The Heuristic Core: Understanding Subnetwork Generalization in Pretrained Language Models

Anonymous ACL submission

Abstract

Prior work has found that pretrained language models (LMs) fine-tuned with different random seeds can achieve similar in-domain performance but generalize differently on tests of syntactic generalization. In this work, we show that, even within a single model, we can find multiple subnetworks that perform similarly in-domain, but generalize vastly differently. To better understand these phenomena, we investigate if they can be understood in terms of "competing subnetworks": the model initially represents a variety of distinct algorithms, corresponding to different subnetworks, and generalization occurs when it ultimately converges to one. This explanation has been used to account for generalization in simple algorithmic tasks. Instead of finding competing subnetworks, we find that all subnetworks—whether they generalize or not—share a set of attention heads, which we refer to as the *heuristic core*. Further analysis suggests that these attention heads emerge early in training and compute shallow, non-generalizing features. The model learns to generalize by incorporating additional attention heads, which depend on the outputs of the "heuristic" heads to compute higher-level features. Overall, our results offer a more detailed picture of the mechanisms for syntactic generalization in pre-trained LMs.[1]

1 Introduction

A central question in machine learning is to understand how models generalize from data that supports different possible solutions. McCoy et al. (2020) investigated this question in the context of pre-trained language models (LMs) like BERT (Devlin et al., 2019) trained on natural language inference (NLI; Williams et al., 2018), showing that models trained with different random seeds perform very similarly on in-domain (ID) evaluation sets but generalize very differently to adversarial out-of-domain (OOD) evaluation sets. However, we have limited understanding of how any individual model learns to generalize.

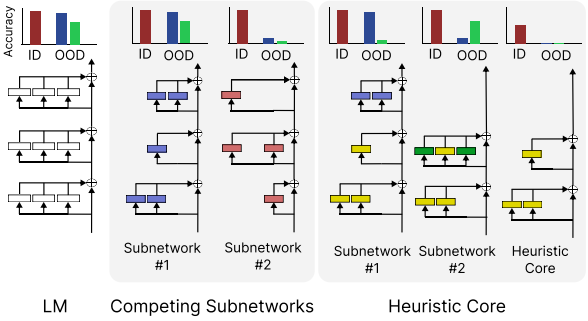


Figure 1: Different subnetworks in a BERT model achieve similar in-domain performance but generalize differently. Prior work has explained similar generalization phenomena in synthetic tasks in terms of distinct subnetworks that compete during training. We instead find evidence of a *heuristic core*: a set of attention heads that appear in all generalizing subnetworks but, on their own, do not generalize.

In this work, we use subnetwork analysis to better understand how BERT models generalize on NLP tasks. Specifically, we use structured pruning (Wang et al., 2020; Xia et al., 2022) to isolate various subnetworks—subsets of attention heads and MLP layers—that approximate the behavior of the full model. Our main finding is that, that even within a single BERT model, there exist multiple subnetworks that all match the model's performance closely on in-domain evaluation sets, but exhibit vastly different generalization. This result has practical implications—for example, underscoring the importance of OOD evaluation for pruning—and also raises a number of questions about the mechanisms underlying generalization, which we proceed to investigate.

One possible explanation for our results is that the model initially consists of a variety of disjoint subnetworks, corresponding to distinct possible

[1]We release our code publicly to foster further research.

solutions for the task; the model ultimately generalizes if it converges to a subnetwork that generalizes. This model of generalization has been used to explain the "grokking" phenomenon on toy algorithmic tasks, whereby a neural network initially overfits a training set—but, after continued training, suddenly undergoes a phase shift to perfect generalization (Barak et al., 2022; Nanda et al., 2023; Varma et al., 2023). We refer to this as the *Competing Subnetworks* explanation, following Merrill et al. (2023). Our setting bears a similarity to grokking, originally documented by Tu et al. (2020): models converge early on in-domain evaluations and only start to generalize after training for additional epochs. Therefore, we test if generalization in this case also arises from competition between disjoint subnetworks.

First, we check if these behaviorally different subnetworks consist of disjoint subsets of model components. To the contrary, we find a small set of attention heads that occur in *all* subnetworks—even the ones that do not generalize at all. Furthermore, we find that this same set of attention heads also appear when we prune earlier checkpoints of the model, before it starts to generalize. Generalization is accompanied by an increase in *effective size*—defined as the size of the smallest subnetwork that matches the model's performance on the generalization sets (Merrill et al., 2023)—as the model learns to incorporate additional attention heads on top of these core components. This differs from the grokking setting, where generalization is marked by a reduction in effective size, as the model switches from a dense, memorizing subnetwork to a sparse, generalizing subnetwork. In contrast, our results suggest that, instead of selecting between competing subnetworks, the model initially learns a "core" of attention heads that implement simple heuristics. The model ultimately generalizes by learning additional attention heads that interact with these heads.

We refer to this core set of attention heads as the *heuristic core*, and conduct further analysis to better understand what roles they play in the model. We find that these attention heads are associated with heuristics—namely, attending to words that are repeated across sentences—supporting our characterization as these components as heuristic core. On the other hand, ablating these attention heads from the original model leads to a dramatic *decrease* in performance on the generalization set. This result further supports the idea that generalization arises from the interaction between heuristic components, which are associated with simple features, and additional components. At intermediate sparsity levels, subnetworks contain different "counter-heusitic" components, leading to different patterns of partial generalization.

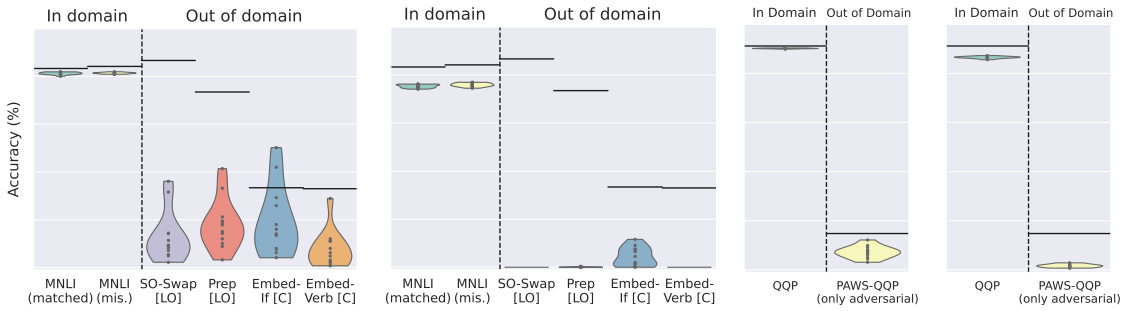
Our results paint a more detailed picture into the mechanisms for generalization in a natural language understanding task. These results have practical implications—for example, suggesting that we cannot make the model more robust by ablating the "heuristic" components—and raise intriguing questions about how models learn to generalize by composing simple parts.

2 Problem Setup

Tasks. We focus on two sentence pair classification tasks. MNLI (Williams et al., 2018) is a natural language inference (NLI) dataset, which involves predicting whether a *premise* sentence logically entails a *hypothesis* sentence. QQP (Iyer et al., 2017) is a paraphrase identification dataset. Prior work has noted that these datasets admit "heuristics": solutions that achieve high in-domain accuracy but do not generalize well. In particular, McCoy et al. (2019) introduced the HANS dataset, which tests whether NLI models use specific heuristics, such as lexical overlap (predicting that sentences entail each other if they share many words); see Appendix Table 3 for examples. Similarly, the QQP-PAWS dataset (Zhang et al., 2019) tests whether paraphrase identification models are susceptible to a word-overlap heuristic (Appendix Table 4).

Models. We fine-tune BERT models (Devlin et al., 2019) on MNLI and QQP (see Appendix B for more model and training details). Our results reproduce the observations of Tu et al. (2020): in-domain accuracy saturates early, and out-of-domain accuracy increases much later. In this paper, we mainly focus on the four subcases in Table 3 (Appendix A), as the model generalizes well on them. Other subcases are discussed in Appendix D. The effect is smaller, but still consistent, on PAWS-QQP.

Finding subnetworks via pruning. For the BERT model, we define a subnetwork as a subset of the attention heads and feed-forward layers. An important motif in our analysis will be the search for subnetworks that preserve model performance. We originally considered circuit-finding methods like



(a) MNLI, 50% (b) MNLI, 70% (c) QQP, 30% (d) QQP, 60%

Figure 2: Pruning a BERT model with different random seeds results in subnetworks that perform similarly in-domain but generalize differently. MNLI/HANS: At 50% sparsity, the subnetworks operform within 3% of the model on MNLI but show varying generalization. At 70% sparsity, the subnetworks behave as pure heuristics despite respectable MNLI accuracy. The trend also holds for QQP/PAWS, with sparsities of 30% and 60%. Solid lines line indicate full model performance. Figure 12 in Appendix D shows the plot for all subcases of HANS.

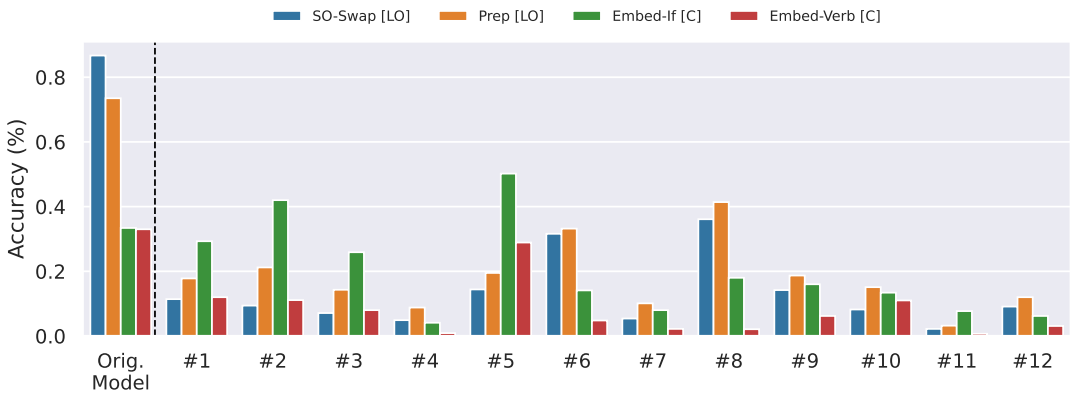


Figure 3: Different subnetworks generalize partially to different subcases of HANS. For instance, subnetwork #5 generalizes well to the *Constituent* subcases, whereas #8 generalizes to the *Lexical Overlap* subcases. Subnetwork #2 does well on the *Embed-If* subcase of *Constituent*, but not *Embed-Verb*. These subnetworks are pruned with a 50% sparsity threshold.

AC/DC (Conmy et al., 2023) to achieve this, but some of our experiments rely on the injection of randomness via a seed. In addition, pruning allows us to directly optimize the KL loss with respect to the original model, instead of identifying important components locally. Therefore, we chose to use pruning instead. We use structured pruning rather than unstructured pruning for two reasons. First, unstructured pruning can be expressive enough to find subnetworks that have completely different behavior from the original model (Wen et al., 2023). Second, structured pruning lets us compare subnetworks in terms of semantically meaningful components, namely attention heads.

We prune via optimization of a binary *mask*, where 0 and 1 indicate dropping and retention, respectively. Our runs prune attention heads and entire MLP layers, each corresponding to one bit in this mask. Pruned layers are replaced by an addition of their mean activation calculated over the training data, as per recommendations from work in Mechanistic Interpretability (Zhang and Nanda, 2023). Following CoFi Pruning (Xia et al., 2022), we relax binary masks to floats in $[0, 1]$ and then perform gradient descent to optimize them. The objective optimized is the KL loss between the predictions of the full model and those of the pruned subnetwork. Finally, the mask entries are discretized to $\{0, 1\}$ based on a threshold. L0 regularization is used to enforce a target sparsity. We adopt Louizos et al. (2018)'s recipe to model the masks and furnish the details in Appendix C. More details and hyperparameters, along with a more detailed discussion of the method, are provided in Appendix B. We freeze model weights and only train the masks to preserve faithfulness to the model.

3 Comparing Subnetwork Generalization

We start by examining how different subnetworks generalize on MNLI/HANS and PAWS/QQP. Specifically, we compare the generalization behavior of subnetworks with the same sparsity level, but pruned with different seeds; and we compare subnetworks pruned with different sparsity levels. We prune the BERT models fine-tuned on MNLI and QQP with 12 random seeds for different target sparsities (more details in Appendix B). The resulting subnetworks are evaluated on the ID (MNLI and QQP validation sets) and OOD (HANS subcases, and PAWS-QQP's adversarial subset) evaluation sets. In our runs, we observe that the MLP layers are never pruned (although they are allowed to be).

Different subnetworks generalize differently. Figures 2a and 2c plot the results for a target sparsity of 50% and 30% in the case of MNLI and QQP, respectively. The pruned models perform close to each other in-domain, and are all within $2-3\%$ of the full model's accuracy on both the MNLI and QQP validation splits. However, their accuracies on the out-of-domain evaluation splits varies widely. In the most extreme case, the accuracies on the *Embed-If (Constituent)* subcase of HANS vary from just 4.1% to 50.2%. Moreover, different subnetworks generalize to different subcases, as Figure 3 illustrates. Subnetwork #8 achieves 41.4% at *SO-Swap (Lexical Overlap)* but only 18.0% at *Embed-If (Constituent)*. Subnetwork #5, on the other hand, achieves only 19.5% at the former but 50.2% at the latter. Thus, pruning different parts of the model seems to sacrifice generalization on the OOD subcases to varying degrees. The results in the case of QQP and PAWS-QQP are similar.

Sparser subnetworks generalize worse. We also observe that sparser subnetworks consistently generalize worse. First, we prune the model to a higher sparsity (70% for MNLI and 60% for QQP) with 12 random seeds. As Figures 2b and 2d show, virtually *all* subnetworks at this sparsity show a complete lack of generalization on almost every OOD subcase. The ID accuracy stays at a respectable $74-76\%$, indicating that a large portion of model performance is explained by these sparse subnetworks. In Figure 4, we plot the generalization accuracy of subnetworks pruned at different sparsity levels, for three random seeds. The generalization accuracy generally drops steadily at higher sparsity levels and is devoid of any drastic phase transitions.

4 Competing Subnetworks Hypothesis

In the previous section, we observed that different subnetworks perform similarly to the full model in-domain but generalize differently. Moreover, no sparse subnetwork generalizes as well as the full model. We now seek to understand how these subnetworks emerge during training and interact to give rise to generalization.

Specifically, we explore whether our findings can be explained in terms of *competing subnetworks*: the model initially represents multiple solutions as different subnetworks, and it ultimately converges to one. Whether it generalizes is dependent on which of the subnetworks it converges to. This hypothesis has been found to explain the curious phenomenon of grokking (Merrill et al., 2023). Additionally, as Tu et al. (2020) note, the ID performance of our BERT models saturates early in training (Figure 6). The OOD accuracy starts to rise much later. Motivated by the similarity to grokking, we test if generalization in our case also arises from competition between disjoint subnetworks.

We consider the Competing Subnetwork Hypothesis to consist of two main predictions, mirroring Merrill et al. (2023). First, the model can be composed into subnetworks that are *disjoint*, representing distinct algorithms consistent with the training data. Second, a rise in generalization is accompanied by a reduction in the model's *effective size*, which occurs when the model switches from a dense, non-generalizing subnetwork to a sparse, generalizing subnetwork.

All subnetworks share a common set of components. We start by measuring how many model components are shared between different subnetworks. In particular, we compute the frequencies of each attention head in the 50% (partially generalizing) subnetworks and in the 70% (non-generalizing) subnetworks of the MNLI model. These frequencies are visualized in Figure 5 (a corresponding plot for QQP is presented in Figure 11 of Appendix D). The Spearman's Rho correlation between the two frequencies if 0.82 (p-value = $1.6 \cdot 10^{-36}$), i.e., very strong agreement. (The corresponding value for QQP is 0.74, corresponding to strong agreement; see Figure 14 in Appendix D.) In particular, we observe that **nine attention heads** appear in *all* subnetworks, even the sparsest ones—that is, the subnetworks that behave most consistently with the heuristics. In other words, instead of finding disjoint subnetworks, we find a high degree

(a) MNLI/HANS

(b) QQP/PAWS-QQP

Figure 4: The OOD accuracy decreases fairly smoothly with sparsity (3 seeds). The drop in ID accuracy is slow and has low variance. We find no subnetworks sparser than 30% generalizing as well as the full model on either dataset.

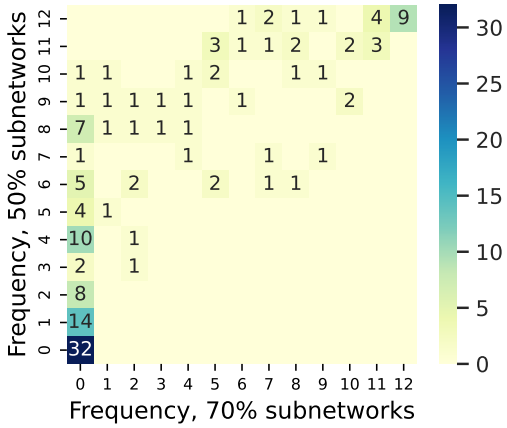


Figure 5: This heatmap quantifies the frequencies of attention heads in the 50% and 70% sparsity subnetworks (MNLI). Entry (i, j) corresponds to the number of heads appearing in i/12 50% subnetworks and j/12 70% subnetworks. In particular, we note that *nine* attention heads appear in all of the subnetworks.

of overlap (the expected intersection of 12 random 70% sparsity subnetworks is $7.7 \cdot 10^{-5}$ heads) between the subnetworks that partially generalize and the subnetworks that do not generalize at all.

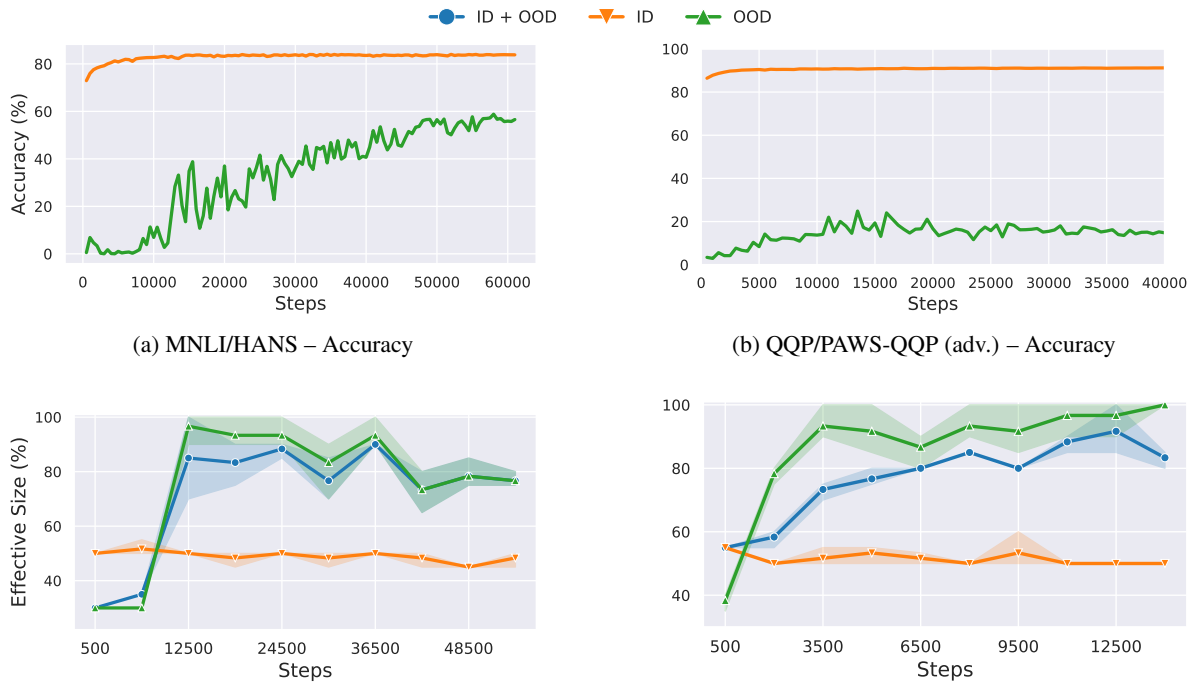
Effective size increases as the model generalizes. Next, we investigate how these subnetworks emerge over the course of training. Specifically, we conduct a similar analysis to Merrill et al. (2023) and plot the *effective size* of the model, defined as the size of the smallest subnetwork that is *faithful* to the model on a particular evaluation dataset D. We say that a subnetwork is *faithful* to a model on dataset D if their accuracies are within 3%. We report faithfulness using different choices of D—either the in-domain validation set, the out-of-domain validation sets, and the equally weighted combination of the two. We prune each checkpoint to sparsities that are multiples of 5% and pick the highest sparsity that yields a faithful subnetwork.

The results are in Figure 6. The effective size needed to approximate the model on the in-domain evaluation remains relatively flat over the course of training. However, on out-of-domain and mixed data, generalization is accompanied by an *increase* in effective size. We also examine the set of attention heads that appear in the MNLI subnetworks at the checkpoint immediately before the model starts to generalize. We find that the same nine attention heads that appear in all subnetworks at the end of training appear in these subnetworks as well. This suggests that, rather than switching between competing subnetworks, the model first learns a set of attention heads that compute simple, non-generalizing features, and then generalizes by incorporating more components into this core.

5 Understanding the Heuristic Core

Instead of finding evidence for distinct, competing subnetworks, we have found that all subnetworks share a set of common components. In particular, nine attention heads appear in all MNLI subnetworks, including subnetworks that do not generalize at all. Moreover, we find the same nine attention heads in subnetworks when we prune checkpoints from early in training, before the model begins to generalize. We refer to these nine attention heads as the *heuristic core* of the model. In this section, we inspect these attention heads in more detail, with the goal of understanding what features they compute, and how they interact with the rest of the model to ultimately enable the model to generalize.

How do the heuristic core attention heads behave? First, we look at the attention patterns exhibited by these attention heads. We inspect example



(a) MNLI/HANS – Accuracy

(b) QQP/PAWS-QQP (adv.) – Accuracy

(c) MNLI/HANS – Effective size

(d) QQP/PAWS-QQP (adv.) – Effective size

Figure 6: The ID and OOD accuracies, and effective size of the model during fine-tuning, computed by pruning with 3 seeds. Generalization is associated with a steep *increase* in effective size for OOD/mixed datasets. The effective size for ID datasets remains constant, showing that these extra effective parameters go towards generalization.

attention patterns and observe that most heads exhibit a simple attention pattern, and we quantify these in Figure 9. Four out of nine heads attend to tokens that repeat across the premise and hypothesis, suggesting that they extract token overlap features. Figure 7 shows example attention patterns. Interestingly, we find that a number of these attention heads also appear in all QQP subnetworks. Figure 8 presents an example from PAWS-QQP, indicating that these heads play a similar role in this dataset, attending between repeated words.

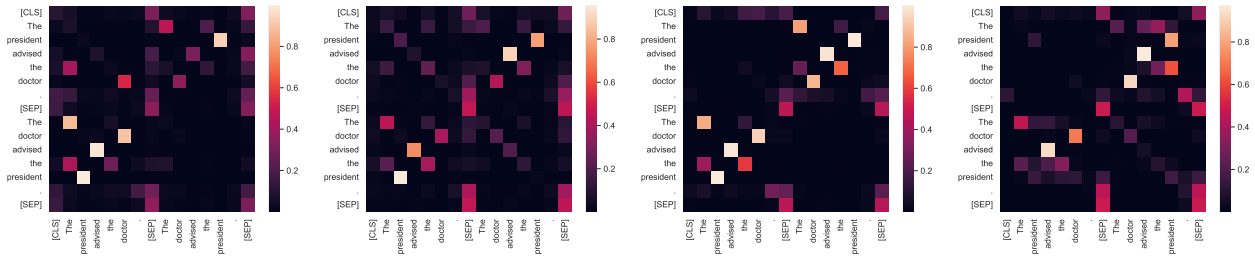
Other low-layer heads attend to the previous or subsequent token. The heads in the higher layers are more difficult to characterize but often attend to the special separator token, perhaps aggregating information from across the sequence. Overall, these findings are consistent with the notion that these attention heads calculate simple, shallow features.

How do the heuristic core attention heads interact with the rest of the model? Next, we ablate each heuristic core attention head from the full model and observe how the in-domain and out-of-domain accuracies change, in Table 1. None of the heads' ablation affects in-domain accuracy greatly. Surprisingly, however, many heads seem to be critical to performing well on HANS: for example, ablating a single head (4.0) reduces performance on the *subject-object swap* subcase by more than 40%. Moreover, the drops observed on ablating these heads is *super-additive*: ablating only three of the heads we identified leads to the OOD accuracy plummeting. This indicates that heads outside the heuristic core rely on the heuristic core in order to implement more complex behavior.

Table 2 paints a similar picture for QQP. Perhaps most interestingly, the heads 2.5 and 4.0 that had the maximum OOD accuracy reduction on ablation from the MNLI model also stand out here. Along with the fact that 5 of these 8 heads also exist in the MNLI model's core, this highlights the possibility that these attention heads already extract word overlap features in the BERT model prior to fine-tuning. We leave this question for future work.

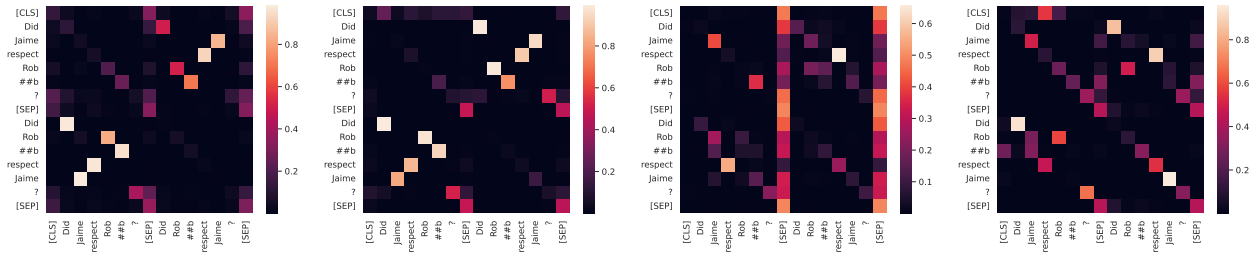
6 Related Work

Pruning. Pruning (LeCun et al., 1989) removes layers or parameters from a network for space efficiency and speed-ups. Structured Pruning (Wang et al., 2020; Ma et al., 2023; Xia et al., 2022, 2023) preserves some structure in the model, e.g. equal fraction of surviving parameters in each attention head, accepting lower compression in exchange for



(a) Layer 2, Head 5 (b) Layer 3, Head 7 (c) Layer 4, Head 0 (d) Layer 8, Head 11

Figure 7: Many attention heads occurring in all subnetworks (MNLI) attend to repeated tokens across the premise and context. Since the order of, *e.g.*, "doctor" and "president" is reversed, the heads seem to identify lexical overlap.



(a) Layer 2, Head 5 (b) Layer 4, Head 0 (c) Layer 8, Head 11 (d) Layer 9, Head 1

Figure 8: Many attention heads occurring in all subnetworks (QQP) also attend to repeated tokens across the premise and context. Interestingly, there is a significant overlap between these heads and those identified for MNLI.

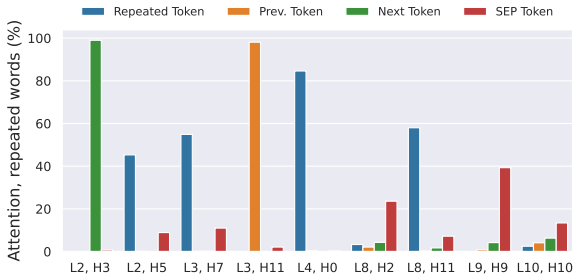


Figure 9: Of the 9 heads occurring in all subnetworks, several assign the majority of their attention weight between words repeated across the premise and hypothesis. Others either attend to the previous, next, or separator token. Here, Lx.Hy indicates Head y of Layer x.

higher speed-ups. Unstructured pruning (Sun et al., 2023) approaches like Magnitude Pruning (LeCun et al., 1989; Hassibi and Stork, 1992) achieve better compression at the cost of lower potential speedup. Proposals such as Layer dropping (Zhang and He, 2020) and Block Pruning (Lagunas et al., 2021) have explored pruning at coarser granularities.

Interpretability & pruning. Mechanistic Interpretability seeks to understand models through their components, e.g., neurons, attention heads, and MLPs. Such analysis uncovers intricate interactions between different atomic components, which form *circuits* (Elhage et al., 2021; Olah et al., 2020) implementing various tasks. Pruning has occasionally been used as part of a larger interpretability effort (Lepori et al., 2023a), though mostly in an ad-hoc manner. Jain et al. (2023) use pruning and probing techniques to show that fine-tuning learns a thin "wrapper" around model capabilities. Lepori et al. (2023b) use pruning to identify subnetworks which they then show form modular building blocks of model behavior. In a similar vein, (De Cao et al., 2020) learn differentiable masks over the input to investigate model behavior. While most of these approaches replace missing layers with an identity function, work in Mechanistic Interpetability (Zhang and Nanda, 2023; Conmy et al., 2023) has shown that adding their mean activations might preserve faithfulness better. Prior work has examined attention patterns to decipher model behavior (e.g. Clark et al., 2019), although attention patterns have also been shown to be misleading in some cases (Jain and Wallace, 2019).

Generalization. The question of how a neural network generalizes has remained a compelling research direction for the NLP community (Yang et al., 2023). McCoy et al. (2020) find that fine-

Ablated Head	Accuracy (%) ↑				
	MNLI (matched)	SO-Swap [LO]	Prep [LO]	Embed-If [C]	Embed-Verb [C]
None	83.3	86.7	73.5	33.0	33.4
Layer 2, Head 3	$83.2_{\downarrow 0.1}$	$79.3_{\downarrow 7.4}$	$72.3_{\downarrow 1.2}$	$28.8_{\downarrow 4.2}$	$28.9_{\downarrow 4.5}$
Layer 2, Head 5	$83.4_{\uparrow 0.1}$	$70.3_{\downarrow 16.4}$	$58.6_{\downarrow 14.9}$	$28.5_{\downarrow 4.5}$	$23.6_{\downarrow 9.8}$
Layer 3, Head 7	$83.5_{\uparrow 0.2}$	$68.6_{\downarrow 18.1}$	$60.2_{\downarrow 13.3}$	$28.2_{\downarrow 4.8}$	$23.0_{\downarrow 10.4}$
Layer 3, Head 11	$83.5_{\uparrow 0.2}$	$81.8_{\downarrow 4.9}$	$70.1_{\downarrow 3.4}$	$31.0_{\downarrow 2.0}$	$32.9_{\downarrow 0.5}$
Layer 4, Head 0	$82.9_{\downarrow 0.4}$	$45.4_{\downarrow 41.3}$	$41.8_{\downarrow 31.7}$	$21.2_{\downarrow 11.8}$	$12.7_{\downarrow 20.7}$
Layer 8, Head 2	$83.7_{\uparrow 0.4}$	$84.5_{\downarrow 2.2}$	$69.5_{\downarrow 4.0}$	$31.6_{\downarrow 1.4}$	$27.5_{\downarrow 5.9}$
Layer 8, Head 11	$81.0_{\downarrow 2.3}$	$88.3_{\uparrow 1.6}$	$76.5_{\uparrow 3.0}$	$37.6_{\uparrow 4.6}$	$34.1_{\uparrow 0.7}$
Layer 9, Head 9	$83.4_{\uparrow 0.1}$	$87.7_{\uparrow 1.0}$	$74.2_{\uparrow 0.7}$	$36.1_{\uparrow 3.1}$	$32.4_{\downarrow 1.0}$
Layer 10, Head 10	$81.5_{\downarrow 1.8}$	$84.7_{\downarrow 2.0}$	$72.2_{\downarrow 1.3}$	$41.1_{\uparrow 8.1}$	$38.7_{\uparrow 5.36}$
L2.H5 & L3.H7 & L4.H0	$80.4_{\downarrow 2.9}$	$3.6_{\downarrow 83.1}$	$11.1_{\downarrow 62.4}$	$20.1_{\downarrow 12.9}$	$4.7_{\downarrow 28.7}$

Table 1: The effects of ablating the 9 heads occurring in all MNLI subnetworks individually from the full model. Interestingly, some are also important for counter-heuristic behavior, as their ablation leads to reduced generalization. In the final row, Lx.Hy refers to Head y in Layer x.

Ablated Head	Accuracy (%) ↑	
	QQP	QQP-PAWS (adv.)
None	91.2	14.8
Layer 2, Head 3	$91.1_{\downarrow 0.1}$	$14.1_{\downarrow 0.7}$
Layer 2, Head 5	$91.1_{\downarrow 0.1}$	$9.9_{\downarrow 4.9}$
Layer 3, Head 3	$91.2_{\downarrow 0.0}$	$13.9_{\downarrow 0.9}$
Layer 4, Head 0	$90.2_{\downarrow 1.0}$	$3.3_{\downarrow 11.6}$
Layer 8, Head 11	$91.0_{\downarrow 0.2}$	$14.6_{\downarrow 0.2}$
Layer 9, Head 1	$91.2_{\downarrow 0.0}$	$16.6_{\uparrow 1.8}$
Layer 9, Head 9	$91.1_{\downarrow 0.1}$	$16.2_{\uparrow 1.4}$
Layer 11, Head 6	$91.1_{\downarrow 0.1}$	$17.1_{\uparrow 2.3}$

Table 2: The effects of ablating the 8 heads occurring in all QQP subnetworks individually from the full model. The heads with the strongest reduction are the same ones that had the largest effect on the MNLI model.

tuning BERT with different random seeds leads to disparate generalization despite similar in-domain accuracies. Other works have identified NLP tasks where OOD accuracies increase long after in-domain accuracy saturates, and attributed them to hard minority examples (Tu et al., 2020; Bhargava et al., 2021). Recently, a phenomenon called Grokking (Power et al., 2022) was discovered, where test accuracy on a toy algorithmic task increases abruptly, long after overfitting to the training data. This phenomenon has been explained in terms of competing subnetworks (Merrill et al., 2023; Varma et al., 2023). Grokking-like phenomena have also been identified in other settings (Liu et al., 2023) as well as for small transformers (Murty et al., 2023) on hierarchical depth generalization. Simultaneously, questions about memorizing in pre-trained LMs have been asked in other settings apart from grokking (Tänzer et al., 2022).

Subnetworks as modular components. An emerging body of work (Lepori et al., 2023b; Choenni et al., 2023; Hupkes et al., 2020) suggests that Neural Networks and Language Models may consist of modular subnetworks each performing different subtasks. Bayazit et al. (2023) find a subnetwork in GPT-2 responsible for most of its factual knowledge – removing it does not lead to loss of performance on non-knowledge-intensive tasks. Li et al. (2023) report the possibility of ablating away undesired behavior in a model (e.g. toxicity).

7 Conclusion

We showed that pruning a BERT model with different random seeds can yield subnetworks that perform similarly to the original model in-domain but generalize differently. Moreover, sparser subnetworks generalize worse. To explain this result, we first considered the *Competing Subnetworks Hypothesis*, which has been used to explain grokking on algorithmic tasks; it suggests that the model consists of disjoint subnetworks representing different solutions, some which generalize and some which do not. We found that our results are instead more consistent with the existence of a *Heuristic Core*: a subset of attention heads that appear in all subnetworks and calculate shallow, non-generalizing features. The model generalizes by incorporating additional heads that interact with the heuristic core. Our results have practical implications about the effect of pruning on generalization and raise intriguing questions about how language models learn to generalize by composing simple components.

Limitations

Our approaches are not without limitations. Our analysis is based on a top-down approach of identifying subnetworks. Thus, we do not look for or discover "circuits" that are atomically interpretable. Our experiments only use text classification tasks, although we believe that such a task is ideal due to clear definitions of heuristics. Additionally, we only work with BERT models—it is possible that larger transformers, or those pretrained differently, exhibit different behavior. Finally, although we have analyzed the mechanisms of generalization in this paper, the question of why they are undertaken remains elusive for the moment. Our experiments also raised many novel questions, which require further research to answer.

Ethical Considerations

We do not foresee any ethical considerations raised due to our work. We work with pre-trained Language Models, which can at times hallucinate or demonstrate other undesirable behavior. Our work, however, does not interact with these facets of model behavior. Additionally, we only work with English datasets in our paper.

References

Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham M. Kakade, eran malach, and Cyril Zhang. 2022. Hidden progress in deep learning: SGD learns parities near the computational limit. In *Advances in Neural Information Processing Systems*.

Deniz Bayazit, Negar Foroutan, Zeming Chen, Gail Weiss, and Antoine Bosselut. 2023. Discovering knowledge-critical subnetworks in pretrained language models.

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in NLI: Ways (not) to go beyond simple heuristics. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 125–135, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rochelle Choenni, Ekaterina Shutova, and Dan Garrette. 2023. Examining modularity in multilingual lms via language-specialized subnetworks.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Nicola De Cao, Michael Sejr Schlichtkrull, Wilker Aziz, and Ivan Titov. 2020. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3243–3255, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Babak Hassibi and David Stork. 1992. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? (extended abstract). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 5065–5069. International Joint Conferences on Artificial Intelligence Organization. Journal track.

Shankar Iyer, Nikhil Dandekar, Kornél Csernai, et al. 2017. First Quora dataset release: Question pairs. *quora.com*.

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P. Dick, Hidenori Tanaka, Edward Grefenstette, Tim Rocktäschel, and David Scott Krueger. 2023. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Pa-*

pers), pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander Rush. 2021. Block pruning for faster transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10619–10629, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

Michael A. Lepori, Ellie Pavlick, and Thomas Serre. 2023a. Neurosurgeon: A toolkit for subnetwork analysis.

Michael A. Lepori, Thomas Serre, and Ellie Pavlick. 2023b. Break it down: Evidence for structural compositionality in neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Maximilian Li, Xander Davies, and Max Nadeau. 2023. Circuit breaking: Removing model behaviors with targeted ablation.

Ziming Liu, Eric J Michaud, and Max Tegmark. 2023. Omnigrok: Grokking beyond algorithmic data. In *The Eleventh International Conference on Learning Representations*.

Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. LLM-pruner: On the structural pruning of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Association for Computational Linguistics (ACL)*, pages 3428–3448.

William Merrill, Nikolaos Tsilivis, and Aman Shukla. 2023. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher Manning. 2023. Grokking of hierarchical structure in vanilla transformers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 439–448, Toronto, Canada. Association for Computational Linguistics.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. 2022. Grokking: Generalization beyond overfitting on small algorithmic datasets.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2023. A simple and effective pruning approach for large language models.

Michael Tänzer, Sebastian Ruder, and Marek Rei. 2022. Memorisation versus generalisation in pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7564–7578, Dublin, Ireland. Association for Computational Linguistics.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. 2023. Explaining grokking through circuit efficiency.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2020. Structured pruning of large language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6151–6162, Online. Association for Computational Linguistics.

Kaiyue Wen, Yuchen Li, Bingbin Liu, and Andrej Risteski. 2023. Transformers are uninterpretable with myopic methods: A case study with bounded Dyck grammars. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1112–1122.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@NeurIPS 2023)*.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1513–1528, Dublin, Ireland. Association for Computational Linguistics.

Linyi Yang, Yaoxian Song, Xuan Ren, Chenyang Lyu, Yidong Wang, Jingming Zhuo, Lingqiao Liu, Jindong Wang, Jennifer Foster, and Yue Zhang. 2023. Out-of-distribution generalization in natural language processing: Past, present, and future. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4533–4559, Singapore. Association for Computational Linguistics.

Fred Zhang and Neel Nanda. 2023. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*.

Minjia Zhang and Yuxiong He. 2020. Accelerating training of transformer-based language models with progressive layer dropping. In *Advances in Neural Information Processing Systems*, volume 33, pages 14011–14023. Curran Associates, Inc.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

A Examples from the HANS and PAWS-QQP datasets

We list examples from some of the HANS cases and subcases in Table 3. Table 4 lists examples for PAWS-QQP. Note that each subcase of HANS has 1000 validation examples. PAWS-QQP has 12663 examples in all, of which 8696 bear the verdict "not equivalent". We work with the latter subset.

B Model and Training Details

BERT Architecture Our experiments use fine-tuned BERT models (Devlin et al., 2019). In the simplest setting, a Transformer Model (Vaswani et al., 2017) consists of L blocks, where each block is in turn composed of a Multi-Head *Attention* layer followed by an *MLP* layer. Given an input X, the Attention Layer outputs

$$X + \sum_{i=1}^{N_h} \text{Attn}(W_Q^{(i)}, W_K^{(i)}, W_V^{(i)}, X) W_O^{(i)}$$

where N_h is the number of attention heads, and $W_K^{(i)}$, $W_Q^{(i)}$ and $W_V^{(i)}$ denote the Key, Query, and Value matrices, respectively. $W_O^{(i)}$ performs a linear output projection, before adding the result to the *residual stream* via a skip connection. On the other hand, an MLP layer first up-projects its input X by multiplying with $W_U \in \mathbb{R}^{d \times d_h}$, then passes the result through an activation function such as GeLU, then down-projects the result back via $W_D \in \mathbb{R}^{d \times d_h}$. Here d is the hidden dimension and d_h typically equals $4d$. This is then added to the residual stream to produce

$$X + \text{GeLU}(XW_U)W_D$$

Layer Normalization precedes both layers.

BERT Fine-Tuning We fine-tuned BERT (`bert_base_cased`, 125M parameters) on MNLI for 5 epochs (61360 steps with batch size $= 32$), with a learning rate of $2 \cdot 10^{-5}$, a and a maximum sequence length of 128. Checkpointing and evaluation were performed every 500 steps. For QQP, we opted for 40000 steps of fine-tuning with an effective batch size of 256 and a learning rate of $2 \cdot 10^{-5}$. Evaluation and checkpoint were once again performed every 500 steps. All runs used the Adam optimizer (Kingma and Ba, 2015), with $\beta = (0.9, 0.999)$ and $\epsilon = 10^{-8}$.

Each run takes approximately 3 hours to complete on one NVIDIA A5000 GPU.

C Details of Pruning and L0 Regularization

Our formulation of the masks is the same as CoFi Pruning (Xia et al., 2022)'s, and is built on top of Louizos et al. (2018)'s work. Specifically, the masks z are modeled as hard concrete distributions:

$$\mathbf{u} \sim \text{Uniform}(\epsilon, 1 - \epsilon)$$

Subcase	Name	Abbreviation	Example
Lexical Overlap (LO)			
Subject-Object Swap	`lexical_overlap_ ln_subject-object_swap`	SO-Swap	The doctor advised the president. $\not\rightarrow$ The president advised the doctor.
Preposition	`lexical_overlap_ ln_preposition`	Prep	The tourist by the manager saw the artists. $\not\rightarrow$The artists saw the manager.
Constituent (C)			
Embedded under if	`constituent_ cn_embedded_under_if`	Embed-If	If the artist slept, the actor ran. $\not\rightarrow$ The actor ran.
Embedded under verb	`constituent_ cn_embedded_under_verb`	Embed-Verb	The lawyers believed that the tourists. shouted $\not\rightarrow$ The tourists shouted.

Table 3: Example cases and adversarial subcases from the HANS (McCoy et al., 2020) dataset. We focus mostly on these subcases in the main text as the model generalizes best to them. We introduce abbreviated names for ease of discussion.

Question #1	Question#2	Equivalent?
Is a contagious yawn also fake?	Is a fake yawn also contagious?	✗
How are noble gases stable?	How are noble gases stable?	✗
How is Perth better than Melbourne?	How is Melbourne better than Perth?	✗

Table 4: Example questions from the PAWS-QQP dataset (Zhang et al., 2019) dataset. We only work with questions whose label is "not equivalent", as the others are heuristic-friendly.

$$\mathbf{s} = \sigma\left(\frac{1}{\beta} \cdot \frac{\mathbf{u}}{1-\mathbf{u}} + \log \alpha\right)$$
$$\tilde{\mathbf{s}} = \mathbf{s} \times (r - l) + l$$
$$\mathbf{z} = \min(1, \max(0, \tilde{\mathbf{s}}))$$

Here, σ denotes the sigmoid function, and $\epsilon = 10^{-6}$ is chosen to avoid division by 0 errors. The temperature is fixed at $\frac{1}{\beta} = \frac{2}{3}$. The result is then stretched to the interval $[l, r] = [-0.1, 1.1]$, with the probability mass from $[-0.1, 0]$ and $[1, 1.1]$ accumulated to 0 and 1 in the end. This ensures that the mask is incentivized to contain values close to 0 or 1. Here, the log alphas $\log \alpha$ are the main learnable parameters.

Following Wang et al. (2020) and Xia et al. (2022), we enforce a target sparsity via a Lagrangian Term. Supposing the target and current sparsity to be s and t, we add

$$\mathcal{L}_s = \lambda_1(t - s) + \lambda_2(t - s)^2$$

to the loss (CoFi uses $s - t$ instead of $t - s$, but the two are functionally the same; consider the transform $\lambda_1 \longrightarrow -\lambda_1$). Along with training the model parameters, a gradient *ascent* is performed on λ_1 and λ_2, so that the updates of the lambdas always keep adjust the contribution of $\mathcal{L}_s$ to keep the sparsity in check.

Pruning - General Commentary We found that the hyperparameters were very sensitive to the choice of learning rate. Deviation even by a factor of 3 would cause instability in training. Specifically, if the learning rate for the ascent on the Lagrangian Parameters (λ_1 and λ_2) was too low, the model would stay under a sparsity of 10^{-5} until the target sparsity climbed to 0.15 or so, and would then jump to a value beyond the target, sacrificing a lot of performance in the process. On the other hand, a large value also leads to insufficient time for "settling into" the intermediate sparsities. The exact choice of the number of optimization steps was also important, although not as sensitive as the learning rate.

BERT Pruning We used a maximum sequence length of 128 as before, with an effective batch size of 128 including gradient accumulation. The learning rates for log alphas and lambdas were 0.1 and 1, respectively. We use the formulae

$$\texttt{warmup_steps} = 6500 + 50 \times \texttt{sparsity_pct}$$

$$\texttt{total_steps} = \texttt{warmup_steps} + 6 \times \texttt{sparsity_pct}$$

to compute the number of steps taken to linearly warm up the target sparsity, and the total number of steps, respectively. For sparsity targets of 50%, these translate to 9300 training steps, of which the first 9000 warm up the target sparsity. For 70% target, these change to 10500 and 10000, respectively. Evaluation and checkpointing were done every 64

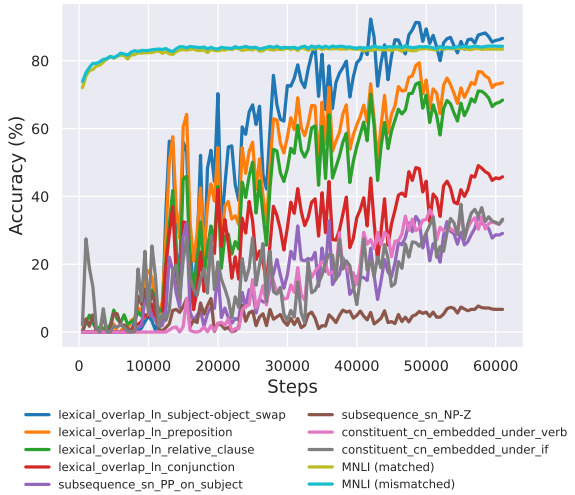


Figure 10: Model performance saturates early on MNLI, but generalizing traits are learned much later.

steps but we always used the final checkpoint to report results. The thresholds were chosen with a grid search on $[0, 1]$ with stride 0.05 to match the target sparsity as closely as possible. They were found to always be 0.5 and 0.35 for target sparsities of 0.5 and 0.7.

Each pruning run takes approximately 90 minutes to complete on an A5000 GPU. We estimate our total computational budget to be 750 GPU hours.

D More results

In this section, we provide further exposition on some of the results from the main text. For these results, we use the full names of the HANS subcases instead of abbreviations.

Figure 10 showcases the in-domain v/s out-of-domain training dynamics on MNLI/HANS for more subcases. The trend remains the same—OOD accuracy starts increasing much after ID accuracy saturates. For the other subcases not shown here, the model never gets off the ground.

Figures 13 and 14 show that the frequency of the heads is strongly correlated between the 50% and 70% sparsity subnetworks for MNLI/HANS, and the 30% and 60% sparsity subnetworks for QQP. This point is further elaborated by the frequency heatmap of Figure 11, the counterpart of Figure 5. Essentially, the heads most important to the heuristic subnetworks also appear often in the generalizing subnetworks, once again showing that the heuristic subnetworks are subsets of generalizing subnetworks.

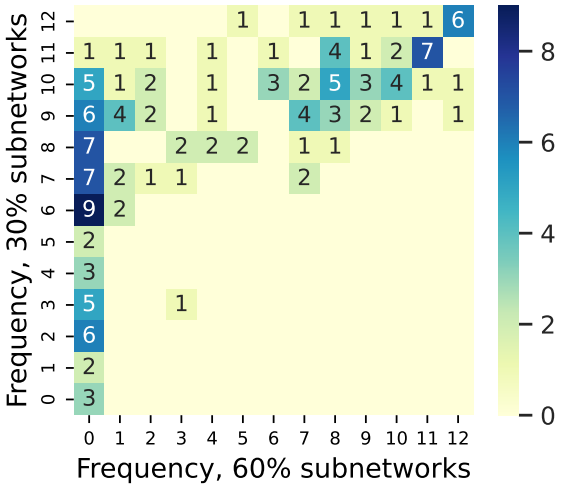
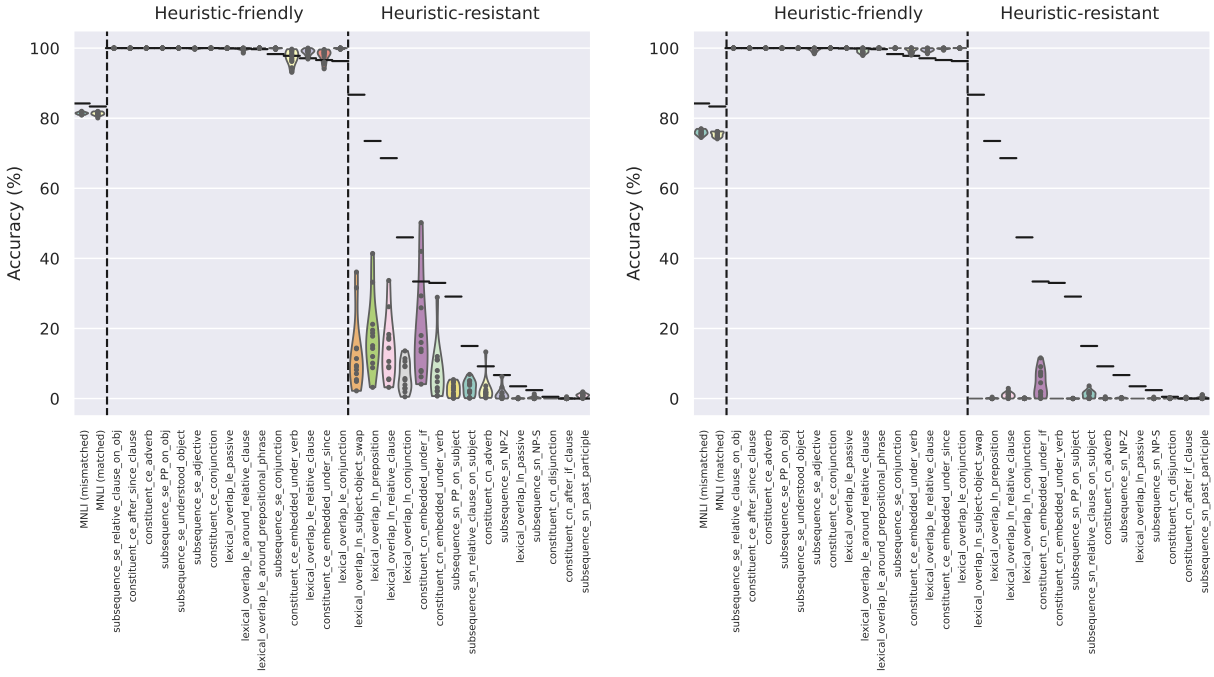


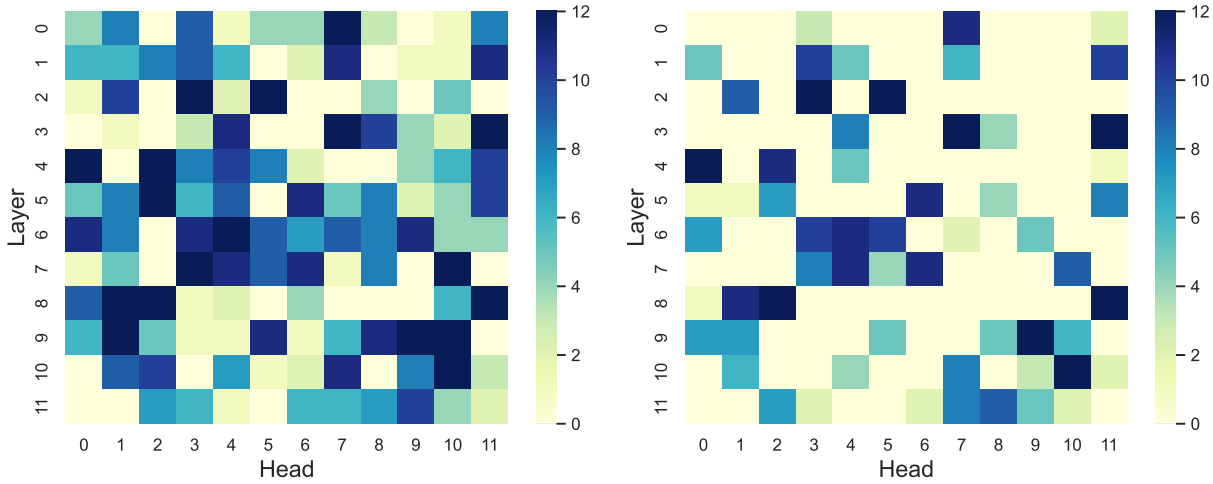
Figure 11: This heatmap quantifies the frequencies of attention heads in the 30% and 60% sparsity subnetworks of the QQP model. Entry (i, j) denotes the number of heads appearing in $i/12$ 30% subnetworks and $j/12$ 60% subnetworks. Eight attention heads appear in all subnetworks.



(a) Sparsity = 50%

(b) Sparsity = 70%

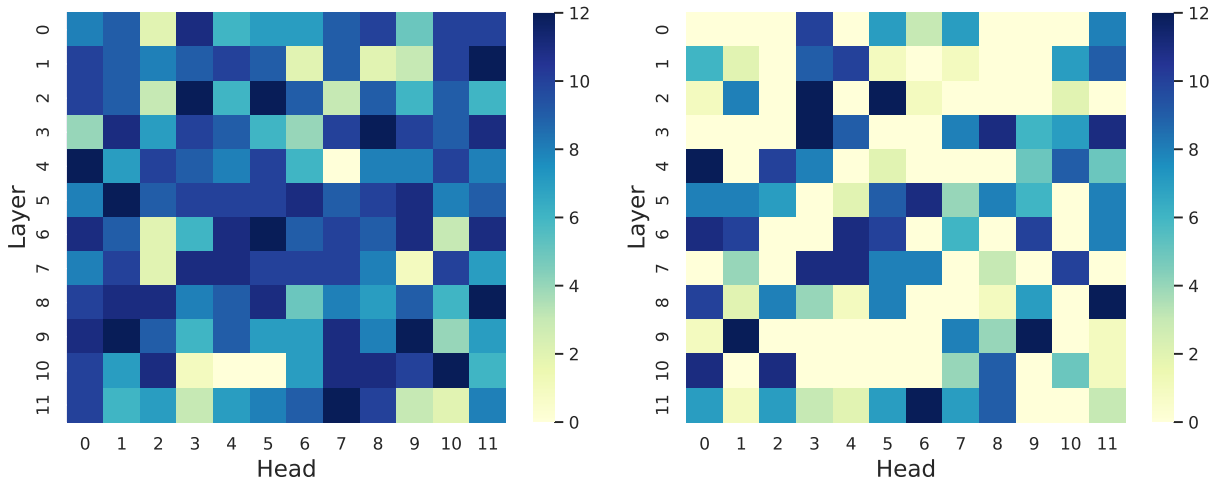
Figure 12: The full results of the multi-seed pruning experiment on MNLI/HANS. The subnetworks perform close to the model both in-domain and on heuristic-friendly subcases of HANS. Ay 50% sparsity, they show disparate generalization to the heuristic-resistant subcases, while at 70%, they degenerate to heuristic behavior. This is also corroborated by the fact that they slightly overperform the full-model on the heuristic-friendly cases.



(a) Sparsity = 50%

(b) Sparsity = 70%

Figure 13: The frequency of various attention heads across the subnetworks pruned with 12 random seeds. The heads part of the 70% sparsity subnetworks (non-generalizing) are also popular in 50% sparsity (partially generalizing) subnetworks. The Separman's Rho agreement between the frequencies of attention heads at the two sparsities is 0.82 (p-value = $1.6 \cdot 10^{-36}$), corresponding to very strong agreement.



(a) Sparsity = 30%

(b) Sparsity = 60%

Figure 14: The frequency of various attention heads across in the QQP subnetworks, pruned with 12 random seeds. Once again, the frequencies of the heads are highly correlated between the two sparsities. The Separman's Rho agreement in this case is 0.74 (p-value = $7.7 \cdot 10^{-26}$), corresponding to strong agreement.