Sometimes I am a Tree: Data Drives Unstable Hierarchical Generalization

Tian Qin
Harvard University
Cambridge, MA
tqin@g.harvard.edu

Naomi Saphra
Harvard University
Cambridge, MA
nsaphra@fas.harvard.edu

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

Abstract

When training deep neural networks, models can adopt various heuristics, leading to different out-of-distribution (OOD) behaviors. Previous works have attributed these preferences to choices of model architecture or training objective, but the role of training data is less explored. Here, we examine how data composition impacts a model's generalization behavior and accounts for inconsistent training outcomes across random seeds. Using the question formation task as a case study, we show that hierarchical rule — the correct rule in English grammar — is induced by grammatically complex sequences with center embedding structures, whereas the linear rule — the surface level heuristic — is learned from simpler right branching sequences. Additionally, we show that models stabilize their OOD behavior in training only when committing to a rule. When the data contains a mix of simple and complex examples, potential rules compete, leading to unstable dynamics in training runs that fail to commit. Our findings highlight how training data shapes generalization patterns and how competition between data subsets can lead to inconsistent training results.

1 Introduction

Deep neural networks trained to solve a specific task often find "shortcut" heuristics instead of the intended comprehensive solution. While these heuristics fit well to the training data, they often lead to different out-of-distribution (OOD) behaviors. In this work, we study the source of inductive bias towards different generalization rules in language models, in particular the source of syntactic structure. Recent work [1, 12, 13] uses cases studies in learning English grammar to show that in cases where the training data is ambiguous between two heuristics—a linear rule and a hierarchical rule—the language model favors the hierarchical rule. They further hypothesize that learning the hierarchical rule allows the model to generalize to unseen sentences. However, these studies have not adequately addressed two critical aspects: (1) the inconsistencies observed across different training runs, and (2) the role of training data in shaping a model's preference towards the hierarchical rule. In this work, we use the question formation task to investigate how training data affects model's preference towards different heuristics and its inconsistency under random variation in seeds.

First, we demonstrate that inclusion of an auxiliary task in the training data —forming declaration sentences— is necessary to induce *any* consistent rule preferences on OOD data. We then identify the key data component in the auxiliary task, namely sentences with center embeddings, that drives a model's preference towards the hierarchical rule. While the above analysis suggests that different data compositions can significantly influence a model's bias towards hierarchical generalization, it does not fully explain the inconsistent generalization behaviors observed across random seeds. By examining training dynamics, we find that the training data composition determines the likelihood of a model committing to a rule. Moreover, only runs that commit to a simple rule exhibit stable OOD performance during training. Finally, models trained on a mix of hierarchical-inducing samples and

NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward.

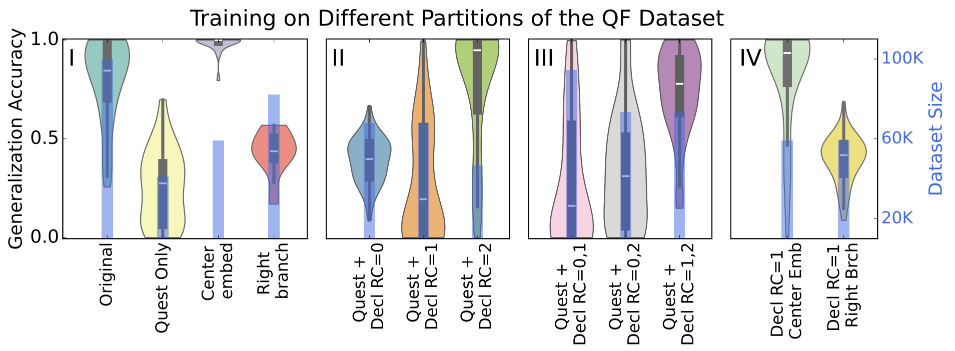

Figure 1: **Components of training data drive different generalization behaviors.** *Panel I:* Center-embedded sentences, appearing only in declaration copying task, induce hierarchical generalization. *Panel II&III:* Partitioning declaration sentences by number of relative clauses. *Panel IV:* Sentences 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.

linear-inducing samples vary the most across random seeds. In summary, our findings demonstrate that data plays a critical role in shaping model's preference towards certain heuristics and in shaping model's OOD generalization behaviors.

2 Experimental Setup

2.1 Question Formation Task

Task type	Example
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 rule output**: Doesn't my unicorn who sing does move? **Hierachical rule output**: Does my unicorn who doesn't sing move?
Decl	**Input**: My unicorn who doesn't sing does move. **Output**: My unicorn who doesn't sing does move.

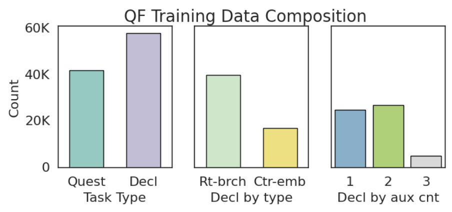

Figure 2: **Details on the training data.** *Left:* A table showing examples of question formation task, and declaration copy task. *Right:* Breakdown of different tasks and declaration types in the dataset.

In the **question formation (QF)** task, a declarative sentence is transformed into a question (see Figure 2) by moving the main auxiliary verb (such as "does" in "does move") to the front. There are two strategies for identifying which auxiliary to move: (1) a linear rule that moves the first auxiliary, or (2) 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 with specific dependency relationships (e.g., subject, preposition, object), as shown in Figure 3. The model leverages this tree representation to determine which auxiliary to move.

In Figure 2, 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 yields the correct result. The training data contains only ambiguous samples, while the OOD generalization set includes only unambiguous samples. If a model has learned a hierarchical representation of syntax, it should achieve 100% accuracy on both the in-distribution (ambiguous questions) and OOD generalization (unambiguous questions) sets. Conversely, a model relying solely on linear rules will score 0% on the OOD generalization set, but still score 100% accuracy on the in-distribution set. In our experiments, we use the model's performance on the OOD generalization set as a metric for hierarchical generalization.

2.2 Model and Training

We use a decoder-only Transformer architecture with 12M parameters: 6 layers of 8 heads with a 512-dimensional embedding. All models are trained from scratch on a causal language modeling

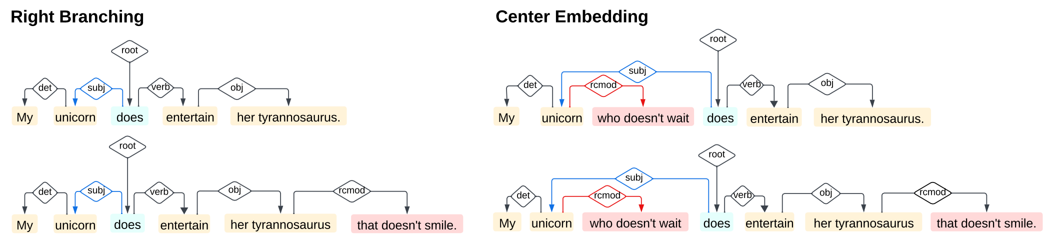

Figure 3: **Sentence Examples.** *Left:* Right-branching sentence examples. The linear progression of the main phrase is not interrupted by the relative clause. *Right:* Center-embedded sentence examples. When the relative clause modifies the subject, it interrupts the linear progression of the main clause.

objective using the Adam optimizer [10], a learning rate of 1e-4, and a linear decay schedule and train for 300K steps. We run all experiments on the same 50 random seeds.

3 Data Characteristics Drive Generalization Behaviors

3.1 Center embeddings induce hierarchical generalization

Center embeddings Center embedding occurs when a clause — often acting as a modifier — is placed within another clause or phrase. In Figure 3 left, we illustrate two center-embedded sentences examples. In those two sentences, the embedded clause interrupts the syntactic dependencies between sentence components (for example, the subject-verb-object relationship). Right branching is the opposite of center-embedding. Right branching does not exclude the use of modifying clauses — they can be appended at the end of main clause without interrupting the linear progression of the main clause. We illustrate two right-branching sentences in Figure 3 right. Linguists have studied the types of data required for humans to acquire grammatical rules, and center-embedding is central to the study. According to Chomsky's generative grammar framework [4], center-embedded clauses give rise to the hierarchical, tree-like structures of syntax. Wexler and Culicover [28] further posits that all English syntactic rules can be learned using "degree 2" data: namely, sentences containing no more than one embedded clause nested within another.

We now aim to understand whether the same type of data can induce a LM to acquire the hierarchical grammar rule. To correctly predict the distribution of next tokens, LMs also need to keep track of the dependency between different sentence components. When modeling right-branching sentences, LMs can identify the dependency relationship by linear proximity. For example, in Figure 3, the LM simply needs a linear bigram model to capture the the subject-auxiliary relationship. In contrast, when modeling center-embedded sentences, the length and syntax of the relative clause can vary, rendering the linear n-gram model inefficient for capturing the subject-verb dependency. For those sentences, using a tree structure to model the subject-verb relationship is more compact and efficient.

A secondary task Specified in Section 2.1, the training data needs to be ambiguous between the linear rule (i.e., move the first auxiliary) and the hierarchical rule (i.e., move the main auxiliary). Center-embedded sentences do not fulfill the ambiguity requirement, and cannot appear in question formation training samples. To ensure that the model is exposed to a variety of sentence types during training, McCoy et al. [13] introduces a secondary task during training: declaration copying. Similar to the question formation, the declaration-copying training sample starts with a declarative sentence, but instead of transforming it into a question, the model simply needs to repeat it. Since the ambiguity requirement does not impose restrictions on the declaration-copying task, center-embedded sentences appear in this secondary task. See Appendix B for concrete examples of both tasks.

Experiment results We now train models on three subsets of original training data by varying components of the declaration-copying task. In the first one, we remove the declaration-copying task altogether. In the second one, we only keep center-embedded sentences. In the third subset, we only keep right-branching sentences. For all three subsets, the question formation samples remain unchanged. Training on all three subsets, the in-distribution validation accuracy reaches 100%. On the other hand, the OOD generalization performances, shown in Figure 1 (I), exhibit distinct behaviors. By removing the declaration-copying task altogether, none of the 50 training runs reaches a OOD generalization accuracy higher than 75%, indicating the declaration-copying task is essential to induce the hierarchical rule. Furthermore, by only training on center-embedded sentences in the declaration-copying task, the model has a strong tendency towards the hierarchical rule. In contrast, by only training on right-branching sentences in the declaration-copying task, the model again fails to generalize hieratically. This evidence suggests that center-embedded sentence induce a model's preference towards the hierarchical rule.

3.2 Center embedding is the necessary requirement

An alternative hypothesis Center embedding imposes two restrictions on the syntax structure: first, the sentence must have a relative clause and second, the relative clause must modify the subject token. On the other hand, right-branching sentences may or may not have a relative clause. Experiments from the previous section indicates that center-embedded sentences can induce the model to learn the hierarchical rule and generalize OOD. However, we have not eliminated the possibilities that other data partitions can also induce hierarchical generalization. Specifically, we have not imposed control on number of relative clauses in the previous experiment.

We now partition declarations based on the number of relative clauses present. We train models on three datasets, each retains only declarations with a specific number of relative clauses. We also conduct experiments on subsets where we mix two of the three data partitions while excluding the third. Results are shown in Figure 1 (II and III). Models trained on declarations without any relative clauses fail to generalize hierarchically, and those trained sentences with two relative clauses are likely to succeed in doing so. This observation give rise to an alternative hypothesis: the number of relative clauses determines sentence complexity and complex sentence (ones with more relative clauses) induces the model to learn the tree-like representation of syntax.

Experiment results However, experiments on declaration sentences with one relative clause will show that this alternative hypothesis is in fact incorrect. With only one relative clause present, the clause either modifies the subject or the object. The ones modifying the subject are center-embedded while the other modifying the object are right-branching. We keep samples with one relative clause in declaration-copying task, and further partition them based on center-embedding / right-branching. The distinct generalization behaviors, shown in Figure 1 (IV), suggest that when controlling the number of relative clauses, only center-embedded sentences induces the hierarchical rule. For sentences with two relative clause, both the subject and object have a relative clause modifier, therefore, they are always center-embedded. This explains why the number of relative clauses may seem to be an alternative data partition to induce different generalization behaviors.

4 Stability Comes from Commitment to a Simple Rule

4.1 Instability during training

A model's OOD behavior can be highly unstable during training. Instead of evolving monotonically, generalization accuracy often exhibits large swings. To quantify this intra-run instability, we use the total variation as a metric. 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:

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

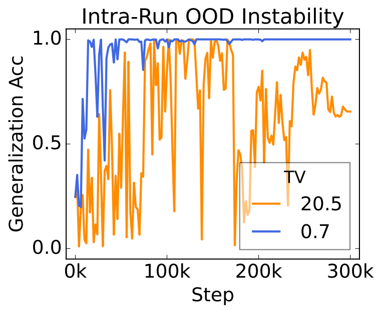

Figure 4: Total variation measures intra-run instability.

In Figure 4, we visualize the OOD behavior of two runs during training and their total variation. Both trained on the original data, the more stable run (blue line) reaches 100% generalization accuracy within 50K steps and maintains it. In contrast, the less stable run (orange line) occastionaly reaches 100% accuracy but fluctuates, eventually converging around 65%.

4.2 Training stability ties to rule commitment

Our next step is to investigate whether all models with stable (i.e., small TV) OOD behavior during training commit to the hierarchical rule. We validate this hypothesis by analyzing runs trained on

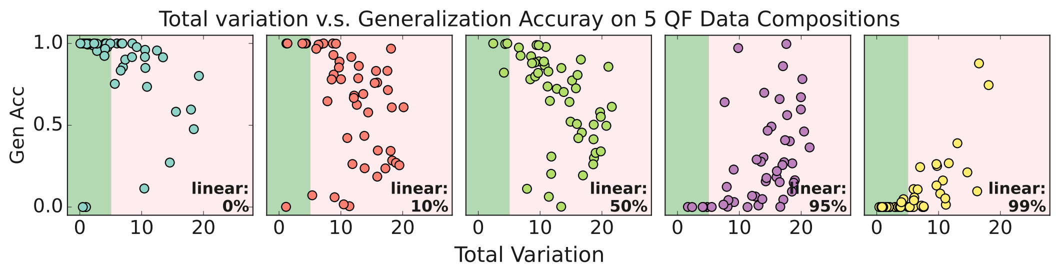

Figure 5: **Total Variation v.s. final generalization accuracy.** OOD training stability ties to committing to a simple rule, and mixing declaration types leads to variations in rule learning across random seeds. “Linear" denotes the proportion of linear-inducing declarations in the data.

different data compositions. Each dataset contains a different proportion of hierarchical-inducing and linear-inducing declarations (further details on the dataset in Appendix C). In Figure 5, we visualize the relationship between a training run’s total variation and its final generalization accuracy.

Across 5 datasets, we observe two distinct regimes of OOD behavior. For runs with a total variation below 5, the final generalization accuracy is either 100% or 0%. Furthermore, data composition (i.e., proportions of hierarchical/linear inducing data) determines which rule is favored by stable runs. Interestingly, in mixed data scenarios (e.g., linear=10% case), the final generalization accuracy of stable runs shows a bimodal distribution, clustering around 100% or 0%. This bimodality suggests that training stability is tied to committing to a simple rule, even when the data is “ambiguous" between rules. In Appendix D, we visualize sample runs with different total variation values.

In addition to different rule preference, we also observe different distributions of stable and unstable runs across 5 datasets. We now examine how data composition not only drives rule preferences but also also impact training instability. In Figure 6, we visualize the distribution of total variation across different seeds to assess whether heterogeneous data mixes lead to training instabilities. When the data composition is dominated by either hierarchical-inducing (e.g., linear=0%) or linear-inducing (e.g., linear=99%) data, we observe more stable runs, signified by small total variations. This suggests that data homogeneity results in both training stability and consistency across random seeds. On the other hand, as the data becomes more heterogeneous (e.g., linear=50%), a higher proportion of runs exhibit unstable OOD behavior. This indicates that mixed data types contribute to training instabilities and inconsistent OOD behavior at convergence.

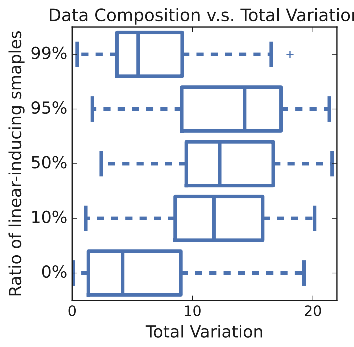

Figure 6: Competition between subsets of data drives training instability.

5 Conclusion

We have shown how training data — at least in a specific setting — shapes the OOD generalization patterns and how competition between different components in the data lead to training inconsistency. In fact, both during-training and at-convergence random variations on OOD data have been observed in the wild [29]. For future work, it would be interesting to extend our analysis that links data, simplicity bias and training inconsistency to these tasks. We have also shown that for English grammatical rules, specific structures identified in linguistic studies play a significant role in shaping an LM’s inductive bias toward hierarchical grammar rules. For future work, it would be interesting to extend this analysis to other syntactic structures, examining how diverse grammatical patterns influence LMs’ language acquisition behaviors.

References

[1] K. Ahuja et al. "Learning syntax without planting trees: Understanding when and why transformers generalize hierarchically". *arXiv [cs.CL]* (Apr. 2024).

[2] B. Barak et al. "Hidden progress in deep learning: SGD learns parities near the computational limit". *arXiv [cs.LG]* (July 2022).

[3] A. Chen et al. "Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs". *arXiv [cs.CL]* (Sept. 2023).

[4] N. Chomsky. *Aspects of the theory of syntax*. en. 50th ed. The MIT Press. London, England: MIT Press, 2015.

[5] L. Choshen, G. Hacohen, D. Weinshall, and O. Abend. "The grammar-learning trajectories of neural language models". *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jan. 2022, pp. 8281–8297.

[6] A. D'Amour et al. "Underspecification Presents Challenges for Credibility in Modern Machine Learning". *Journal of Machine Learning Research* 23.226 (2022), pp. 1–61.

[7] J. Dodge et al. "Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping". *arXiv [cs.CL]* (Feb. 2020).

[8] K. L. Hermann and A. K. Lampinen. "What shapes feature representations? Exploring datasets, architectures, and training". *arXiv [cs.LG]* (June 2020).

[9] J. Hewitt and C. D. Manning. "A Structural Probe for Finding Syntax in Word Representations". *Proceedings of the 2019 Conference of the North*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 4129–4138.

[10] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". *arXiv [cs.LG]* (Dec. 2014).

[11] B. MacWhinney. *The childes project: Tools for analyzing talk, volume II: The database*. 3rd ed. London, England: Psychology Press, Jan. 2014.

[12] R. T. McCoy, R. Frank, and T. Linzen. "Does syntax need to grow on trees? Sources of hierarchical inductive bias in sequence-to-sequence networks". en. *Trans. Assoc. Comput. Linguist.* 8 (Dec. 2020), pp. 125–140.

[13] R. T. McCoy, R. Frank, and T. Linzen. "Revisiting the poverty of the stimulus: hierarchical generalization without a hierarchical bias in recurrent neural networks". *arXiv [cs.CL]* (Feb. 2018).

[14] R. T. McCoy, E. Pavlick, and T. Linzen. "Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference". *arXiv [cs.CL]* (Feb. 2019).

[15] W. Merrill, N. Tsilivis, and A. Shukla. "A tale of two circuits: Grokking as competition of sparse and dense subnetworks". *arXiv [cs.LG]* (Mar. 2023).

[16] A. Mueller and T. Linzen. "How to plant trees in language models: Data and architectural effects on the emergence of syntactic inductive biases". *arXiv [cs.CL]* (May 2023).

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

[18] A. Mueller et al. "Coloring the blank slate: Pre-training imparts a hierarchical inductive bias to sequence-to-sequence models". *Findings of the Association for Computational Linguistics: ACL 2022*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2022.

[19] S. Murty, P. Sharma, J. Andreas, and C. Manning. "Grokking of hierarchical structure in vanilla transformers". *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2023.

[20] S. Murty, P. Sharma, J. Andreas, and C. D. Manning. "Characterizing intrinsic compositionality in transformers with tree projections". *arXiv [cs.CL]* (Nov. 2022).

[21] A. Naik et al. "Stress Test Evaluation for Natural Language Inference". *Proceedings of the 27th International Conference on Computational Linguistics*. 2018, pp. 2340–2353.

[22] N. Nanda et al. “Progress measures for grokking via mechanistic interpretability”. *arXiv [cs.LG]* (Jan. 2023).

[23] C. Olsson et al. “In-context learning and induction heads”. *arXiv [cs.LG]* (Sept. 2022).

[24] J. Petty and R. Frank. “Transformers generalize linearly”. *arXiv [cs.CL]* (Sept. 2021).

[25] A. Power et al. “Grokking: Generalization beyond overfitting on small algorithmic datasets”. *arXiv [cs.LG]* (Jan. 2022).

[26] N. Saphra and A. Lopez. “Understanding learning dynamics of language models with”. *Proceedings of the 2019 Conference of the North*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jan. 2019, pp. 3257–3267.

[27] T. Sellam et al. “The MultiBERTs: BERT Reproductions for Robustness Analysis”. *International Conference on Learning Representations*. Oct. 2021.

[28] K. Wexler and P. W. Culicover. *Formal principles of language acquisition*. en. London, England: MIT Press, July 1980.

[29] X. Zhou, Y. Nie, H. Tan, and M. Bansal. “The Curse of Performance Instability in Analysis Datasets: Consequences, Source, and Suggestions”. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Nov. 2020.

A Related Work

A.1 Syntax and Hierarchical Generalization

McCoy et al. [13] 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. [12] found that tree-structured architectures consistently induce hierarchical generalization. Petty and Frank [24] and Mueller et al. [18] further investigated inductive biases and concluded that, like RNNs, transformers tend to generalize linearly. This view was challenged by Murty et al. [19], 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. [1] showed that hierarchical generalization is achieved only with models trained on a language modeling objective.

While the previous work focused on models trained from scratch, another line of research examined the inductive bias of pre-trained models. Mueller and Linzen [16] and Mueller et al. [17] pre-trained transformers on text corpora such as Wikipedia and CHILDES [11] 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 [9] and Murty et al. [20] directly interpreted model's internal representation to understand whether transformers constrain their computations to to follow tree-structure patterns. Hewitt and Manning [9] demonstrated that the syntax tress are embedded in model's representation space. Similarly, Murty et al. [20] projects transformers into a tree-structured network, and showed that transformers become more tree-like over the course of training on language data.

Previous work primarily attributed the source of inductive bias toward hierarchical rule to model architecture, whereas our study highlights two overlooked aspects: training inconsistency and the impact of data. McCoy et al. [13] and Ahuja et al. [1] observed inconsistencies across training runs, but the underlying causes remain unexplored. While McCoy et al. [13] emphasized the importance of declarative sentences, their conclusions were based on RNNs trained with a seq-to-seq objective.

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. D'Amour et al. [6], Naik et al. [21], and Zhou et al. [29] reported significant performance differences across model checkpoints on various analysis and stress test sets. Zhou et al. [29] further found that instability extends throughout the training curve, not just in final outcomes. To investigate the source of this inconsistency, Dodge et al. [7] 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. [27] 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 [8], which is also common in LMs. Choshen et al. [5] found that early LMs behave like n-gram models, and Saphra and Lopez [26] observed that early LMs learn simplified versions of the language modeling task. McCoy et al. [14] showed that even fully trained models can rely on simple heuristics, like lexical overlap, to perform well on Natural Languge Inference (NLI) tasks. Chen et al. [3] further explored the connection between training dynamics and simplicity bias, showing that simpler

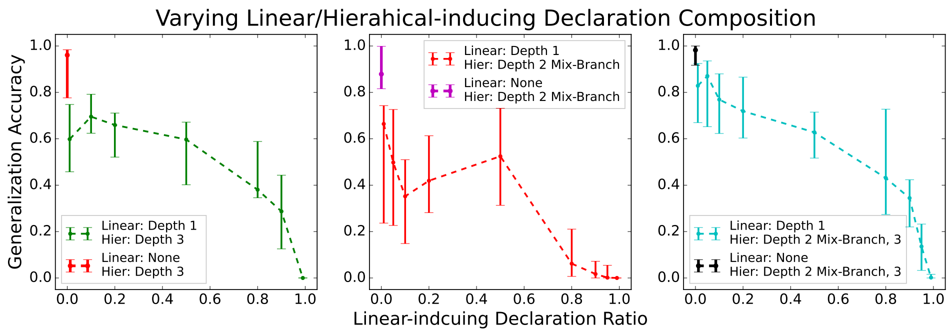

Figure 7: **Hierarchical generalization is sensitive to data compositions.** Datasets are standardized to contain 50K question formation samples and 50K declaration samples. For declaration samples, we experiment mixing different proportions of hierarchical-inducting declarations (depth 3 and depth 2 with mixed-branching) and linear-inducing declarations (depth 1).

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. [23] 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 [15, 25] and arithmetic tasks [2, 22].

In the context of hierarchical generalization, Ahuja et al. [1] 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.

B Training Data Samples

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 caucal language modeling objective.

C Sensitivity to Data Compositions

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. To generate additional declarations, we follow the distribution of the original dataset by resampling words from the vocabulary while preserving the syntactic structure. These datasets are used for the experiments in Section 4.2.

Additional Experiments We also use these datasets to examine how different mix ratios affect the model's preference for hierarchical generalization. The median generalization accuracy, along with error bars representing the 35th and 65th percentiles, is shown in Figure 7. Including as little as 1

D Sample Training Runs

We visualize several training runs with different total variation values in Figure 8.

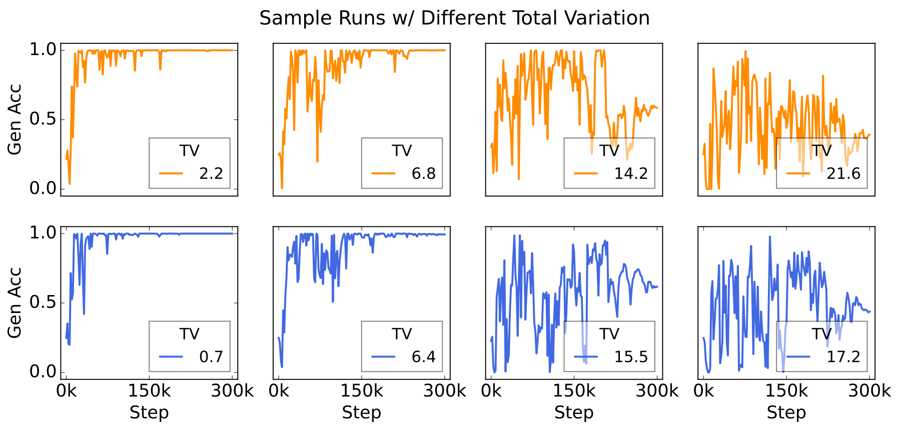

Figure 8: Sample Runs with different total variation

E Details on Training Dynamics

In Figure 9, we visualize the training dynamics for 30 independent runs when trained on the original data. Each run differs in both initialization and data order. These runs share a similar progression in training loss, validation accuracy, and generalization accuracy up until moment when the training loss converges. Pink line marks the median training steps when the model reaches 100% validation (ID) accuracy, indicating linear rule is applied ID. Immediately after, all runs reach 0% on the generalization set (brown line), indicating that the model strictly prefers linear rules on OOD data. After that, models start to achieve non-trivial performance in generalization accuracy. Gray line marks then median training steps when model reaches 100% for the first time on OOD data, indicating the hierarchical rule is preferred. 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.

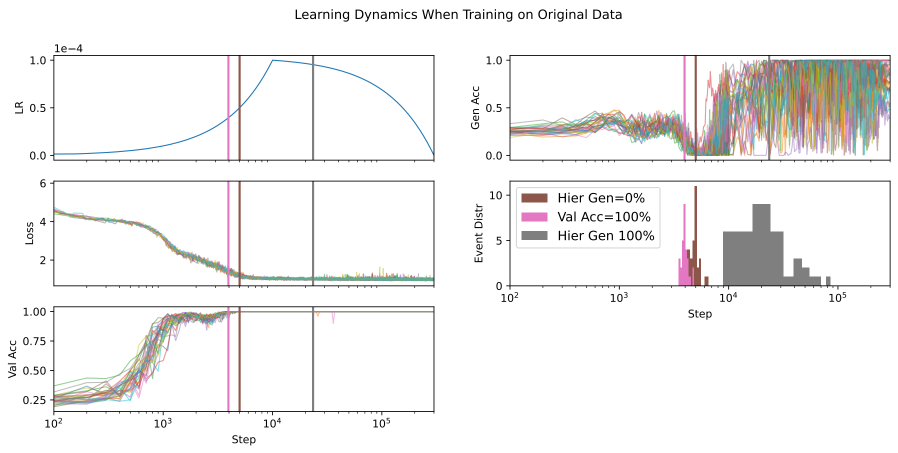

Figure 9: **Training Dynamics on original data.** After a training momen when models apply linear rule on OOD data (brown line), we observe large training inconsistency on OOD data both within a single run and across different runs.