Twin Studies of Factors in OOD Generalization

Victoria Li[*]
Dept. of Computer Science
Harvard University
vrli@college.harvard.edu

Jenny Kaufmann[*]
Dept. of Mathematics
Harvard University
jkaufmann@math.harvard.edu

David Alvarez-Melis
Dept. of Computer Science
Harvard University
dam@seas.harvard.edu

Naomi Saphra
Kempner Institute
Harvard University
nsaphra@g.harvard.edu

Abstract

Studies of model behavior often neglect the role of random variation in out-of-distribution (OOD) behavior. To enable research on the interplay between random factors and training conditions in determining model behavior, we describe a simple ambiguous setting where models can learn either a counting-based or hierarchical classification rule. We find LSTMs consistently learn the hierarchical rule, while transformer models demonstrate diverse generalization rules across different hyperparameter and random seed settings. We analyze this model population across and at the end of training, using natural variation to draw conclusions about determinants of model performance and generalization. In particular, we quantify the impact of particular hyperparameter choices, finding that different model depths favor different rules and that regularization drives multimodally distributed generalization capabilities. We also release the weights of the 270 transformer models we trained spanning a wide range of OOD behavior, which can serve as a sandbox for theoretical and interpretability investigations.

1 Introduction

In the empirical science of deep learning, random variation is a scourge. Failed training runs are discarded from analysis. Clean theories lean on convex linear networks or infinite parameter sizes to guarantee uniform behavior across seeds. Only converged behavior is treated as a tractable target of study, while dynamics with oscillation or irregularity are best overlooked.

In the traditional sciences, by contrast, variation is not a nuisance, but a controllable factor—and even an analytic tool. A gold standard in genetics research, for example, is a *twin study* comparing identical and fraternal twins. These studies are used to quantify not only the broad heritability of traits, but the role of specific genes and even variable environmental factors. In this paper, we likewise use "identical" model sets—varying only by random seed—to explore the contributions of hyperparameter settings and random data order conditions.

Existing work often argues the impact of particular hyperparameters on training based on a small group of models in particular conditions-of-interest (see e.g. Wei et al. (2022) and other work on scaling laws). Herein, we describe a proof of concept that by running more tests across a sweep of settings, we can determine when differences in model training dynamics are attributable to certain hyperparameters or simply random variation. Furthermore, the random variation itself provides a

[*] Joint first author.

Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.

rich dataset of model behaviors, which can enable correlational studies of differences in interpretable circuits and training dynamics.

Using a novel setting with an ambiguous sequence classification rule, i.e., where models can achieve perfect in-distribution (ID) accuracy through either a counting (EQUAL-COUNT) or hierarchical (NESTED) rule, we explore variation in rule learning. Studying a large set of 270 Transformer models and 83 LSTM models, we investigate the conditions under which consistent generalization rules are learned and applied out of distribution (OOD). The rich behaviors present in our experimental setting provide a number of surprising conclusions:

- We introduce a toy setting for exploring random and nonrandom factors in OOD generalization behavior (Section 2). Our visualizations reveal the distinct OOD generalization rules that create clusters—including an outlier OOD heuristic FIRST-SYMBOL that fails even ID, revealing that some OOD generalization behavior can expose hidden mistakes that are otherwise compensated or cancelled by other subnetworks (Section 3).
- When LSTM models converge, they almost always converge to a hierarchical rule. Transformers, however, can converge to different heuristics depending on their random seed (Section 3.1). This result supports existing claims that LSTMs have a stronger hierarchical inductive bias than Transformers do. It also contributes to a growing body of evidence that OOD generalization—but not ID generalization—is subject to random variation in training.
- When Transformers converge, 1-layer Transformers learn either the EQUAL-COUNT rule or a FIRST-SYMBOL heuristic. 2-layer Transformers show a preference for the NESTED rule, while 3-layer Transformers show no strong rule preference (Section 3.2). This result supports the notion that 1-layer Transformers cannot learn latent hierarchical structure.
- Weight decay leads to more consistent systematic OOD generalization, closer to the extremes of specific generalization rules like EQUAL-COUNT and NESTED (Section 3.4). Our findings suggest vestigial features and circuits—otherwise pruned by regularization—may shape OOD generalization, especially without adequate regularization.
- One-layer models learn an OOD generalization rule nearly simultaneously upon learning to generalize ID, whereas deeper models exhibit what Murty et al. (2023) call *structural grokking*, a phenomenon in which OOD generalization changes well after ID accuracy converges (Appendix C).

2 Experiments

Our synthetic setting trains binary classifiers on input strings of (and) parentheses. Our in-distribution train and test data are chosen to be ambiguous between two possible rules:

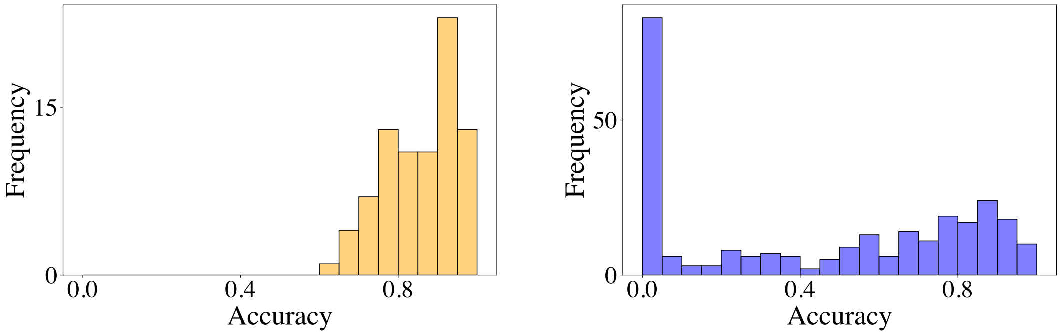
- A string s is a positive example of the NESTED rule if its parentheses are properly nested and balanced.
- A string s is a positive example of the EQUAL-COUNT rule if it has the same number of of (and) parentheses.

Our 1M datapoint train and 1K datapoint ID test datasets are split between 50% nested equal-count strings (with positive labels under both rules) and 50% non-nested unequal-count strings (with negative labels under both rules). Thus, models may learn either rule—or some combination of the two—and achieve perfect ID accuracy. To disambiguate what rule was learned, we test models on an OOD dataset: OOD strings have equal numbers of (and) parentheses, but these parentheses are not hierarchically nested. In other words, OOD test strings obey EQUAL-COUNT but not NESTED.

We create our datasets by generating uniformly random parentheses sequences with string lengths sampled randomly from a Binomial$(40, 0.5)$ distribution (see Appendix B). We train all models with learning rate 0.0001, optimizer Adam, dropout 0, batch size 8, and an embedding dimension of 64. We use 5 random seeds and 3 data shuffling seeds, yielding a total of 15 models for each hyperparameter setting. We also vary hyperparameters, considering three weight decays: $0, 0.001$ and 0.01. For transformers, we train $1, 2$ and 3 layer models with 2 or 4 heads using the minGPT architecture (Karpathy, 2020). For LSTMs we train 2 and 3 layer models. All transformers achieve an in-distribution test accuracy of 0.99 early in training, and maintain it for at least 85% of total

	NESTED	EQUAL-COUNT	FIRST-SYMBOL
`()(())`	`True`	`True`	`True`
`)()(`	`False`	`True`	`False`
`())()`	`False`	`False`	`True`

Table 1: Labels for example strings under each of the OOD generalization rules our models learn.



(a) OOD-test accuracy for LSTMs.

(b) OOD-test accuracy for Transformers.

Figure 1: OOD test accuracy for LSTM and Transformer models that achieve 99%+ ID accuracy. When LSTMs converge to near-perfect ID accuracy, they consistently also apply the NESTED rule to OOD data. Transformers, meanwhile, apply a variety of rules and exemplar-based behavior OOD.

training. Around half of LSTMs training runs yield these conditions, while the other half never reach 0.99 ID accuracy at any point during training.

3 Factors in rule selection

We investigate the effects of hyperparameter values on rule selection in our ambiguous training setting. In addition to the NESTED and EQUAL-COUNT rules, we empirically observe a third heuristic governing OOD generalization behavior, which we refer to as FIRST-SYMBOL. This heuristic labels an OOD (non-nested/equal-count) string `s` `True` if `s[0] = (` and `False` if `s[0] = )`. No models follow FIRST-SYMBOL on ID data, as it cannot correctly classify all ID sequences; it classifies about half of OOD sequences as positive.

For consistency, we define OOD accuracy with respect to the NESTED rule. Thus a model achieving 100% OOD accuracy classifies each non-nested, equal-count string in our OOD test set as `False`. Correspondingly, models with 0% OOD accuracy learn EQUAL-COUNT, classifying every OOD example as `True` (Table 1).

3.1 Architecture

By comparing Transformers to LSTMs, we confirm existing findings (Abnar et al., 2020; McCoy et al., 2020a; Tran et al., 2018; Saphra & Lopez, 2020) that LSTMs are intrinsically hierarchical while Transformers are not (Figure 1). The inductive bias of the LSTM architecture places every successfully trained model at more than 60% accuracy on the OOD generalization set, indicating that none of these models learn EQUAL-COUNT and all are closer to the hierarchical NESTED rule. In contrast, Transformer models exhibit an OOD accuracy distribution with two peaks: 24% of models have an OOD accuracy near 0% (indicating perfect application of the EQUAL-COUNT rule) while the others fall around a peak at 90% OOD accuracy (indicating a tendency towards NESTED).

3.2 Depth

Depth is a significant factor in rule selection, determining the peaks of the OOD behavior distribution. For instance, the 24% of Transformer models which achieve perfect EQUAL-COUNT behavior, i.e., 0% OOD accuracy (Figure 1), include 68%, 0% , and 5% respectively of 1-, 2-, and 3-layer models.

Our 1-layer models cluster around two rules: EQUAL-COUNT (yielding 0% OOD accuracy) and FIRST-SYMBOL (yielding ∼55% OOD accuracy). The 12 models exhibiting FIRST-SYMBOL match each other's decisions almost identically, returning `True` label if the input begins with `(`. This heuristic, unlike either EQUAL-COUNT or NESTED, cannot achieve perfect ID accuracy alone because it would incorrectly classify about half of negative training examples. Because we only consider models with at least 99% validation accuracy, they cannot rely on FIRST-SYMBOL ID.

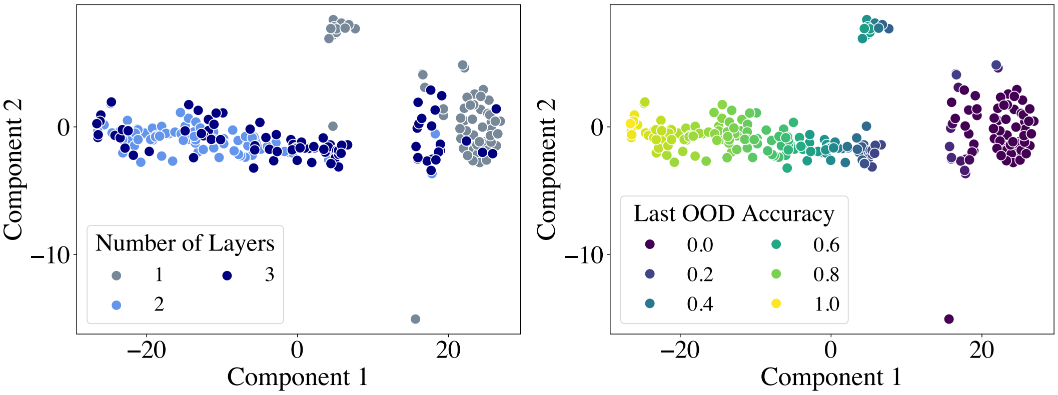


Figure 2: T-SNE of models' final OOD classifications colored by model depth and OOD test accuracy.

We use T-SNE to visualize models based on their OOD test set labels (Figure 2). While EQUAL-COUNT and NESTED clearly form distinct clusters, FIRST-SYMBOL also forms an outlier cluster in model judgements. We also observe one outlier model (lower right), which switches its OOD generalization rule from FIRST-SYMBOL to EQUAL-COUNT partway through training and may remain slightly closer to FIRST-SYMBOL.

Only 2% of 2-layer models are close to EQUAL-COUNT (determined by $< 20\%$ accuracy OOD) and none learn FIRST-SYMBOL. The mode of this model distribution is instead at 90% accuracy, firmly suggesting that most 2-layer models learn to approximate NESTED. Among 3-layer models, behavior varies enormously, with the distributional mode defined by the 20 models with < 0.1 OOD accuracy and learn EQUAL-COUNT. Over all 270 runs, 10.4% of models achieve at least 90% accuracy, and no one-layer models do. Shallow models instead learn simpler counting rather than hierarchical rules, and intermediate depth 2-layer models tend more toward hierarchy relative to 3-layer models, supporting the inverted U-shape of tree-structuredness hierarchical inductive biases in Transformers from Murty et al. (2023).

3.3 Width

Using the non-parametric Mann-Whitney U test to detect differences between distributions, we find layer width has no significant effect on the distribution of OOD accuracies for any depth of model (Figure 4(a)). This result adds to a growing body of evidence across settings that transformer width, unlike depth, has little effect on model expressivity and OOD generalization (Petty et al., 2024; Tay et al., 2022).

3.4 Regularization

Weight decay significantly impacts the rules learned. Without weight decay, models can converge on a variety of OOD generalization behaviors with varying accuracy. With weight decay, models cluster around the OOD accuracy modes (Figure 4(b)).

Also, with weight decay, all 1-layer models converge to the EQUAL-COUNT rule. Without it, 15.6% converge to FIRST-SYMBOL with ∼55% accuracy on OOD-test (Figure 4(b)). The presence of all FIRST-SYMBOL-learning models in 1-layer models with weight decay 0 indicates regularization can help prune away vestigial model features unnecessary for ID generalization. The presence of circuits supporting FIRST-SYMBOL may not impact ID performance, but in the absence of regularization, such features significantly decrease OOD performance.

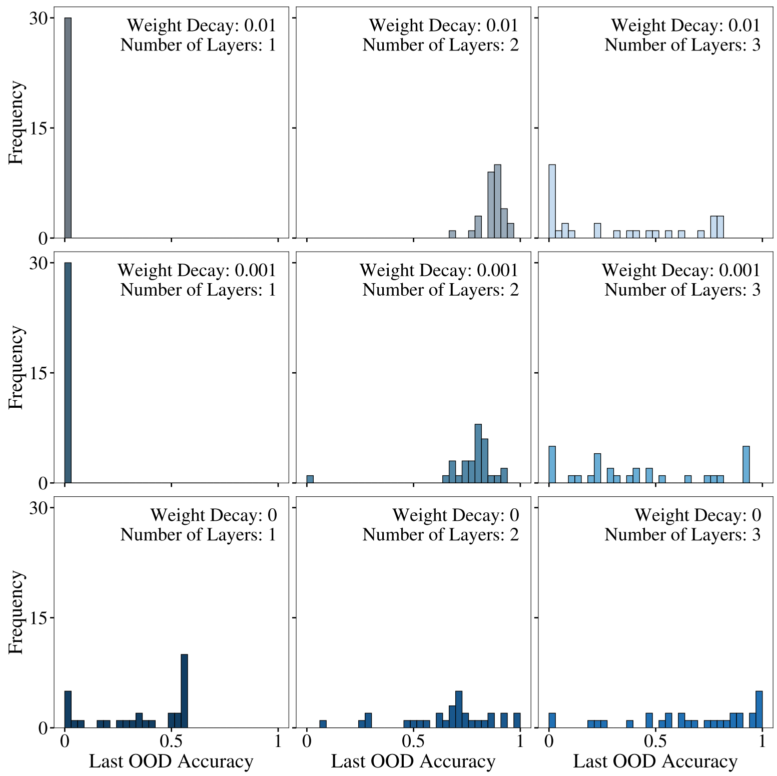
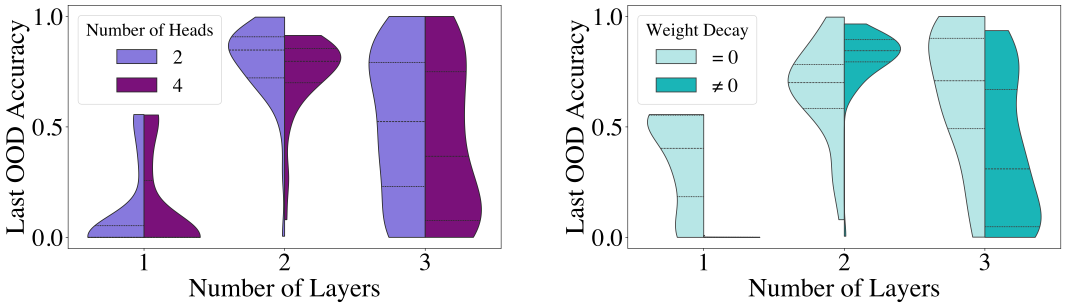


Figure 3: Final accuracy on OOD-test for Transformers of varying depths and weight decays. 1-layer Transformers learn EQUAL-COUNT or FIRST-SYMBOL. Deeper Transformers can learn EQUAL-COUNT or approximate NESTED, with 2-layer Transformers most likely to learn NESTED and 3-layer Transformers instead exhibiting more complex OOD generalization behavior.



(a) Depth of Transformers

(b) WD of Transformers

Figure 4: Width is not a substantial factor in final OOD rule selection in this setting, but weight decay is. In particular, the Mann-Whitney U test finds no statistically significant differences in the distribution of last OOD accuracy over width (all $p > 0.05$), but statistically significant differences between the absence and presence of weight decay (all $p \ll 0.01$).

4 Discussion and Conclusions

We introduce a novel setting for studying the training dynamics and learned rules of transformer models. We show LSTMs and transformers behave very differently at our classification task, suggesting our setting could be leveraged to better understand the inductive biases of transformers, possibly central to surpassing LSTMs in language modeling tasks (Bhattamishra et al., 2023). The presence of the FIRST-SYMBOL heuristic in one layer models also indicates we could use this setting to study internal representations, competitive dynamics, and OOD generalization.

Acknowledgments and Disclosure of Funding

This work was enabled in part by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence. Jenny Kaufmann is supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 2140743. David Alvarez-Melis acknowledges support from the Kempner Institute, the Aramont Fellowship Fund, and the FAS Dean's Competitive Fund for Promising Scholarship.

References

Samira Abnar, Mostafa Dehghani, and Willem Zuidema. Transferring inductive biases through knowledge distillation, 2020. URL https://arxiv.org/abs/2006.00555.

Kabir Ahuja, Vidhisha Balachandran, Madhur Panwar, Tianxing He, Noah A. Smith, Navin Goyal, and Yulia Tsvetkov. Learning syntax without planting trees: Understanding when and why transformers generalize hierarchically, 2024. URL https://arxiv.org/abs/2404.16367.

D. B. Arnold and M. R. Sleep. Uniform random generation of balanced parenthesis strings. *ACM Trans. Program. Lang. Syst.*, 2(1):122–128, 1980. ISSN 0164-0925. doi: 10.1145/357084.357091. URL https://doi.org/10.1145/357084.357091.

Satwik Bhattamishra, Arkil Patel, Varun Kanade, and Phil Blunsom. Simplicity bias in transformers and their ability to learn sparse boolean functions, 2023. URL https://arxiv.org/abs/2211.12316.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping, 2020. URL https://arxiv.org/abs/2002.06305.

Michael Y. Hu, Angelica Chen, Naomi Saphra, and Kyunghyun Cho. Latent state models of training dynamics, 2024. URL https://arxiv.org/abs/2308.09543.

Jeevesh Juneja, Rachit Bansal, Kyunghyun Cho, João Sedoc, and Naomi Saphra. Linear connectivity reveals generalization strategies, 2023. URL https://arxiv.org/abs/2205.12411.

Andrej Karpathy. Mingpt transformer model, 2020. URL https://github.com/karpathy/minGPT.

R. Thomas McCoy, Robert Frank, and Tal Linzen. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140, 2020a. doi: 10.1162/tacl_a_00304. URL https://aclanthology.org/2020.tacl-1.9.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance, 2020b. URL https://arxiv.org/abs/1911.02969.

Aaron Mueller, Albert Webson, Jackson Petty, and Tal Linzen. In-context learning generalizes, but not always robustly: The case of syntax, 2024. URL https://arxiv.org/abs/2311.07811.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D. Manning. Grokking of hierarchical structure in vanilla transformers, 2023. URL https://arxiv.org/abs/2305.18741.

Jackson Petty, Sjoerd van Steenkiste, Ishita Dasgupta, Fei Sha, Dan Garrette, and Tal Linzen. The impact of depth on compositional generalization in transformer language models, 2024. URL https://arxiv.org/abs/2310.19956.

Naomi Saphra and Adam Lopez. LSTMs compose—and Learn—Bottom-up. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2797–2809, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.252. URL https://aclanthology.org/2020.findings-emnlp.252.

Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, Ian Tenney, and Ellie Pavlick. The multiberts: Bert reproductions for robustness analysis, 2022. URL https://arxiv.org/abs/2106.16163.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale efficiently: Insights from pre-training and fine-tuning transformers, 2022. URL https://arxiv.org/abs/2109.10686.

Ke Tran, Arianna Bisazza, and Christof Monz. The importance of being recurrent for modeling hierarchical structure, 2018. URL https://arxiv.org/abs/1803.03585.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022. URL https://arxiv.org/abs/2206.07682.

Yongchao Zhou, Uri Alon, Xinyun Chen, Xuezhi Wang, Rishabh Agarwal, and Denny Zhou. Transformers can achieve length generalization but not robustly, 2024. URL https://arxiv.org/abs/2402.09371.

A Related Work

Prior work on random variation has explored the influence of randomness on ID and OOD behavior. Dodge et al. (2020) found that weight initialization and data order are equally influential random factors. Meanwhile, Sellam et al. (2022) performed a study of random variation in performance generally and found that OOD behavior varies more by seed or architecture than ID performance does.

Other work has found that OOD generalization behavior displays more variation and bimodality than ID behavior, even within a single architecture trained on the same dataset (McCoy et al., 2020b; Juneja et al., 2023; Mueller et al., 2024; Zhou et al., 2024). Random variation in the timing of generalization during training is another case of greater variation in OOD behavior compared with ID (Hu et al., 2024)

Inductive bias towards hierarchy has also been previously studied. (McCoy et al., 2020a; Murty et al., 2023; Ahuja et al., 2024; Saphra & Lopez, 2020). LSTMs have been observed to generalize hierarchically, while transformers display less consistent inductive biases.

B Dataset generation and model training

We investigated a Dyck-1 setting generating datasets by sampling string lengths from a Binomial$(40, 0.5)$ distribution, discarding repeats. Since we discard repeats, the empirical distribution of string lengths is skewed towards longer strings since the probability of a repeated string is larger for shorter relative to longer strings. With this sampling method, our maximum string length is 40, and we tokenize the (and)characters in addition to start, end, and padding tokens (appended to the end of the string) to ensure each sequence for classification had length 42 (including start and end tokens).

Our training and ID sets only have nested/equal-count strings labeled as `True` (ex: ()(())) and non-nested/unequal-count strings labeled as `FALSE` (ex: (()). Our OOD test set consists of non-nested/unequal-count parentheses sequences—strings with the same number of open and closed parentheses characters but arranged to be non-nested (ex:))((). Following this sampling scheme, we use ID and OOD test sets with 1000 examples.

Nested (ID positive) strings were generated via the classical uniform random nested parentheses generation algorithm Arnold & Sleep (1980). Non-nested/unequal-count (ID negative) strings and non-nested/equal-count (OOD) strings were produced through uniform random string generation, discarding nested strings.

We train models on H100 GPUs across 5 random seeds determining model weights and random operations. We also use 3 random shuffle seeds with our train set of size 200000, with 100000

positive examples—nested/equal-count parentheses strings—and 100000 negative examples—non-nested/unequal-count strings. During training, we repeat this set five times, so all models were exposed to 1 million total parentheses sequences (including 5 repeats of each) across the course of training. The random shuffle seed determines the order of data within each block of 200000 training examples, so during training, models are exposed to the same training data in five different orderings. Models with the same shuffle seed hyperparameter encounter the 1 million total training datapoints in exactly the same order (data within each block is shuffled in a consistent way).

C Training dynamics of different rules

Some OOD generalization rules can converge simultaneously with ID performance, whereas others take long to learn after the model successfully learns ID. In our setting, models can acquire and stabilize into the EQUAL-COUNT rule, but often take much longer to but generally take much longer to converge to the NESTED rule.

Although overall, models tend to classify OOD strings as `False` at the outset of training—likely because the `False` training examples, being sampled uniformly at random, are far more diverse—they rarely stabilize immediately at high rates of `False` (i.e., equivalent to a NESTED rule). The models that stabilize at a NESTED rule, as seen in Figure 5, usually stabilize long after ID convergence. In other words, we see an example of *structural grokking* (Murty et al., 2023). These results support the idea that NESTED is a more difficult rule to fully learn. Indeed, only three Transformer models adhere completely to the NESTED rule by classifying all OOD examples as `False`.

Furthermore, it appears that EQUAL-COUNT is used as an ID rule—learned as part of the initial fitting process—whereas NESTED often emerges much later than ID fitting. We therefore view NESTED as the product of a structural grokking transition enabled by regularization.

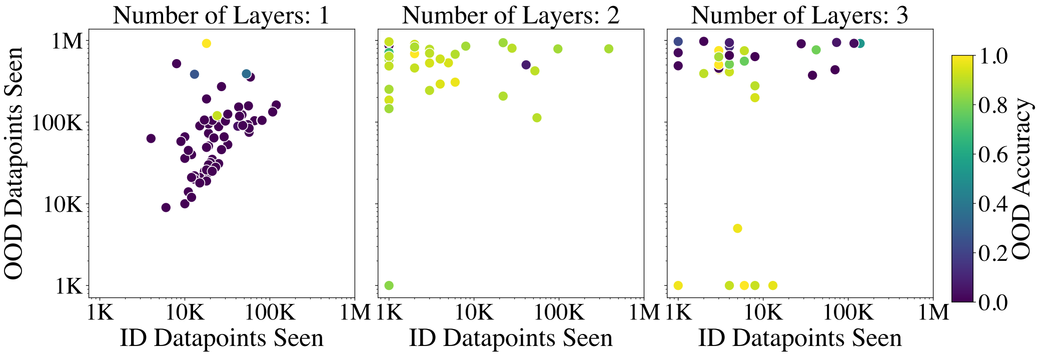
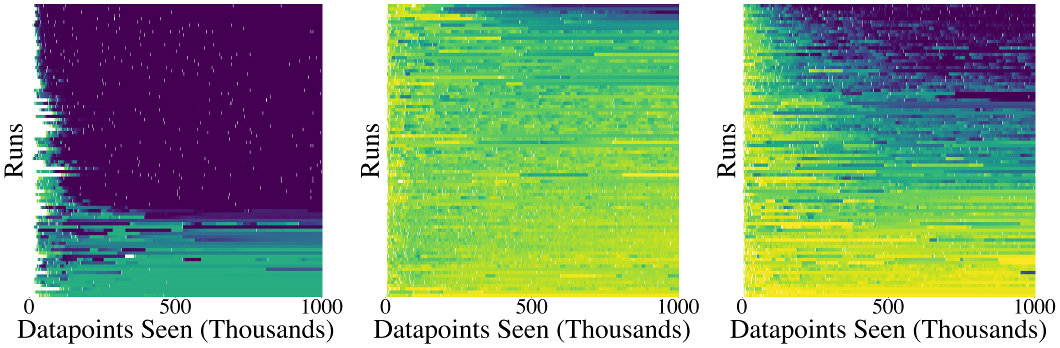


Figure 5: Illustration of generalization rules across training. ID convergence occurs when models achieve $> 99\%$ ID accuracy for $> 99\%$ of remaining datapoints seen—all Transformers achieve this metric after 385K datapoints. We define OOD convergence to either EQUAL-COUNT or NESTED as Transformers achieving < 0.2 or > 0.8 accuracy for $> 99\%$ of the rest of the model run, respectively, after seeing at most 975K datapoints—55% of transformers achieve this metric.

In most cases, models which converged to a rule did not flip-flop to a different rule. However, we observe that rarely, a model can change its generalization behavior partway through training (Figure 6) even after apparently converging to a rule. In particular, one 1-layer model initially generalized according to FIRST-SYMBOL, but later switched to adhering to EQUAL-COUNT. This flip-flopping behavior is associated with a sudden temporary reduction in ID performance, suggesting that the model has "forgotten" what it has learned, and then quickly re-learns a different rule.

C.1 Breakdown of training dynamics by depth and regularization

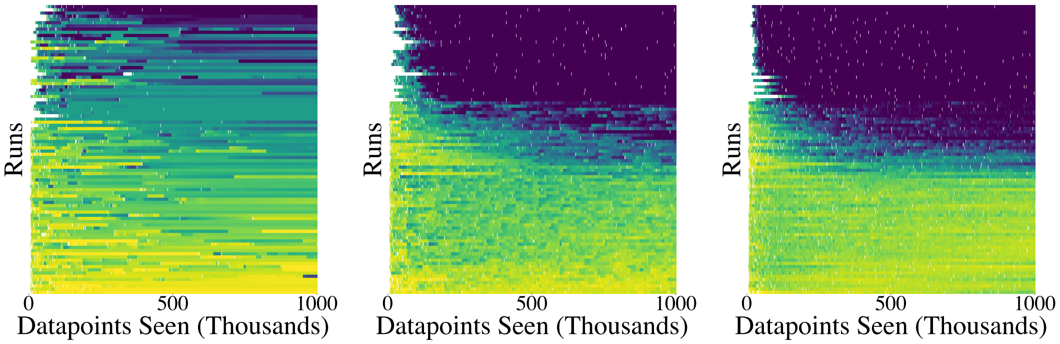
We can breakdown the training dynamics of models further by weight decay (Figure 6 and 7). As described in Figure 3, 1-layer Transformers primarily learn EQUAL-COUNT or FIRST-SYMBOL, 2-layer Transformers lergely largely learn NESTED, and 3-layer Transformers have a diverse set of OOD behaviors. Breaking down training dynamics by weight decay, we also see how more



(a) 1-layer Transformers (b) 2-layer Transformers (c) 3-layer Transformers

Figure 6: OOD accuracy of models across training, where each colored cell indicates the OOD accuracy of a particular run when ID accuracy is at least 0.99. Heatmaps showing model training dynamics broken down by depth, where more purple indicates adherence to the EQUAL-COUNT rule and more yellow to the NESTED rule (i.e., the same scale as Figure 5).

regularization makes models achieve different OOD performance for shorter amounts of time. With 0 weight decay, models maintain the same OOD accuracy even as they are exposed to more datapoints, but with 0.001 and 0.01 weight decay, they display more instability in their OOD performance.



(a) Weight decay 0 Transformers (b) Weight decay 1e-3 Transformers (c) Weight decay 0.01 Transformers

Figure 7: Heatmap showing model training dynamics broken down by weight decay where each colored cell indicates the OOD accuracy when ID accuracy is at least 0.99. Colors indicating OOD accuracy are on the same scale as Figures 5 and 6.