Can Interpretation Predict Behavior on Unseen Data?

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Interpretability research often aims to predict how a model will respond to targeted interventions on specific mechanisms. However, it rarely predicts how a model will respond to unseen *input data*. This paper explores the promises and challenges of interpretability as a tool for predicting out-of-distribution (OOD) model behavior. Specifically, we investigate the correspondence between attention patterns and OOD generalization in hundreds of Transformer models independently trained on a synthetic classification task. These models exhibit several distinct systematic generalization rules OOD, forming a diverse population for correlational analysis. In this setting, we find that simple observational tools from interpretability can predict OOD performance. In particular, when in-distribution attention exhibits hierarchical patterns, the model is likely to generalize hierarchically on OOD data—even when the rule's implementation does not *rely on* these hierarchical patterns, according to ablation tests. Our findings offer a proof-of-concept to motivate further interpretability work on predicting unseen model behavior.

1 Introduction

When can we claim to understand a system? One standard, that of the classic scientific method [14], requires *testable predictions of behavior under unseen conditions*. Accordingly, interpretability research often assesses proposed mechanisms by predicting the effect of a test-time mechanistic intervention [13] such as activation steering [32, 34, 19] or patching [20, 18, 37, 40]. In contrast, researchers rarely use their interpretations to predict the effect of a test-time *data* intervention or to predict model behavior on unseen *inputs*. This paper focuses on the latter objective; we interpret hidden representations to infer the model's implemented rules and predict its outputs on unseen data.

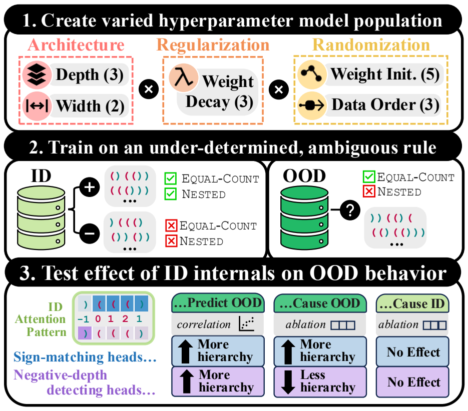


Figure 1: **Our approach.** In a population of independently trained classifiers, we correlate internal structures with OOD behaviors.

If we could predict model behavior under data distribution shifts, it would unlock entirely new interpretability applications. Well-understood instruments come with engineering tolerances; these limits correspond to edge cases where the instrument may fail. By providing tolerances and edge cases for AI models, we might debug them, offer recommendations for their reliable use, and even identify deployment failures in advance. For example, structures associated with specific languages and tasks may provide clues as to whether an LM can reliably compose them to fluently handle a given task in a particular language. However, current techniques may fall short of addressing this challenge

Submitted to Mechanistic Interpretability Workshop at NeurIPS 2025. Do not distribute.

because they fail to generalize on novel domains [17, 30]. Can we still use interpretations to predict a model's response to unseen data?

Using a synthetic task, we show that simple analysis of attention patterns can reveal a model's algorithm *even if* they are not used in its implementation. In other words, our interpretations are not *faithful* in a causal or mechanistic sense, but nonetheless reveal informative traces left by the algorithm. Interpretability-based intuitions then allow us to guess what algorithm is being executed, and therefore to predict how it will treat unseen data. Current interpretability work often seeks to reverse engineer a model by identifying features that allow us to *control* it, but we look for representational structures that allow us to *simulate* it. Either aim demonstrates understanding.

We wish to predict out-of-distribution (OOD) model behavior using only interpretations of internal model representations of in-distribution (ID) data. To this end, we train a large set of models with perfect ID validation accuracy and diverse OOD behavior in a synthetic setting. Our model population is trained on data with an ambiguous classification rule: models can achieve perfect ID accuracy using either a parenthesis **counting** rule (Equal-Count) or a hierarchical parenthesis **nesting** rule (Nested). We use an OOD test set to determine which rule each model follows.

In this diverse model population, we correlate internal behaviors with systematic rules, finding:

- **Independently trained models cluster around systematic generalization rules.** We identify clusters of models which implement Equal-Count and Nested by visualizing their OOD judgements (Section 3). Some unregularized training runs learn a simplistic heuristic instead of a valid rule, but only apply this heuristic OOD (Section 3.1). Further investigation provides evidence that models often pass through transient heuristic phases early in training, suggesting that vestigial circuits can still affect OOD judgments. In addition to weight decay, model depth and random variation influence rule learning (Section 3.2).
- **OOD generalization rules are intuitively predicted by internal representations of ID data.** We identify heads that encode hierarchical structure in their attention activations (Section 4.1). We show that models with these heads will usually apply Nested on unseen OOD data (Section 4.2).
- **Internal structures predict generalization rules even when the rule's implementation doesn't rely on them.** We differentiate circuits that are *causally* necessary in implementing a rule from those that *correlate* with the rule across a model population. Although Nested is associated with multiple types of hierarchical attention heads, ablation tests show that some types suppress, rather than support, the Nested rule (Section 4.3). Furthermore, the effect of ablation is only weakly correlated between ID and OOD data conditions, calling into question the robustness of findings in causal interpretability (Section 4.3.3). Drawing on the realism debate in philosophy of science, we advocate for instrumentalist alternatives to mechanistic intervention in validating model understanding (Section 5).

2 Methods and Experiment Setup

To create models that vary in their OOD generalization, we use a training dataset compatible with at least two distinct OOD generalization rules (Section 2.1). We study the resulting variation by training a large collection of models with different hyperparameters and random seeds (Section 2.2.).

2.1 Data setting

Our setting is inspired by work on systematic generalization from ambiguous training rules [22, 23, 24]. We base our dataset on the parentheses-balancing task Dyck-1 [33, 12, 24, 41]. Unlike standard parentheses-balancing settings, our training dataset is compatible with either Equal-Count, an unordered counting rule, or Nested, a hierarchical parentheses-balancing rule.

Our models are classifiers, not sequence generators. They verify that the input follows some rule and output a binary class $y \in \{\texttt{True}, \texttt{False}\}$. For a given n-length sequence $s = s_1 s_2 \ldots s_n$, where $s_i \in \{(,)\}$, each rule labels s as follows, where $\mathbf{1}(\cdot)$ is the indicator function.

- Equal-Count is `True` if s has the same number of open and close parentheses:

$$\sum_{i=1}^{n} \mathbf{1}(s_i = () = \sum_{i=1}^{n} \mathbf{1}(s_i =)). \tag{1}$$

- Nested is True if s forms a recursively nested tree. Note that all Nested sequence are Equal-Count, but the converse is not always true. In addition to Equation 1, s must fulfil:

$$\forall j \in \{1, \ldots, n\} \qquad \sum_{i=1}^{j} \mathbf{1}(s_i = \text{(}) \geq \sum_{i=1}^{j} \mathbf{1}(s_i = \text{)}). \tag{2}$$

Our training set is compatible with both Equal-Count and Nested: every input sequence satisfies either *both* or *neither* of Equations 1 and 2. Thus, *a model can perfectly classify ID data by learning either rule*. We test which rule each model learned using an OOD set of sequences that are Equal-Count but not Nested, so they have the same number of (and) tokens but not in a nested order. As a convention, we define accuracy according to Nested, i.e., a model has 100% OOD accuracy if all OOD labels are False and 0% if all labels are True.

We create ID sequence-label pairs compatible with both rules and we create OOD sequences where the rules disagree. Each sequence is generated randomly with length sampled $n \sim \text{Binomial}(40, 0.5)$ (see Appendix B). The OOD test set contains 1K sequences which fulfill Equal-Count but not Nested (e.g.,))((()). Our 1M-example train set and 1K-example ID validation set are label-balanced, containing 50% negative examples which fulfill neither Equal-Count nor Nested (e.g., (()))) and 50% positive examples which fulfill both Equal-Count and Nested (e.g., ()(())).

2.2 Models and attention

We train Transformers with causal self-attention and an input length of $L = 42$ (see Section **??** for details). Each sequence s consists of a BOS (beginning-of-sequence) token at s_0, a sequence of $n \leq 40$ parentheses, an EOS (end-of-sequence) token at s_{n+1}, and $L - n - 2$ padding tokens starting at EOS. We output a binary class $\hat{y} \in \{\texttt{True}, \texttt{False}\}$ at the index of the EOS token in the final layer.

Let $A \in \mathbb{R}^{k \times k}$ represent the attention activations of a given head on input s. Because our models output a classification label at the EOS token, we are interested in attention activations at its index $n + 1$. For index $i \in \{1, \ldots, n\}$, we therefore define $a_{\texttt{EOS}}(i)$ as attention to token s_i:

$$a_{\texttt{EOS}}(i) = A_{n+1,i}. \tag{3}$$

We train a population of classifier models based on the minGPT architecture [16] with hidden dimension 64 and causal attention. Following Vaswani et al. [39], we use reshaping for multi-head attention, so the model's overall parameter count is the same regardless of per-layer head count W. We set the learning rate $\eta = 0.0001$ with no dropout. All trained models stablize to an ID validation accuracy of at least 99% after at most 900K training examples (Appendix Figure 11).

We grid sweep over other hyperparameters to create a diverse model population. We train models with depths of $D \in \{1, 2, 3\}$ layers and widths of $W \in \{2, 4\}$ attention heads per layer. We also vary optimizer weight decay $\lambda \in \{0, 0.001, 0.01\}$. In each hyperparameter setting, we train models with 5 random seeds for weight initialization and 3 seeds for dataset shuffle order. This grid sweep results in 15 models per hyperparameter configuration and 270 models in total.

3 Generalization Behavior

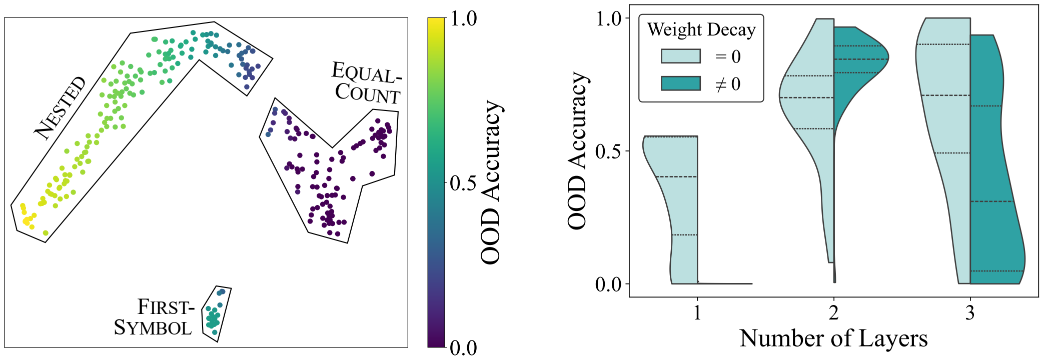
All Transformer models achieve high ID validation accuracy, but their OOD behaviors vary widely.

3.1 Models cluster around systematic rules

We visualize all models according to their OOD output probabilities in Figure 2a. One cluster has near-0% OOD accuracy and another has high OOD accuracy, supporting existing claims [28, 43] that *rule-following is a categorical phenomenon, not a continuum*. Note, however, a third cluster in which models exhibit nearly identical judgments. We now characterize this cluster.

3.1.1 The heuristic cluster

The outlier cluster represents a simple heuristic which we call First-Symbol: a sequence is labeled True if its first symbol is (and False if its first symbol is). Some models use this heuristic OOD, but they do *not* apply it to ID data.



(a) T-SNE visualization of each model's output probabilities on the OOD test set. Observe three rule clusters: (1) Equal-Count with low OOD accuracy; (2) Nested with higher OOD accuracy; (3) First-Symbol with approx. 0.55% OOD accuracy.

(b) Weight decay and depth have a strong impact on OOD rule selection. The Mann-Whitney U test finds significant differences in OOD accuracy distribution between the absence and presence of weight decay across all depths ($p \ll 0.01$ across layers).

Figure 2: **Similarly trained models vary in systematic generalization rules.**

All models have high (>99%) accuracy on the ID validation set, whereas First-Symbol would incorrectly label half of all negative examples (Appendix Table 2).

We hypothesize that the First-Symbol heuristic is implemented by a vestigial circuit which the model eventually learns to suppress on ID—but not OOD—data. Prior literature [36, 5, 10] suggested that vestigial circuits are pruned by weight decay and that their presence limits ID generalization from small training sets. As predicted by our hypothesis, then, the First-Symbol heuristic only governs 1-layer models trained *without* weight decay. In similar models *with* weight decay, the heuristic does not survive regularization, so final OOD accuracy is always 0% (Figure 2b). As further evidence of visibility, 1-layer Equal-Count models (Figure 3) often pass through apparent heuristic phases while training, during which OOD accuracy matches that of First-Symbol models. We therefore conclude that *a vestigial circuit—which has no detectable impact on ID behavior—can still affect OOD judgments.*

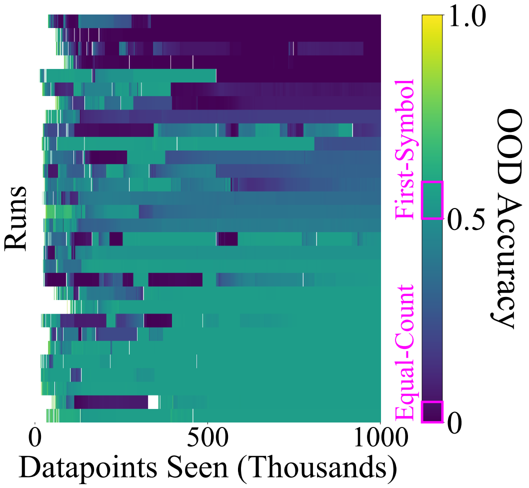


Figure 3: **Unregularized models can enter transient heuristic stages during training.** Intermediate checkpoints of all one-layer models without weight decay, sorted by final OOD accuracy. The gradient represents an intermediate checkpoint's OOD accuracy. Each cell is left white if ID accuracy falls below 0.99, i.e., if the checkpoint cannot model its training distribution. The color bar is marked in magenta for sections corresponding to First-Symbol and Equal-Count behavior. (For further training analysis, see Appendix D.)

3.2 Factors in rule selection

The dominant factors in rule selection are model depth and weight decay regularization. Other factors are detailed in Appendix C, including an extension of existing findings [1, 22, 38, 29] that recurrent architectures, but not Transformers, favor hierarchical generalization rules.

Model depth Overall, Figure 2b shows that 2- and 3-layer models can learn either Nested or Equal-Count, depending on training conditions and random variation. In contrast, 1-layer models only learn First-Symbol or Equal-Count. Our results indicate shallower models learn simple counting rather than hierarchical rules, and that 2-layer models tend more toward hierarchy than

3-layer models. We therefore observe an inverted U-shape of hierarchical inductive bias by model depth, mirroring results from previous work [24].

Weight decay Regularization pushes a model towards the distribution's modes and therefore to more consistently systematic rules. In particular, higher weight decay leads to more concentrated distributions of OOD behaviors across the model population (Figure 2b). Among 1- and 3-layer models, non-zero weight decay increases the preference for EQUAL-COUNT, but among 2-layer models it increases the preference for NESTED. Although these findings support the notion that regularization promotes simple systematic rules, they complicate reductionist narratives around what rules might be described as simpler. Instead of promoting a universally simple rule, regularization promotes different rules depending on the architectural hyperparameters. Intuitively, universal systematic rules are simpler than example memorization, but which rule is simplest—and therefore most favored by regularization—depends on the setting.

4 Using Interpretability to Predict Behavior

Next, we inspect the activations produced by each attention head. Using intuitive explanations of how ID inputs are processed, we predict model decisions on unseen OOD inputs. Appendix F confirms that attention patterns add predictive value even if a model's hyperparameter settings are known.

In particular, we will show that models following the hierarchical rule NESTED also display hierarchical attention patterns on ID data. These models, despite their shared outputs, can employ subtly different internal mechanisms with different causal roles. Not only do *some mechanisms lead to better OOD generalization then others*, but *some modules generalize better by maintaining the same systematic role in- and out-of-distribution*. However, hierarchical attention patterns predict that the model will follow NESTED even when these attention patterns do not directly implement NESTED.

4.1 Hierarchical attention patterns

Certain heads exhibit systematic, interpretable attention patterns at the EOS token. We specifically focus on patterns based on the **depth** of a position's token within the input sequence's latent tree. We will connect these intrinsically hierarchical heads to the NESTED generalization rule.

4.1.1 Tracking token depth

We identify heads that encode an input's hierarchical structure. These heads attend to tokens based on their depth in the latent tree structure of input s, defined as follows. Let the number of open (and close) tokens that appear through index j be $o(j) = \sum_{i=1}^{j} \mathbf{1}(s_i = \texttt{(})$ and $c(j) = \sum_{i=1}^{j} \mathbf{1}(s_i = \texttt{)})$, respectively. Then the depth at index $j \in \{1, \dots, n\}$ is:

$$d(j) = o(j) - c(j) \tag{4}$$

Note that if a sequence contains any negative depth tokens, it cannot be properly nested, as it fails the NESTED criterion (Equation 2). An EQUAL-COUNT sequence is also NESTED if and only if it contains no negative-depth tokens, so *all* examples in the OOD set—composed of EQUAL-COUNT, but not NESTED), sequences—must have at least one negative depth index. Intuitively, then, attention patterns which track position depth can signal that a model is behaving hierarchically, closer to NESTED.

There are two ways for an attention output to **track depth** on a given sequence: a head can preferentially attend either to negative or to non-negative depth tokens. Given an input s, we say:

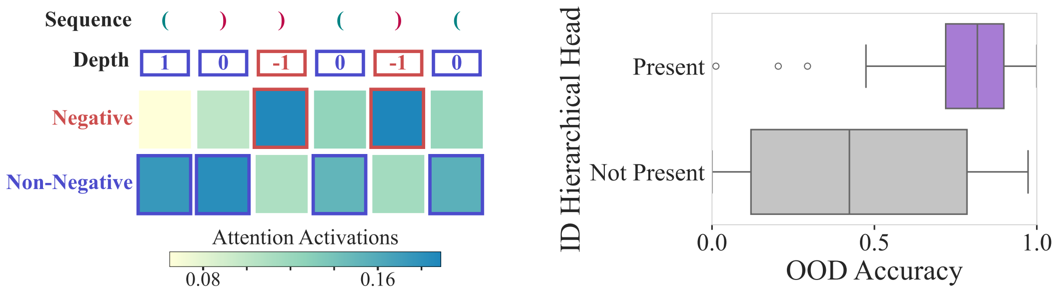
- An attention head favors *negative depth tokens* on input s if there exists threshold $t > 0$ such that:

$$\forall j \in \{1, \dots, n\}: \qquad a_{\text{EOS}}(j) \geq t \text{ iff } d(j) < 0 \tag{5}$$

- An attention head favors *non-negative depth tokens* on input s if the conditions are reversed, i.e., there exists $t > 0$ such that:

$$\forall j \in \{1, \dots, n\}: \qquad a_{\text{EOS}}(j) \geq t \text{ iff } d(j) \geq 0 \tag{6}$$

Examples of each of each pattern are shown in Figure 4a. We say an attention head is **depth-tracking on a given input** if it favors *either* negative *or* non-negative depth tokens on that input.



(a) Examples of attention patterns favoring negative (top) and non-negative (bottom) depth tokens. The input sequence is shown above the heatmaps, and each token's tree depth is displayed below.

(b) OOD accuracy of 2- and 3-layer models with and without ID hierarchical heads. Across models (and for each hyperparameter; see Appendix F), these heads correlate with OOD accuracy.

Figure 4: **Hierarchical attention patterns correlate with hierarchical generalization rules.**

4.1.2 Hierarchical head types

Some heads reliably reflect hierarchical structure by tracking depth on each input, making them hierarchical heads across a dataset. A head is a **hierarchical head** on a given dataset if it tracks depth on at least 80% of **mixed-depth inputs**, defined as sequences containing both negative and non-negative depth tokens (a category which includes all OOD examples). Heads which are depth-tracking ID typically behave as hierarchical heads OOD, but not always: 23% of ID hierarchical heads do not behave as OOD hierarchical heads. We divide hierarchical heads into two subtypes:

- **Negative-depth detector heads.** These heads consistently assign more attention to negative-depth indices, i.e., positions preceded by more) than (tokens. We say a head is a *negative depth detector* on a given dataset if it favors negative depth tokens on at least 80% of mixed-depth sequences.
- **Sign-matching heads.** These heads favor negative or non-negative depth depending on the sign of the final token's depth in the sequence. If the final token has negative depth, the head will favor negative depth tokens; likewise, if the final token has non-negative depth (as in all OOD sequences), the head will favor non-negative depth tokens. We say an attention head is a *sign-matching head* on a given dataset if it follows this rule on at least 80% of mixed-depth sequences.

Among models with ID hierarchical heads, 81% have sign-matching heads, 9% have negative depth detecting heads, and 8% have both; only 2% of models with hierarchical heads have neither subtype. We therefore focus on these subtypes, which cover almost all ID hierarchical heads.

4.2 Hierarchical attention on ID data *predicts* hierarchical rules on OOD data

We return to our overarching question: Can these model internals intuitively suggest which function the model implements, thereby predicting its treatment of unseen OOD data? If we claim to understand a model, we should know its behavior under many unseen conditions. We now demonstrate that expectations based on our interpretations can, in fact, reliably predict OOD model behavior.

As seen in Figure 4b, models which contain at least one ID hierarchical head are more likely to follow the Nested rule. This result also holds separately across 2- and 3-layer models (Appendix F). These findings confirm our intuitive hypothesis: that ID hierarchical representational structure is associated with OOD hierarchical generalization behavior.

We find that 1-layer models, which notably *do not* learn the Nested rule (Figure 2b), possess no hierarchical heads; indeed, these heads do not occur in the first layer of any model. Therefore, our models only learn depth-tracking if they have multiple layers, which may explain why 1-layer models never learn Nested.

Because our OOD sequences all end at token depth zero, a sign-matching head favors non-negative depth tokens OOD. We do not observe that any *ID negative-depth detector heads* switch to favoring non-negative depth tokens on OOD sequences. However, 25% of *ID sign-matching heads* become negative-depth detectors OOD. Given that a mechanism's behavior can change so substantially between input datasets, it would be challenging to describe any internal mechanism in a way that applies to its behavior in new domains. Nonetheless, these structures hint at a *holistic* understanding of the algorithm implemented by the model. This holistic understanding, unlike any specific mechanistic interpretation, applies across distribution shift.

4.3 Hierarchical attention may not *cause* hierarchical rules on OOD data

Although hierarchical heads correlate with the NESTED rule, correlation alone doesn't establish them as causal mechanisms. To determine causality, we intervene on attention activations and examine how model performance responds. These experiments demonstrate that an attention pattern might correlate with a systematic rule without supporting it causally—-in fact, we will see that the pattern may even, counter-intuitively, suppress the rule. Preventing models from displaying these attention patterns can thus enhance, rather than reduce, the correlated output behavior.

4.3.1 Ablation method

To investigate whether models rely on particular attention patterns, we measure their accuracy after *uniform* attention ablation. This ablation replaces every attention activation with a uniform activation which attends equally to all prior tokens.

Our intervention preserves all other components of the Transformer and its activation, but strips any influence of depth-tracking attention or other attention activation patterns. Wen et al. [41] demonstrated that uniform attention is sufficient to implement a parentheses-balancing task, the generative form of our NESTED classification rule. In such cases where the model does not rely on its attention patterns, replacing attention activations with uniform attention will not harm OOD accuracy, and could even improve OOD accuracy in cases where attention does not support NESTED behavior.

Note that when we apply uniform attention ablation, we apply it to *all* attention across all heads. By flattening all attention patterns, we control for all head types simultaneously and uniformly across models. Our ablation applies to attention as a whole, and is not a targeted ablation of the hierarchical heads alone. Then, one limitation to keep in mind is that some hierarchical head types may frequently co-occur with other head types that are more important. Therefore, it is difficult to solidify precise claims about causal dependencies on *specific* heads through universal uniform ablation. All heads must, however, be ablated because in some models, multiple heads are the same hierarchical type.

4.3.2 Ablation results

We find that certain types of depth-tracking attention, although correlated with NESTED, actually lower the OOD accuracy of the model when present (Figure 5). Ablating the attention of models with OOD *negative-depth* heads damages OOD accuracy, as might be expected if negative depth tracking is a key mechanism in implementing NESTED. By contrast, ablating the attention of models with OOD *sign-matching* heads actually *improves* OOD accuracy. The latter type of hierarchical head, although just as correlated with hierarchical generalization as the former type, is not a key mechanism in the rule's implementation. Instead, it interferes with systematic hierarchical generalization.

Our findings resist simple narratives about the mechanisms which implement a model's underlying function. Although all hierarchical heads are similarly correlated with hierarchical generalization (Appendix Figure 14), not all of them implement that rule under causal tests. Therefore, the complexity of even simple models can present situations in which a model is resistant to causal analysis, but its correlational interpretations still provide predictive value.

Unlike the FIRST-SYMBOL circuit, OOD sign-matching attention heads are *promoted* by regularization. They occur more frequently in models trained with weight decay than without (Appendix Figure 13). We therefore reject the position that these attention patterns are vestigial. Instead, we conjecture that these heads either develop as spandrels (side effects of learning NESTED) or that the hierarchical head somehow malfunctions under distribution shift. Disentangling module generalization failure like this is a ripe topic for future work.

4.3.3 Robustness to ablation is data dependent

Interpretability researchers commonly test proposed explanatory mechanisms by intervening on those specific mechanisms. We call this approach into question by demonstrating that a model can be *robust to an ablation on ID data but not OOD data*. Uniform attention ablation has substantial effects on

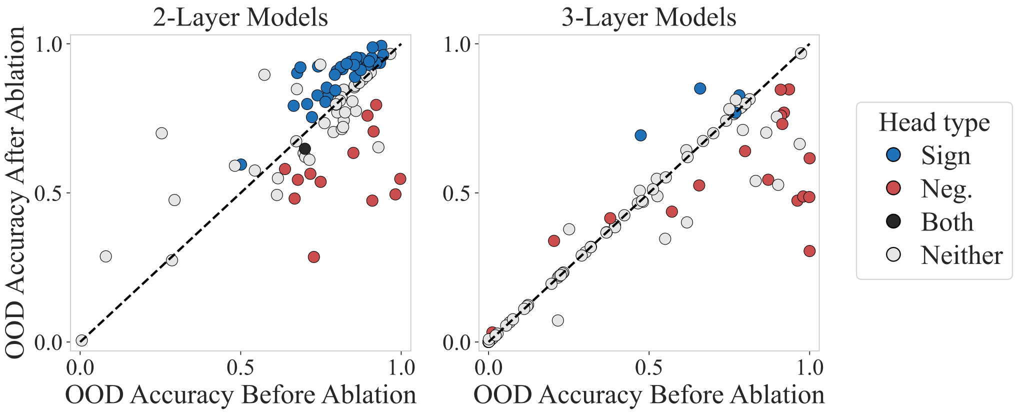


Figure 5: **Some hierarchical attention patterns damage the implementation of hierarchical rules.** OOD accuracy before and after applying uniform attention ablation to all attention heads. Each point represents a single model, colored by presence of an OOD sign-matching and/or negative depth detecting head. Ablation damages OOD performance in models represented by points below the diagonal, but improves OOD performance in those above the diagonal. In the former case, the attention pattern can be said to support the implementation of the NESTED rule, whereas in the latter case, the pattern suppresses the implementation of the rule. Although both head types are correlated with NESTED, only negative depth detection causally supports the rule.

OOD accuracy (Figure 5), but leaves ID validation accuracy nearly unchanged, reduced by only 0.5% on average (Appendix H); moreover, effects of ablation on ID and OOD accuracy are only weakly correlated ($\rho = 0.24$, $p < 0.01$). These results suggest that a model might be able to compensate for the loss of a mechanism on most data, while still heavily relying on the mechanism when judging OOD edge cases. Although the model may be applying the same rule ID and OOD at the output, the distribution shift reveals brittle elements of its implementation.

We posit that models could contain many redundant backup circuits that compensate for ablation ID, but become unreliable under distribution shift. Without redundancy, model judgments become more sensitive to ablations on the remaining circuits. Based on these findings, we propose that a negative result from an ID causal intervention provides weak evidence against our proposed mechanisms.

5 Discussion and Conclusions

Our findings show that interpretability can be used to predict future model behavior on unseen inputs. If we can identify similar cases in real world settings, the consequences for model evaluation could be enormous. In modern machine learning, data presents the main bottleneck to performance, so if we could propose edge cases where a model is likely to fail, we could efficiently evaluate and improve model robustness.

Along with this new application of interpretability, our findings also suggest new ways of evaluating interpretations. We judge interpretations by their ability to predict model behavior on unseen inputs, as we have demonstrated. We can also evaluate causal interpretations based on their robustness to distribution shift, as we have shown some ablation results to be surprisingly brittle. These desiderata can be added to the current evaluation toolkit, which often focuses on in-distribution model behaviors when measuring ablation response or correlation with data properties.

5.1 Interpretability and causality

Our approach is inspired by correlational studies in natural sciences like biology. In genetics, for example, correlational “twin studies” that compare fraternal and identical twin populations are highly valued. Some interventional studies that rely on genetic engineering may actually be *less* informative because single-gene editing can sabotage unrelated processes through complex interactions between genes. Likewise, rather than focusing solely on mechanistic interventions, we leverage correlations over model populations to test our interpretations of model representations.

In contrast, prior work in mechanistic interpretability commonly characterizes the role of individual modules by measuring damage to model performance under ablation. The resulting findings are often compatible with multiple analyses that can only be differentiated by labor-intensive, precisely targeted ablations; moreover, proposed interpretation methods may not transfer to new settings [31, 42, 20]. By focusing on causal intervention alone to test the faithfulness of explanations, the literature on understanding attention has produced skepticism and fierce debate [3].

Our work underscores the importance of caution in overinterpreting results from causal interventions. We provide an example of a case in which a causal intervention—attention activation patching—barely affects in-distribution performance. Naive interpretation of this result might suggest that the patched attention patterns have little effect on model behavior. However, this causal analysis fails to capture the relevance of these attention patterns to the model's out-of-distribution generalization. By instead conducting a correlational analysis across a model population, we discover that depth-tracking attention patterns are predictive of hierarchical generalization behavior.

We believe interpretations of model internals can be valuable independent of causal analyses, as representational geometry may give clues as to which capabilities a model has and how it may perform under distribution shift. A model's algorithm can leave observable traces, which constitute meaningful signals regardless of their causal role. In other words, hidden representations can provide a *proxy* for the model's algorithm, even if they are not employed in its implementation.

5.2 Interpretability and scientific realism

Of the many controversies in philosophy of science, few are more central than the rivalry between realism and instrumentalism [25]. Realists claim that the concepts used in scientific models, from quarks to gravity, are specific objects and forces acting on the world [27]. Instrumentalists, by contrast, argue that the epistemic goal of science is not to uncover fundamental truths about the world, but to make predictions about their observable outcomes [11]. To an instrumentalist, a quark might not be a real object, but simply a convenient variable in a predictive model. In the philosophy of mind, instrumentalists may even deny that internal beliefs, desires, or intentions are real phenomena, while still acknowledging that these concepts might help predict a person's behavior [6, 8, 7].

Our objective is an instrumentalist one. Rather than insisting that we have identified a true hierarchical mechanism, we treat hierarchical structure as part of our scientific model of network behavior. Even at a small scale in a simple synthetic environment, complete causal analysis is a challenge. From an instrumentalist perspective, however, we do not need a true understanding of the objects and mechanisms within a model, as long as we can make useful inferences from the traces they leave.

5.3 Interpretability and levels of analysis

If we are using interpretability to understand a model, we must consider what level that understanding operates at. In Marr's levels of analysis [21], mechanistic interpretability research typically works at the *implementational* level. By contrast, our approach only seeks the *algorithmic* level of analysis, a high-level description of model behavior. In our view, full understanding of a model requires multiple levels of analysis. We hope to see the interpretability field continue to develop through a diversity of objectives and approaches; algorithmic-level interpretability is only one under-explored direction.

Limitations

This paper offers a proof of concept that, in some circumstances, simple correlational studies of model internals can provide valuable insights which would require intensive labor—and possibly be intractable—to accomplish through a causal analysis. Our study, however, uses a synthetic setting and it it would take further effort to repeat a similar analysis in a realistic scenario.

Because the models we study are necessarily small, the particular patterns we observe might apply less in large models. In particular, larger scales may reduce the substantial impact of random variation and the isolated role of attention heads. Both random variation and specialized attention heads have been observed in larger models, but they become increasingly challenging to assess when models are expensive to train and high dimensional in their representations.

References

[1] S. Abnar, M. Dehghani, and W. Zuidema. Transferring inductive biases through knowledge distillation, 2020. URL https://arxiv.org/abs/2006.00555.

[2] 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.

[3] A. Bibal, R. Cardon, D. Alfter, R. Wilkens, X. Wang, T. François, and P. Watrin. Is attention explanation? an introduction to the debate. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3889–3900, 2022.

[4] S. C. Y. Chan, A. Santoro, A. K. Lampinen, J. X. Wang, A. Singh, P. H. Richemond, J. McClelland, and F. Hill. Data distributional properties drive emergent in-context learning in transformers, 2022. URL https://arxiv.org/abs/2205.05055.

[5] X. Chen, R. Pan, X. Wang, F. Tian, and C.-Y. Tsui. Late breaking results: Weight decay is all you need for neural network sparsification. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–2, 2023. doi: 10.1109/DAC56929.2023.10247950.

[6] P. M. Churchland. Eliminative materialism and the propositional attitudes. *the Journal of Philosophy*, 78(2):67–90, 1981.

[7] D. C. Dennett. *The intentional stance*. MIT press, 1989.

[8] D. C. Dennett. Real patterns. *The journal of Philosophy*, 88(1):27–51, 1991.

[9] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping, 2020. URL https://arxiv.org/abs/2002.06305.

[10] D. Doshi, A. Das, T. He, and A. Gromov. To grok or not to grok: Disentangling generalization and memorization on corrupted algorithmic datasets, 2024. URL https://arxiv.org/abs/2310.13061.

[11] P. Duhem. *The aim and structure of physical theory*. na, 1954.

[12] J. Ebrahimi, D. Gelda, and W. Zhang. How can self-attention networks recognize Dyck-n languages? In T. Cohn, Y. He, and Y. Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4301–4306, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.384. URL https://aclanthology.org/2020.findings-emnlp.384.

[13] A. Geiger, D. Ibeling, A. Zur, M. Chaudhary, S. Chauhan, J. Huang, A. Arora, Z. Wu, N. Goodman, C. Potts, and T. Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability, 2024. URL https://arxiv.org/abs/2301.04709.

[14] B. Hepburn and H. Andersen. Scientific Method. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2021 edition, 2021.

[15] J. Juneja, R. Bansal, K. Cho, J. Sedoc, and N. Saphra. Linear connectivity reveals generalization strategies. In *International Conference on Learning Representations*, 2023.

[16] A. Karpathy. MinGPT transformer model, 2020. URL https://github.com/karpathy/minGPT.

[17] C. Kissane, robertzk, N. Nanda, and A. Conmy. SAEs are highly dataset dependent: a case study on the refusal direction. *Alignment Forum*, 2024. URL https://www.alignmentforum.org/posts/rtp6n7Z23uJpEH7od/saes-are-highly-dataset-dependent-a-case-study-on-the.

[18] J. Kramár, T. Lieberum, R. Shah, and N. Nanda. Atp*: An efficient and scalable method for localizing llm behaviour to components, 2024. URL https://arxiv.org/abs/2403.00745.

[19] S. Liu, H. Ye, L. Xing, and J. Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering, 2024. URL https://arxiv.org/abs/2311.06668.

[20] A. Makelov, G. Lange, and N. Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching, 2023. URL https://arxiv.org/abs/2311.17030.

[21] D. Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010.

[22] 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. *Transactions of the Association for Computational Linguistics*, 8:125–140, 2020. doi: 10.1162/tacl_a_00304. URL https://aclanthology.org/2020.tacl-1.9.

[23] R. T. McCoy, J. Min, and T. Linzen. Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance, 2020. URL https://arxiv.org/abs/1911.02969.

[24] S. Murty, P. Sharma, J. Andreas, and C. Manning. Grokking of hierarchical structure in vanilla transformers. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 439–448, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.38. URL https://aclanthology.org/2023.acl-short.38.

[25] S. Okasha. *Philosophy of science: very short introduction*. Oxford University Press, 2016.

[26] J. Petty, S. van Steenkiste, I. Dasgupta, F. Sha, D. Garrette, and T. Linzen. The impact of depth on compositional generalization in transformer language models, 2024. URL https://arxiv.org/abs/2310.19956.

[27] H. Putnam. *Mathematics, Matter and Method: Volume 1, Philosophical Papers*, volume 1. cup Archive, 1975.

[28] T. Qin, N. Saphra, and D. Alvarez-Melis. Sometimes I am a tree: Data drives unstable hierarchical generalization, 2024. URL https://arxiv.org/abs/2412.04619.

[29] N. Saphra and A. Lopez. LSTMs compose—and Learn—Bottom-up. In T. Cohn, Y. He, and Y. Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2797–2809, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.252. URL https://aclanthology.org/2020.findings-emnlp.252.

[30] L. Smith, S. Rajamanoharan, A. Conmy, CallumMcDougall, T. Lieberum, J. Kramár, R. Shah, and N. Nanda. Negative results for SAEs on downstream tasks and deprioritising SAE research (GDM mech interp team progress update #2), 2025. URL https://www.alignmentforum.org/posts/4uXCAJNuPKtKBsi28/negative-results-for-saes-on-downstream-tasks.

[31] D. Stander, Q. Yu, H. Fan, and S. Biderman. Grokking group multiplication with cosets, 2024. URL http://arxiv.org/abs/2312.06581.

[32] N. Subramani, N. Suresh, and M. E. Peters. Extracting latent steering vectors from pretrained language models, 2022. URL https://arxiv.org/abs/2205.05124.

[33] M. Suzgun, S. Gehrmann, Y. Belinkov, and S. M. Shieber. Memory-augmented recurrent neural networks can learn generalized Dyck languages, 2019. URL https://arxiv.org/abs/1911.03329.

[34] D. Tan, D. Chanin, A. Lynch, D. Kanoulas, B. Paige, A. Garriga-Alonso, and R. Kirk. Analyzing the generalization and reliability of steering vectors, 2025. URL https://arxiv.org/abs/2407.12404.

[35] Y. Tay, M. Dehghani, J. Rao, W. Fedus, S. Abnar, H. W. Chung, S. Narang, D. Yogatama, A. Vaswani, and D. Metzler. Scale efficiently: Insights from pre-training and fine-tuning transformers, 2022. URL https://arxiv.org/abs/2109.10686.

[36] H. Tessier, V. Gripon, M. Léonardon, M. Arzel, T. Hannagan, and D. Bertrand. Rethinking weight decay for efficient neural network pruning. *Journal of Imaging*, 8(3):64, Mar. 2022. ISSN 2313-433X. doi: 10.3390/jimaging8030064. URL http://dx.doi.org/10.3390/jimaging8030064.

[37] E. Todd, M. L. Li, A. S. Sharma, A. Mueller, B. C. Wallace, and D. Bau. Function vectors in large language models, 2024. URL https://arxiv.org/abs/2310.15213.

[38] K. Tran, A. Bisazza, and C. Monz. The importance of being recurrent for modeling hierarchical structure, 2018. URL https://arxiv.org/abs/1803.03585.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[40] J. Vig, S. Gehrmann, Y. Belinkov, S. Qian, D. Nevo, Y. Singer, and S. Shieber. Investigating gender bias in language models using causal mediation analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

[41] K. Wen, Y. Li, B. Liu, and A. Risteski. Transformers are uninterpretable with myopic methods: a case study with bounded Dyck grammars. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2023. URL https://par.nsf.gov/biblio/10489627.

[42] F. Zhang and N. Nanda. Towards best practices of activation patching in language models: Metrics and methods, 2024. URL https://arxiv.org/abs/2309.16042.

[43] R. Zhao, N. Saphra, and S. M. Kakade. Distributional scaling laws for emergent capabilites, 2024. URL https://openreview.net/pdf?id=e8eo9iEFaO.

Software and Data

We release the weights for the 270 models we train at five checkpoints, as well as our training and evaluation data and code, in order to facilitate future work into the variability of Transformer behavior and its underlying factors: https://anonymous.4open.science/r/id-predict-ood-D6F0.

A Glossary

Term	Definition and Usage
Depth	Parentheses sequence token chracteristic. At index j, the token depth is $o(j) - c(j)$ where $o(j)$ and $c(j)$ are the cumulative counts of (and) up to j (Equation 4).
Hierarchical head	Head that tracks a particular type of hierarchical attention pattern. Specifically, we evaluate if the EOS attention weights persistently favor either negative or non-negative depth tokens (e.g. Equation 5 or 6) on $\geq$80% of mixed-depth sequences. This head type encompasses negative-depth detectors and sign-matching heads, and it is typically found in 2- and 3-layer models.
EOS	The end-of-sequence token where attention patterns are inspected.
Equal-Count	Rule that a model can learn. `True` iff the number of open and close parentheses in a sequence are equal, regardless of their ordering (Equation 1). **Marker**: Predicts `True` for all OOD inputs (0% OOD accuracy).
First-Symbol	Rule that a model can learn. Predicts `True` if first token is (, `False` otherwise. This rule occurs in 1 layer models with 0 weight decay. **Marker**: $\sim$55% OOD accuracy.
Depth-tracking attention	A hierarchical attention pattern which reflects the tree structure of the preceding input sequence at an EOS token, as described in Equations 5 and 6.
ID data	In-distribution training and validation data where negative examples satisfy neither the Equal-Count nor Nested rules.
Mixed-depth sequence	Sequence with at least one negative- and one non-negative-depth token. Note all OOD examples are mixed-depth.
Nested	Rule that a model can learn. Returns `True` iff the input is properly nested (i.e, satisfies both Equal-Count and depth non-negativity, Equation 2). **Marker**: Predicts `False` on all OOD examples (100% OOD accuracy).
Negative-depth detector head	Depth-tracking head that consistently attends to negative-depth tokens on $\geq$ 80 % of mixed depth sequences. **Behavior**: Their presence is correlated with the hierarchical Nested rule OOD, but ablating this head decreases models' hierarchical behavior.
OOD data	Out-of-distribution test set where negative examples satisfy Equal-Count but not Nested. Higher accuracy is associated with Nested.
Sign-matching head	Depth-tracking head that attends to tokens matching the sign of the final parentheses token's depth. In other words, these heads attend to negative depth tokens if the last token is negative and non-negative depth tokens if the last is non-negative on $\geq$ 80% of mixed-depth sequences. **Behavior**: Their presence is correlated with the hierarchical Nested rule OOD, but ablating this head increases models' hierarchical behavior.
Attention ablation	Flattens attention to a uniform weight distribution. Used to test causal importance of particular attention heads and patterns.
Vestigial circuit	A sub-circuit used early in training, but is not necessary for model performance at the end (e.g., First-Symbol circuit in unregularized 1-layer models). At the end of training, such circuits may be associated with rules applied OOD but not ID.

Table 1: Glossary of key terms used in this paper.

B Dataset Generation Details

We can think of the sequence length distribution as though we are generating NESTED trees with a 50% probability of recursing at each node, but discarding identical sequences. Each sequence of symbols, sampled uniformly at random, is then sorted according to which rule it follows.

B.1 Dataset parentheses sampling

We create datasets as follows:

1. Sample a sequence length n from a $\text{Binomial}(40, 0.5)$ distribution, with mean 20 and variance 10. These properties ensure that our samples are concentrated around a reasonable center, reducing extreme sequence lengths that could occur with other distributions like the Uniform. Also note that our maximum sequence length is 40.
2. Generate a uniformly random parentheses sequence of length n with the desired attributes.
   - To generate a uniformly random sequence that is neither EQUAL-COUNT nor NESTED, we choose each character independently from the set { (,) }. If the resulting sequence satisfies EQUAL-COUNT, we discard it and generate a new one.
   - To generate a uniformly random sequence that is EQUAL-COUNT but not balanced, we randomly permute $n/2$ (parentheses and $n/2$) parentheses. If the resulting sequence is NESTED, we discard it and generate a new one.
   - To generate a uniformly random sequence that is NESTED, we use the algorithm of Arnold and Sleep [2].
3. If the sequence generated does *not* already appear in the dataset, add it to the dataset.

Thus, each length-n sequence s with the desired attributes is equally likely to be chosen, and it is chosen at most once. Since we discard repeats, the empirical distribution of sequence lengths is skewed towards longer sequences, as short sequences are likely to be repeated.

We tokenize the (and) characters in addition to start, end, and padding tokens (`BOS`, `EOS` and `PAD`, respectively, with `PAD` appended to the end of the sequence) to ensure each sequence for classification has length 42 (including start and end tokens). In other terms, we create a sequence of form:

$$s_0 s_1 \dots s_n \dots s_{41},$$

where s_0 is the beginning-of-sequence token `BOS`, s_1 through s_n make up the n-length parentheses sequence s, s_{n+1} is the end-of-sequence `EOS` token, and s_{n+2} through s_{41} are `PAD` tokens.

Our ID datapoints are randomly split into training and validation datasets. Each ID set contains the same number of `True` examples (following both EQUAL-COUNT and NESTED) and `False` examples (following neither EQUAL-COUNT nor NESTED). Our OOD test set consists of parentheses sequences which follow EQUAL-COUNT but not NESTED, i.e., sequences with the same number of open and closed parentheses characters, but in which the parentheses are not properly nested (ex:))((). See Table 2 for examples.

Empirically, we find some models classify sequences by a FIRST-SYMBOL heuristic OOD. These models check whether $s_1 =$ (and label an OOD sequence as `True` if the first character is (and `False` if it is). (In-distibution, no models follow FIRST-SYMBOL, which would fail to achieve full accuracy ID.)

B.2 Random data order implementation

Our train set contains 200000 distinct datapoints. 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 seed used for data ordering, or "shuffle seed," determines the order of data within each block of 200000 training examples, so during training, a single model is exposed to the same examples 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).

Dataset	EQUAL-COUNT	NESTED	Possible s_1	Example	FIRST-SYMBOL
ID	`True`	`True`	(	()(())	`True`
ID	`False`	`False`	( )	(()())((()	`True` `False`
OOD	`True`	`False`	( )	())())()(	`True` `False`
—	`False`	`True`	—	DNE	—

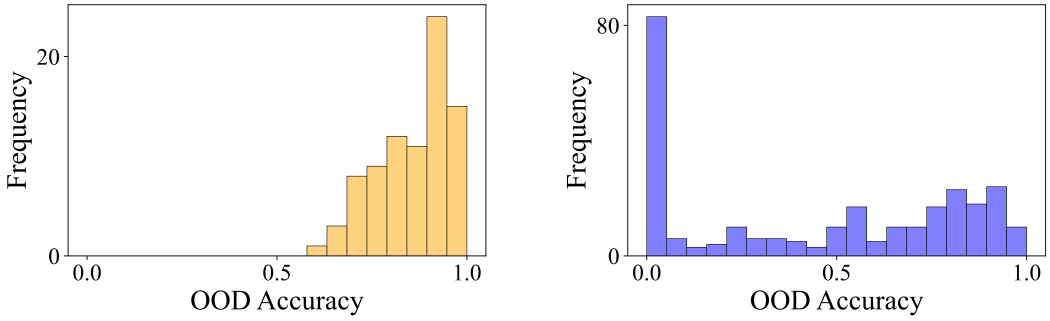
Table 2: The EQUAL-COUNT and NESTED rules applied to example parentheses sequences in our ID and OOD test sets. Also, classifications of the same parentheses sequences according to the OOD FIRST-SYMBOL heuristic. Notice that this rule does not achieve perfect accuracy ID.

C Factors in rule selection

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 sequence in our OOD test set as `False`. Correspondingly, models with 0% OOD accuracy learn EQUAL-COUNT, classifying every OOD example as `True` (Table 2).

C.1 Architecture

By comparing Transformers to LSTMs, we confirm existing findings [1, 22, 38, 29] that LSTMs are intrinsically hierarchical while Transformers are not. The inductive bias of the LSTM architecture places every 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: one near 0% (indicating perfect application of the EQUAL-COUNT rule) and a smaller one at 90% (indicating a tendency towards NESTED). Overall, 24.4% of transformers learn EQUAL-COUNT perfectly, achieving zero OOD accuracy at the last step in training (Figure 6).



(a) OOD-test accuracy for LSTMs.

(b) OOD-test accuracy for Transformers.

Figure 6: Last 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.

C.2 Width

Using the non-parametric Mann-Whitney U test to detect differences between distributions, we find that the number of Transformer heads has no significant effect on the distribution of OOD accuracies for any depth of model (Figure 7). This result, which we show holds over randomness in model initialization and data exposure order, adds to a growing body of evidence across settings that changing transformer width has little effect on model expressivity and OOD generalization [26, 35].

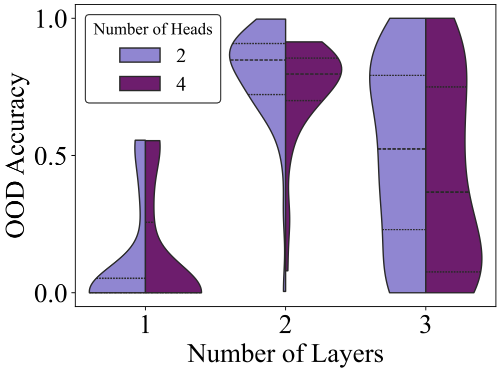


Figure 7: Unlike weight decay and depth (Figure 2b), width is not a substantial factor in final OOD rule selection in this setting. The Mann-Whitney U test finds no statistically significant differences in the distribution of last OOD accuracy over width (all $p > 0.05$).

C.3 Depth

In this setting, depth—unlike width—is a significant factor in rule selection, determining the peaks of the OOD behavior distribution. For instance, among the 66 out of 270 transformers (24%) which achieve perfect EQUAL-COUNT behavior with 0% OOD accuracy (Figure 6), there are 61 1-layer models, no 2-layer models, and 5 3-layer models. (Note our overall model population is evenly split into 90 each of 1-, 2-, and 3-layer models.)

We find that 1-layer models fall into a bimodal distribution centering on two rules: EQUAL-COUNT (characterized by 0% OOD accuracy) and FIRST-SYMBOL (which gives ∼55% OOD accuracy). The 12 models that generalize according to FIRST-SYMBOL all exhibit nearly identical judgments on specific examples, following a rule associated with returning a `True` label if the input begins with (. Because we only consider models with at least 99% validation accuracy, the models in question must also have other subroutines that are successfully applied ID but dominated by FIRST-SYMBOL OOD.

We group models via T-SNE (perplexity 12) based on their judgments on the OOD test set (Figure 2a and 9). FIRST-SYMBOL forms an outlier cluster in model judgements of 1-layer models, which otherwise primarily vary in their adherence to NESTED or EQUAL-COUNT.

By contrast, only 2% of 2-layer models are EQUAL-COUNT-leaning (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 have approximately learned NESTED. Among 3-layer models, behavior varies enormously, with the distributional mode defined by the 20 models with < 0.1 OOD accuracy which learn EQUAL-COUNT. Across all 270 model training runs, 10.4% of models achieve at least 90% accuracy, including 15 2-layer and 13 3-layer models.

C.4 Regularization

Weight decay has a significant impact on the distribution of rules models learn. Without any weight decay, models that generalize ID can converge on a variety of OOD generalization rules with wide distributions. With weight decay, particularly among smaller models, models have more similar OOD behaviors. For example, while 2-layer transformers show a consistent tendency to prefer NESTED, they achieve higher OOD accuracy more reliably when weight decay is applied (Figure 8, Figure 2b).

Among 1-layer models, training with weight decay always results in convergence to the EQUAL-COUNT rule. Without weight decay, 15.6% of 1-layer models converge to FIRST-SYMBOL with ∼55% accuracy on the OOD test set (Figure 2b). 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.

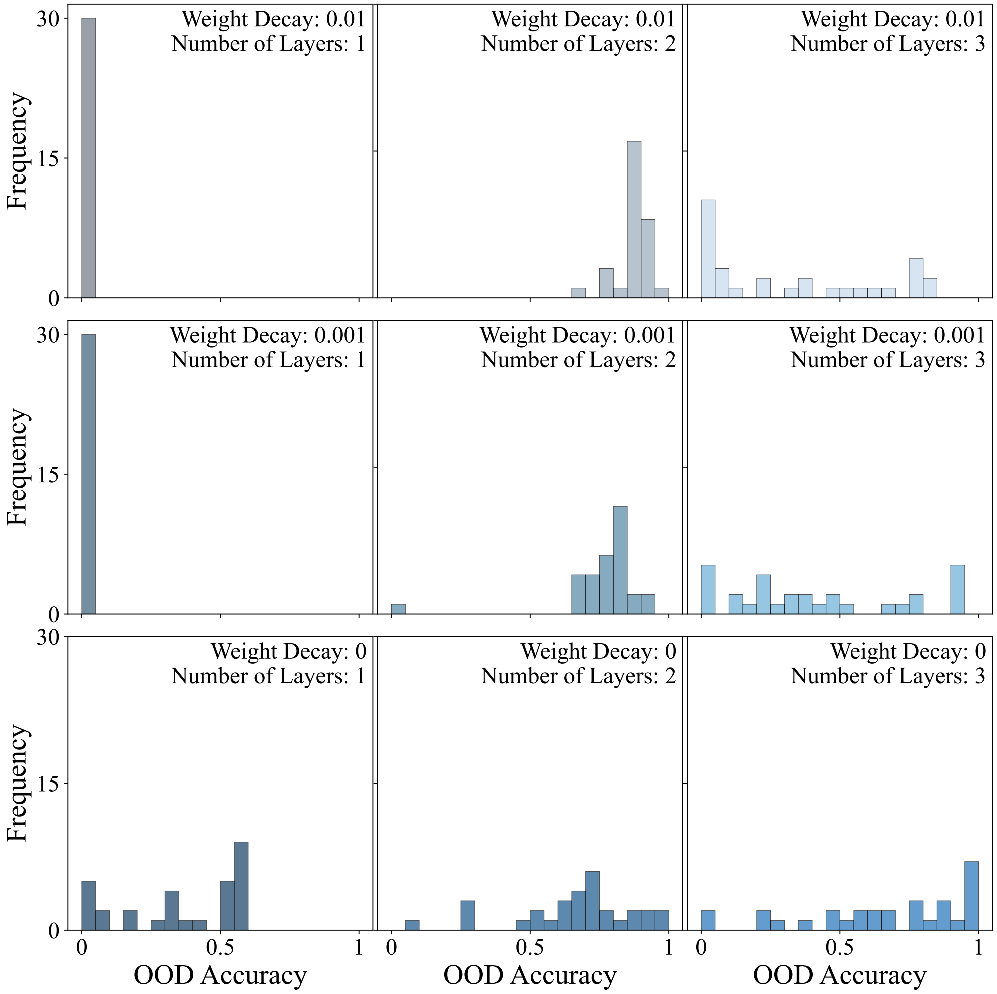


Figure 8: Final accuracy on OOD-test for Transformers of varying depths and weight decays. 1-layer Transformers learn EQUAL-COUNT or FIRST-SYMBOL. Deeper Transformers can learn EQUAL-COUNT or approximate NESTED, with 2-layer Transformers most likely to learn NESTED and 3-layer Transformers instead exhibiting more complex OOD generalization behavior.

C.5 Randomization in Weight Initialization and Data Order

Existing work [28, 4, 43, 15] has investigated the impact of random weight initialization and data order on model performance. Dodge et al. [9] varied these factors in BERT fine-tuning, finding that modifying either factor had significant, comparable impacts on model performance. We investigate the impact of the two sources of random variation that account for differing model behaviors, and find that while they do not affect ID accuracy, both dataset ordering and model initialization affect generalization behavior.

Since we train models across three data shuffle seeds and 5 model random seeds, for a fair comparison between ranges of the impact of these factors on final OOD accuracy, we randomly select three random seeds to plot and compare the ranges of performance across shared hyper-parameter conditions.

In Figure 10, we show that in a plurality of models trained, OOD performance was impacted by at least 10%, with the maximum difference due to either one of the two random factors reaching an above 90% difference in OOD behavior. The similar distribution in the impact of random initialization and data order is aligned with previous work and indicates both factors are important to determining model OOD performance and should be accounted for in building robust ML systems.

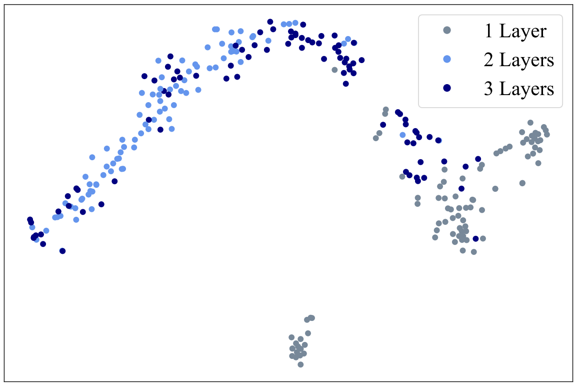


Figure 9: T-SNE of models' final OOD classifications colored by model depth.

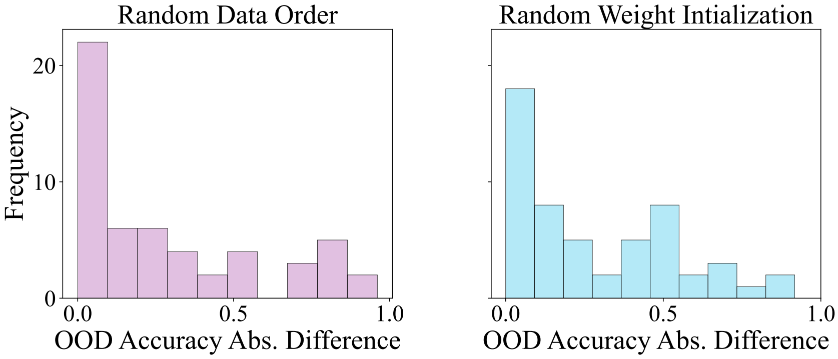


Figure 10: Data order (left) and weight initialization (right) are equally influential random factors in OOD behavior, as shown by the distribution of the ranges of final OOD accuracy between models trained with differing data ordering and weight initialization.

605 D Training dynamics of different rules

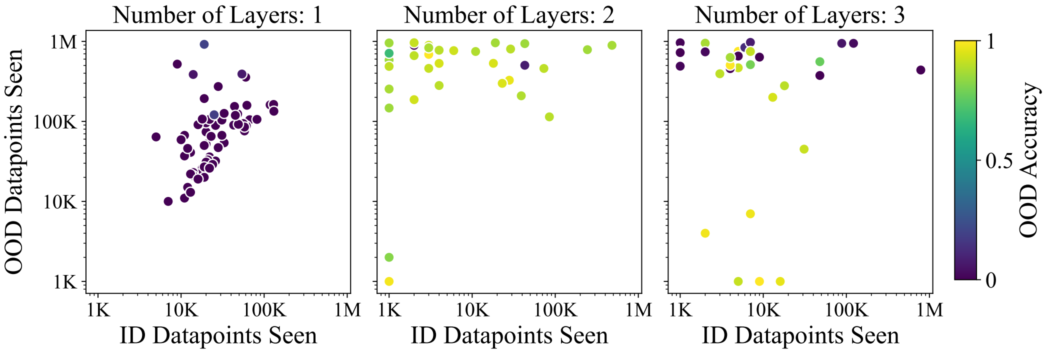
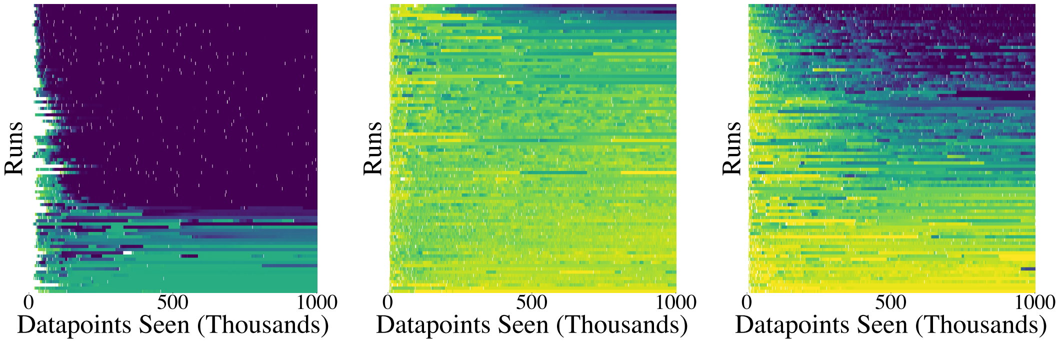


Figure 11: Illustration of generalization rules across training. ID convergence occurs when models achieve $\geq 99\%$ ID accuracy for $> 99\%$ of remaining datapoints seen—all Transformers achieve this metric after 900K datapoints. We define OOD convergence to either EQUAL-COUNT or NESTED as Transformers achieving ≤ 0.2 or ≥ 0.8 accuracy for $> 99\%$ of the rest of the model run, respectively, after seeing at most 975K datapoints—53% of transformers achieve this metric. Using these metrics, this plot shows the number of datapoints seen before OOD and ID convergence, excluding models that do not converge OOD.



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

Figure 12: Heatmaps showing model training dynamics broken down by depth, where purple and yellow indicates model adherence to the EQUAL-COUNT and NESTED rule, respectively (using the same scale as Figures 2a, 3, and 11). Colored cells indicate the OOD accuracy of a particular run when ID accuracy is at least 0.99.

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 generally take longer to converge to the NESTED rule.

Although overall, models tend to classify OOD sequences 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 11, often stabilize long after ID convergence. In other words, we see an example of *structural grokking* [24]. 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. The training dynamics of different rules broken down in-detail by model depth are also shown in Figure 12. Notably, deeper models have higher final OOD accuracy variance and also display greater variance during training, possibly because of their higher expressivity.

D.1 Evaluating the Heuristic

The FIRST-SYMBOL heuristic, in contrast to the NESTED and EQUAL-COUNT rules, is not used by any models ID, where it would produce a low accuracy. However, it is used by some 1-layer models OOD, and it produces an OOD accuracy of ≈ 0.55, reflecting the fact that around 55% of our OOD test examples begin with close brackets.

As discussed in Section 3.1.1, a majority of 1-layer models appear to pass through a FIRST-SYMBOL heuristic phase, although this heuristic does not persist until the end of training among models trained with weight decay. In this case, we say the models “appear" to pass through this phase, rather than asserting that they do, because we are only able to verify their behavior on individual datapoints at our five saved model checkpoints. However, at those model checkpoints, we are able to confirm that 1-layer models whose OOD accuracy is between 0.54 and 0.56 indeed almost always make OOD judgments based on first symbol. We therefore posit that throughout training, instances of OOD accuracy in this range likely reflect the FIRST-SYMBOL heuristic, though we cannot rule out that some such instances reflect some other heuristic which coincidentally produces the same OOD accuracy. Notably, 2 and 3 layer models also reach accuracies in this range while training, but, checking their at saved checkpoints, we find they do not learn FIRST-SYMBOL.

E Attention Heads Classified by OOD Behavior

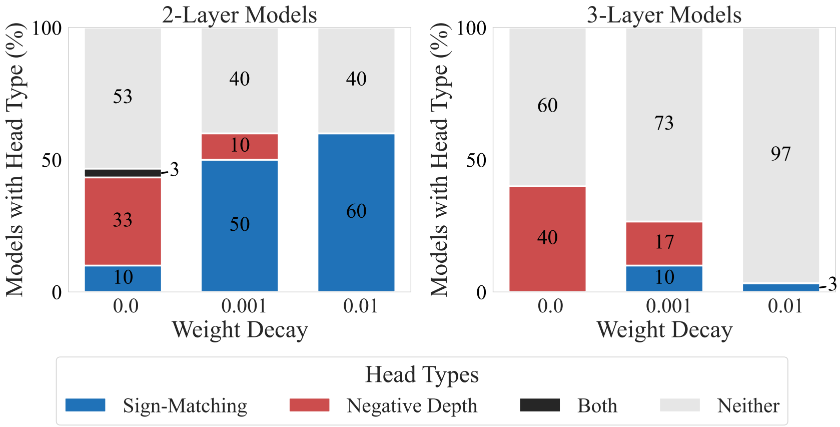


Figure 13: Percentage of 2- and 3-layer models containing each head type, by weight decay. Head types are classified according to their OOD behavior.

We breakdown the presence of types of head across 2 and 3 layer models (Figure 13).

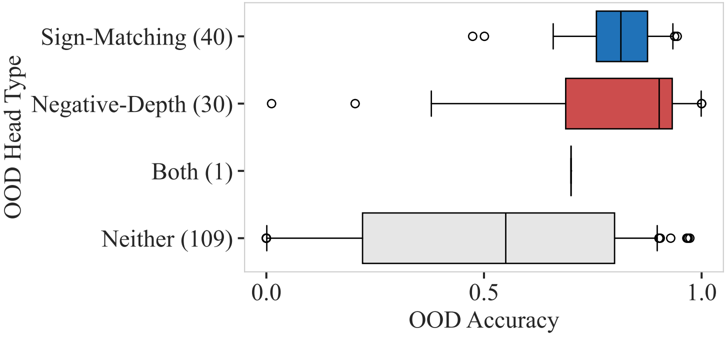


Figure 14: OOD accuracy of 2- and 3-layer models with and without OOD hierarchical heads, classified by head subtype.

In Section 4.2, we showed that classifying hierarchical heads according to their behavior on the ID validation set is predictive of generalizing according to the balanced rule. It is also possible to conduct the same analysis using behavior on the OOD test set. We find that the presence of OOD hierarchical heads is similarly predictive of generalizing according to the NESTED rule (Figure 14). All types of depth head appear to correlate similarly strongly with NESTED generalization.

F Breakdown of Hierchical Heads by Layer and Weight Decay

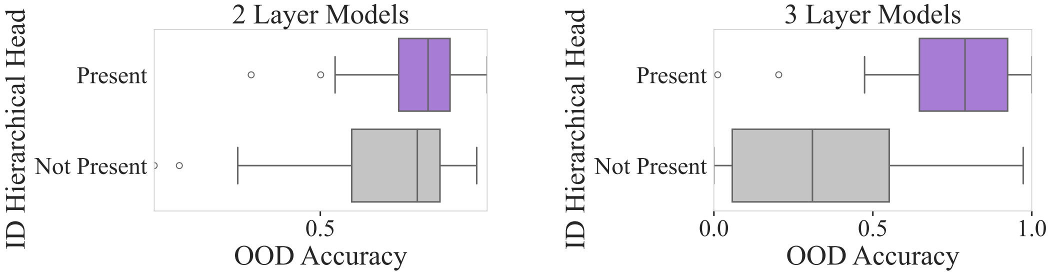


Figure 15: Last OOD test accuracy of 2 and 3-layer models with and without ID hierarchical heads.

Models with ID hierarchical heads consistently have higher OOD accuracy across both 2 and 3 layer models, indicating that model internals can provide additional predictive power to hyperparameters alone (Figure 15). Particularly for 3 layer models, which have the greatest diversity in OOD performance, the presence of ID hierarchical heads in a model provides additional insight into predicting hierarchical NESTED-like generalization.

Model internals continue to improve predictive power even when fixing both depth and weight decay simultaneously. Some hyperparameter combinations lead to one rule or the other relatively consistently: for example, “1 layer, weight decay $>$ 0" is completely predictive of EQUAL-COUNT (see Figure 8, Appendix C.3). However, in hyperparameter settings with diverse OOD behavior, presence of hierarchical heads is predictive of NESTED generalization behavior.

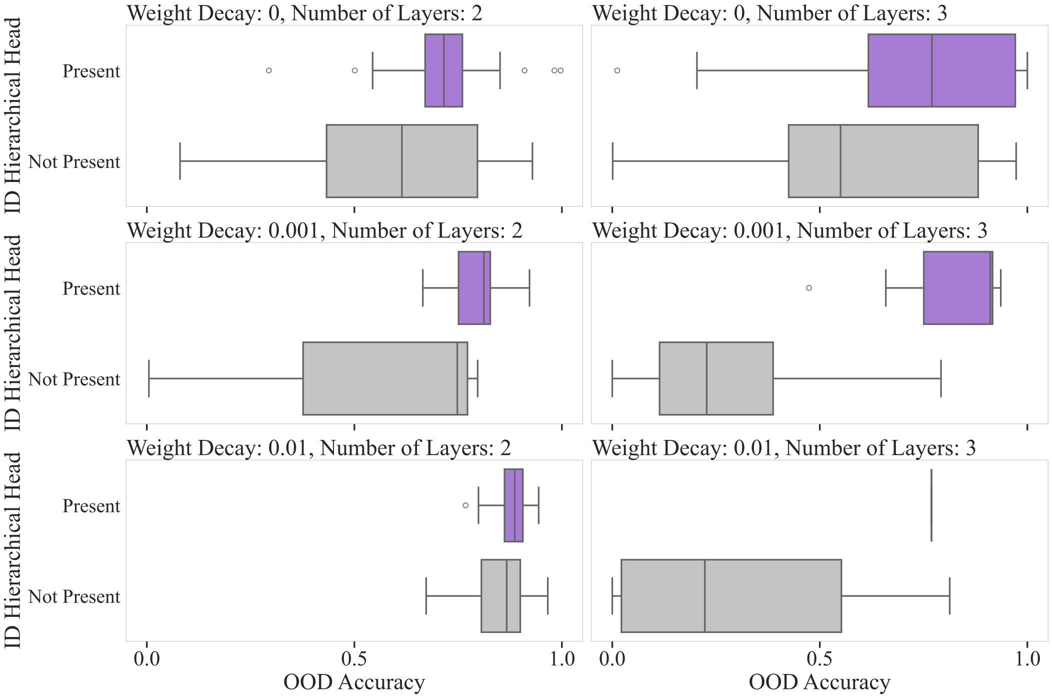


Figure 16: OOD accuracy of 2 and 3-layer models with and without ID hierarchical heads, by number of layers and weight decay value. Over the four populations, (1) 2 and 3 layer models with non-zero weight decay, (2) all 2 layer models, (3) all 3 layer models, and (4) 3 layer models with 0.001 WD, a Mann-Whitney U test shows a significant difference in OOD accuracy distributions.

In Figure 16, consider any multilayer setting producing a diverse range of OOD accuracy values, including values both above and below 50% (in other words, any setting except the “2 layer, 0.01

weight decay" setting, where all models learn NESTED). In every such setting, models without hierarchical heads have a median OOD accuracy that falls at or below the bottom quartile of the score for models with such heads. For some settings, the distributions barely overlap at all.

Thus, even if we consider the effect of hyperparameter settings, the presence of a hierarchical (i.e., depth-tracking) head is highly informative. Of course, some of these settings might lead to more hierarchical runs *because* they enable the learning of hierarchical mechanisms, meaning that the effect of hierarchical heads is far greater than any one setting would suggest.

G Effects of Causal Intervention on Attention

Our causal experiments involve uniform ablation of the attention distribution in all attention heads (Section 4.3). We ablate all attention in order to uniformly and symmetrically intervene on all models. This is in contrast to ablating exclusively hierarchical heads, which would require us to compare models on an "unequal footing," in the sense that some models would have 0 heads ablated, some models 1 head ablated, and some models 2 or more. Ablating all attention allows us to definitively eliminate all influence from hierarchical heads without introducing asymmetric interventions.

We also examined the effects of ablating individual heads one at a time. We found effects that were generally generally very similar (if weaker), in comparison with full attention ablation; in particular, in models with 2 or more hierarchical heads, full attention ablation tended to affect OOD accuracy more strongly than one-at-a-time attention ablation. Ultimately, one-head-at-a-time ablation demonstrates the same trend as full attention ablation: ablating negative depth heads reduces NESTED behavior, while ablating sign-matching heads increases this behavior. See Figure 17.

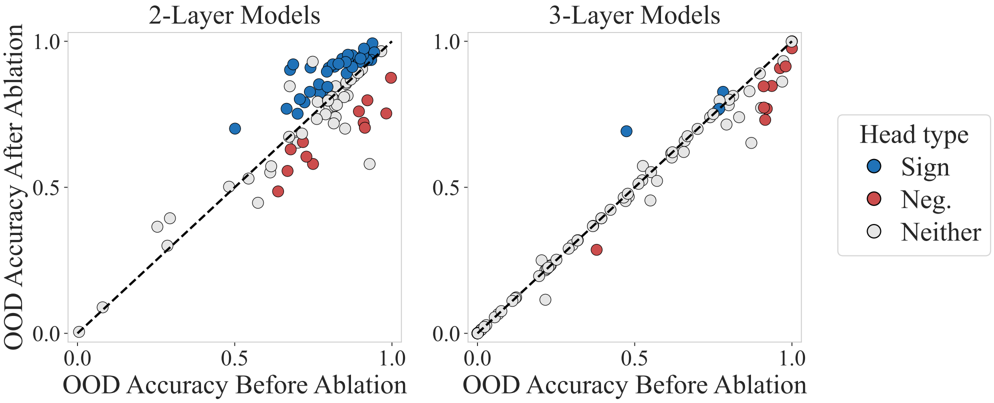


Figure 17: OOD accuracy before and after applying uniform attention ablation to one head in each model. For each model, we plot the head whose ablation affects OOD accuracy most (in absolute value), colored by whether it is a sign-matching or negative depth head (or neither). In comparison with full attention ablation (Figure 5), effects on OOD accuracy are smaller. However, our analysis of differing effects of ablating different types of depth heads remains the same as in Section 4.3.2.

H Change in OOD and ID Accuracy After Ablating Heads

Ablating either ID or OOD Sign-matching heads and Negative-depth detecting heads, has little to no effect on ID accuracy (Figures 18 and 19, left panels). The maximum effect of ablating an ID or OOD head type on the ID data is 77/1000, both for Negative-Depth detectors, but the median impact is around 10/1000 for all classified head types. Across ID head types, ablation tends to decrease OOD accuracy, but for OOD head types, as seen in Figure 5, ablating negative-depth heads decreases while ablating sign-matching heads increases OOD accuracy (Figures 18 and 19, right panels).

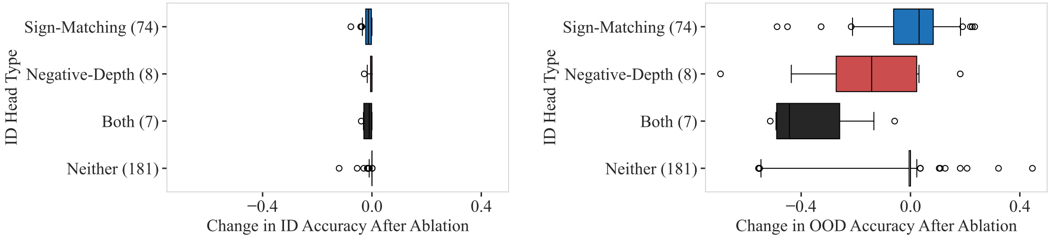


Figure 18: For heads classified by type based on their ID behavior, accuracy ID and OOD after ablation subtracted by baseline. Ablation has little impact on ID accuracy, and tends to decrease OOD accuracy across ID sign-matching and negative-depth heads. The number of models in each category are included in parentheses after the label.

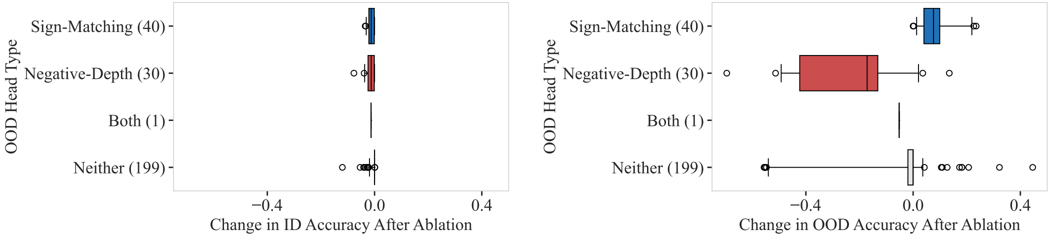


Figure 19: For heads classified by type based on their OOD behavior, accuracy ID and OOD after ablation subtracted by baseline. Ablation has little impact on ID accuracy, but ablating negative depth OOD heads decreases and ablating sign-matching OOD heads increases OOD accuracy. The number of models in each category are included in parentheses after the label.