Fine-Grained Analysis of Shared Syntactic Mechanisms in Language Models

Ryoma Kumon[1,2] **Hitomi Yanaka**[1,2,3]
[1]The University of Tokyo [2]RIKEN [3]Tohoku University
{kumoryo9, hyanaka}@is.s.u-tokyo.ac.jp

Abstract

While language models demonstrate sophisticated syntactic capabilities, the extent to which their internal mechanisms align with cross-constructional principles studied in linguistics remains poorly understood. This study investigates whether models employ shared neural mechanisms across different syntactic constructions by applying causal interpretability methods at a granular level. Focusing on filler-gap dependencies and negative polarity item (NPI) licensing, we utilize activation patching to identify the functional roles of specific attention heads and MLP blocks. Our results reveal a highly localized and shared mechanism for filler-gap dependencies located in the early to middle layers, whereas NPI processing exhibits no such unified mechanism. Furthermore, we find that these mechanisms identified by activation patching generalize to out-of-distribution, while distributed alignment search, a supervised interpretability method, is susceptible to overfitting on narrow linguistic distributions. Finally, we validate our findings by demonstrating that the manipulation of the identified components improves model performance on acceptability judgment benchmarks.[1]

1 Introduction

Recent advancements in language models (LMs) have demonstrated their ability to process linguistic expressions with complex syntactic structures. However, it remains unclear to what extent their internal mechanisms align with linguistic theory or resemble human processing mechanisms. While linguistics has long studied underlying elements shared across constructions, clarifying whether LMs employ shared syntactic mechanisms across different constructions will lead to a deeper understanding of both linguistic structure and language modeling (Futrell and Mahowald, 2025).

To investigate the internal processes, causal abstraction methods (Vig et al., 2020; Geiger et al., 2021, 2024) have emerged for identifying the causal impact of internal components on model behavior. Previous studies have utilized these methods to analyze syntactic mechanisms (Arora et al., 2024; Elazar et al., 2021), specifically investigating the commonality of mechanisms in subject-verb agreement (Finlayson et al., 2021) and filler-gap dependencies (Boguraev et al., 2025).

However, several challenges remain. First, existing analyses primarily focus on examining representations in the residual stream and comparing their scores across tokens, without identifying the components that impact the residual stream, leaving the underlying mechanisms at the level of layers and attention heads largely unexplored. As a result, it remains unclear at what level of granularity shared mechanisms operate. For example, Boguraev et al. (2025) analyzes representations in the residual stream, but this approach may fail to distinguish between mechanisms that rely on different sets of attention heads or MLP blocks while producing similar residual representations.

Second, causal analysis methods that require training are susceptible to overfitting, which makes verifying out-of-distribution (OOD) generalization essential (Wu et al., 2024b). Nevertheless, prior research applying these methods to the analysis of syntactic mechanisms (Arora et al., 2024; Boguraev et al., 2025) has not sufficiently validated the OOD generalization of the identified mechanisms. Thus, their faithfulness, which concerns whether the identified components truly reflect the model's internal mechanisms rather than artifacts of specific datasets or lexical distributions, remains unestablished.

To address these issues, this study employs activation patching (Vig et al., 2020; Geiger et al., 2021), which is a causal analysis method without any additional training, and investigates the

[1]All code and datasets are avilable at https://github.com/ynklab/shared_syntactic_mechanism.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 44866–44891
July 2-7, 2026 ©2026 Association for Computational Linguistics

commonality of mechanisms across different constructions at the granularity of layers and attention heads. We focus on two well-studied phenomena, filler-gap dependencies (FGDs) and negative polarity item (NPI) licensing. Both phenomena appear across diverse surface constructions while posing distinct linguistic challenges. FGDs require long-range structural processing, whereas NPI licensing involves integrating syntactic structure with semantic properties of licensors, which may give rise to different or construction-specific mechanisms. We analyze whether LMs employ any shared mechanisms across constructions for each of these phenomena and, if so, how the mechanisms work. Furthermore, to validate the identified mechanisms, we evaluate changes in model accuracy on acceptability judgment benchmarks by manipulating the identified components. Additionally, we assess the OOD generalization of these mechanisms by comparing our results with distributed alignment search (DAS; Geiger et al., 2024), a training-based causal interpretability method.

Our analysis reveals that, while LMs have a shared mechanism in FGDs, such commonality was not observed in NPI licensing. Specifically, we found that the mechanisms contributing to FGDs are localized within a small number of attention heads situated in the early to middle layers. Furthermore, amplifying the activation values of these identified components led to improved accuracy on acceptability judgment tasks. Additionally, while DAS results fluctuated significantly in OOD settings, suggesting a potential overfitting to specific lexical or syntactic distributions, activation patching yielded consistent results across different distributions. These findings demonstrate the validity and faithfulness of our analytical approach.

2 Related Work

2.1 Analysis of Syntactic Mechanisms

Analysis of syntactic mechanisms in LMs is broadly categorized into probing, which examines the extent to which models encode syntactic knowledge (Hewitt and Manning, 2019; Clark et al., 2019), and causal analysis, which investigates the mechanisms that causally influence model behavior during syntactic processing (Elazar et al., 2021; Lasri et al., 2022; Arora et al., 2024). To elucidate the mechanisms directly governing model behavior, this study adopts the latter causal approach.

Shared mechanisms across different constructions have been explored through both causal methods (e.g., subject-verb agreement (Finlayson et al., 2021) and FGDs (Boguraev et al., 2025)) and probing (Kryvosheieva et al., 2025). However, studies using causal methods mainly focus on token-level analysis, leaving the degree of commonality at finer granularities, such as attention heads, largely unexplored. Moreover, learning-based causal methods face the risks of overfitting, which requires verification of OOD generalization (Wu et al., 2024b), and yet existing studies (Arora et al., 2024; Boguraev et al., 2025) have not sufficiently validated this aspect. Conversely, probing-based analyses (Kryvosheieva et al., 2025) fail to verify the direct causal impact of identified components on actual behavior. In this work, we primarily utilize training-free causal analysis to investigate both OOD generalization and the functional impact of internal components on model behavior.

2.2 Filler-Gap Dependencies and Negative Polarity Items

FGDs refer to relationships between a filler and a gap, where the filler is typically a wh-word or phrase, and the gap is an empty position corresponding to the filler. These dependencies underlie various constructions. For example, in the EWHK construction in Table 1, the filler "who" is interpreted as the direct object of "liked" and creates the gap before ".", making the pronoun "her" ungrammatical. Several studies (Wilcox et al., 2018; Ozaki et al., 2022; Wilcox et al., 2024; Lan et al., 2024) have conducted surprisal-based analyses using FGDs to investigate the syntactic capabilities of LMs and have shown the sensitivity of LMs to structural constraints. Generalizations across constructions of FGDs have also been analyzed. It has been reported that long short-term memory (LSTM) networks struggle to generalize to filler-gap constructions beyond those seen in training (Howitt et al., 2024) and that LMs trained on child-language data struggle to generalize across constructions (Chang et al., 2025).

NPIs are expressions that must be licensed by certain contexts, such as negation and questions. For instance, in the DNEG construction in Table 1, the NPI "any" is licensed by the licensor "No". Early formal accounts attempted to uniformly explain that NPIs are licensed by downward entailing environments (Ladusaw, 1979). However, Zwarts (1998) and Giannakidou (1998) proposed a more nuanced hierarchy of NPIs based on the strength

Filler-Gap Dependencies								
Construction	Abbr.	Prefix	Filler	NC	Article	Noun	Verb	Output
Emb. Wh-q. (know)	EWHK	The man knows	[**who/that**]		the	teacher	liked	[**./her**]
Emb. Wh-q. (wonder)	EWHW	The boy wondered	[**who/if**]		the	doctor	admired	[**./him**]
Matrix wh-q	MWH	Then,	[**who/**ϕ]	did	the	girl	choose	[**?/them**]
Restr. Rel. Clause	RELCL	The customer	[**who/and**]		the	lady	sounded like	[**./me**]
Cleft	CLEFT	It was	[**the man/clear**]	that	the	boss	scared	[**./him**]
Pseudo-cleft	PCLEFT		[**Who/That**]		the	dancer	is listening to	[**is/you**]
Topicalization	TOPIC	Actually,	[**the kid/**ϕ]		the	guest	hated	[**./them**]
Negative Polarity Item Licensing								
Construction	Abbr.	Prefix	Licensor	NC			Last	Output
Conditionals	COND	The host will sleep	[**if/while**]	the guest			eats	[**any/some**]
Determiner Negation	DNEG		[**No/The**]	patient have			liked	[**any/some**]
Scope of Only	SONLY		[**Only/Even**]	the boys			have	[**any/some**]
Quantifers	QNT	These are	[**all/some**]	of the students who			showed	[**any/some**]
Embedded Questions	EMBQ	The senators	[**wonder whether /know that**]	the man has			found	[**any/some**]
Simple Questions	SMPQ		[**Has/**ϕ]	the actor			sold	[**any/some**]
Superlatives	SUP	This is the	[**fastest/fast**]	kid that had			liked	[**any/some**]
Only	ONLY	They are the	[**only/upset**]	teachers that			makes	[**any/some**]
Control								
Construction	Abbr.	Prefix	Filler/Licensor	NC	Article	Noun	Verb/Last	Output
Capital Knowledge	CTRL		[**Paris/Rome**]	is	the	captital	of	[**France/Italy**]

Table 1: Examples of minimal pairs of the constructions in the dataset. NC stands for “No comparison", and is not used for analysis.

of their licensing requirements. This was motivated by the observation that different NPIs require different licensing contexts and many are licensed by non-downward entailing environments, such as questions. Regarding LMs, Warstadt et al. (2019) showed that masked LMs can handle NPIs. Jumelet et al. (2021) argued that a two-layer LSTM language model processes NPI constructions uniformly through a monotonicity-based mechanism based on probing the internal representation. Under this mechanism, the model evaluates NPI licensing by determining whether the items occur within a downward entailing environment. DeCarlo et al. (2023) showed that LMs handle NPI licensing in a more complex and non-uniform way, based on analysis of output probabilities. In contrast, we study the internal workings of LMs in processing FGDs and NPI licensing by employing causal interpretability methods rather than probing or behavior analysis focused on model output.

3 Experimental Settings

3.1 Dataset

We construct the dataset for analysis, which consists of minimal pairs of constructions for FGDs and NPI licensing. Table 1 shows examples of minimal pairs of each construction in FGDs and NPI licensing, seven and eight each. We determine the construction coverage based on the previous studies for FGDs (Boguraev et al., 2025) and for NPI licensing (Warstadt et al., 2019; DeCarlo et al., 2023). These constructions have lexical variations, which makes it hard for interpretability methods to capture shallow lexical patterns. For example, the constructions in FGDs have several words for filler, which prevents us from identifying a mechanism sensitive to a single lexical item such as “who”.

To distinguish mechanisms associated with syntactic structures from those impacted by unrelated factors such as linear distance, we incorporate a control construction that involves the retrieval of world knowledge. Specifically, we use sentences that require the model to associate a capital city with its country name. The sentences are designed so that the linear distance between the dependent tokens is similar to that in FGDs and NPI constructions.

We generate the dataset by extending the data generation scripts[2] by Warstadt et al. (2020). The generation scripts use pre-defined structural templates, with lexical items sampled from an exten-

[2] https://github.com/alexwarstadt/data_generation

sive vocabulary. This approach enables the creation of sentences that span a wider distribution of vocabulary and constructions than previous studies (Arora et al., 2024; Boguraev et al., 2025). Notably, using only one output word per construction pattern may lead to the identification of a mechanism specific to that lexical item and cause supervised interpretability methods to overfit to the output token. To mitigate these issues, we use multiple words for output tokens. Note that, for the NPI constructions, we only use "ever" and "any" for output tokens because these are the only items that are licensed across all eight selected licensing contexts.

For the training and evaluation of DAS, we also split the dataset into the training, in-distribution (ID), and OOD test sets. The training and ID test sets are generated from the same vocabulary set, and the OOD test set is generated from an entirely disjoint vocabulary to evaluate the robustness[3]. We ensure that no sentences overlap between the training and ID test sets. Unless otherwise specified, we report the results averaged across the ID and OOD test sets. The examples in Table 1 are all drawn from the ID test set.

3.2 Models

We use Pythia 1B, 2.8B, 6.9B (Biderman et al., 2023) and Gemma 3 1B, 12B (Gemma Team, 2025). Using different model sizes and model families allows for analyses of their effect on the development of syntactic mechanisms. Furthermore, we use multiple checkpoints at different training steps for Pythia models at 1k, 2k, 3k, 4k, 10k, and 143k (final checkpoint) to compare how the mechanism is acquired throughout the pre-training. The Pythia family is particularly suited for this comparison because all models are trained on the same training data in the same order. Unless otherwise specified, we report the results of Pythia 1B.

4 Analysis of Shared Mechanisms

4.1 Activation Patching

To analyze mechanisms employed in the task described in the previous section, we use activation patching (Vig et al., 2020; Geiger et al., 2021). Activation patching is a method to identify which component is responsible for the model behavior and is defined by the following:

$$f_{\text{interv}}(\boldsymbol{b}, \boldsymbol{s}) = f(\boldsymbol{s})$$

It replaces the activation $f(\boldsymbol{b})$ of a component f during an inference of a base input $\boldsymbol{b}$ with cached activations $f(\boldsymbol{s})$ from another inference with a source input $\boldsymbol{s}$, and observes how this affects the model's output. A source input is designed to be minimally different from a base input, the only difference being the concept of interest, to isolate the causal role regarding the concept.

4.2 Evaluation Metric

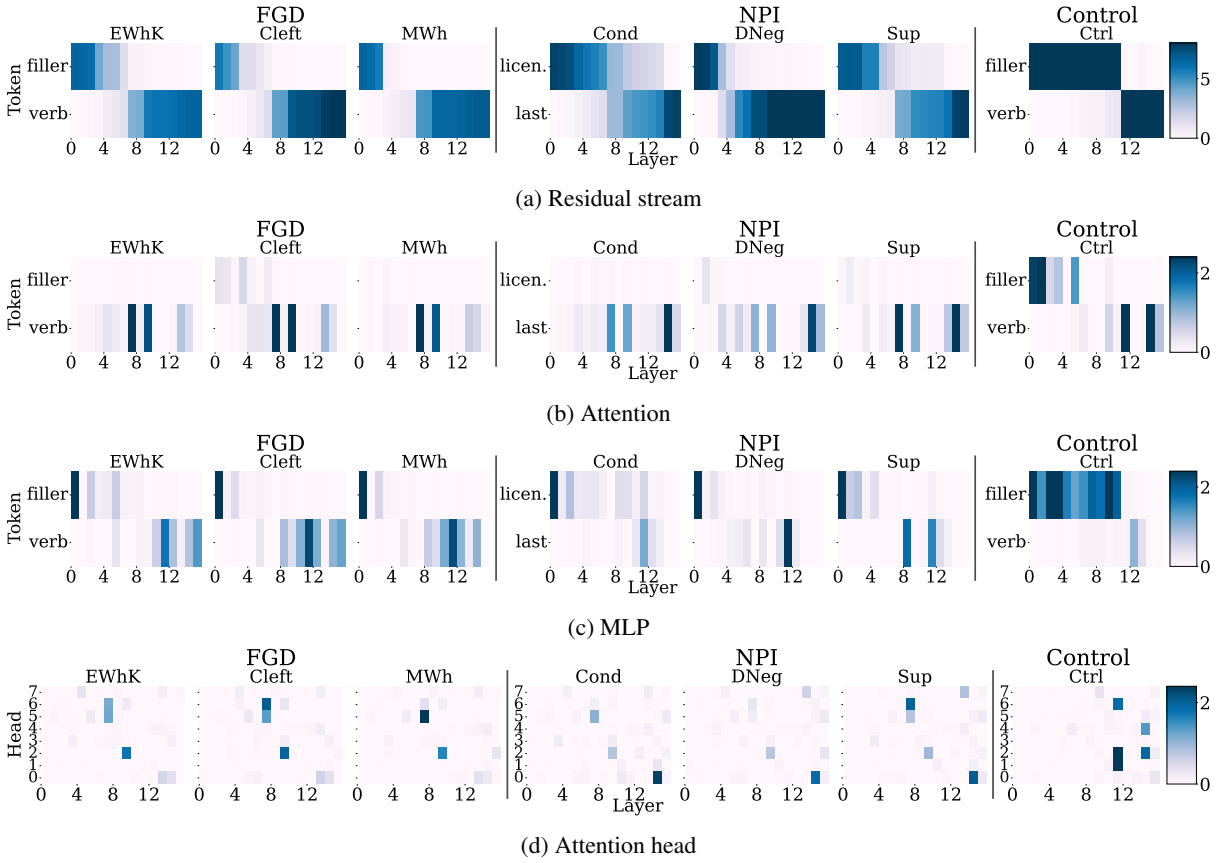
To quantify causal effect, we employ a modified version of log-odds ratio (ODDS), originally introduced by Arora et al. (2024). It is defined as:

$$\text{ODDS}(p, p_{f_{\text{interv}}}, T) = \frac{1}{|T|} \sum_{(\boldsymbol{b},\boldsymbol{s},y_b,y_s)\in T} \log\left(\frac{p(y_b|\boldsymbol{b})}{p(y_b|\boldsymbol{s})} \frac{p_{f_{\text{interv}}}(y_b|\boldsymbol{s},\boldsymbol{b})}{p_{f_{\text{interv}}}(y_b|\boldsymbol{b},\boldsymbol{s})}\right),$$

where y_b and y_s are the output tokens of the inputs $\boldsymbol{b}$ and $\boldsymbol{s}$, respectively, and T denotes the test set. While the original metric compares the relative probabilities of two tokens y_b and y_s given a single input, we modify it to measure the shift in probability for a specific token y_b for two inputs $\boldsymbol{b}$ and $\boldsymbol{s}$ before and after the targeted intervention. This adjustment is critical because the grammatical licensing of an NPI is largely independent of the choice of alternative, non-NPI lexical items. We use only minimal pairs that have NPIs as y_b and non-NPIs as y_s for analysis, which leads to measuring the shift in probability of NPIs across two inputs. Note that these restrictions do not apply to the minimal pairs in FGDs, and the ODDS score in FGDs is equivalent to the one originally defined by Arora et al. (2024) (see details in Appendix C).

This metric captures the extent to which a specific model component contributes to the next token prediction. A higher ODDS score indicates that the intervened component is more causally efficacious. Thus, we expect higher scores in components that propagate information from critical tokens, such as fillers or licensors, to the final layers at the last token. If mechanisms are shared across constructions, score distributions will be similar across constructions at each component level. In contrast, distinct mechanisms should yield different score distributions.

[3] Boguraev et al. (2025) examine the impact of animacy differences in the inputs. However, in their setting, some of the output tokens overlap, and therefore their evaluation is not strictly out-of-distribution.



(a) Residual stream

(b) Attention

(c) MLP

(d) Attention head

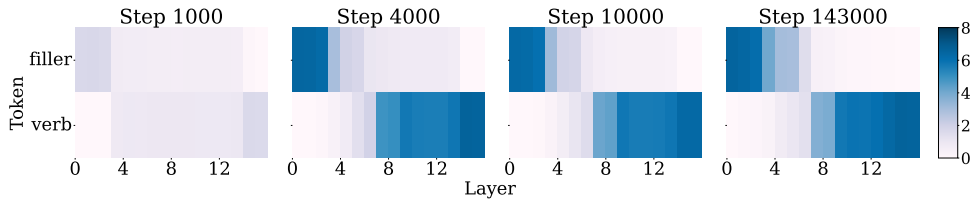
Figure 1: ODDS scores with activation patching in Pythia 1B. Note that layer numbers are zero-indexed and token names correspond to those in Table 1. The full results including other constructions and tokens are presented in Appendix D.

4.3 Results

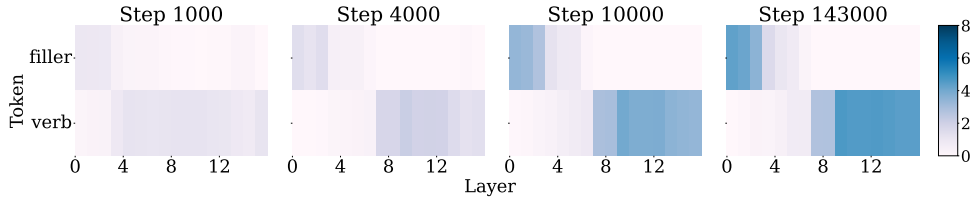
Figure 1 illustrates the results of the activation patching for each component. The distribution of ODDS scores across components is largely consistent across different FGD constructions, whereas the control pattern exhibits a distinct trend. Specifically, we observe that attention heads contributing to the prediction at the final token are sparsely located in the middle layers, particularly heads 7.5, 7.6, and 9.2, and this localization is shared across constructions. Furthermore, the ODDS scores of the residual stream at the final token show a sharp increase around the seventh layer, aligning with the depth of these heads. We also observe the same trend in naturally occurring sentences (see details in Appendix D.6).

This pattern reflects a consistent information flow across FGD constructions. The filler token is processed in the early layers, with MLP blocks encoding its syntactic role, and this information is then transferred to the final (verb) token via attention in the middle layers, where the residual stream scores increase. In the later layers, MLP blocks further process this information to produce the final output. The sparsity and localization of contributing attention heads, together with this flow from filler to final token, are consistently observed across constructions.

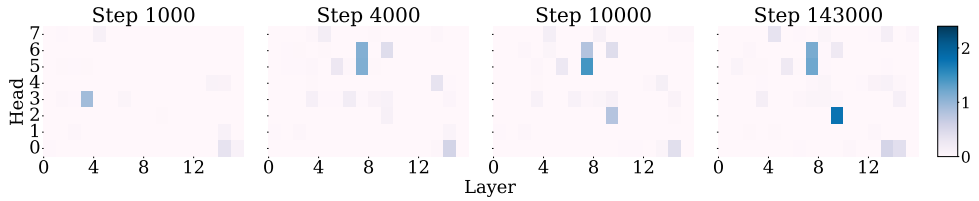
In contrast, a slightly different trend was observed in the constructions regarding NPI. The distribution of ODDS scores varies across constructions, indicating differences in the underlying mechanisms. For example, in DNEG, some contributing attention output are located in earlier layers, leading to earlier propagation of information and a corresponding increase in residual stream scores at earlier layers. In COND and SUP, the highest-scoring attention and MLP blocks at the final token differ across constructions. These variations suggest that, although similar types of components are involved, their functional roles differ across NPI constructions. A possible explanation is that NPI licensing requires integrating both syntactic and semantic information, which varies across constructions, and thus does not exhibit a single shared mechanism such as a monotonicity-based



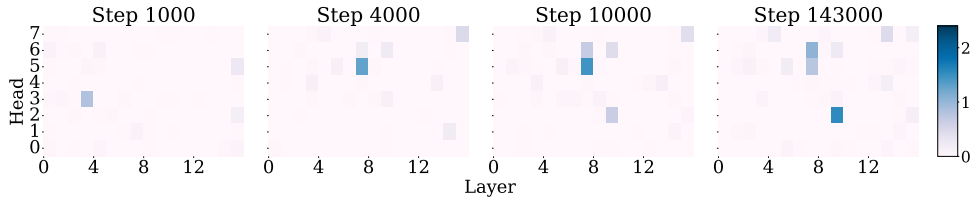
(a) EWHK, Residual stream



(b) PCLEFT, Residual stream



(c) EWHK, Attention head



(d) PCLEFT, Attention head

Figure 2: ODDS scores with activation patching of the residual stream in Pythia 1B with various training steps in filler-gap dependencies.

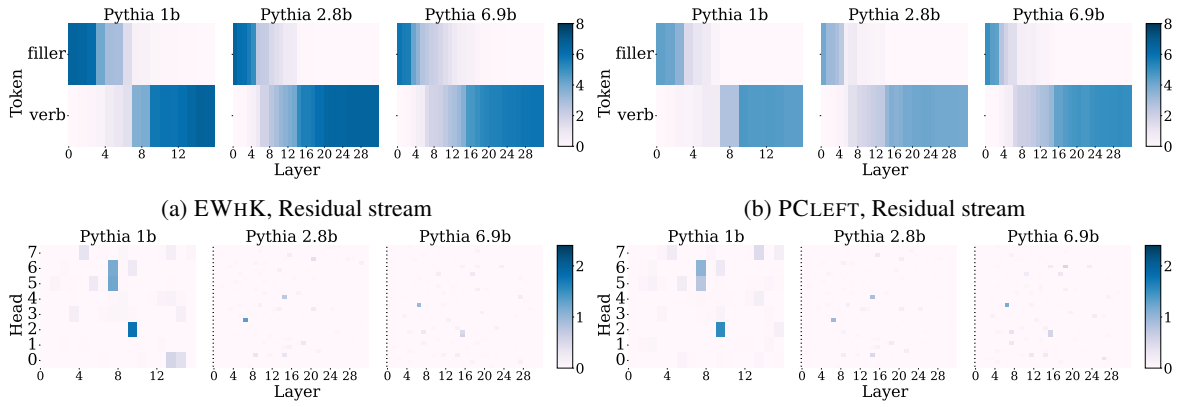
account (Jumelet et al., 2021) in decoder-based LMs.

4.4 Comparison with Different Training Steps

We next analyze the emergence of the mechanisms in FGDs during training, specifically focusing on the activation patching of the residual stream and attention heads in the context of embedded wh-questions EWHK and pseudo-clefts PCLEFT. According to Boguraev et al. (2025), embedded wh-questions occur approximately 15 times more frequently than pseudo-clefts in the English-EWT Universal Dependencies dataset (de Marneffe et al., 2021; Nivre et al., 2020; Silveira et al., 2014). We investigate whether this discrepancy in frequency impacts the training dynamics.

The results are illustrated in Figure 2. The ODDS scores in both constructions increase monotonically with the training steps. In particular, scores in the earlier layers improve more rapidly than those in the later layers, suggesting that the mechanism is established hierarchically, beginning with lower-level representations. However, the two constructions differ in their convergence. In the residual stream, the scores in EWHK reach levels comparable to the final training step by step 4,000, while the scores in PCLEFT at step 10,000 remain lower than their final values. A similar trend was observed in attention heads, although the improvement of the scores was quicker in both constructions compared to that in the residual stream.

These findings suggest that high-frequency constructions are learned during the initial stages of training, whereas less frequent constructions are acquired more gradually. Although both constructions eventually converge upon a shared mechanism, the development toward that mechanism depends on their frequency. This trend was consistently observed across other constructions and MLP components (see Appendix D.3 for details).



(a) EWHK, Residual stream

(b) PCLEFT, Residual stream

(c) EWHK, Attention head

(d) PCLEFT, Attention head

Figure 3: ODDS scores with activation patching of residual stream in Pythia models with various sizes in filler-gap dependencies.

4.5 Comparison with Different Parameter Sizes and Model Family

Next, we analyze how the mechanism varies with respect to the parameter size of the models. Figure 3 shows the results. It can be seen that the models with more layers process FGDs in earlier layers, while the mechanism is still shared and localized in each parameter size. Also, the distribution of the scores was similar between Pythia 2.8B and 6.9B. It indicates that the hidden dimension of the model, which is the only architectural difference between 2.8B and 6.9B, does not impact the processing mechanism of FGDs.

We also saw similar trends in Gemma 3 1B and 12B, which have 10 and 32 more layers than Pythia 1B respectively, where contributing attention heads are located in earlier layers, and the ODDS scores increase in earlier layers than Pythia 1B, while there is a shared mechanism across FGD constructions (see full results in Appendix D.4).

5 Steering Based on the Mechanism

5.1 Method

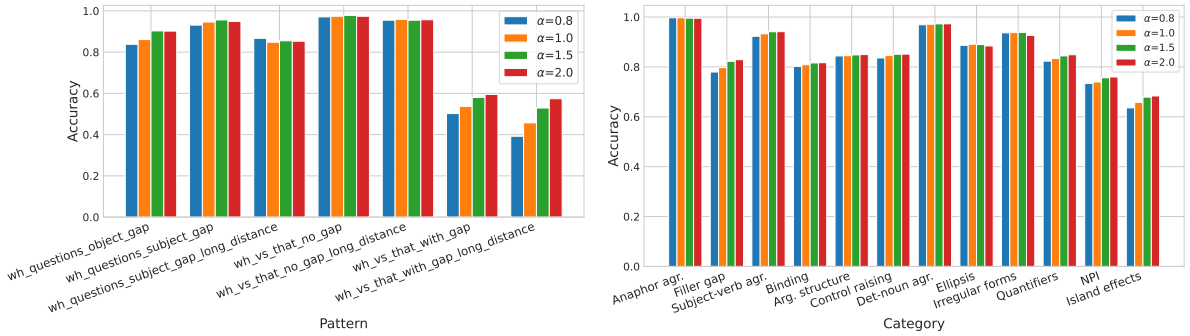
The previous sections revealed a shared mechanism underlying FGDs. However, it is possible that the analysis failed to distinguish heuristic patterns from the core syntactic phenomena if the dataset or methods were insufficiently robust. To address this, we investigate whether model behavior changes in an existing targeted syntactic evaluation benchmark when we manipulate only the specific components identified as contributors to the found mechanism. Specifically, we scale the activation values of the attention heads 7.5, 7.6, and 9.2 by a factor α and evaluate the resulting performance in BLiMP (Warstadt et al., 2020).

BLiMP assesses the extent to which an LM assigns a higher probability to a grammatical sentence compared to its ungrammatical counterpart within a minimal pair. The BLiMP task setting, which calculates probability over an entire sentence, allows us to verify that the identified mechanisms are not merely localized heuristics specific to individual tokens. It should be noted that we restrict our analysis to minimal pairs with identical tokenized lengths. This is necessary because different token lengths have been shown to skew the probability assignments, thereby hindering an accurate evaluation of the model's capability (Ueda et al., 2024).

5.2 Results

First, we focus on how the model performance changed in the categories involving FGDs. As illustrated in Figure 4a, amplifying the activation of the three attention heads with $\alpha > 1$ improved or maintained performance, the latter occurring in cases of score saturation. Moreover, for most unsaturated patterns, performance improved monotonically with increasing α from 0.8 to 1.5, although there was one exception pattern, where the score was higher with $\alpha = 0.8$ than with $\alpha = 1.0$. This result shows that the manipulated attention heads indeed contribute to assigning higher probabilities to grammatical sentences involving FGDs, not only in those with an object gap that we use for analysis, but also those with a subject gap, a prepositional object gap, or long-distance dependencies.

We next examine the model performance in other categories, as shown in Figure 4b. There were no-



(a) FILLER GAP category

(b) All categories

Figure 4: Accuracy in the category involving filler-gap dependencies (left) and in all categories (right) in BLiMP after multiplying activations of several attention heads in Pythia 1B by α.

ticeable improvements in categories such as island effects, binding, NPI, quantifiers, and center embedding. Also, the improvements in subject-verb agreement in BLiMP all came from the patterns involving agreement beyond the distractors with prepositional phrases or relative clauses. The results thus indicate that the attention heads manipulated here might serve a more general function in capturing hierarchical structural dependency rather than being specialized solely for FGDs.

We also tested with SyntaxGym (Hu et al., 2020), and the overall trends were similar (see the full results in Appendix E). In addition, to investigate performance gains in downstream tasks that require syntactic dependency processing, we conducted a steering experiment on HANS (McCoy et al., 2019), an NLI benchmark designed for analyzing syntactic heuristics. The results show that amplifying the activation values of the same attention heads leads to improved model performance (see details in Appendix F).

6 Comparison Between DAS and Activation Patching

6.1 Distributed Alignment Search

To assess the robustness of our findings, we compare activation patching with DAS, which is used by previous studies (Arora et al., 2024; Boguraev et al., 2025) for identifying causally efficacious components in syntactic tasks. DAS (Geiger et al., 2024) is a supervised causal localization method designed to identify subspaces that mediate specific task behavior. Following Arora et al. (2024) and Boguraev et al. (2025), we employ the one-dimensional variant of DAS, which learns a direction that causally impacts the output distribution. The intervention with DAS is defined by the following transformation:

$$f_{\text{interv}}(\boldsymbol{b}, \boldsymbol{s}) = f(\boldsymbol{b}) + (f(\boldsymbol{s})\boldsymbol{a} - f(\boldsymbol{b})\boldsymbol{a})\boldsymbol{a}^T,$$

where $\boldsymbol{a}$ is the vector to be learned.

During the training, the objective is to optimize $\boldsymbol{a}$ such that the intervention shifts the model's output toward the correct token associated with the source input. The loss function is defined as:

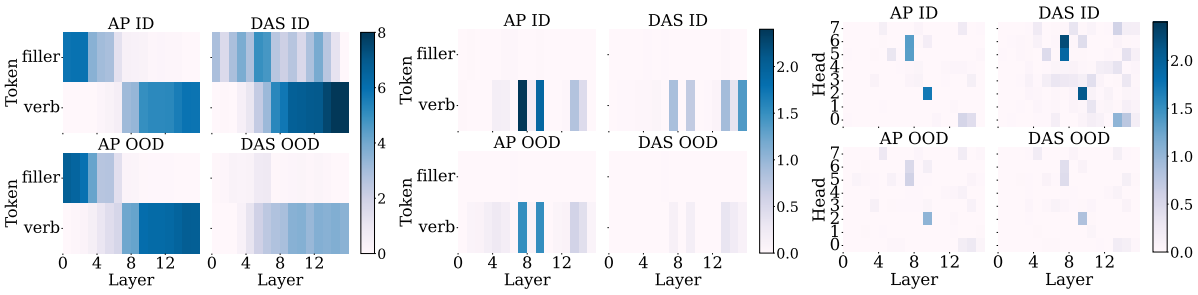
$$\min_{\boldsymbol{a}} \left\{ - \sum_{(\boldsymbol{b},\boldsymbol{s},y_b,y_s)\in D} \log p_{f_{\text{interv}}}(y_s|\boldsymbol{b},\boldsymbol{s}) \right\},$$

where D denotes the training set.

Following Boguraev et al. (2025), we apply DAS to analyze shared mechanisms through a leave-one-out training paradigm. In this setup, the direction is trained using a subset of the constructions $\{c_j \mid j \neq i, 0 \leq i, j \leq n\}$ and then evaluated on the remaining held-out construction c_i. This approach allows us to assess whether the learned direction for one set of constructions generalizes to another, thereby measuring the similarity of the mechanism across constructions.

6.2 Results

Figure 5 presents the comparative results between DAS and activation patching. First, we observe that DAS fails to generalize to the OOD test set, whereas training-free activation patching revealed a consistent distribution of scores between the ID and OOD test sets. This discrepancy suggests that the direction learned in DAS is likely overfit to the training set, capturing features specific to the dataset rather than the structural dependencies. Second, the scores obtained when learning a direction in attention output with DAS were lower than those



(a) Residual stream (b) Attention (c) Attention head

Figure 5: Comparison of the ODDS scores of Pythia 1B between activation patching (AP) and DAS, evaluated on the EWHK ID and OOD test sets.

with activation patching, even in the ID test set. This trend was more salient in the early to middle layers compared to the late ones. It significantly changes the interpretation of the mechanism, as otherwise, we would see more active contribution in the components of later layers in DAS. Interestingly, applying DAS to attention heads yields results more closely aligned with activation patching, despite lower ODDS scores in the OOD test set and a slight overestimation of causal efficacy in the later layers. This indicates that DAS may be more suitable for learning a direction in the representation space of individual attention heads.

These results suggest that interpretability methods requiring supervision necessitate rigorous OOD evaluation to ensure that the identified mechanism is functionally relevant to the task. Moreover, using DAS in datasets with relatively limited diversity may not always detect the most relevant mechanisms. The full comparison with DAS is provided in Appendix D.5, and the above trend was consistent in other constructions.

7 Conclusion

This paper investigated the internal mechanisms of LMs to determine whether they utilize shared mechanisms across different syntactic constructions. We identified the existence of a shared mechanism for FGDs, primarily mediated by a small number of attention heads located in the early to middle layers. In contrast, NPI licensing appears to rely on construction-specific mechanisms, reflecting its complex syntactic and semantic requirements. We also validated the causal efficacy of the identified components through steering experiments, where activation scaling consistently improved model performance on BLiMP. Furthermore, we compared activation patching and DAS, which revealed poor OOD generalization with DAS. These insights not only advance our understanding of how LMs process language but also can provide a basis for developing methods to improve the syntactic capability of LMs.

Limitations

First, our study is restricted to English, a limitation shared by much of the current work on mechanistic interpretability. It remains unclear whether our findings generalize to other languages with different linguistic features. For example, factors such as word order may require different processing strategies. Moreover, as the frequency of each language in the pretraining corpus likely impacts the internal mechanisms of language models, models may employ different mechanisms for low-resource languages compared to English. We leave these questions to future work.

Second, our dataset used for our main analysis is synthetically generated rather than sampled from natural corpora, following previous studies (Arora et al., 2024; Boguraev et al., 2025). Although this approach allows for the precise control of minimal pairs, it limits the diversity of language expressions observed in the wild. However, we made efforts to diversify the dataset compared to the previous studies (Arora et al., 2024; Boguraev et al., 2025), and conducted experiments by using a small set of naturally occurring sentences.

Ethical Considerations

Our study investigates the internal workings of language models for a deeper understanding of their language processing and ultimately utilizes these insights to improve model capabilities and reliability. However, we acknowledge that such insights could potentially be misused to modify the behav-

ior of models in unsafe or harmful ways. We warn model developers and users against the malicious application of these techniques.

The dataset does not contain any information that names or uniquely identifies individual people or offensive content, as it is generated with our templates.

Acknowledgments

We appreciate the anonymous reviewers on ACL Rolling Review, who helped refine this paper from various perspectives. We also thank Adam Nohejl, Taisei Yamamoto, Amane Watahiki, Tomoki Doi, Daiki Matsuoka, Gaëtan Margueritte, and Koki Ryu for their helpful comments on this paper. This work was supported by JST CREST Grant Number JPMJCR2565 and JST BOOST Program Grant Number JPMJBY24H5, Japan.

References

Aryaman Arora, Dan Jurafsky, and Christopher Potts. 2024. CausalGym: Benchmarking causal interpretability methods on linguistic tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14638–14663, Bangkok, Thailand. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Sasha Boguraev, Christopher Potts, and Kyle Mahowald. 2025. Causal interventions reveal shared structure across English filler–gap constructions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25032–25053, Suzhou, China. Association for Computational Linguistics.

Chi-Yun Chang, Xueyang Huang, Humaira Nasir, Shane Storks, Olawale Akingbade, and Huteng Dai. 2025. Mind the gap: How BabyLMs learn filler-gap dependencies. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15060–15076, Suzhou, China. Association for Computational Linguistics.

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.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Deanna DeCarlo, William Palmer, Michael Wilson, and Bob Frank. 2023. NPIs aren't exactly easy: Variation in licensing across large language models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 332–341, Singapore. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843, Online. Association for Computational Linguistics.

Jaden Fiotto-Kaufman, Alexander R Loftus, Eric Todd, Jannik Brinkmann, Caden Juang, Koyena Pal, Can Rager, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Michael Ripa, Adam Belfki, Nikhil Prakash, Sumeet Multani, Carla Brodley, Arjun Guha, Jonathan Bell, Byron Wallace, and David Bau. 2024. Nnsight and ndif: Democratizing access to foundation model internals.

Richard Futrell and Kyle Mahowald. 2025. How linguistics learned to stop worrying and love the language models. *Behavioral and Brain Sciences*, page 1–98.

Atticus Geiger, Hanson Lu, Thomas F Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. 2024. Finding alignments between interpretable causal variables and distributed neural representations. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pages 160–187. PMLR.

Gemma Team. 2025. Gemma 3.

Anastasia Giannakidou. 1998. Polarity sensitivity as (non) veridical dependency.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Katherine Howitt, Sathvik Nair, Allison Dods, and Robert Melvin Hopkins. 2024. Generalizations across filler-gap dependencies in neural language models. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 269–279, Miami, FL, USA. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Jaap Jumelet, Milica Denic, Jakub Szymanik, Dieuwke Hupkes, and Shane Steinert-Threlkeld. 2021. Language models use monotonicity to assess NPI licensing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4958–4969, Online. Association for Computational Linguistics.

Daria Kryvosheieva, Andrea de Varda, Evelina Fedorenko, and Greta Tuckute. 2025. Different types of syntactic agreement recruit the same units within large language models. *Preprint*, arXiv:2512.03676.

William Allen Ladusaw. 1979. *Polarity sensitivity as inherent scope relations.* The University of Texas at Austin.

Nur Lan, Emmanuel Chemla, and Roni Katzir. 2024. Large language models and the argument from the poverty of the stimulus. *Linguistic Inquiry*, pages 1–28.

Karim Lasri, Tiago Pimentel, Alessandro Lenci, Thierry Poibeau, and Ryan Cotterell. 2022. Probing for the usage of grammatical number. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8818–8831, Dublin, Ireland. Association for Computational Linguistics.

Lovish Madaan, David Esiobu, Pontus Stenetorp, Barbara Plank, and Dieuwke Hupkes. 2025. Lost in inference: Rediscovering the role of natural language inference for large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9229–9242, Albuquerque, New Mexico. Association for Computational Linguistics.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Satoru Ozaki, Dan Yurovsky, and Lori Levin. 2022. How well do LSTM language models learn filler-gap dependencies? In *Proceedings of the Society for Computation in Linguistics 2022*, pages 76–88, online. Association for Computational Linguistics.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Naoya Ueda, Masato Mita, Teruaki Oka, and Mamoru Komachi. 2024. Token-length bias in minimal-pair paradigm datasets. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16224–16236, Torino, Italia. ELRA and ICCL.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019. Investigating BERT's knowledge of language: Five analysis methods with NPIs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the*

Association for Computational Linguistics, 8:377–392.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Ethan Gotlieb Wilcox, Richard Futrell, and Roger Levy. 2024. Using computational models to test syntactic learnability. *Linguistic Inquiry*, 55(4):805–848.

Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah Goodman, Christopher Manning, and Christopher Potts. 2024a. pyvene: A library for understanding and improving PyTorch models via interventions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 158–165, Mexico City, Mexico. Association for Computational Linguistics.

Zhengxuan Wu, Atticus Geiger, Jing Huang, Aryaman Arora, Thomas Icard, Christopher Potts, and Noah D. Goodman. 2024b. A reply to makelov et al. (2023)'s "interpretability illusion" arguments. *Preprint*, arXiv:2401.12631.

Frans Zwarts. 1998. Three types of polarity. In *Plurality and quantification*, pages 177–238. Springer.

A Details of the Dataset

We adopt the dataset sizes from the setting used by Arora et al. (2024). For each construction pattern, the training set consists of 200 examples, and each of the ID and OOD tests contains 50 examples. To maintain a strict balance between labels, we swap the base and source inputs and labels to double the size. Thus, our combined analysis across the ID and OOD test sets utilizes 200 examples in total, which is a larger sample size than the test sets used in related work (Arora et al., 2024; Boguraev et al., 2025).

The sentences are generated using a vocabulary set of 4,275 words, which was originally created by Warstadt et al. (2020). To properly test generalization, we create two templates for each construction, one for the training and ID test sets and another for the OOD test set. This ensures that the model is evaluated on a wide range of lexical and structural variations.

B Hyperparameters

In the training of DAS, we use the hyperparameters that Arora et al. (2024) used based on tuning. We set learning rate as 5×10^{-3} with linear warmup until 10% of the training steps. We set the batch size to 4 and train for 100 steps. We also experimented with other learning rates ($5 \times 10^{-2}, 5 \times 10^{-4}$) and training steps ($50, 200$) and observed largely consistent trends across settings.

C Details of the ODDS score

In this section, we demonstrate that for FGDs, our ODDS score is mathematically equivalent to the one used by related work (Arora et al., 2024; Boguraev et al., 2025), which is defined as follows:

$$\begin{aligned}&\textsc{Odds}^*(p, p_{f_{\text{interv}}}, \langle \boldsymbol{b}, \boldsymbol{s}, y_b, y_s \rangle) \\ &= \log \left(\frac{p(y_b|\boldsymbol{b})}{p(y_s|\boldsymbol{b})} \frac{p_{f_{\text{interv}}}(y_s|\boldsymbol{b}, \boldsymbol{s})}{p_{f_{\text{interv}}}(y_b|\boldsymbol{b}, \boldsymbol{s})} \right)\end{aligned}$$

As described in Appendix A, our dataset includes both the original minimal pair $(\boldsymbol{b}, \boldsymbol{s}, y_b, y_s)$ and its flipped counterpart $(\boldsymbol{b}, \boldsymbol{s}, y_b, y_s)$ and $(\boldsymbol{s}, \boldsymbol{b}, y_s, y_b)$. Summing the ODDS scores for these two samples yields the following derivation:

$$\begin{aligned}&\textsc{Odds}(p, p_{f_{\text{interv}}}, \langle \boldsymbol{b}, \boldsymbol{s}, y_b, y_s \rangle) \\ &\quad + \textsc{Odds}(p, p_{f_{\text{interv}}}, \langle \boldsymbol{s}, \boldsymbol{b}, y_s, y_b \rangle) \\ &= \log \left(\frac{p(y_b|\boldsymbol{b})}{p(y_b|\boldsymbol{s})} \frac{p_{f_{\text{interv}}}(y_b|\boldsymbol{s}, \boldsymbol{b})}{p_{f_{\text{interv}}}(y_b|\boldsymbol{b}, \boldsymbol{s})} \right) \\ &\quad + \log \left(\frac{p(y_s|\boldsymbol{s})}{p(y_s|\boldsymbol{b})} \frac{p_{f_{\text{interv}}}(y_s|\boldsymbol{b}, \boldsymbol{s})}{p_{f_{\text{interv}}}(y_s|\boldsymbol{s}, \boldsymbol{b})} \right) \\ &= \log \left(\frac{p(y_b|\boldsymbol{b})}{p(y_s|\boldsymbol{b})} \frac{p_{f_{\text{interv}}}(y_s|\boldsymbol{b}, \boldsymbol{s})}{p_{f_{\text{interv}}}(y_b|\boldsymbol{b}, \boldsymbol{s})} \right) \\ &\quad + \log \left(\frac{p(y_s|\boldsymbol{s})}{p(y_b|\boldsymbol{s})} \frac{p_{f_{\text{interv}}}(y_b|\boldsymbol{s}, \boldsymbol{b})}{p_{f_{\text{interv}}}(y_s|\boldsymbol{s}, \boldsymbol{b})} \right) \\ &= \textsc{Odds}^*(p, p_{f_{\text{interv}}}, \langle \boldsymbol{b}, \boldsymbol{s}, y_b, y_s \rangle) \\ &\quad + \textsc{Odds}^*(p, p_{f_{\text{interv}}}, \langle \boldsymbol{s}, \boldsymbol{b}, y_s, y_b \rangle).\end{aligned}$$

Therefore, our ODDS score is equivalent to the one defined in previous studies when we use symmetric pairs. It should be noted that because we use only one of the two for NPI licensing, as mentioned in Section 3.1, the ODDS score in NPI licensing differs from the previous one.

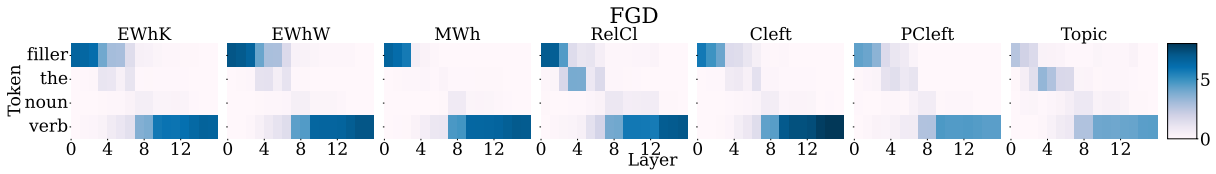
D Detailed Activation Patching Results

D.1 All Constructions in FGDs

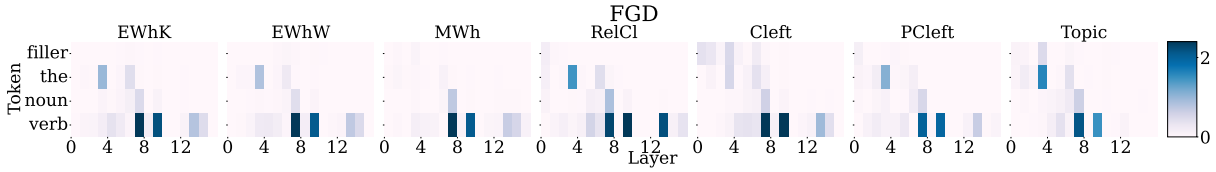
The results are shown in Figure 6.

D.2 All Constructions in NPI Licensing

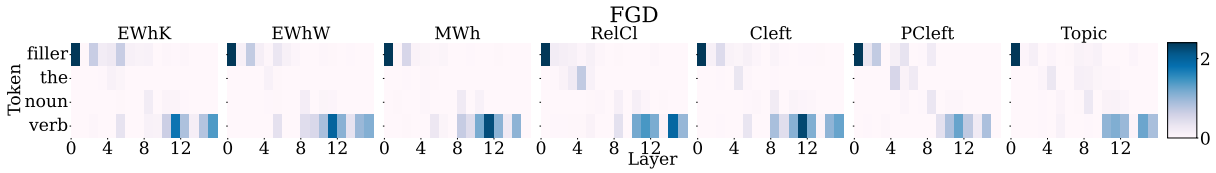
The results are shown in Figure 7.



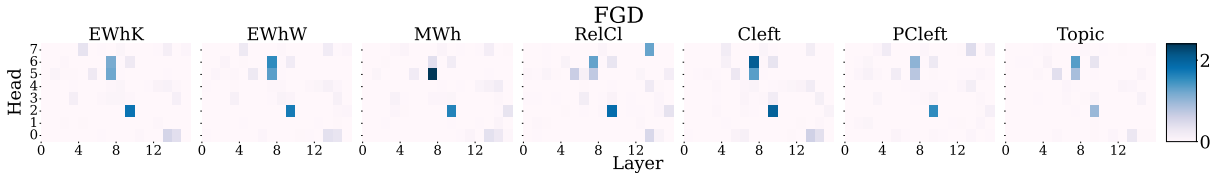
(a) Residual stream



(b) Attention

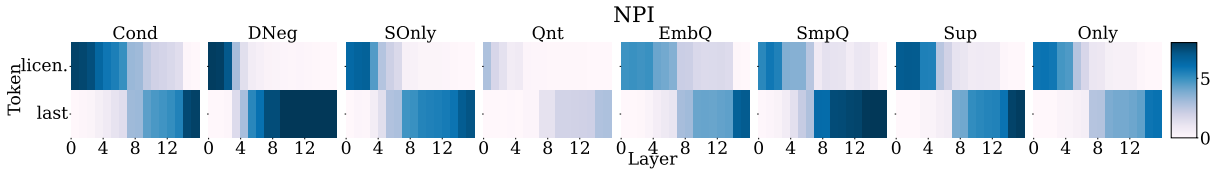


(c) MLP

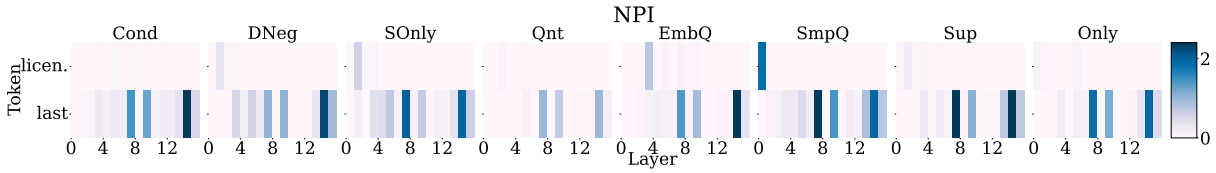


(d) Attention head

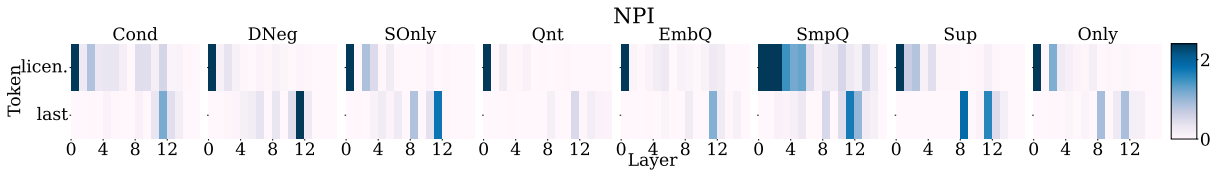
Figure 6: ODDS scores with activation patching of Pythia 1B in all the constructions in FGDs.



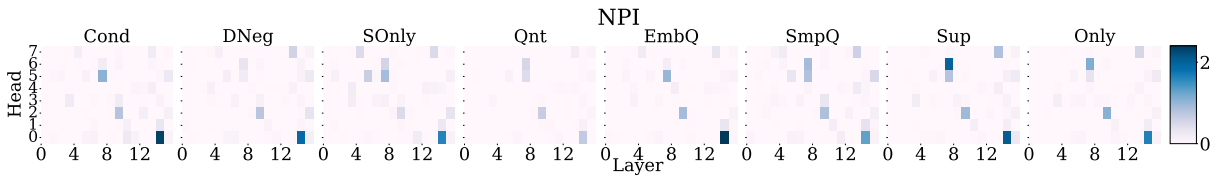
(a) Residual Stream



(b) Attention

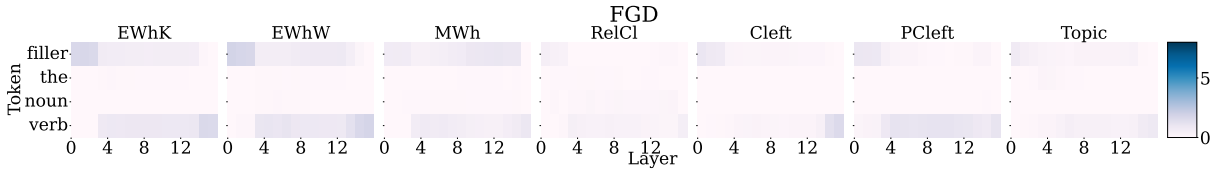


(c) MLP

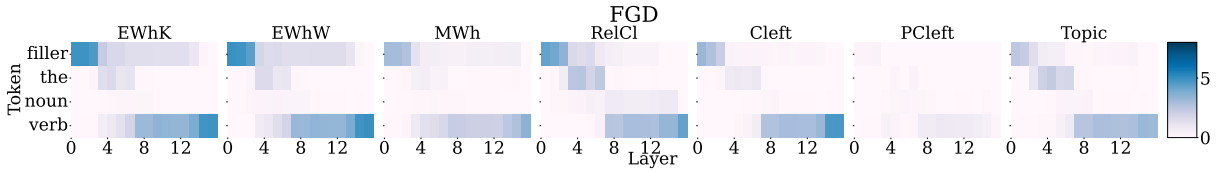


(d) Attention Heads

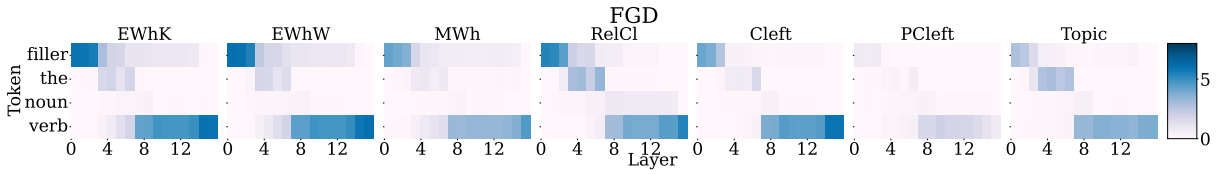
Figure 7: ODDS scores with activation patching of Pythia 1B in all the constructions of NPI licensing.



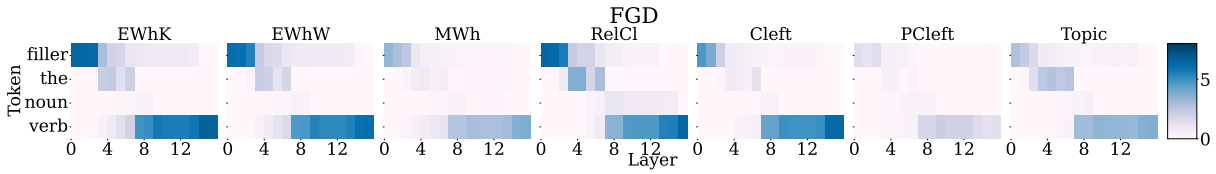
(a) Training Steps: 1000



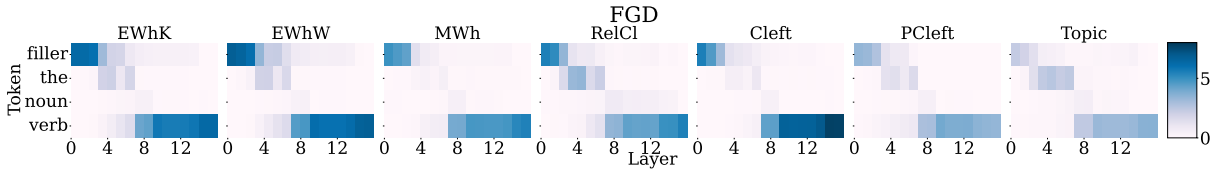
(b) Training Steps: 2000



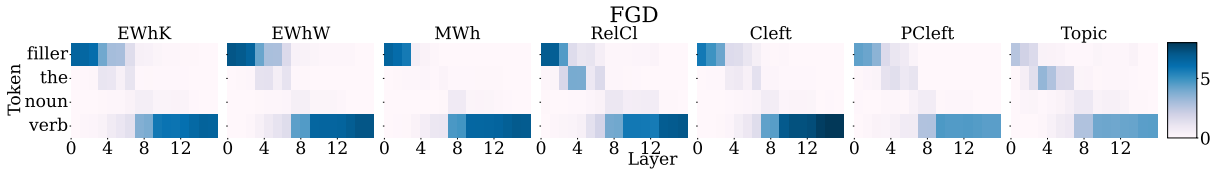
(c) Training Steps: 3000



(d) Training Steps: 4000

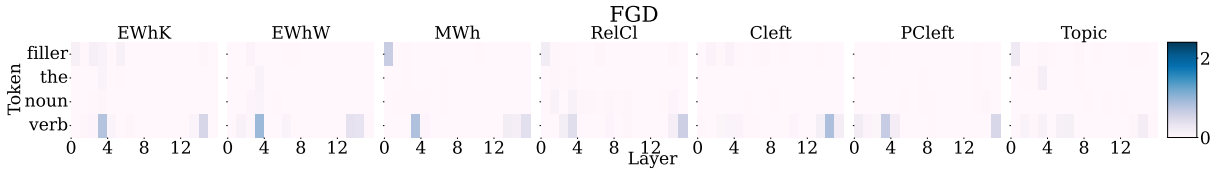


(e) Training Steps: 10000

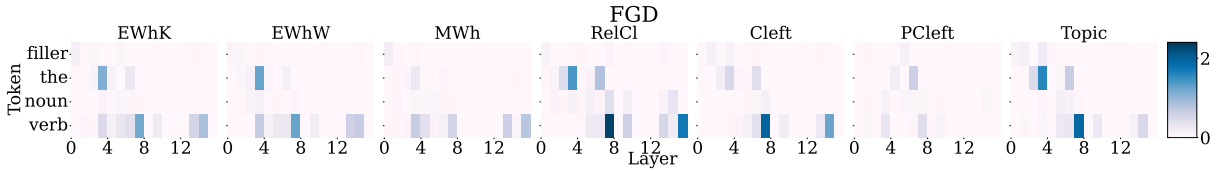


(f) Training Steps: 143000 (Final)

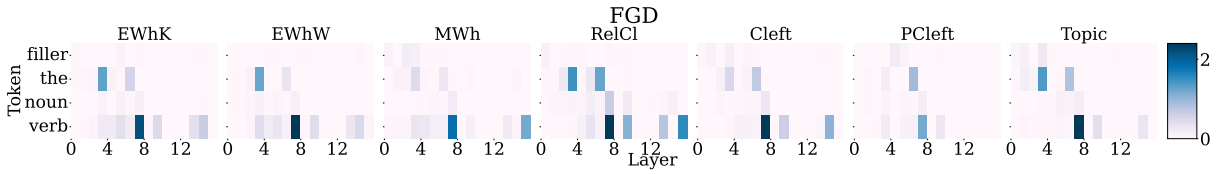
Figure 8: ODDS scores with activation patching of residual stream of models with various training steps in filler-gap dependencies.



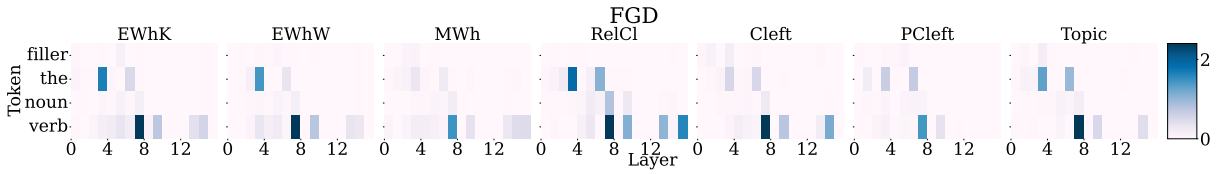
(a) Training Steps: 1000



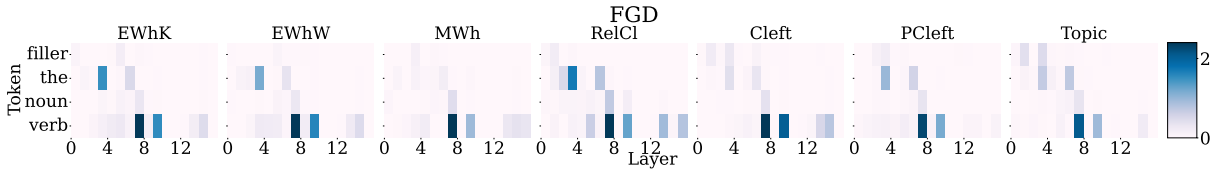
(b) Training Steps: 2000



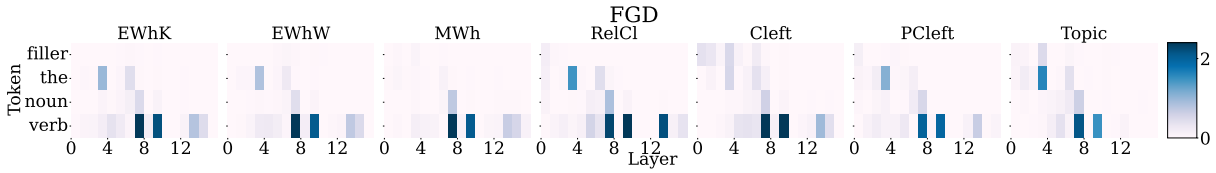
(c) Training Steps: 3000



(d) Training Steps: 4000

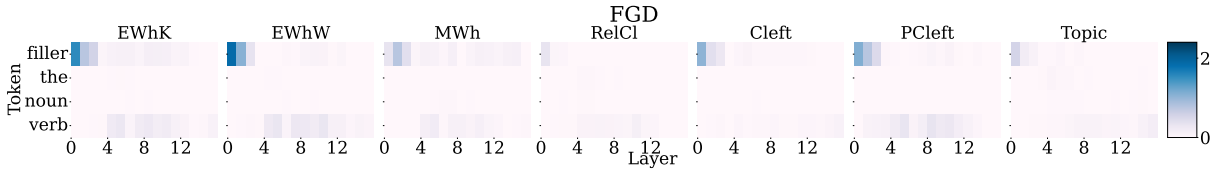


(e) Training Steps: 10000

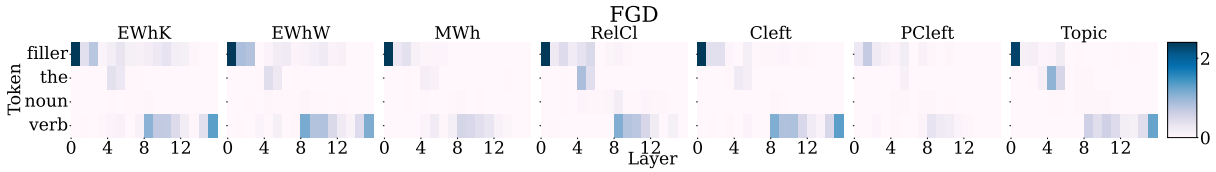


(f) Training Steps: 143000 (Final)

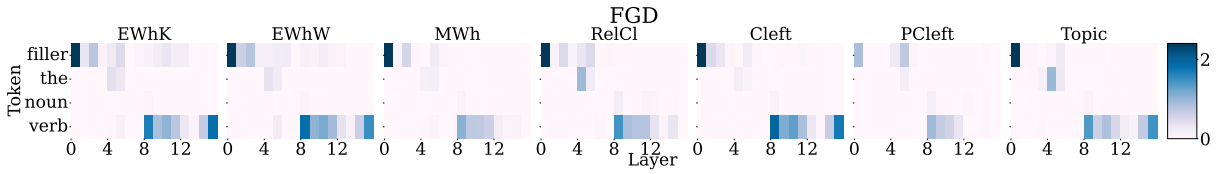
Figure 9: ODDS scores with activation patching of attention output of models with various training steps in filler-gap dependencies.



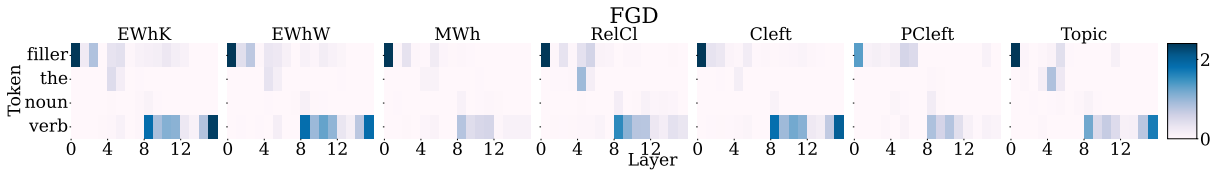
(a) Training Steps: 1000



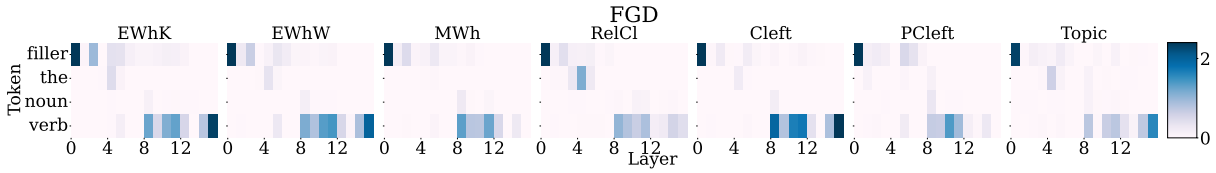
(b) Training Steps: 2000



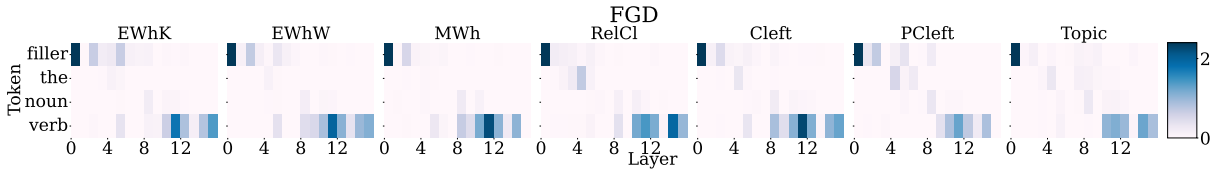
(c) Training Steps: 3000



(d) Training Steps: 4000

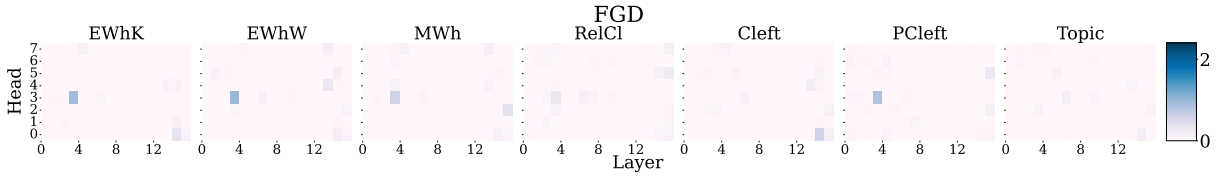


(e) Training Steps: 10000

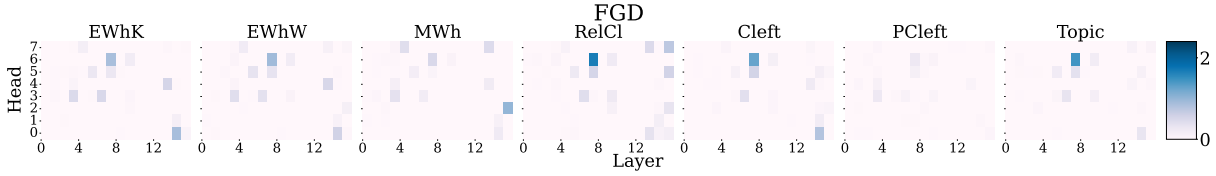


(f) Training Steps: 143000 (Final)

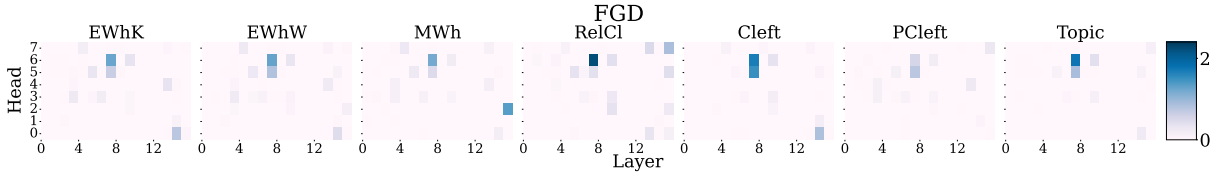
Figure 10: ODDS scores with activation patching of MLP output of models with various training steps in filler-gap dependencies.



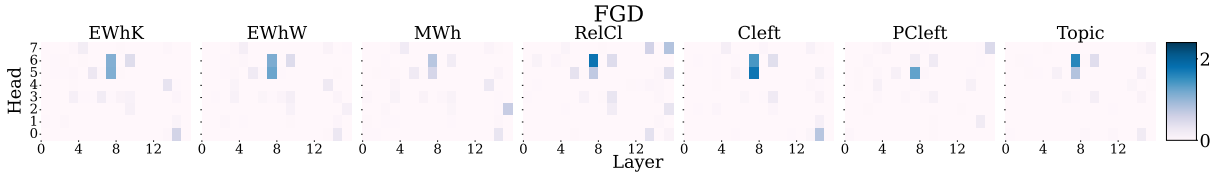
(a) Training Steps: 1000



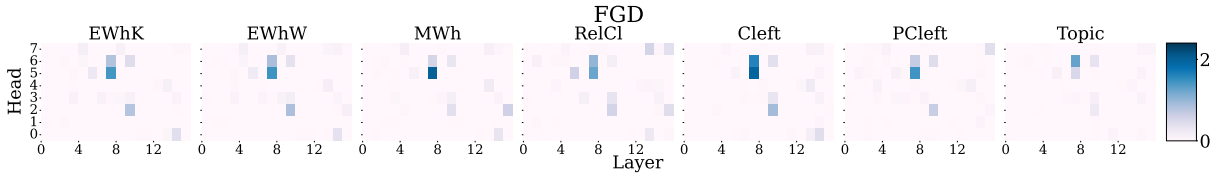
(b) Training Steps: 2000



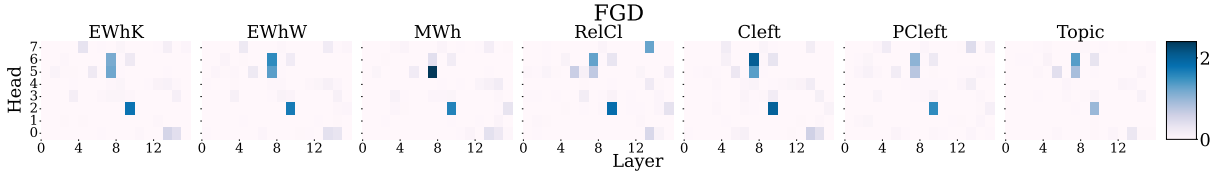
(c) Training Steps: 3000



(d) Training Steps: 4000



(e) Training Steps: 10000



(f) Training Steps: 143000 (Final)

Figure 11: ODDS scores with activation patching of attention output of models with various training steps in filler-gap dependencies.

D.3 Training Dynamics of All Constructions in FGDs

We present the results of activation patching applied to each component of Pythia 1B across several training steps. Figure 8 through 11 show the results of residual stream, attention output, MLP output, and attention heads.

D.4 Gemma 3

We apply activation patching to Gemma 3 1B and 12B. The results are shown in Figure 12 and 13. Similar to Pythia models, a shared mechanism can be seen among FGD constructions in Gemma 3, although the scores were higher in RELCL than in other constructions.

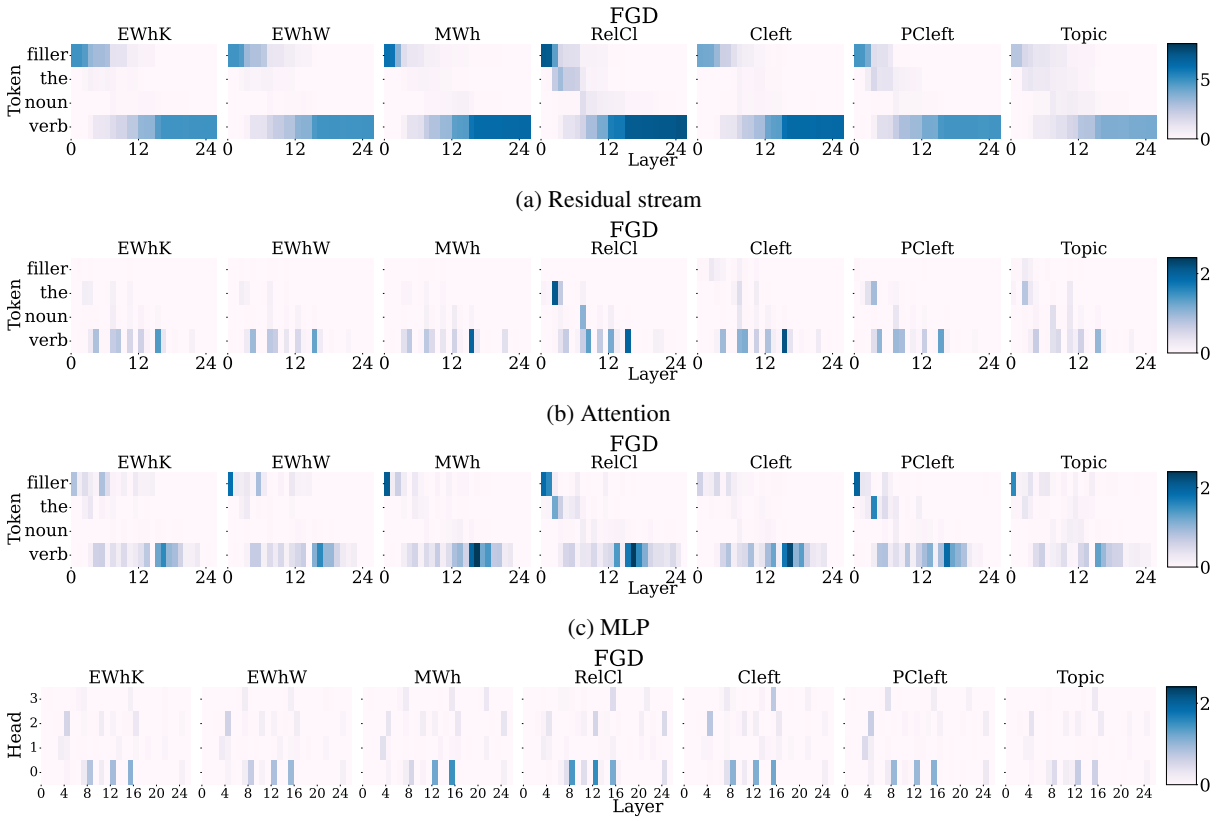
D.5 Comparison of Activation Patching and DAS in All Constructions in FGDs

We present the results of comparisons between activation patching and DAS applied to Pythia 1B. Figure 14 through 17 show the results of residual stream, attention output, MLP output, and attention heads.

D.6 Naturally Ocurring Data

To investigate the mechanism of language models in processing natural sentences, we extract 20 naturally occurring sentences containing wh-embedding structures from the English-EWT Universal Dependencies dataset (de Marneffe et al., 2021; Nivre et al., 2020; Silveira et al., 2014) and performed activation patching on Pythia-1B.

The results, shown in Figure 18, demonstrate the exact same trends observed with our artificially constructed filler-gap structures, such as EWHK and EWHW. Specifically, attention heads 7.5, 7.6, and 9.2 exhibited high scores, and the distribution across the residual stream remained identical. These findings confirm that our analysis generalizes to naturally occurring sentences.



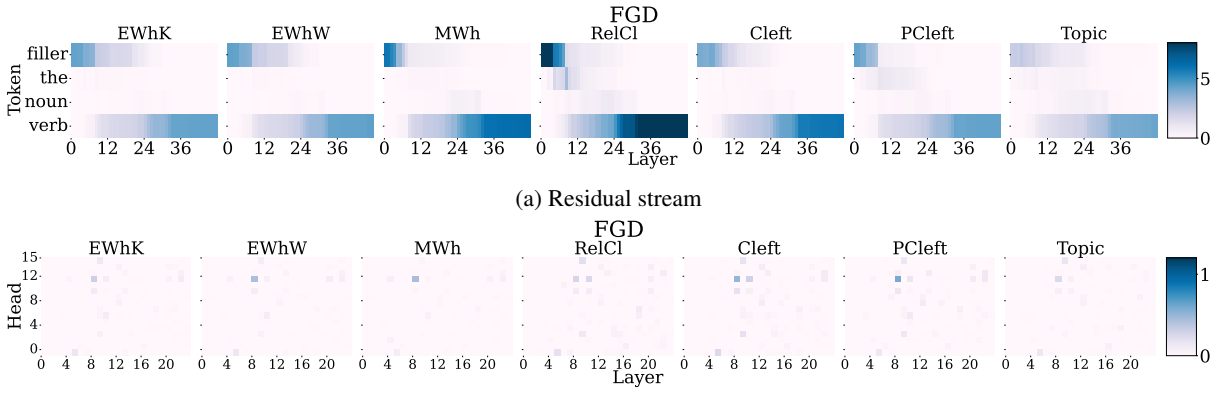
(a) Residual stream

(b) Attention

(c) MLP

(d) Attention head

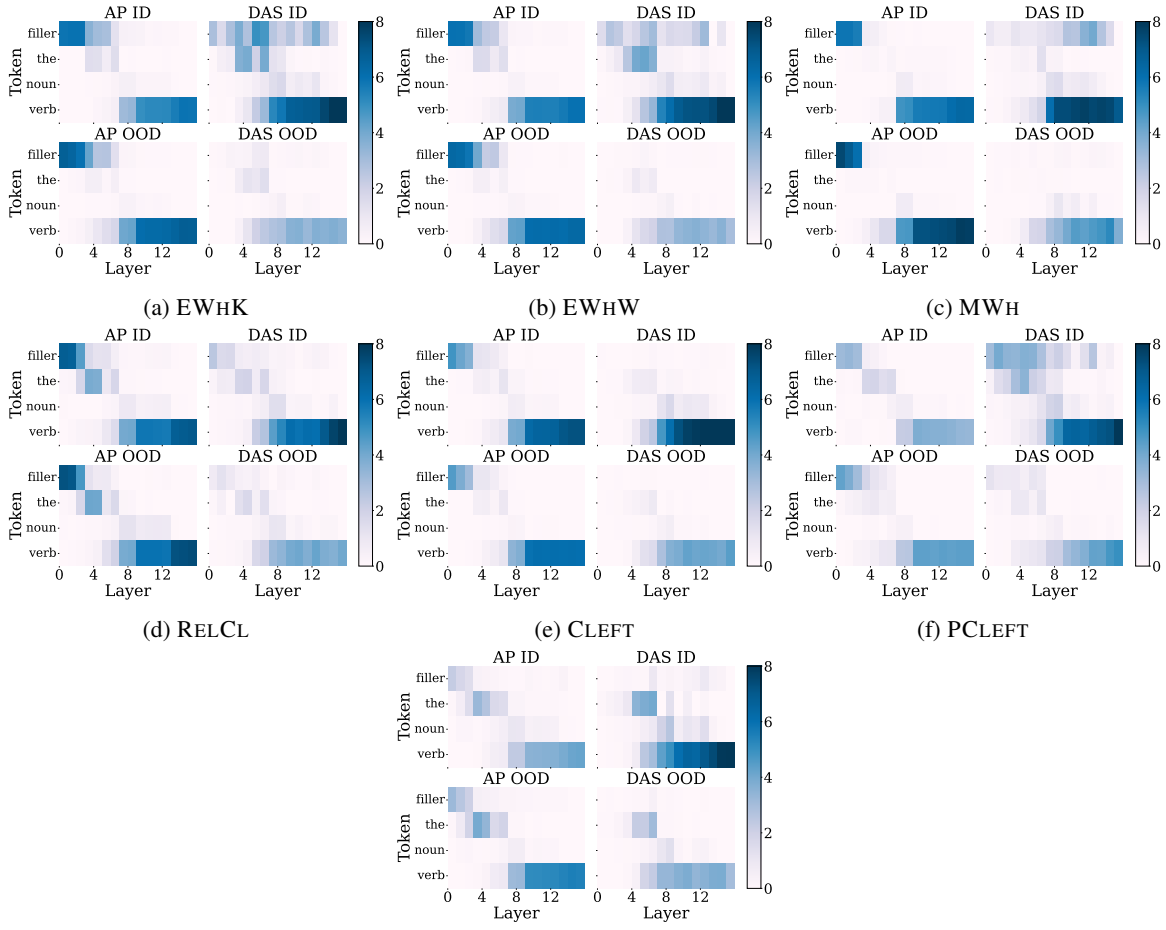
Figure 12: ODDS scores with activation patching in Gemma 3 1B in FGDs.



(a) Residual stream

(b) Attention head

Figure 13: ODDS scores with activation patching in Gemma 3 12B in FGDs. The results of attention head are limited to layers 0-24 for clarity.

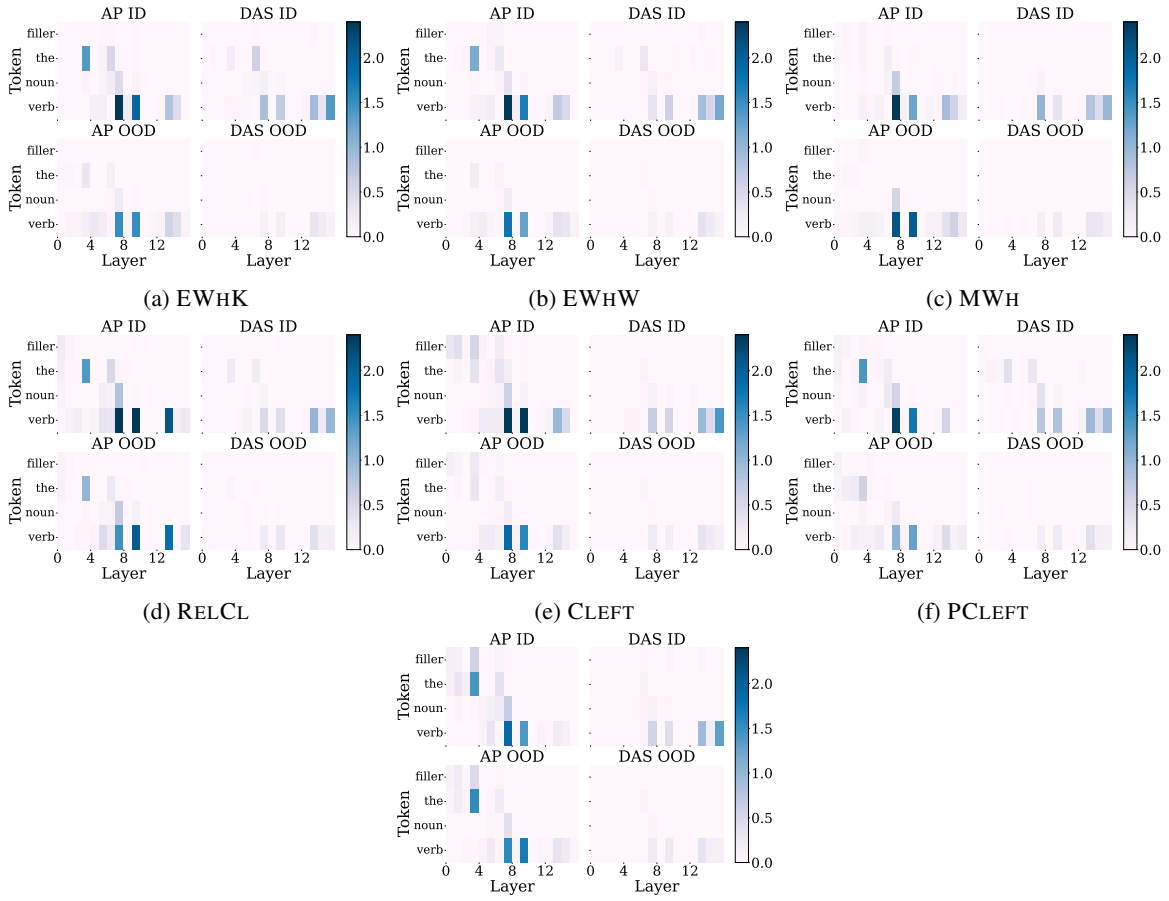


(a) EWHK (b) EWHW (c) MWH

(d) RELCL (e) CLEFT (f) PCLEFT

(g) TOPIC

Figure 14: Comparison of ODDS scores between activation patching (AP) and DAS of residual stream in Pythia 1B, evaluated on all the constructions in FGDs.

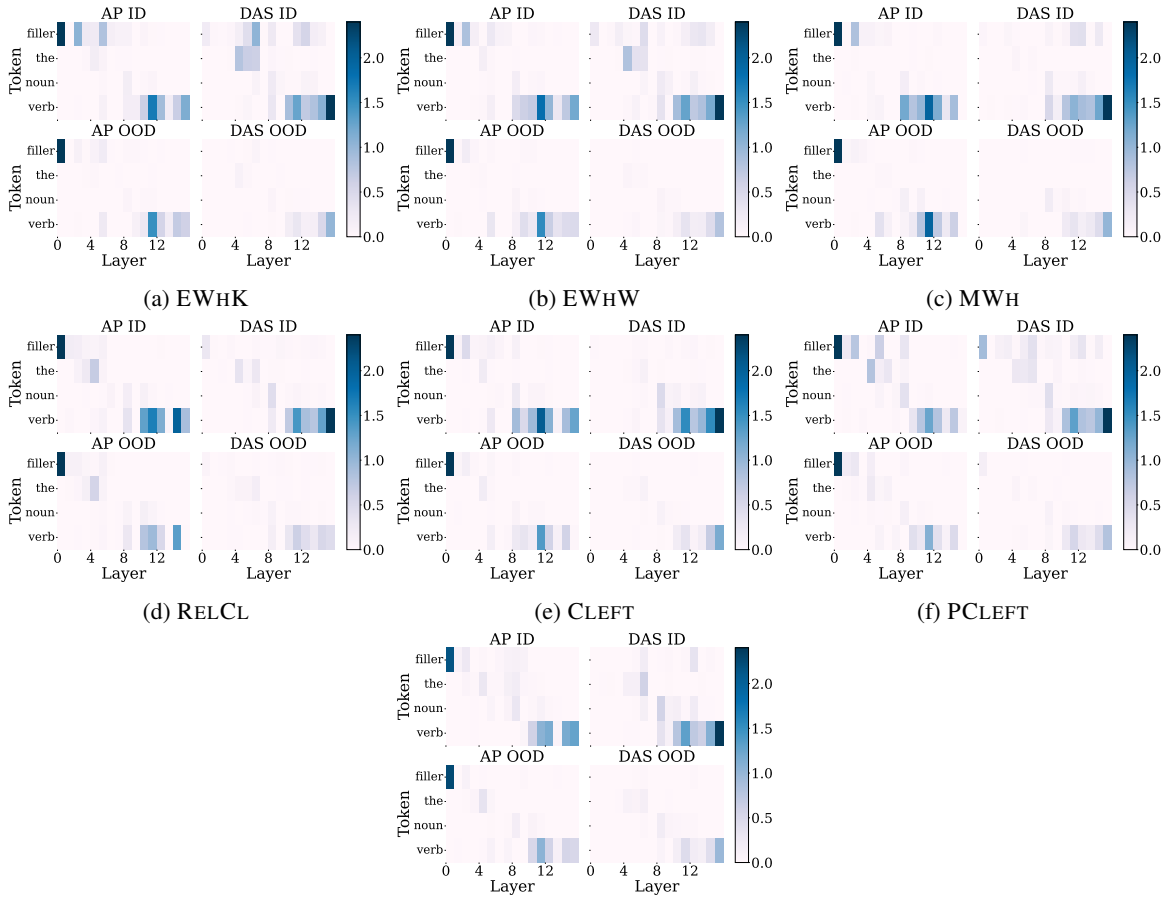


(a) EWHK (b) EWHW (c) MWH

(d) RELCL (e) CLEFT (f) PCLEFT

(g) TOPIC

Figure 15: Comparison of ODDS scores between activation patching (AP) and DAS of attention output in Pythia 1B, evaluated on all the constructions in FGDs.

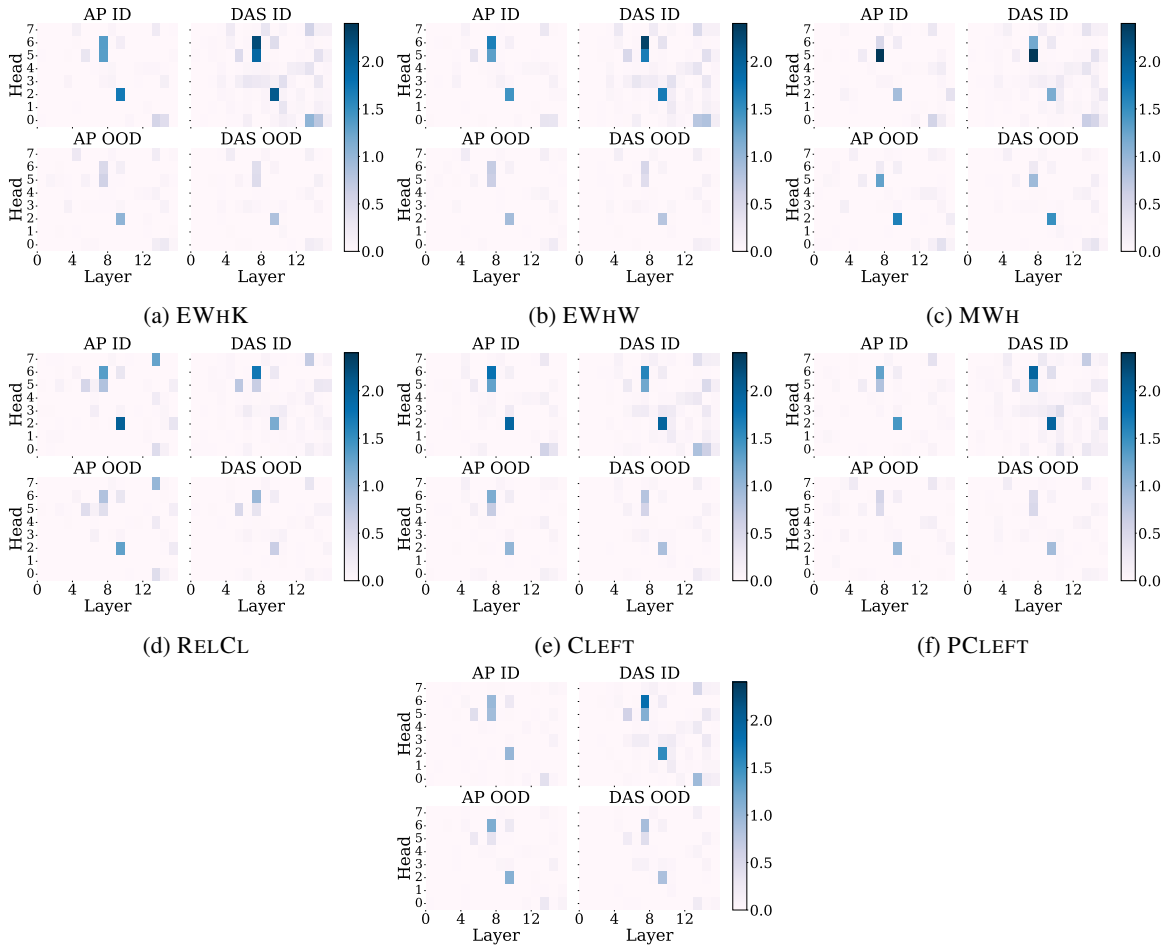


(a) EWHK (b) EWHW (c) MWH

(d) RELCL (e) CLEFT (f) PCLEFT

(g) TOPIC

Figure 16: Comparison of ODDS scores between activation patching (AP) and DAS of MLP output in Pythia 1B, evaluated on all the constructions in FGDs.

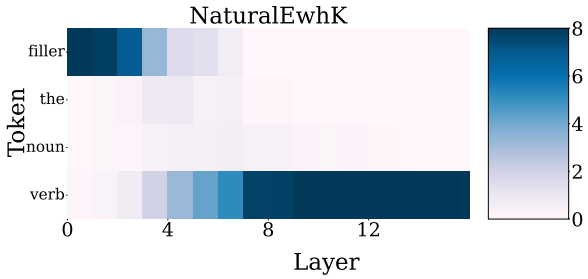


(a) EWHK (b) EWHW (c) MWH

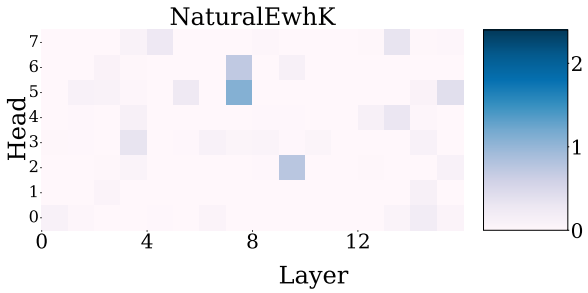
(d) RELCL (e) CLEFT (f) PCLEFT

(g) TOPIC

Figure 17: Comparison of ODDS scores between activation patching (AP) and DAS of attention heads in Pythia 1B, evaluated on all the constructions in FGDs.



(a) Residual Stream



(b) Attention head

Figure 18: ODDS scores with activation patching in filler-gap dependencies in naturally occuring sentences.

E Steering Results on SyntaxGym

We extend the steering experiments described in Section 5 to SyntaxGym (Hu et al., 2020). SyntaxGym differs from BLiMP in that it calculates the probability of only regions that differ among sentences in comparison, instead of using the probability of the entire sentence. For example, in a subject-verb agreement pattern where the distractor is a prepositional phrase:

(1) The author next to the senators is$_{V_{\text{sg}}}$ good.

(2) * The author next to the senators are$_{V_{\text{pl}}}$ good.

(3) The authors next to the senator are$_{V_{\text{pl}}}$ good.

(4) * The authors next to the senator is$_{V_{\text{sg}}}$ good.

The evaluation criteria are defined by the raw output probabilities $P(\cdot)$ at the verb position:

$$P_1(V_{\text{sg}}) > P_2(V_{\text{pl}}) \wedge P_3(V_{\text{pl}}) > P_4(V_{\text{sg}})$$

However, we argue that certain SyntaxGym patterns for FGDs are unsuitable for evaluating the syntactic competence of causal language models. Consider the subject extraction pattern:

(5) You remember that $\overbrace{\text{the businessman}}^{\alpha}$ showed the presentation to the visitors after lunch.

(6) * You remember who $\overbrace{\text{the businessman}}^{\alpha}$ showed the presentation to the visitors after lunch.

(7) You remember who $\overbrace{\text{showed}}^{\beta}$ the presentation to the visitors after lunch.

(8) * You remember that $\overbrace{\text{showed}}^{\beta}$ the presentation to the visitors after lunch.

SyntaxGym use surprisals $S(\cdot)$ for evaluation:

$$S_6(\alpha) > S_5(\alpha) \wedge S_8(\beta) > S_7(\beta)$$

Because causal language models process text incrementally, none of the strings above leading up to α or β are ungrammatical, and their grammaticality depends on the following tokens. Thus, comparing surprisals at these positions may not faithfully reflect the model's grammatical judgment.

Moreover, SyntaxGym has a considerably smaller sample size, around 20 to 40 per pattern, compared to the 1,000 samples per pattern in BLiMP. This limits the sensitivity to capture subtle improvements in language models' syntactic capability.

Despite these caveats, we report results on SyntaxGym when the attention heads 7.5, 7.6, and 9.2 are manipulated. The results are shown in Figure 19. The overall trends were similar to those of BLiMP, as $\alpha > 1$ led to improvement of the scores in many of the patterns and categories. In the patterns where the scores did not improve, such as the "fgd_subject" pattern, the results may have reflected the limitations of the evaluation criteria mentioned above rather than a failure of the manipulated mechanism.

F Steering Results on HANS

α	Accuracy (%)
0.8	51.6
1.0	52.3
1.2	53.5
1.5	54.9

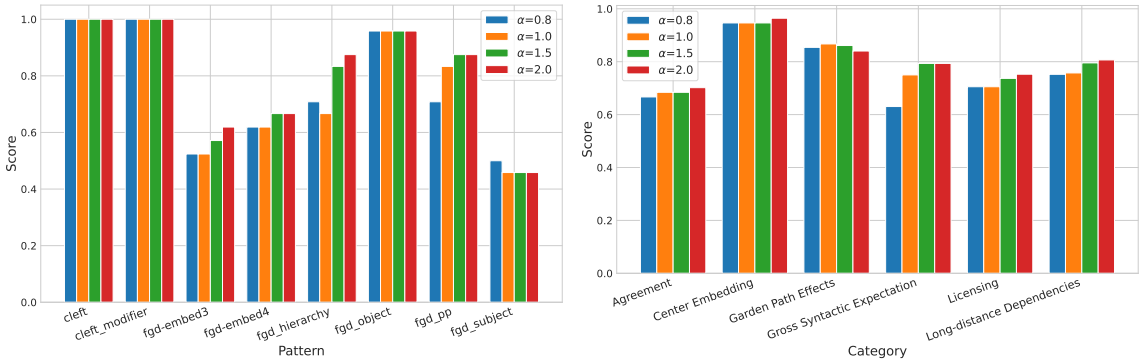
Table 2: Accuracy in HANS after multiplying activations of several attention heads in Gemma 3 1B IT by α.

Our findings suggest potential performance gains in downstream tasks that rely on syntactic dependency processing. To investigate this, we conducted additional steering experiments using HANS (McCoy et al., 2019), an NLI benchmark. We multiply the activation values of the attention heads that contribute to the processing of filler-gap dependencies by a factor of α and observe the resulting changes in NLI performance. We follow Madaan et al. (2025) for prompt and few-shot examples for the evaluation with HANS, and we use Gemma 3 1B IT as the model.

The results of the performance are shown in Table 2. The results show that amplifying the activation values of identified attention heads leads to a better performance in NLI, and the findings in the paper are applicable to application tasks.

G Details of Computational Experiments

We used nnsight (Fiotto-Kaufman et al., 2024) and pyvene (Wu et al., 2024a) for the implementation



(a) LONG-DISTANCE DEPENDENCIES category

(b) All categories

Figure 19: Scores in the category involving filler-gap dependencies (left) and in all categories (right) in SyntaxGym after multiplying activations of several attention heads in Pythia 1B by α.

of the experiments. We ran each experiment one time. We used A100 GPUs for around 300 hours for the experiments.