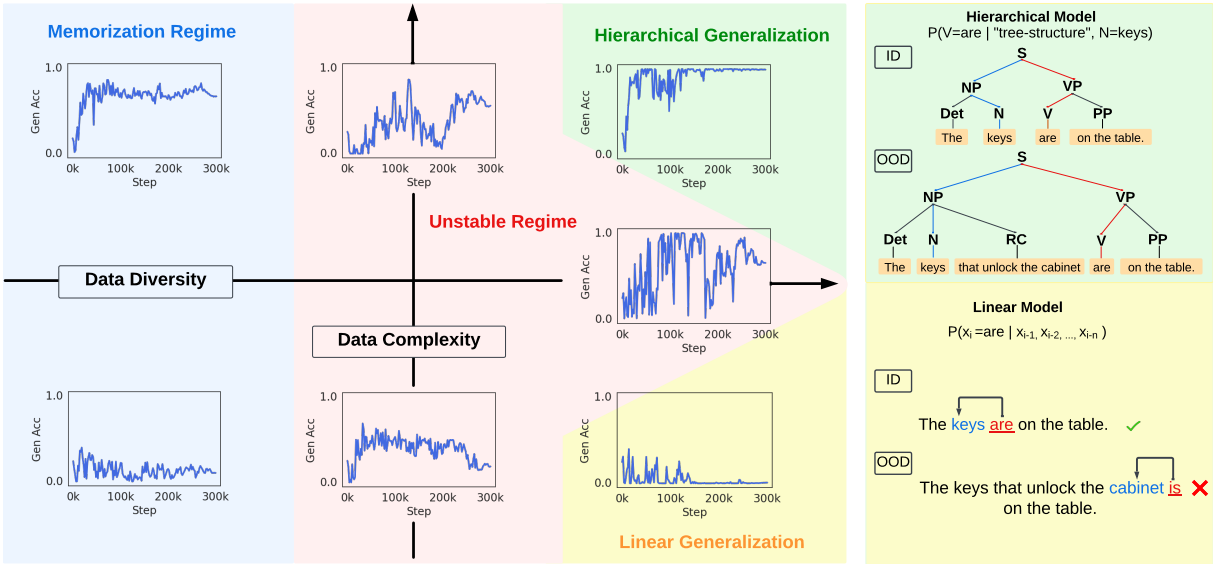

Figure 1: **Data plays a critical role in generalization behaviors and training stability.** *Left:* Along the data diversity x-axis, low data diversity (as measured by variation in syntactic structure) leads the model to memorize unreliable sample-specific patterns, whereas high data diversity promotes commitment to a general rule. Along the data complexity y-axis, high data complexity (as measured by the proportion of center-embedded sentences) 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 of syntax 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.

a relative clause. Our experiments show that such complex structures are the key driver of hierarchical OOD generalization; models trained without them consistently fall back on shortcuts. This finding echoes a classic claim in linguistics (Wexler, 1980) that center embeddings play a central role in human syntax acquisition.

After identifying center embeddings as the driver of hierarchical generalization, we next ask how data composition influences rule competition during training. Ahuja et al. (2024) showed that linear and hierarchical rules can coexist and compete during learning. We show that the outcome of this competition depends on the training data. While in-distribution accuracy is always stable across time and consistent across seeds, OOD accuracy is often *unstable* during training and *inconsistent* across seeds. The two rules compete, and only runs that fully commit to one rule—linear or hierarchical—exhibit stable OOD performance. These commitments are determined by two data properties: **complexity** (presence of center embeddings) and **diversity** (number of distinct syntax trees). We find that commitment consistently occurs only under both high complexity and high diversity, leading to stable generalization. At intermediate levels of either property, models fail to commit, resulting in unstable training and inconsistent outcomes across seeds. As summarized in Fig. 1, diversity promotes general rules over memorization, while complexity determines which rule is learned.

Our controlled synthetic settings allows us to precisely manipulate data properties and isolate their effects on rule learning. The resulting insights extend beyond grammar learning: they highlight how data complexity and diversity govern whether models rely on memorization, heuristics, or systematic generalization. These dynamics connect to broader themes in the study of grokking and training instability (Power et al., 2022; Ahuja et al., 2024), random variation between training runs (Juneja et al., 2022; Hu et al., 2023), and phase transitions in neural networks (Schaeffer et al., 2023; Theunissen et al., 2020). The mechanisms we identify in a simplified domain may inform our understanding of generalization in larger-scale models. Our contribution can be summarized as follows:

- We show that data complexity—that is, center embedding frequency—drives models to adopt hierarchical syntax rather than n-gram heuristics (Section 3).
- We find that models only achieve stable OOD

performance when they commit to a general rule, whether linear or hierarchical. Competing rules lead to unstable training and inconsistent outcomes across seeds (Section 4).

- We show that data diversity separates memorization, instability, and generalization regimes (Section 5).

2 Experimental Setup

The question formation and tense inflection tasks were first proposed by Frank and Mathis (2007) and Linzen et al. (2016) as canonical tests of language modeling ability. We use existing synthetic datasets for question formation from McCoy et al. (2018) and tense inflection from McCoy et al. (2020a).

2.1 Question Formation Task

In the **question formation (QF)** task, the model transforms a declarative sentence into a question by moving the main auxiliary verb (such as *does* in *does move*) to the front. Our training data (based on McCoy et al. (2018)) permits two strategies for choosing which verb to move: (1) a linear rule that moves the first auxiliary verb (Fig. 1 *top right*), or (2) a hierarchical rule—the correct rule in English grammar—based on the sentence's syntax tree (Fig. 1 *bottom right*), which places the main auxiliary verb above other verbs.

Examples of each rule are provided in Table 1. The first example is considered *ambiguous* because the hierarchical and linear rules produce the same correct outcome. In contrast, the second example is *unambiguous* because only the hierarchical rule produces the correct outcome. The training and in-distribution test data contain only ambiguous samples, while the OOD generalization set includes only unambiguous samples. Therefore, if a model uses the hierarchical rule, it will achieve 100% accuracy on both the in-distribution (ambiguous questions) and OOD (unambiguous questions) sets. Conversely, if a model uses the linear rule, it will still score 100% accuracy on the in-distribution set, but 0% on the OOD set. We therefore use the model's accuracy on the OOD set to measure hierarchical generalization.

2.2 Tense Inflection Task

In the **tense inflection (TI)** task, the model transforms a past-tense sentence into the present tense by changing the inflection of its main verb. In English, past-tense verbs (*walked*) are not inflected by plurality, so the model must identify the subject and use its plurality to inflect the present-tense verb (*walk*/*walks*). The TI training data again follows either a hierarchical or linear rule for subject-verb agreement (based on McCoy et al. (2020a)). The linear rule inflects the verb based on the most recent noun, while the hierarchical rule correctly inflects according to the subject. As in the QF task, the training and ID validation sets contain ambiguous examples, whereas the OOD set contains unambiguous examples. In the ambiguous example from Table 1, the subject noun *zebra* and the most recent noun *peacock* share the same plurality, so either rule produces the correct answer. In the OOD unambiguous example, 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.

2.3 Models, Data and Training

We run all experiments on the same 50 random seeds using hyperparameter settings from the existing literature (Ahuja et al., 2024; Murty et al., 2023). We use a decoder-only Transformer architecture where each layer has 8 heads with a 512-dimensional embedding. QF models have 6 layers and TI models have 4 layers. All models are trained from scratch on a causal language modeling objective for 300K steps. We use the Adam optimizer (Kingma and Ba, 2014), a learning rate of 1e-4, and a linear decay schedule. We use a word-level tokenizer with a vocabulary of size 72.

We use the original training, validation and OOD test data proposed by McCoy et al. (2018) and McCoy et al. (2020a). Where we expand the training data for our data composition experiments, we mimic the data generation process used for the original QF and TI task. Specifically, the original TI and QF data are generated with Context-Free Grammars (CFGs) using a simplified set of grammatical rules; we reuse the same CFG rules to create variations of the training data.

3 Data Complexity Determines the Rule

We begin by examining how the complexity of training data influences the rules that models learn. Our focus is on center-embedded sentences, a syntactic structure that has long been used in linguistics to characterize complexity. By ablating center

Table 1: **Examples from two grammar case studies.** *Top*: In the question formation task, the model moves the main auxiliary verb to the front to form a question. *Bottom*: In the tense inflection task, the model inflects the main verb from past to present tense, while respecting subject-verb agreement.

Dataset	Rule Type	Examples
Question Formation	Ambiguous	**Input:** My unicorn does move the dogs that do wait. **Output:** Does my unicorn move the dogs that do wait?
	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	Ambiguous	**Input:** My zebra behind the peacock smiled. **Output:** My zebra behind the peacock smiles.
	Unambiguous	**Input:** My zebra behind the peacocks smiled. **Linear output:** My zebra behind the peacocks smile. **Hierarchical output:** My zebra behind the peacocks smiles.

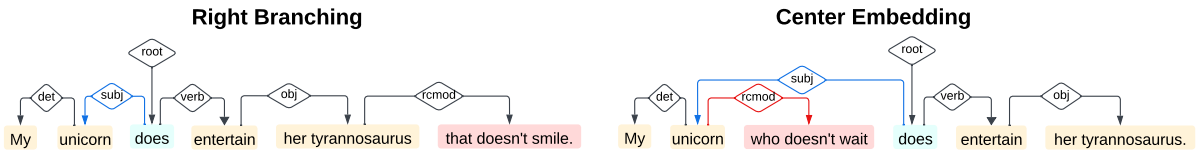

Figure 2: **Sentence Examples.** *Left:* Right-branching sentence example. The linear progression of the main constituent is not interrupted by the relative clause. *Right:* Center-embedded sentence example. When the relative clause modifies the subject, it interrupts the linear progression of the main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

embeddings from training data, we test how data complexity shapes which rule a model learns.

3.1 Center Embedding

Center embedding occurs when a clause is placed recursively within another clause of the same type. Fig. 2 illustrates two examples of center-embedded sentences, where the embedded clause complicates syntactic parsing by placing an additional subject noun in between a verb and its own subject.

Without center embeddings, English syntax only branches recursively to the right. In exclusively right-branching structures, modifying clauses are always appended at the end of the main clause, maintaining linear flow. Therefore, linguists have long argued that center embeddings play a crucial role in grammar acquisition (Wexler, 1980) and lead to tree-like syntactic structures (Chomsky, 2015). We find center embeddings, which are crucial for human language acquisition, also lead an LM to prefer hierarchical grammar rules.

Linguistic theory explains that center embeddings lead to hierarchical rule learning because of their greater computational requirements. To predict the next token, LMs must track syntactic connections between words in the context. In right-branching sentences, LMs can rely on linear proximity to identify these connections; as shown in Fig. 2, a simple bigram model suffices to capture the subject-verb relationship for such sentences. In contrast, center embeddings introduce relative clauses of various lengths, making linear n-gram models inefficient for capturing subject-verb relationships. Because center embeddings are recursive, they require the model to track multiple subject-verb relationships: one for the main clause and a separate one for the embedded relative clause. In these cases, a tree structure is more efficient than a linear one to model subject-verb relationships.

3.2 Question Formation Results

In the QF task (Section 2.1), the training data must remain ambiguous between the linear rule (moving the first auxiliary) and the hierarchical rule (moving the main auxiliary). Because center-embedded sentences are not ambiguous, they cannot appear in QF training examples. To expose the model to such structures, McCoy et al. (2018) introduced a secondary task: declaration copying. Like QF, it begins with a declarative sentence, but instead of transforming it, the model simply repeats it. Only the primary QF task enforces ambiguity, so declaration-copying examples may include center embeddings. In our first set of experiments, we modify the composition of the declaration-copying subset to greate three new training sets: *Quest*

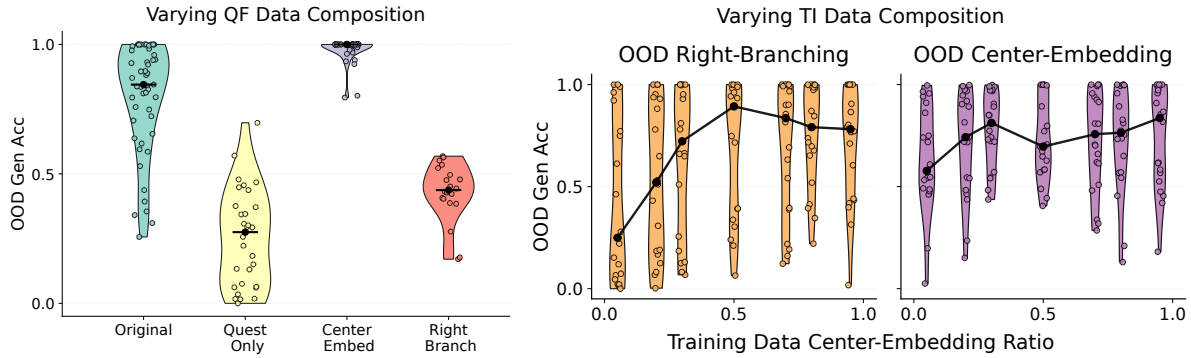

Figure 3: **Components of training data drive different generalization behaviors.** Each dot represents one model trained from a different random seed. *Left:* In QF, center-embedded sentences appear only in declaration-copying data; exposure to these examples induces hierarchical generalization. *Right:* In TI, we vary the mix of right-branching and center-embedded sentences and evaluate OOD performance on right-branching (*orange*) vs. center-embedded (*purple*).

Only (no declaration-copying examples), *Center Embed* (only center-embedded declarations), and *Right Branch* (only right-branching declarations). In all cases, the full set of QF training examples is retained. We train models using 50 random seeds.

Regardless of training set composition, all models achieve 100% accuracy on the in-distribution test data. However, their OOD performance differs sharply (Fig. 3, left). Models trained without any center-embedded declaration-copying examples universally (across all 50 seeds) fail to learn the hierarchical rule, even when right-branching declarations are included. By contrast, models trained with only center-embedded declaration-copying examples strongly favor the hierarchical rule. These results demonstrate that exposure to center-embedded sentences is essential for inducing hierarchical generalization.

3.3 Tense Inflection Results

In the TI task (Section 2.2), both right-branching and center-embedded sentences are ambiguous as long as the verb's subject shares the same plurality as the distractor noun between the subject and verb. For right-branching sentences, the distractor occurs in a prepositional phrase; for center-embedded sentences, it occurs in a relative clause. We show two concrete examples below.

1. **Right Branching**: The noun (*cabinet*) in the prepositional phrase (*to the cabinet*) acts as the distractor for the TI task.
   ID: *The keys to the* ***cabinets*** *are here.*
   OOD: *The keys to the* ***cabinet*** *are here.*
2. **Center Embedding**: Either the subject or the object (*cabinet*) in the relative clause (*that unlock the cabinet*) acts as the distractor.
   ID: *The keys that unlock the* ***cabinets*** *are here.*
   OOD: *The keys that unlock the* ***cabinet*** *are here.*

To test the effect of center embeddings on rule learning, we create training sets that vary the ratio of right-branching to center-embedded sentences from 5% to 95%, while keeping the total number of samples fixed.[2] For evaluation, we split the original OOD test set (McCoy et al., 2020a) into right-branching and center-embedded subsets. This split ensures we can separately measure how well models generalize on each kind of tree.

All seeds achieve perfect accuracy on the in-distribution test set, but OOD performance depends strongly on the proportion of center-embedded sentences (Fig. 3, right). On the least complex training dataset, where only 5% of sentences have center embeddings, most seeds achieve below-random performance on the right-branching OOD subset and around random on the center-embedded subset. As the proportion of center embeddings increases (up to 50%), the majority of seeds succeed on both OOD subsets, indicating that models have learned the hierarchical rule. These findings confirm that increasing data complexity through center embeddings promotes hierarchical generalization. In Appendix C, we further show that different subtypes of center embedding vary in their effect on TI rules.

4 Rules stabilize training

Why do some runs fail to reliably generalize hierarchically even when trained on hierarchy-inducing

[2]The original dataset also included a secondary past-tense copying task, parallel to declaration copying in QF. We show in App. E.1 that this task is not necessary and omit it here.

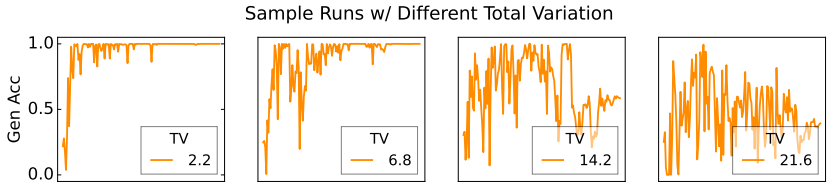

Figure 4: **Each training run either stabilizes in a simple OOD generalization rule or oscillates in its OOD accuracy.** The OOD generalization behaviors can be either stable or unstable when trained on different seeds. We use total variation to quantify the instability within one training run.

data? This section will show that these poor runs are oscillating between competing rules; models only stabilize OOD if they commit to a general rule, whether hierarchical or linear. Under our training conditions, some random seeds fail to stabilize regardless of training set composition.

4.1 Instability During Training

Across all our model runs, training loss and ID performance are both consistent across random seeds and stable during training (See F for example training curves). Despite stable ID accuracy during training, some random seeds lead to highly unstable OOD accuracy accuracy oscillating during training. We measure OOD instability across training time using a standard metric from signal processing: **total variation** (TV). Specifically, we checkpoint the model every 2K steps and measure the generalization accuracy Acc_t at each checkpoint timestep $t \in T$. TV is defined as:

$$\text{TV} = \frac{1}{|T|} \sum_{t \in T} |\text{Acc}_t - \text{Acc}_{t-1}| \qquad (1)$$

In Figure 4, we show the OOD performance of four runs trained on different seeds - ranging from highly stable OOD performance to high unstable ones. We also report corresponding TV to demonstrate that TV quantifies training instability.

4.2 Stability Ties to Rule Commitment

Why do some runs exhibit stable OOD performances while others oscillate wildly? We now demonstrate that regardless of training data mix, training runs with stable OOD behavior always commit to a universal systematic rule.

Setup: We construct five variations of the training data using the following procedure. Each new dataset contains 50K questions from the original data and 50K declarations with a controllable ratio between center-embedded (hierarchy-inducing) and right-branching (linearity-inducing) sentences. Our newly generated declarations maintain the original dataset's distribution of unique syntax trees. Specifically, for each sentence in the original data, we keep its CFG-based syntax tree but resample words from the CFG's vocabulary. We then train models from 50 random seeds using the hyperparameters from Section 2.3. For each seed, we report model OOD performance and TV in Figure 5.

Results: As shown in Figure 5, regardless of the data mix, the final OOD generalization accuracy for all stable models is either 100% or 0%—that is, they all commit to a universal systematic rule. While either rule can be implemented by a stable model, training data composition determines the likelihood of training outcomes—whether a run stabilizes and whether its systematic rule is hierarchical or linear. As seen previously, if the training data is dominated by either center-embedded or right-branching sentences, the resulting models usually commit to the hierarchical or linear rule, respectively. However, when the data is heterogeneous (e.g., 10% of examples are linearity-inducing right-branching sequences), the final generalization accuracy of stable runs is bimodally distributed across random seeds, clustering around 100% or 0%. In fact, the horseshoe-shaped curves in Figure 5 illustrate that the less stable a QF training run is, the less universally the model applies its OOD rules. (See Appendix E.2 for matching results on the TI task.)

In summary, by mixing linear-inducing and hierarchical-inducing data, we create heterogeneous training data mixture. When training on heterogeneous data mixtures, more seeds lead to unstable runs. However, even with heterogeneous mixes, some runs can still stabilize by committing to one of the competing rules.

5 Data Diversity Leads to General Rules

In the previous section, we showed that training runs stabilize if the model commits to a system-

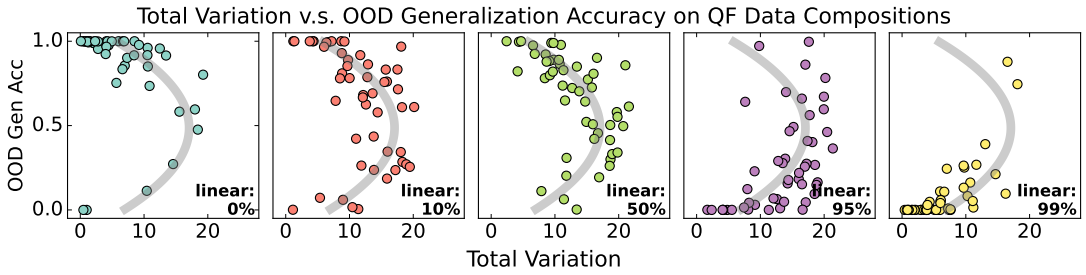

Figure 5: **Training stability vs. final generalization accuracy for QF task.** We create heterogeneous training data by mixing different proportions of linear-inducing data and hierarchical-inducing data. For each data mix, we train models on 50 random seeds and for each seed, we examine training instability (x-axis) and final OOD generalization performance (y-axis). Grey line indicates the smoothed average curve across all five datasets.

atic rule. We now extend this finding by showing that data diversity promotes commitment to these systematic universal rules. We find that training can stabilize in two ways: with *less* diverse data, models stabilize by memorizing training examples, while with *more* diverse data, they stabilize by committing to systematic rules (studied in the previous section). At intermediate diversity levels—as at intermediate complexity levels (Section 4.2)—training becomes unstable.

5.1 Measuring Data Diversity

We define a dataset's diversity according to the syntactic similarity between its sentences. We measure a sentence pair's similarity by the tree-edit distance (TED) between their respective latent syntax trees (Chomsky, 2015). When two sentences share the same syntax tree, we can transform one into the other while only changing their leaf-node vocabulary. For example, *My unicorn loves her rabbit* and *Your zebra eats some apples* have different vocabulary but identical syntax trees. We define a dataset's **diversity** as the number of unique syntactic trees it contains, similar to diversity metrics previously used in both natural language (Huang et al., 2023; Gao and He, 2024; Ramírez et al., 2022) and code (Song et al., 2024).

5.2 Diversity and stability

In Section 4, we showed that training stabilizes if models commit to a systematic rule. Here, we show that data diversity shapes training instability and rule commitment. Specifically, we find that models trained on less diverse data can stabilize through memorization without committing to any systematic rules, whereas those trained on more diverse data stabilize through systematic rules. Previously, we found that *which* rule a model commits to depends on whether the training set is dominated by hierarchy-inducing data or linearity-inducing data, so we will analyze these two settings separately.

Hierarchy-inducing data To study the effect of diversity in hierarchy-inducing settings, we construct multiple training sets with varying levels of syntactic diversity. Each set contains 50K question samples and 50K center-embedded declarations. We control diversity by changing the number of unique syntactic trees in the declarations. For each dataset, we train 50 random seeds and measure instability using total variation (Section 4.1). We also report a **rule commitment ratio**: the proportion of runs achieving OOD accuracy either above 95% or below 5%, which both indicate that the model has committed to a systematic rule.

Figure 6 (*left*) shows three distinct regimes. At low diversity (diversity < 5), models enter the **memorization regime**: training is stable, but they fail to commit to a rule. In Appendix G.1, we confirm that these memorizing models apply the hierarchical rule to syntactic structures seen during training, but cannot extrapolate to unseen structures. At high diversity (at least 50 unique trees), models enter the **hierarchical generalization regime**, where training reliably stabilizes by committing to the hierarchical rule. Between these extremes lies an **unstable regime** (5–20 unique trees), where the data is too diverse to memorize but not diverse enough to teach a universal rule, leading to oscillating OOD curves.

Linearity-inducing data Unlike the hierarchy-inducing setting, linearity-inducing data (right-branching sentences) permits little syntactic variation: the auxiliary always follows the subject noun, leaving no natural way to increase diversity. Instead, we build on a mix of 99% right-branching

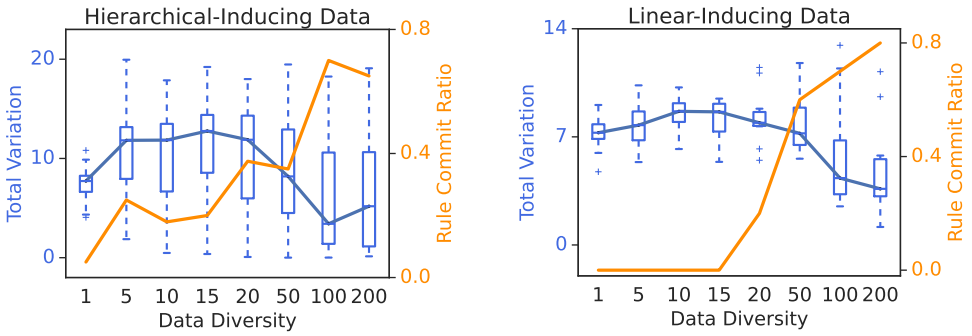

Figure 6: **Effect of data diversity on training stability and rule commitment.** The x-axis measures syntactic diversity as the number of unique syntax trees in the training set (Section XX). The left y-axis shows training instability, measured with total variation (Section 4.1). The right y-axis shows the rule commitment ratio, defined as the proportion of 50 training seeds that achieve OOD accuracy either above 95% or below 5%, indicating commitment to a systematic rule. *Left*: hierarchy-inducing data; *Right*: linearity-inducing data, where diversity is varied through the small fraction of hierarchical examples. In both settings, low diversity leads to stable memorization, intermediate diversity to instability, and high diversity to stable rule commitment.

and 1% center-embedded data, which we find reliably induces the linear rule. Keeping this mixture ratio, we control diversity by varying the syntactic structures of the center-embedded sentences. As in the hierarchy-inducing setting, we use sets of 50K questions and 50K declarations and measure training instability and rule commitment as before.

Figure 6 (*right*) shows that the same three regimes appear in the linearity-inducing case. At low diversity, models memorize familiar syntactic structures without committing to a rule. At intermediate diversity, training becomes unstable; memorization is no longer possible but the signal is too weak for to induce a systematic generalization rule. At high diversity, models stabilize in the **linear generalization regime**, committing to the linear rule. This mirrors the U-shaped pattern observed with hierarchy-inducing data, confirming that syntactic diversity is essential for systematic generalization—regardless of which rule is preferred.

6 Related Work

See App. A for a more detailed literature review.

Syntax and Hierarchical Generalization McCoy et al. (2018) first used the question formation task to study hierarchical generalization in neural networks, showing that attention mechanisms improved generalization performance in recurrent neural networks (RNNs). Later, McCoy et al. (2020a) found that tree-structured architectures consistently induce hierarchical generalization. Petty and Frank (2021) and Mueller et al. (2022) further concluded that transformers tend to generalize linearly. This view was challenged by Murty et al. (2023), who attributed their results to insufficient training: decoder-only transformers can generalize hierarchically, but only after in-distribution performance has plateaued. They named this transition from surface-level heuristics to hierarchical generalization structural grokking. Expanding on their findings, Ahuja et al. (2024) showed that models only generalize hierarchically when trained on a language modeling objective. All of this prior work attributed hierarchical inductive bias to model architecture or objective, whereas our study highlights the impact of data. While previous work observed some inconsistency across seeds (McCoy et al., 2018, 2020b), we characterize this random variation more specifically.

Training Dynamics and Grokking During *grokking*, a neural network generalizes to a test set long after it has overfitted to its training data. Power et al. (2022) first observed this phenomenon in simple arithmetic tasks. This *classic* grokking is different from our main focus of *structural* grokking (Murty et al., 2023). In classic grokking, the model transitions from memorization to generalization, allowing it to achieve non-trivial performance on unseen data from the same distribution as the train set. In structural grokking, a model transitions from the simple linear rule to the hierarchical rule, allowing non-trivial performance on OOD data. However, our findings also relate to classic grokking by connecting data diversity to memorization.

Zhu et al. (2024) studied the role of data and found that grokking only occurs when training set

is sufficiently large, and thus more diverse. Berlot-Attwell et al. (2023) studied how data diversity leads to OOD compositional generalization in multimodal models and Lubana et al. (2024) showed that diversity also induces compositional behaviors late in LM training. Liu et al. (2022) showed grokking can be induced by forcing a specific weight norm, a measurement of model—though not data—complexity. Huang et al. (2024) and Varma et al. (2023) have shown that different circuits compete during training and that different data and model sizes lead to different competition and training dynamics. Competition also shapes other phase transitions, such as transient in-context learning (Park et al., 2024).

Random Variation Although choices like hyperparameters, architecture, and optimizer all shape model outcomes, training remains inherently stochastic. Models are sensitive to random initialization and the order of training examples (Dodge et al., 2020; Zhou et al., 2020; D'Amour et al., 2022; Naik et al., 2018; Sellam et al., 2021; Juneja et al., 2022). On text classification tasks, Zhou et al. (2020) observed OOD variability even late in training. We investigate the source of these training inconsistencies and link them more precisely to characteristics of the training data.

7 Discussion and Conclusions

By exploring the role of data in OOD generalization rules, we have also revealed which settings render outcomes unpredictable. Complex data induces more complex rules, but a mix of complex and simple examples leads to instability and inconsistency. Likewise, higher data diversity favors universal generalization rules over example memorization, but intermediate diversity leads to instability and inconsistency. These findings have implications across machine learning and linguistics.

Inconsistent behavior across seeds. While variation in model error is often treated as unimodal Gaussian noise in the theoretical literature (Lakshminarayanan et al., 2016), 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 can create clusters of OOD behaviors. Previously, clusters were seen in text classification heuristics (Juneja et al., 2022) and training dynamics (Hu et al., 2023). In our case, OOD accuracy is clearly bimodal only when we exclude unstable training runs. We suggest that research on variation in training consider training stability in the future, which may expose clustered solutions.

Implications for formal linguistics. In debates about the poverty of the stimulus, linguists have extensively studied the question of what data is necessary and sufficient to learn grammatical rules (McCoy et al., 2018; Berwick et al., 2011). In particular, Wexler (1980) argued that all English syntactic rules are learnable given "degree 2" data: sentences with only one embedded clause nested within another clause. Our center embedding results confirm 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 if they are given sufficiently rich data outside of that rule: the LM extrapolates from degree-1 QF examples if it has seen enough degree-2 declaration examples to induce a hierarchical bias.

Grokking, instability, and latent structure. Classic grokking (Power et al., 2022) is different from structural grokking. While the latter entails a transition between generalization rules, the former entails a transition from memorization to generalization. Our findings clarify both scenarios.

Structural grokking is produced by competition between linear- and hierarchical-inducing training subsets. Without competing subsets, the model immediately learns either the linear or the hierarchical rule without a gradual transition. When these rules compete, training is unstable, leaving an opening for delayed hierarchical generalization. Our study of data diversity has similar implications for classic grokking, where the competition is between memorized heuristics and the systematic rules required to efficiently model diverse training data. Yet again, while a strict memorization regime is relatively stable, intermediate diversity is unstable, leading to potential grokking dynamics.

In a sense, memorization is just another rule to capture the training distribution. This framework unifies the grokking literature with other phase transitions such as emergence (Schaeffer et al., 2023) and benign interpolation (Theunissen et al., 2020). Future work could develop this unified theory of the effects of data diversity and complexity.

Limitations

Our work leverages synthetic controlled settings that are common in the language model analysis literature. These findings, like other prominent results in grokking and training dynamics, may not generalize to larger scale or multitask settings.

References

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

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

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

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

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

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

Leshem Choshen, Guy Hacohen, Daphna Weinshall, and Omri Abend. 2022. 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)*, pages 8281–8297, Stroudsburg, PA, USA. Association for Computational Linguistics.

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, and 21 others. 2022. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61.

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

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

Nan Gao and Qingshun He. 2024. A dependency distance approach to the syntactic complexity variation in the connected speech of alzheimer's disease. *Humanit. Soc. Sci. Commun.*, 11(1).

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

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

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

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

Kuan-Hao Huang, Varun Iyer, I-Hung Hsu, Anoop Kumar, Kai-Wei Chang, and Aram Galstyan. 2023. ParaAMR: A large-scale syntactically diverse paraphrase dataset by AMR back-translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yufei Huang, Shengding Hu, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Unified view of grokking, double descent and emergent abilities: A perspective from circuits competition. *arXiv [cs.LG]*.

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

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2016. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv [stat.ML]*.

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

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

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

Ekdeep Singh Lubana, Kyogo Kawaguchi, Robert P Dick, and Hidenori Tanaka. 2024. A percolation model of emergence: Analyzing transformers trained on a formal language. *arXiv [cs.LG]*.

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

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

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

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

Tom McCoy, Robert Frank, and Tal Linzen. 2020b. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. `https://rtmccoy.com/rnn_hierarchical_biases.html`. Accessed: 2024-11-20.

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

Aaron Mueller, Robert Frank, Tal Linzen, Luheng Wang, and Sebastian Schuster. 2022. 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. Association for Computational Linguistics.

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

Aaron Mueller, Albert Webson, Jackson Petty, and Tal Linzen. 2024. 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)*, pages 4761–4779, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher Manning. 2023. 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. Association for Computational Linguistics.

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

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

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

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, and 7 others. 2022. In-context learning and induction heads. *arXiv [cs.LG]*.

Isabel Papadimitriou and Dan Jurafsky. 2020. 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. Association for Computational Linguistics.

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

Core Francisco Park, Ekdeep Singh Lubana, Itamar Pres, and Hidenori Tanaka. 2024. Competition dynamics shape algorithmic phases of in-context learning. *arXiv [cs.LG]*.

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

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

Jorge Ramírez, Marcos Baez, Auday Berro, Boualem Benatallah, and Fabio Casati. 2022. Crowdsourcing syntactically diverse paraphrases with diversity-aware prompts and workflows. In *Advanced Information Systems Engineering: 34th International Conference, CAiSE 2022, Leuven, Belgium, June 6–10, 2022, Proceedings*, page 253–269, Berlin, Heidelberg. Springer-Verlag.

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

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

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

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. 2021. The MultiBERTs: BERT reproductions for robustness analysis. In *International Conference on Learning Representations*.

Yewei Song, Cedric Lothritz, Daniel Tang, Tegawendé F Bissyandé, and Jacques Klein. 2024. Revisiting code similarity evaluation with abstract syntax tree edit distance. *arXiv [cs.CL]*.

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

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. 2023. Explaining grokking through circuit efficiency. *arXiv [cs.LG]*.

Kenneth Wexler. 1980. Formal principles of language acquisition.

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

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. 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*.

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

A Related Work Extended

A.1 Syntax and Hierarchical Generalization

While works mentioned in Section 6 focused on models trained from scratch, another line of research examined the inductive bias of pretrained models. Mueller et al. (2024); Mueller and 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 and 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 and 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.

Papadimitriou and Jurafsky (2023, 2020) and Mueller et al. (2022) also studied how pretraining data could introduce an inductive bias in language acquisition. Papadimitriou and Jurafsky (2023) specifically identified that by pretraining models on data with a recursive structure their performance when later finetuning them on natural language. This finding is closely related to our conclusions around the importance of recursive center embeddings.

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 and Lampinen, 2020), which is also common in LMs. Choshen et al. (2022) found that early LMs behave like n-gram models, and Saphra and 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

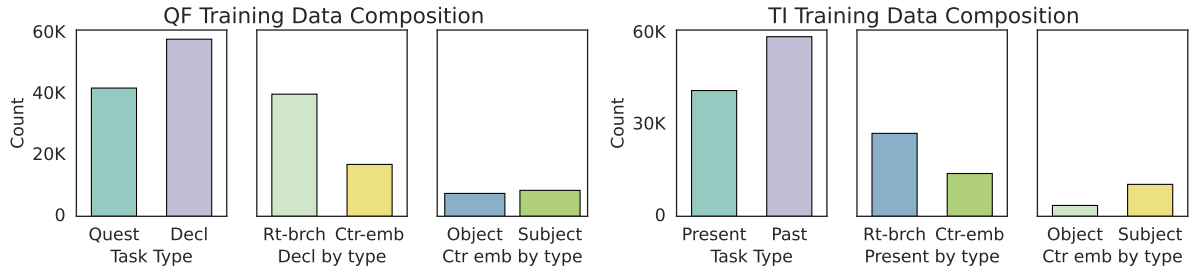

Figure 7: **Components of the original QF and TI training data.** *Left:* QF training data contains samples of two tasks types: question formation and declaration copying. We break down samples in the declaration copying task by branching type. We also breakdown center-embedded sentences based on whether the main subject serves the subject or object in the embedded clause. *Right:* TI training data also contains samples of two task types: tense inflection and past tense copying. Similar to QF, we breakdown tense inflection samples by branching types, and breakdown center-embedded sentences in the tense inflection samples by subject/object type.

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.

B Training Data Samples

B.1 Question Formation

We use the term **declarations** to refer 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 ID validation set and the OOD generalization set only contain question formation samples. In Figure 7 (*left*), we show a breakdown of two task types in QF training data.

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. The secondary task only requires repeating the given sentence, which is always in the past tense, and the copying task is marked by the `PAST` token. In Appendix E.1, show that the past-tense copying task is not necessary.

C Further Partitions on Center-Embedded Sentences

C.1 Two Subtypes of Center-Embedded Sentences

In Section 3, we showed that center-embedded sentences drive hierarchical generalization in both the QF and TI tasks. Here, we further partition center-embedded sentences based on the syntactic role of the *main subject* (i.e., the subject of the main clause) within the modifying clause. Specifically, we classify them into two types:

1. **Subject-type**: The main subject serves as the **subject** within the clause.
   Example: *The keys that unlock the cabinet are on the table.*
2. **Object-type**: The main subject serves as the **object** within the clause.
   Example: *The keys that the bear uses are on the table.*

This partition is motivated by their distinct subject-verb dependency patterns. In subject-type sentences, both the main verb (from the main clause) and the embedded verb (from the relative clause) depend on the main subject. In contrast, object-type sentences exhibit a nested subject-verb structure. Our goal is to investigate whether differences in subject-verb dependency patterns influence the model's preference for the hierarchical

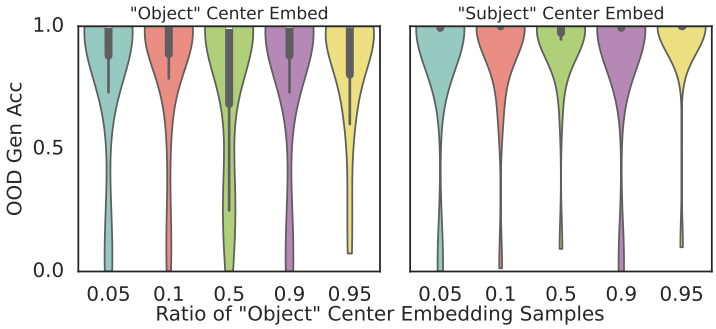

Figure 8: **Both subtypes of center-embedded sentences induce hierarchical generalization in QF.** We train models on datasets containing different ratios of object-type v.s. subject-type center-embedded sentences. We then evaluate on models on two OOD generalization set, one containing unambiguous object-type center-embedded sentences and the other unambiguous subject-type center-embedded sentences.

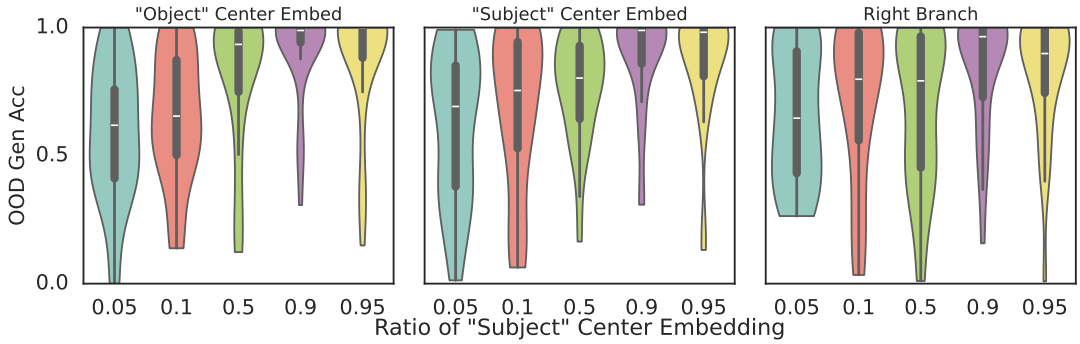

Figure 9: **Subject-type center-embedded sentences gives a stronger bias towards hierarchical generalization in TI.** We train models on datasets containing different ratios of object-type v.s. subject-type center-embedded sentences. We then evaluate on models on three OOD generalization set, one containing unambiguous object-type center-embedded sentences, one unambiguous subject-type center-embedded sentences, and one unambiguous right-branching sentences.

rule.

C.2 QF Task Results

The original QF training data contains roughly equal amount of two subtypes of center-embedded declarations, shown in Figure 7 (*left*). We investigate whether the two subtypes of center-embedded sentences differentially influence the model's preference for the hierarchical rule in the QF task. For all training data variants, we fix 50K question formation samples and 50K declaration copying samples, with the latter containing only center-embedded sentences but varying the ratio between the two subtypes. To analyze generalization behavior on a more granular level, we partition the generalization set (composed solely of center-embedded sentences) into the two subtypes as well. Models are trained on 30 random seeds, and results are shown in Figure 8. Regardless of the data mix, the model consistently favors the hierarchical rule across both partitions of the generalization set. This suggests that, for question formation, both subtypes of center-embedded sentences equally contribute to the model's ability to identify the main auxiliary.

C.3 TI Task Results

The original TI training data contains almost twice amount of subject-type center-embedded sentences than object-type ones, shown in Figure 7 (*left*). We repeat a similar experiment for the TI task, fixing the total number of tense inflection samples to 100K. As shown in Section 3.3, models exhibit the strongest hierarchical generalization when trained on primarily center-embedded sentences. Therefore, in the following data variants, 99% of the samples are center-embedded sentences, with the remaining 1% being right-branching sentences. Within the center-embedded samples, we vary the ratio between the two subtypes. To evaluate generalization, we split the generalization set into three groups: the two subtypes of center-embedded sentences and right-branching sentences.

Models trained on 30 random seeds show that, across all three generalization sets, accuracy is positively correlated with the proportion of subject-type center-embedded sentences (Figure 9). However, even when models are trained predominantly on object-type center-embedded sentences (teal violins in Figure 9), they still show a clear preference toward hierarchical generalization. Thus, while both subtypes drive hierarchical generalization in TI, subject-type center-embedded sentences have a stronger effect.

D Varying Data Ratios for Question Formation

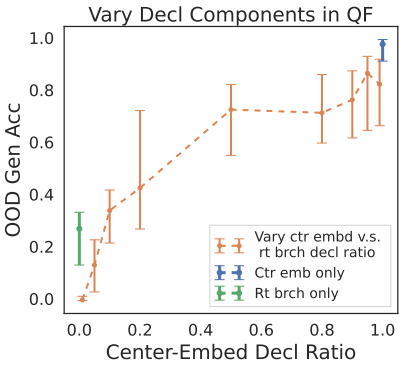

Figure 10: Hierarchical generalization in QF is sensitive to compositions of declaration-copying samples.

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 center-embedded and right-branching sentences. These datasets are used for the experiments in Section 4.2. 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.

Sensitivity to data compositions 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 10. First, note that there is a sharp performance drop between the blue bar and the right-most orange bar. This sharp transition indicates that mixing in as little as 1% of right-branching declarations significantly reduces the model's likelihood of generalizing hierarchically. Interestingly, when the dataset is predominantly right-branching declarations, models consistently achieve 0% generalization accuracy, indicating a strong preference for the linear rule across all training runs. However, note that there is another sharp transition between the green bar and the left-most orange bar. This transition indicates that as soon as we remove the 1% of center-embedded sentences, the model fails to learn either the linear rule or the hierarchical rule. As a result, the generalization accuracy is close to random guess ($\sim 25\%$). This transition is closely studied in Section 5.1, where we examine how data diversity leads to rule commitment.

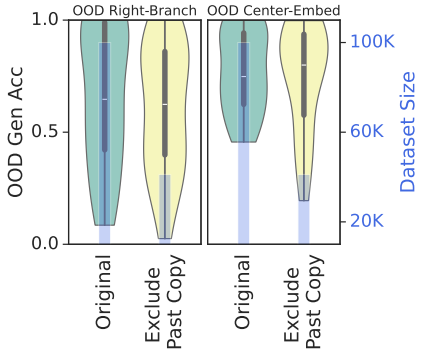

Figure 11: Past-copy task is not necessary to induce hierarchical generalization in TI.

Additional Analysis for Section 4.2 Figure 12 shows the relationship between data homogeneity and training stability. When the training data is dominated by either linearity-inducing (99% linear) or hierarchy-inducing (0% linear) examples, more random seeds lead to stable OOD curves. When the training data is a heterogeneous mix instead, potential rules compete, leading to a higher proportion of unstable training runs.

E Additional Results on Tense Inflection

E.1 A Secondary Task is Not Necessary

In the original of TI training data (McCoy et al., 2020a), 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. Figure 7 (*right*) shows a breakdown of the two tasks in the original TI training data. In experiments conducted in Section 2.2, we

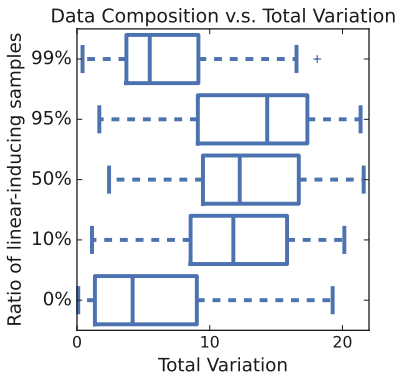

Figure 12: **Training is unstable when different subsets of data compete.** Balanced mixtures of right-branching and center-embedded sentences have higher total variation than mixtures dominated by one or the other subset.

have eliminated the used of this secondary task because center-embedded sentences can be included in the tense inflection training samples *without* violating the ambiguity requirement. Here, we use the training data originally proposed by McCoy et al. (2020a) to confirm that the use of secondary task is indeed not necessary. Specifically, we remove all the past-tense-copying samples from the original training data and train models on the tense-inflection task only. We evaluate the model's generalization performance on two OOD set containing unambiguous right-branching and unambiguous center-embedded sentences, shown in Figure 11. We can see that the model's OOD performances are the same with or without the secondary task.

E.2 Training Instability and Rule Commitment for TI

We repeat the same total variation analysis in Section 4 for the tense inflection task. We use the data mixes from Section 3.3. Specifically, we include only tense inflection samples and vary the ratio between linearity-inducing (i.e., right-branching) and hierarchy-inducing (i.e., center-embedded) sentences. In Section 3.3, 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 13 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 4.2).

F Training Instability

In Figure 14, 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 4.

G Data Diversity and Memorization Patterns

G.1 Memorization Patterns

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 Building on the diversity measure from Section 5.1, 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 and 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

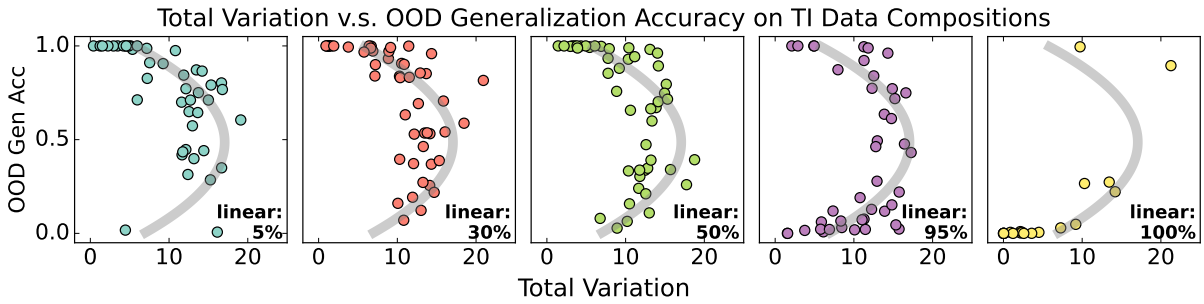

Figure 13: **Total Variation v.s. final generalization accuracy for TI task.** Similar to Figure 5, we observe the same horseshoe shaped behavior between training stability and final generalization accuracy on right-branching sentences for the TI task.

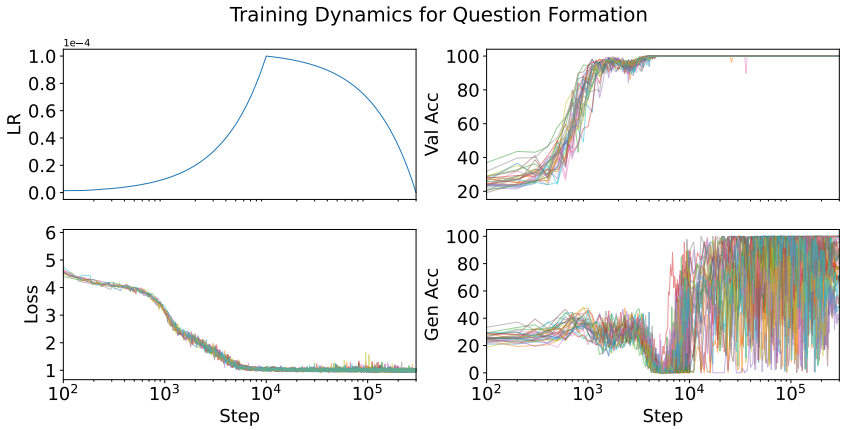

Figure 14: **Training dynamics on original QF data across 50 random seeds.** Training loss (*lower left*) and in-distribution validation accuracy (*top right*) is stable during training and consistent across random seeds. In contrast, the model's performance on OOD generalization set (*lower right*) 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). Even with a learning rate decay (*top right*), the OOD behaviors for some seeds remain unstable throughout training.

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 15, 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 15, *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 15, *right*), generalization accuracy no longer correlates

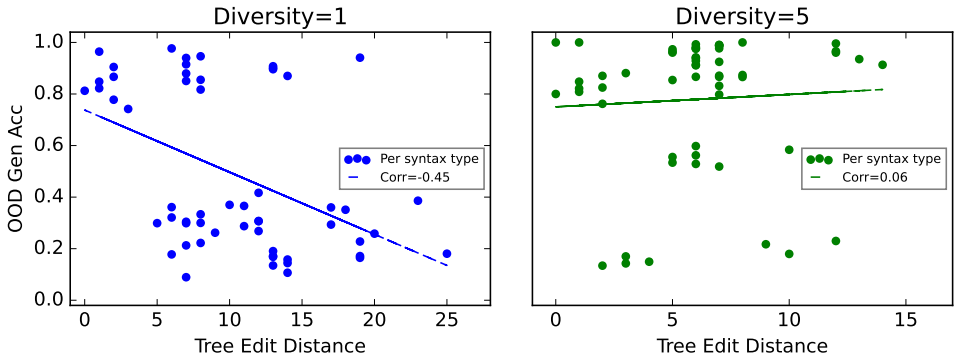

Figure 15: **OOD generalization vs. syntactic similarity to training data.** At low data diversity, model memorizes syntactic patterns and applies the hierarchical rule only on syntax structures similar to ones appeared in the training data. At high data diversity, model can extrapolates the hierarchical rule and can apply it even on unseen syntax structures that are dissimilar to training data.

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.