Revealing Redundant Syntax in Large Language Models through Multi-Hop Dependency Paths

Anonymous ACL submission

Abstract

We present the first systematic analysis of attention heads for syntactic relations in decoder-only Transformer language models. Prior work has demonstrated that encoder-only and encoder-decoder architectures contain attention heads aligned with single-hop syntactic relations, but the internal mechanisms of decoder-only models remain underexplored. Focusing on two representative families (GPT-2 and XGLM) across five model sizes (117M, 345M, 774M, 1.5B, 1.7B parameters), we identify a novel class of attention heads that capture multi-hop dependency paths (MDPs), e.g., "obl+case". Through controlled head-ablation on the BLiMP benchmark, we show that removing 25% MDP heads induces 7.1% drop in average grammaticality accuracy, compared to only 1.6% drop when ablating the same number of conventional, single-hop syntactic heads. Crucially, this pattern holds consistently across all five model sizes, demonstrating the robustness of our findings. Technically, we (i) extend existing head-identification methods—previously limited to encoder-only and encoder-decoder models—to the decoder-only setting, and (ii) propose a formal definition and detection algorithm for MDP heads. Our results reveal that decoder-only Transformers internalize syntactic information in more complex, non-canonical forms than previously understood, underscoring the importance of cross-chain interactions for grammatical competence.

1 Introduction

Understanding how large language models (LLMs) perform syntactic analysis is one of the intriguing topics for revealing the inner workings of LLMs (López-Otal et al., 2025). Previous studies have generally assumed the validity of linguistic dependencies (LD)—that is, syntactic structures as defined in theoretical linguistics—when analyzing LLMs. Some studies have provided several insights into the syntactic capabilities regarding the linguistic dependencies within the attention mechanisms of LLMs, especially for encoder-based models such as BERT (Clark et al., 2019; Kovaleva et al., 2019; Ravishankar et al., 2021), leaving the decoder-only models underexplored.

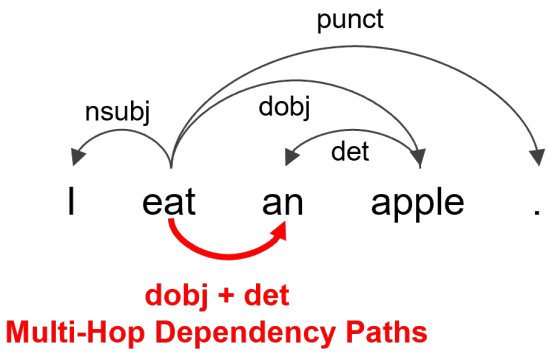


Figure 1: Comparison of linguistic dependencies and multi-hop dependency paths (MDPs) in "I eat an apple." Canonical linguistic dependency grammar typically posits relations like "dobj" between (eat, apple) or "det" between (an, apple). In contrast, LLMs often place significant attention to a shortcut path, e.g., "dobj" + "det" between (eat, an), which we define as MDPs.

This study analyzes linguistic dependencies for decoder-based models, which are the defact standard architecture of recent state-of-the-art LLMs (Grattafiori et al., 2024). Specifically, we focus on two representative models: GPT-2 (Radford et al., 2019) and XGLM (Neelakantan et al., 2022) across five model sizes (117M, 345M, 774M, 1.5B, 1.7B parameters) to identify attention heads that specialize in syntactic structure recognition.

Throughout the analysis, we have found that LLMs acquire not only the dependency structures defined by human linguistic theory, but also multi-hop dependency paths (MDPs)—model-internal syntactic multi-hops that do not strictly follow canonical linguistic dependencies. We define MDPs as dependency-like connections be-

tween two distant tokens that assigns high attention weight, bypassing intermediate tokens that would normally mediate the syntactic relationship. For instance, in the sentence "I eat an apple," canonical linguistic dependency grammar typically posits relations like (eat, apple) and (an, apple). In contrast, LLMs often place significant attention between (eat, an)—a pairing not traditionally considered a direct syntactic dependency—suggesting that such paths serve as alternative cues for grammatical understanding (Figure 1). These multi-hop paths enable a form of redundant syntactic encoding, which we find contributes to the robustness of the model's grammatical reasoning.

To reveal the existance of MDP heads, we first constructed a corpus using the Universal Dependencies (UD) framework, with MDP-based dependency relations added. We then identified attention heads that are particularly sensitive to MDP-style dependencies by extending the existing head-identification methods—previously limited to encoder-based models—to the decoder-only setting. Furthermore, to verify the functional importance of MDPs for grammatical understanding, we conducted attention intervention experiments, selectively disrupting attention weights toward MDP pairs. The results showed that disrupting attention heads specialized in MDPs led to significantly larger performance degradation on grammatical tasks than when intervening on heads associated with traditional dependencies across all the models.

The findings of this study are an important step toward clarifying the differences between the way humans understand language (i.e., linguistics) and the way LLMs understand the same language. We hope these findings provide valuable insights into understanding of the factors behind dramatic performance gains in recent decoder-based LLMs.

2 Background

2.1 Simplicity and Binarity in Syntactic Theory

Linguistic theories often emphasize non-redundant representations of grammar, aiming to capture syntactic structure with the smallest possible set of rules. This principle underlies Chomsky's *Minimalist Program*, which assumes that language is optimally designed and that syntactic derivations should be maximally economical. As Chomsky notes, "only binary branching is permitted" (Chomsky, 1995).

Under this assumption, syntactic trees are strictly binary, with each phrase dividing into two subcomponents. This binary structure not only enforces formal simplicity but also supports recursive compositionality in a uniform and constrained manner. The goal is to derive the full complexity of natural language syntax from a minimal set of generative principles.

2.2 Redundancy and Multi-Hop Dependency Paths in LLMs

Does the grammatical competence of large language models (LLMs) align with the principles of Chomsky's *Minimalist Program*?

The Minimalist Program postulates that language is governed by a minimal set of generative rules, favoring non-redundant and binary branching structures (Chomsky, 1995). In contrast, recent mechanistic studies suggest that LLMs may adopt a fundamentally different strategy. Induction heads—attention mechanisms responsible for pattern matching and copying—emerge in Transformers as additive and redundant circuits (Singh et al., 2024). These heads operate collectively, with overlapping functionalities, indicating that LLMs might not rely on minimal rules for syntactic representation.

As introduced in Section 1, empirical observations reveal that certain combinations of dependency relations frequently co-occur in parsed corpora. For example, paths such as `obj+nmod+case` or `obl+case` are highly recurrent and robust across diverse syntactic contexts. While such composite paths are not primitive relations in the Minimalist framework, they may nonetheless be learned by LLMs alongside linguistically defined dependencies. This suggests that LLMs internalize a hybrid syntactic representation: one that blends formally specified dependencies with statistically redundant co-occurrence patterns.

Example. Consider the sentence:

> *The teacher placed a book of science on the shelf.*

In the UD parse of this sentence, the following MDPs are identified as:

- `obl+case`:

$$on \xrightarrow{\texttt{case}} shelf \xrightarrow{\texttt{obl}} placed$$

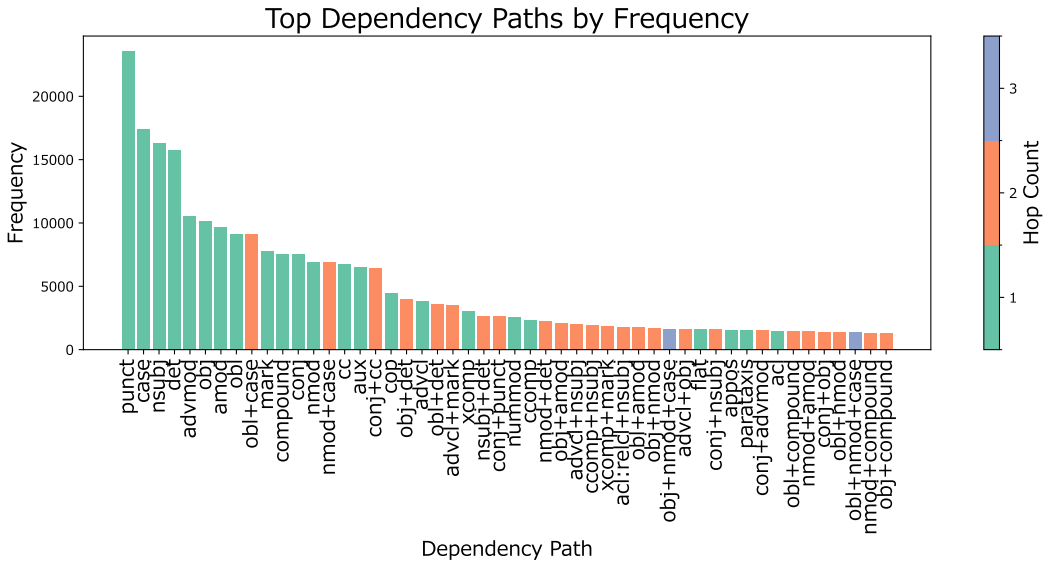


Figure 2: Bar plot of the 50 most frequent dependency paths in the corpus. Each bar's height shows the absolute count, and bar colors (Set2 palette) encode hop counts, highlighting which syntactic hop counts occur most often.

This path represents the oblique phrase *on the shelf*, where the noun *shelf* modifies the verb *placed* via the preposition *on*.

- `obj+nmod+case`:

$$of \xrightarrow{\texttt{case}} science \xrightarrow{\texttt{nmod}} book \xrightarrow{\texttt{obj}} placed$$

This captures an embedded possessive nominal phrase within the object of the verb. The noun *science* is connected to *book* through *of*, and *book* serves as the object of *placed*.

These MDPs reflect compositional structures that are not explicitly encoded in minimalist syntactic theory but nonetheless emerge as stable patterns in LLM training data.

3 Method

3.1 Constructing Multi-Hop Dependency Paths (MDPs)

To capture higher-order syntactic patterns beyond single-step dependencies, we construct a new set of compositional relations, which we call *Multi-Hop Dependency Paths* (MDPs). These MDPs represent frequently co-occurring sequences of syntactic relations along parent-child paths in Universal Dependencies (UD) trees. In this study, we refer to individual dependency relations as *Dependency Paths* (DPs).

We use the English Web Treebank (EWT) (Nivre et al., 2020) as our UD-parsed corpus. From this corpus, we extract all DPs, i.e., single-hop labeled dependency edges. Each sentence in the corpus consists of a sequence of tokens annotated with head indices and dependency labels. Based on the set of observed DPs, we iteratively construct longer MDPs by connecting multiple relations together.

We denote connecting two consecutive hops of DPs as 2-hop DPs. To identify frequently occurring 2-hop DPs, we begin by generating candidate 2-hop DP paths by pairing every DP with every other DP (i.e., DP $\times$ DP), and count their frequency in the corpus. Only those candidate MDPs whose frequency exceeds a predefined threshold τ are retained. In the next step, we extend the 2-hop DP candidates by appending another DP, yielding 3-hop DP = 2-hop DP $\times$ DP. Again, we count occurrences and retain only those that pass the threshold. This procedure is repeated iteratively until no new MDPs exceed the threshold.

The final set of MDPs are represented as atomic labels, such as `obj+amod+case`. Using this final set of MDPs along with the original English Web Treebank, we annotate the corpus to construct a new structured dataset that includes both DPs and the newly defined MDPs (See Figure 2).

3.2 Identifying Attention Heads Sensitive to Syntactic Structure

Using the MDP-annotated corpus as well as the DP corpus as described above, we identify which attention heads capture syntactic structures effectively.

By applying the approach of Clark et al. (2019),

which was originally proposed for encoder-based models, to decoder-only model for our experiment, we analyze attention weights to evaluate if each attention head focuses on syntactically related words.

For each layer l and head h, and each token x_i in a sentence, we determine the token x_j that receives the highest attention weight from x_i. Specifically, for a sentence with attention matrix $A^{(l,h)}$, we define the predicted syntactic connection as follows:

$$\hat{j}_i^{(l,h)} = \arg\max_j A_{ij}^{(l,h)}, \quad j < i. \tag{1}$$

Since the decoder-only model which is our target in our experiment employs causal masked attention, token x_i can only attend to tokens at or before its own position ($j < i$). This is the modification from the original method (Clark et al., 2019).

We then check if the predicted pair (x_i, x_j) matches a syntactic dependency (either DP or MDP) annotated in our corpus. Importantly, we do not consider the directionality of these syntactic dependencies because causal masks in decoder-only models allow only unidirectional attention to past tokens; we only measure whether the predicted pair corresponds to an existing dependency edge, ignoring head-dependent direction.

Because model tokenization typically differs from corpus word-level tokenization, we follow Clark et al. (2019)'s alignment strategy: attention from a word to another word is computed by summing attention scores over all subword tokens for the target word and averaging across subword tokens for the source word (Clark et al., 2019).

We perform this matching procedure for every token in every sentence of the corpus. For each syntactic dependency type r (DP or MDP), we count how many times each head (l, h) correctly identifies a dependency pair:

$$C_r^{(l,h)} = \sum_{\text{sentence} \in \text{corpus}} \sum_{(i,j) \in r} \mathbb{I}\left[\hat{j}_i^{(l,h)} = j\right] \tag{2}$$

Here, the count is summed across all sentences in the corpus. We define the correct rate for a given dependency relation r at attention head (l, h) as:

$$\text{CorrectRate}_r^{(l,h)} = \frac{C_r^{(l,h)}}{|\mathcal{D}_r|} \tag{3}$$

where $C_r^{(l,h)}$ is the number of correctly identified dependency pairs at head (l, h), and $|\mathcal{D}_r|$ is the total number of occurrences of dependency relation r in the corpus.

To identify the head which aligns mostly with a specific dependency relation r, the unlabeled attachment score (UAS) for relation r is defined as the highest correct rate among all attention heads:

$$\text{UAS}_r = \max_{l,h} \left(\text{CorrectRate}_r^{(l,h)}\right) \tag{4}$$

This methodology closely follows Clark et al. (2019)'s original interpretability approach, adapted here without considering dependency directionality for decoder-only model architecture.

3.3 Intervention by Flattening Selected Attention Heads

For each selected head $(l, h) \in \mathcal{H}$, we replace the original attention weight matrix $A^{(l,h)} \in \mathbb{R}^{S \times S}$ with a lower-triangular uniform distribution T, defined as:

$$T_{ij} = \begin{cases} \frac{1}{i+1}, & j \leq i, \\ 0, & j > i, \end{cases} \tag{5}$$

ensuring $\sum_{j=0}^{S-1} T_{ij} = 1$. Thus, the modified attention weight is:

$$A'^{(l,h)} = T. \tag{6}$$

This flattening intervention is inspired by prior work that neutralizes attention distributions to assess head importance (Zhou et al., 2025). Attention weights in all non-selected heads remain unaltered.

4 Experiments

4.1 Identification of MDPs

We use the English Web Treebank (EWT) (Nivre et al., 2020), a publicly available UD-annotated English corpus licensed under CC BY-SA 4.0. The dataset is anonymized and manually curated; we found no personally identifiable or offensive content. It covers a range of syntactic phenomena in web-based English and follows consistent UD annotation guidelines. In our experiments, we used all sentences from the training portion of the EWT corpus. We identified Multi-Hop Dependency Paths (MDPs) using the English Web Treebank (EWT) (Nivre et al., 2020). Specifically, applying the method outlined in Section 3.1, we extracted both single-hop Dependency Paths (DPs) and MDPs that occurred more than 1,000 times within the corpus.

	obl+case	obl	case	nmod+case	nmod	case	conj+cc	conj	cc	obj+nmod+case	obj	nmod	case
Frequency	9095	9150	17417	6883	6888	17417	6413	7523	6757	1655	10170	6888	17417
GPT2	**0.572**	0.158	0.358	**0.785**	0.341	0.358	**0.448**	0.239	0.412	0.353	**0.804**	0.341	0.358
GPT2-Large	**0.576**	0.285	0.416	**0.795**	0.309	0.416	**0.426**	0.323	0.401	0.630	**0.824**	0.309	0.416
GPT2-Medium	**0.673**	0.436	0.348	**0.743**	0.329	0.348	**0.427**	0.273	0.398	0.604	**0.831**	0.329	0.348
GPT2-XL	**0.711**	0.405	0.463	**0.764**	0.423	0.463	**0.456**	0.391	0.424	0.615	**0.831**	0.423	0.463
XGLM-1.7B	0.606	0.230	**0.643**	**0.806**	0.353	0.643	**0.513**	0.317	0.386	0.485	**0.648**	0.353	0.643

Table 1: The UAS results for the three most frequent 2-hop MDPs and their corresponding single-hop DP components, as well as the most frequent 3-hop MDP and its constituent DPs, across each model.

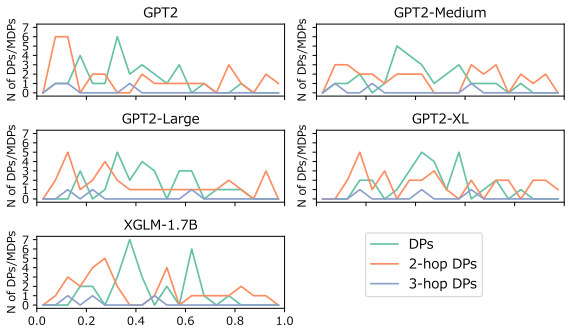


Figure 3: Line-histogram of maximum UAS across attention heads for each dependency hop count. Each subplot corresponds to one model, with colored lines indicating different hop counts. Only dependency relations occurring at least 1000 times are included, and the horizontal axis shows the maximum UAS achieved by any head, while the vertical axis shows the number of relations in each UAS bin.

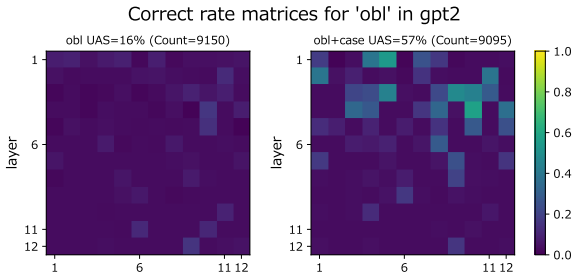


Figure 4: Correct rate matrices for the obl dependency in GPT-2. Left: DPs; Right: MDPs. Rows correspond to transformer layers, columns to attention heads, and the color scale indicates the correct rate per head.

Models used. We analyzed attention heads across the pretrained GPT model family—GPT-small, GPT-medium, GPT-large, and GPT-XL (117M, 345M, 774M, and 1.5B parameters)—as well as the XGLM-1.7B model (Radford et al., 2019; Neelakantan et al., 2022).

4.2 UAS Measurement and Analysis

We computed Unlabeled Attachment Scores (UAS) for the identified DPs and MDPs following the methodology described in Section 3.2. Table 1 summarizes the UAS results for the three most frequent 2-hop MDPs and their corresponding single-hop DP components, as well as the most frequent 3-hop MDP and its constituent DPs, across each model.

From Table 2, we observe that certain frequent 2-hop MDPs exhibit higher UAS compared to their individual DP components. Figure 3 illustrates the distribution of UAS values for all DPs and MDPs, revealing that there is no substantial difference in UAS distributions between the single-hop and multi-hop dependency paths.

4.3 Comparison of DP and MDP Usage

To determine whether models utilize single-hop DPs or multi-hop MDPs, we hypothesize that if a particular dependency is consistently accompanied by another dependency, the model is likely learning this combined dependency set. For each single-hop DP, we assessed whether it occurred predominantly (more than 50%) as a part of an MDP:

$$\text{MDP Usage} = \frac{\text{Count(DP as part of MDP)}}{\text{Total Count(DP)}} > 0.5 \quad (7)$$

We identified nine such DP-MDP pairs (see Table 2 for an overview and Appendix D for detailed definitions). If multiple MDPs surpassed this 50% threshold for a given DP, we selected the most frequently occurring set. We measured the correct rate for each head of each model, separately for these nine DPs and their corresponding MDP sets. Additionally, we recorded the maximum UAS across all heads.

Comparing UAS values model-wise, we found that MDPs consistently outperformed DPs in nearly all cases. Specifically, for the relation obl+case, all models except XGLM-1.7B showed higher UAS for the MDPs. For conj+cc and nmod+case, MDPs outperformed DPs across all evaluated models, as shown in Table 2. Furthermore, qualitative analysis

	obl+case	obl	case	conj+cc	conj	cc	nmod+case	nmod	case
Frequency	9095	9150	17417	6413	7523	6757	6883	6888	17417
GPT2	**0.572**	0.158	0.358	**0.448**	0.239	0.412	**0.785**	0.341	0.358
GPT2-Medium	**0.673**	0.436	0.348	**0.427**	0.273	0.398	**0.743**	0.329	0.348
GPT2-Large	**0.576**	0.285	0.416	**0.426**	0.323	0.401	**0.795**	0.309	0.416
GPT2-XL	**0.711**	0.405	0.463	**0.456**	0.391	0.424	**0.764**	0.423	0.463
XGLM-1.7B	0.606	0.230	**0.643**	**0.513**	0.317	0.386	**0.806**	0.353	0.643

Table 2: UAS scores for the three most frequent multi-hop dependency path (MDP) sets (ranks 1–3) and their corresponding single-hop dependency paths (DPs) across all evaluated models.

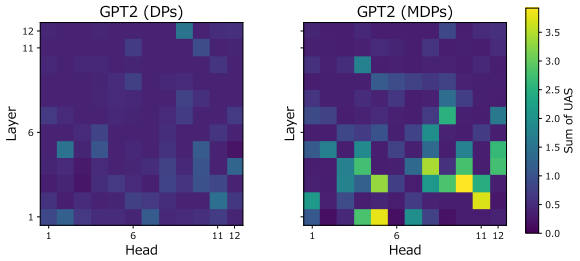


Figure 5: Cumulative correct-rate heatmaps used for head selection, computed by summing correct rate matrices over the nine most frequent single-hop dependency paths (DPs, left) and multi-hop dependency paths (MDPs, right). Rows correspond to layers and columns to attention heads.These matrices were used to identify the top 5–25% of heads for intervention.

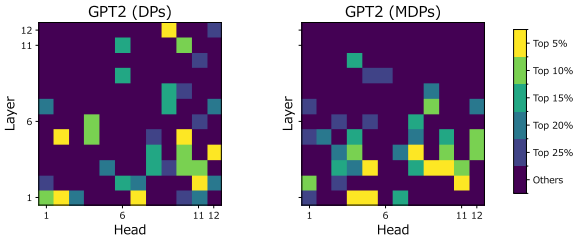


Figure 6: Discrete percentile-based dependency head activation heatmaps. The top 5%, 10%, 15%, 20%, and 25% of attention heads are highlighted using four distinct colors. Left: heads selected based on DPs; Right: heads selected based on MDPs.Rows represent Transformer layers and columns represent attention heads within each layer.

of correct rates revealed that the attention heads utilized for DP and MDP recognition differed notably (Figure 4).

4.4 Intervention Experiments

To empirically verify whether specific attention heads are actively used for grammatical understanding, we conducted intervention experiments using the Benchmark of Linguistic Minimal Pairs (BLiMP; (Warstadt et al., 2020)). BLiMP is a linguistically informed benchmark composed of minimal-pair sentences designed to assess grammatical knowledge across diverse syntactic phenomena.

We applied the intervention method described in Section 3.3. For selecting intervention heads, we summed the correct rate matrices across the nine DPs, then chose the top 5%, 10%, 15%, 20%, and 25% of heads based on the highest cumulative correct rates. In the case of GPT-2, Figure 5 shows the cumulative UAS values, while Figure 6 visualizes the selected heads in a discretized format. Each figure illustrates the differences in attention head selection based on the DP and MDP criteria.

Additionally, we performed control experiments by randomly selecting an equivalent number of heads for intervention. Results indicated that, except for the GPT-2 models, interventions on MDP-selected heads led to significantly larger accuracy reductions compared to interventions on DP-selected or randomly selected heads. However, GPT-2 models exhibited substantial variance, preventing the observation of clear differences.

To assess whether the importance of MDP-selected heads extends beyond syntactic tasks, we also evaluated their impact on long-range language modeling using the LAMBADA benchmark (Paperno et al., 2016). LAMBADA requires models to predict the final word of a passage based on a broad context, thus measuring coherence and semantic integration. Similar to the BLiMP setup, we disabled attention heads selected by DPs, MDPs, and random sampling at varying percentages. We found that disabling MDP-selected heads led to a much larger drop in accuracy ($\Delta = 38.5\%$) compared to DP-selected heads ($\Delta = 23.2\%$), reinforcing the hypothesis that multi-hop dependency paths correspond to functionally central heads involved in both

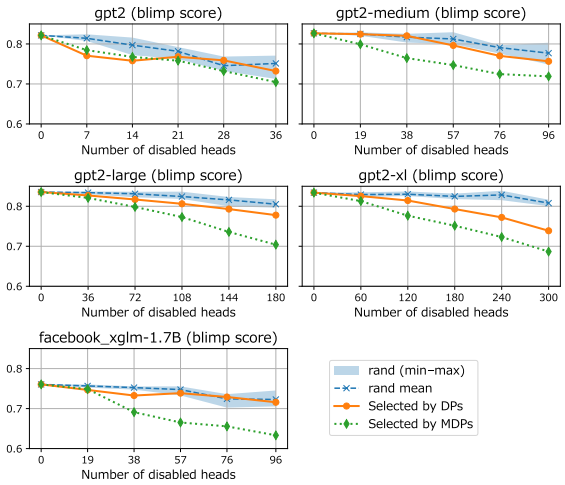


Figure 7: BLiMP grammaticality accuracy as a function of the number of disabled heads for four GPT-2 variants and XGLM-1.7B. Each subplot corresponds to a different model.Selected by DPs: heads selected based on the nine most frequent single-hop dependency paths.Selected by MDPs: heads selected based on the nine most frequent multi-hop dependency paths.The shaded region (min–max) shows the range and mean performance drop when ablating the same number of heads randomly.The horizontal axis indicates the number of ablated heads, and the vertical axis shows BLiMP accuracy.Disabling heads selected by MDPs results in a substantially larger performance drop ($\Delta = 7.1\%$) compared to DPs ($\Delta = 1.6\%$), confirming the stronger functional role of multi-hop paths in capturing grammatical constraints.

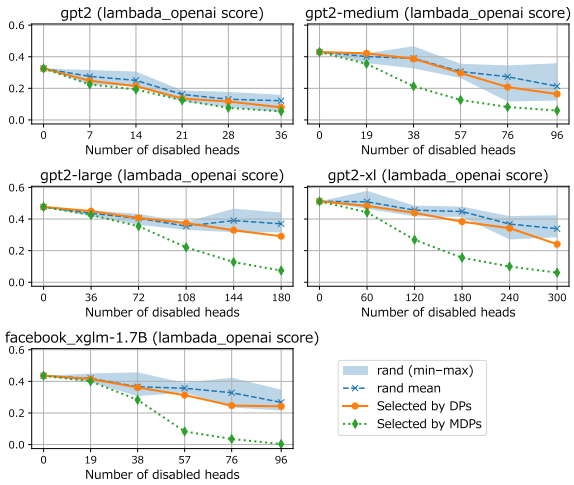


Figure 8: LAMBADA accuracy (open-domain consistency) as a function of the number of disabled heads for four GPT-2 variants and XGLM-1.7B.Setup is identical to BLiMP (Figure 7). Disabling heads selected by MDPs results in a greater accuracy reduction ($\Delta = 38.5\%$) compared to DPs ($\Delta = 23.2\%$), suggesting that multi-hop dependencies also play a crucial role in long-range semantic coherence.

grammatical reasoning and semantic coherence.

5 Related Work

Attention and Syntactic Dependencies Studies on the interpretability of Transformer architectures have suggested that attention heads may capture syntactic dependencies (Clark et al., 2019; Lin et al., 2019; Kovaleva et al., 2019; Lin et al., 2022; Ravishankar et al., 2021). Clark et al. (2019) demonstrated that certain attention heads in BERT correspond to syntactic relations such as subject-verb agreement and coreference resolution. Further, Ravishankar et al. (2021) analyzed multilingual BERT and reported that attention heads reflect syntactic dependencies across multiple languages.

Criticism of Attention Interpretability On the other hand, there have been criticisms regarding whether attention weights truly reflect the importance of features in model decisions. Serrano and Smith (2019) showed that altering attention weights does not necessarily lead to significant changes in model output, raising doubts about attention as an indicator of interpretability. Similarly, (Hassid et al., 2022) reevaluated the role of attention mechanisms and reported that using averaged attention weights does not substantially degrade model performance.

Induction Heads and Grammatical Reasoning There has also been growing interest in the role of induction heads in Transformer models. Induction heads are known to detect patterns in the input sequence and assist in predicting subsequent tokens (Singh et al., 2024). Such mechanisms may play a crucial role in enabling models to learn grammatical structures. For instance, certain attention heads may capture dependencies between multiple tokens, thereby contributing to grammatical reasoning.

Positioning of This Work While prior work has primarily focused on individual syntactic dependencies and localized grammatical relationships, our study focuses on *multi-hop dependency paths* (MDPs), which consist of sequences of dependency relations. These MDPs form structures that differ from traditional linguistic dependencies and may play an important role in how models internalize grammatical knowledge. We demonstrate the exis-

	obl+case	obl	case	conj+cc	conj	cc	nmod+case	nmod	case	advcl+mark	advcl	mark
Frequency	4754	8969	10672	3723	7514	3775	1022	6829	10672	2637	3761	4108
GPT2	**0.448**	0.153	0.182	0.189	**0.239**	0.060	0.328	**0.343**	0.182	**0.222**	0.149	0.131
GPT2-Medium	**0.537**	0.428	0.207	**0.295**	0.273	0.115	0.297	**0.331**	0.207	**0.338**	0.118	0.122
GPT2-Large	**0.498**	0.279	0.310	0.320	**0.323**	0.171	**0.349**	0.311	0.310	**0.276**	0.164	0.161
XGLM-1.7B	0.429	0.231	**0.618**	0.299	**0.317**	0.276	0.254	0.356	**0.618**	0.260	0.202	**0.276**

Table 3: The UAS scores for each of the nine identified single-hop dependency paths (DPs) and their corresponding multi-hop dependency path (MDP) sets across all evaluated models. For each DP–MDP pair, we report the accuracy excluding token pairs that are adjacent in the sentence, i.e., direct neighbors are ignored when computing UAS.

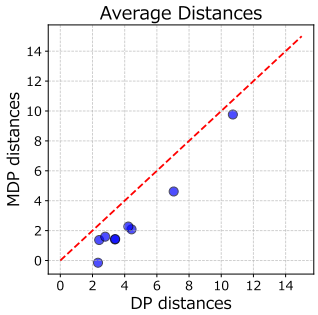


Figure 9: Scatter plot comparing average dependency path (DP) distances (x-axis) against their corresponding multi-hop dependency path (MDP) distances (y-axis) for the nine selected relations. Each point represents one DP–MDP pair, and the red dashed line indicates the identity line (y = x), showing whether MDP distances exceed their DP counterparts.

tence of attention heads that are sensitive to MDPs and show that they contribute to grammatical inference. Furthermore, through direct interventions on attention weights, we reveal that these weights are indeed utilized by the model during inference.

6 Discussion and Conclusions

Previous research primarily focused on the extent to which LLMs reflect linguistically defined dependency structures. In contrast, our findings demonstrate that LLMs also learn dependency structures not explicitly defined by linguistic theory. Specifically, we identified attention heads that are more responsive to Multi-Hop Dependency Paths (MDPs) compared to single-hop Dependency Paths (DPs).

Particularly notable are DPs that rarely occur alone but frequently appear in specific sets; in these cases, we found that models more accurately attend to MDPs containing these DPs than to the DPs alone. One potential explanation for this phenomenon involves token proximity. As indicated in Figure 9, the average token distance in frequently co-occurring DP sets is typically larger than in their corresponding MDPs. Considering the autoregressive nature of GPT and XGLM models, the shorter token distances in MDPs could facilitate easier attention.

It is possible that certain attention heads predominantly attend to adjacent tokens, regardless of grammatical considerations, thereby driving this observed pattern. To investigate this hypothesis, we repeated the UAS analysis while excluding token pairs immediately adjacent to each other.

Table 3 presents DP-MDP pairs occurring more than 1,000 times, filtered to exclude adjacent tokens, using the same criteria as in Section 4.3. This analysis confirms that MDPs still achieve higher accuracy than single-hop DPs, even when adjacent token pairs are excluded. However, it remains plausible that shorter token distances, even when not immediately adjacent, might still bias attention toward MDPs.

7 Limitations

This study has several limitations that provide avenues for future research. First, the complementary relationship between DP and MDP remains unclear. Given that MDPs can often be decomposed into combinations of multiple DPs, it is not yet fully understood what motivates the model to learn these composite structures over individual LDs.

Second, our methodology primarily relies on attention weights. While attention weights offer interpretability advantages, they may not capture all aspects of the underlying syntactic representations. There may be alternative methods, such as analyzing the neural circuit level, that provide deeper insights into how models internally represent syntactic information.

Finally, we primarily explored medium-sized models. Understanding how larger-scale models behave in terms of dependency encoding and grammatical reasoning, particularly regarding their utilization of DP and MDP structures, remains a promising direction for future work.

References

Noam Chomsky. 1995. *The Minimalist Program*. MIT Press, Cambridge, MA.

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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Michael Hassid, Hao Peng, Daniel Rotem, Jungo Kasai, Ivan Montero, Noah A. Smith, and Roy Schwartz. 2022. How much does attention actually attend? questioning the importance of attention in pretrained transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1403–1416, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. 2019. A surprisingly robust trick for the Winograd schema challenge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4837–4842, Florence, Italy. Association for Computational Linguistics.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. 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 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, and 2 others. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Miguel López-Otal, Jorge Gracia, Jordi Bernad, Carlos Bobed, Lucía Pitarch-Ballesteros, and Emma Anglés-Herrero. 2025. Linguistic interpretability of transformer-based language models: a systematic review. *Preprint*, arXiv:2504.08001.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas A. Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David P. Schnurr, Felipe Petroski Such, Kenny Sai-Kin Hsu, and 6 others. 2022. Text and code embeddings by contrastive pre-training. *ArXiv*, abs/2201.10005.

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.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Vinit Ravishankar, Artur Kulmizev, Mostafa Abdou, Anders Søgaard, and Joakim Nivre. 2021. Attention can reflect syntactic structure (if you let it). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3031–3045, Online. Association for Computational Linguistics.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Aaditya K. Singh, Ted Moskovitz, Felix Hill, Stephanie C. Y. Chan, and Andrew M. Saxe. 2024. What needs to go right for an induction head? a mechanistic study of in-context learning circuits and their formation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

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.

Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, Kun Wang, Yang Liu, Junfeng Fang, and Yongbin Li. 2025. On the role of attention heads in large language model safety. *Preprint*, arXiv:2410.13708.

A Example Sentences for Each DP–MDP Pair

To better illustrate the linguistic structures represented by the selected multi-hop dependency paths (MDPs), we provide one example sentence for each of the nine DP–MDP pairs introduced in Section 4.3. These examples were extracted from the English Web Treebank (EWT) and selected to be relatively short (13 words or fewer). For each pair, we show the sentence and the syntactic path that connects the relevant tokens, highlighting how the composed MDPs reflect an interpretable syntactic shortcut such as a prepositional phrase or relative clause.

- `obl+case`:

  The third was being run by the head of an investment firm .

  Path: *by* $\xrightarrow{\texttt{case}}$ *head* $\xrightarrow{\texttt{obl}}$ *run*

- `conj+cc`:

  This item is a small one and easily missed .

  Path: *and* $\xrightarrow{\texttt{cc}}$ *one* $\xrightarrow{\texttt{conj}}$ *missed*

- `nmod+case`:

  The third was being run by the head of an investment firm .

  Path: *of* $\xrightarrow{\texttt{case}}$ *firm* $\xrightarrow{\texttt{nmod}}$ *head*

- `advcl+mark`:

  If someone committed a crime against humanity , prosecute the person .

  Path: *If* $\xrightarrow{\texttt{mark}}$ *committed* $\xrightarrow{\texttt{advcl}}$ *prosecute*

- `xcomp+mark`:

  The situation in Iraq is only going to get better this way .

  Path: *to* $\xrightarrow{\texttt{mark}}$ *get* $\xrightarrow{\texttt{xcomp}}$ *going*

- `ccomp+nsubj`:

  You wonder if he was manipulating the market with his bombing targets .

  Path: *he* $\xrightarrow{\texttt{nsubj}}$ *manipulating* $\xrightarrow{\texttt{ccomp}}$ *wonder*

- `acl:relcl+nsubj`:

  Now that 's a post I can relate to .

  Path: *I* $\xrightarrow{\texttt{nsubj}}$ *relate* $\xrightarrow{\texttt{acl:relcl}}$ *post*

- `parataxis+nsubj`:

  Just go here , it 's simply amazing .

  Path: *it* $\xrightarrow{\texttt{nsubj}}$ *amazing* $\xrightarrow{\texttt{parataxis}}$ *go*

- `acl+mark`:

  There has been talk that the night curfew might be implemented again .

  Path: *that* $\xrightarrow{\texttt{mark}}$ *implemented* $\xrightarrow{\texttt{acl}}$ *talk*

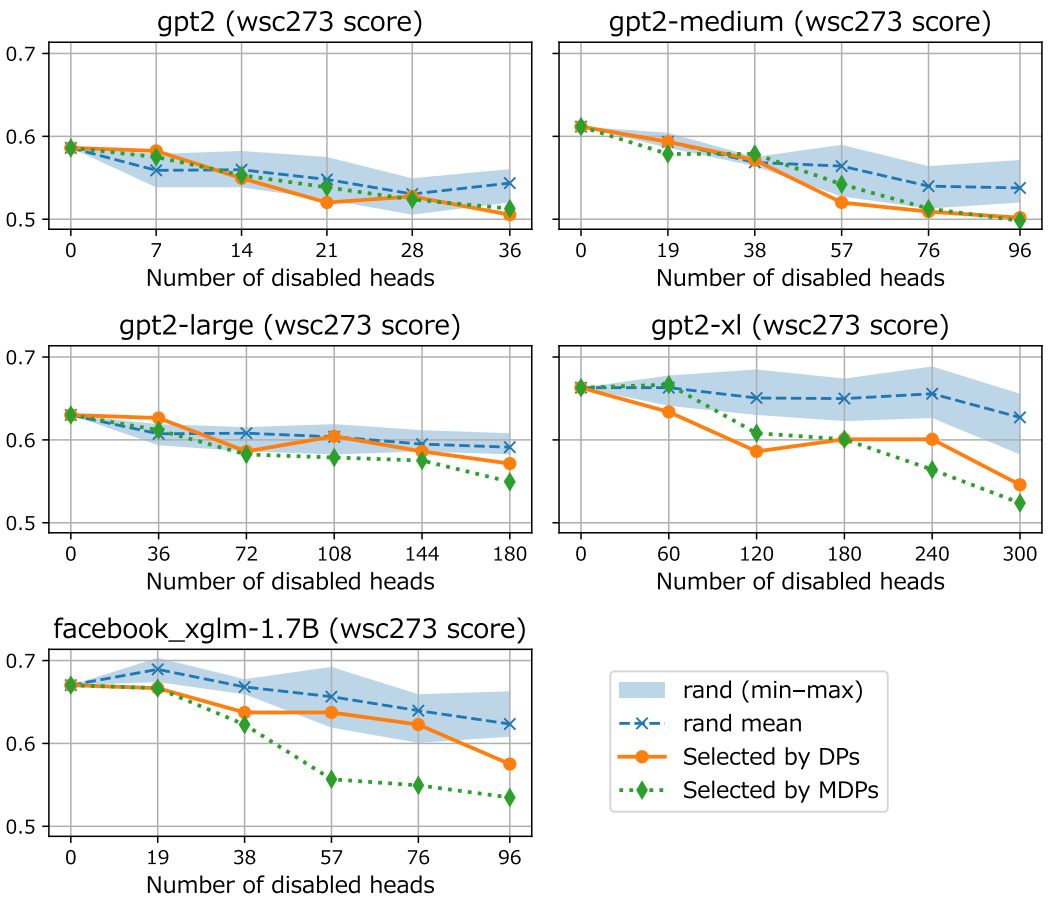


Figure 10: Accuracy on the WSC273 benchmark under head ablation. Left: DPs, Right: MDPs. Disabling MDP-based heads causes a slightly larger drop in accuracy ($\Delta = 14.7\%$) compared to DP-based heads ($\Delta = 13.0\%$).

B Intervention Experiments with Additional Benchmarks

To test whether the effects of dependency-based head interventions generalize beyond BLiMP and LAMBADA, we extended our analysis to two additional widely used benchmarks: **WSC273** and **WikiText**.

WSC273. The Winograd Schema Challenge 273 (WSC273; (Kocijan et al., 2019)) is a coreference resolution benchmark composed of 273 pronoun disambiguation problems. Each item requires reasoning over semantics and world knowledge to resolve ambiguous pronouns correctly. Performance is measured by classification accuracy.

WikiText. WikiText (specifically WikiText-103; (Merity et al., 2016)) is a large-scale language modeling benchmark based on full Wikipedia articles. We evaluate models using perplexity (lower is better) to quantify their ability to model long-form, naturalistic text.

Intervention Setup. We applied the same intervention method described in Section 3.3, comparing heads selected by DPs and MDPs against random baselines. Performance differences are calculated relative to the non-intervened baseline.

C Layer Preferences by Dependency Hop Counts

To investigate whether attention heads that are sensitive to different dependency structures appear at different layers, we analyze the distribution of heads exceeding a UAS threshold across layers. We group these heads by dependency hop count, ranging from 1 to 4, and visualize their cumulative layer-wise

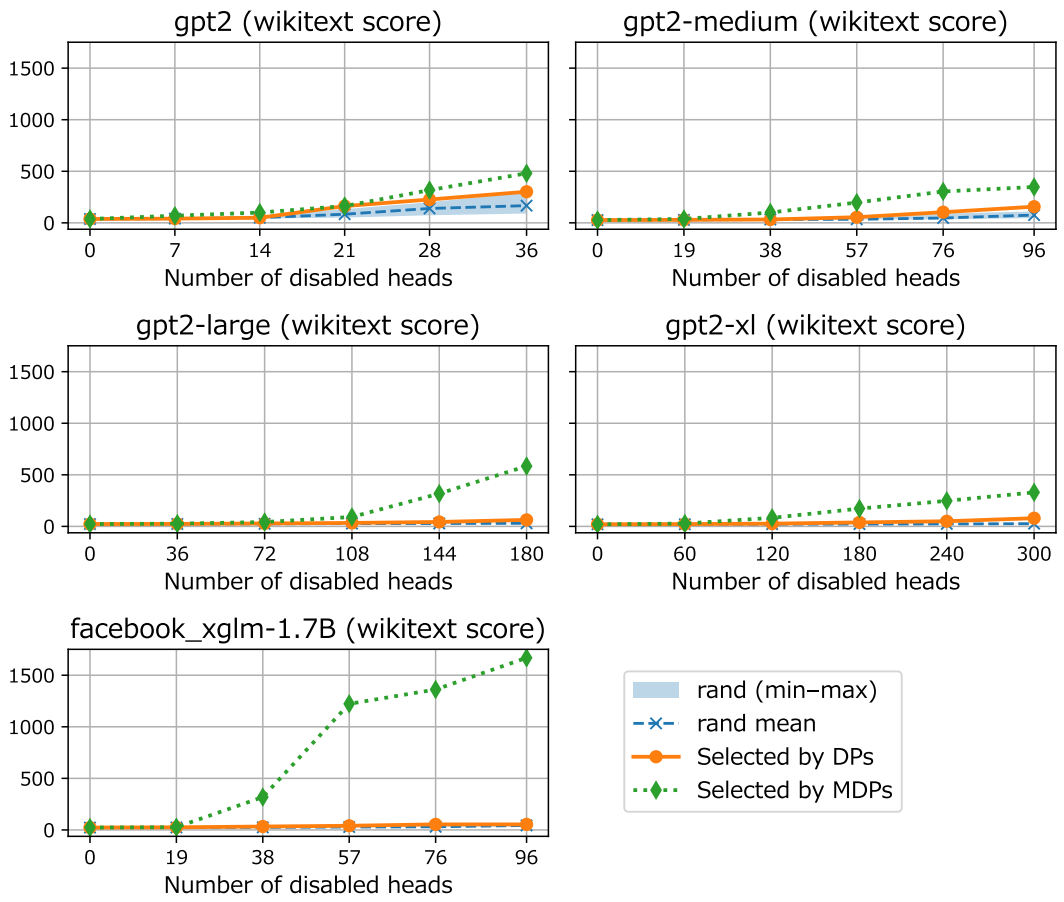


Figure 11: Perplexity on the WikiText benchmark under head ablation. Lower is better. Disabling MDP-based heads leads to a significantly larger increase in perplexity ($\Delta = 659.4$) than disabling DP-based heads ($\Delta = 108.0$).

distribution. This allows us to assess whether higher-hop relations are more likely to be captured in deeper layers.

Concretely, for each dependency pattern r (either DP or MDP) that appears at least 100 times in the corpus (i.e., $|\mathcal{D}_r| \geq 100$), we check whether any attention head (l, h) achieves a correct rate above a fixed threshold $\theta = 0.2$:

$$\text{CorrectRate}_r^{(l,h)} > \theta. \tag{8}$$

If such a head exists for relation r, we count it as *captured*. We then accumulate these counts by hop count k (from 1 to 4) and layer index l, resulting in a cumulative count matrix $N_l^{(k)}$, which represents the number of distinct DPs or MDPs of hop count k captured up to layer l.

Our findings show that although 4-hop relations tend to be more concentrated in shallower layers, overall there is no strong or consistent difference in cumulative distributions across hop counts. That is, syntactic information at varying hop distances does not exhibit a clear tendency to be captured in systematically deeper or shallower layers(Figure12).

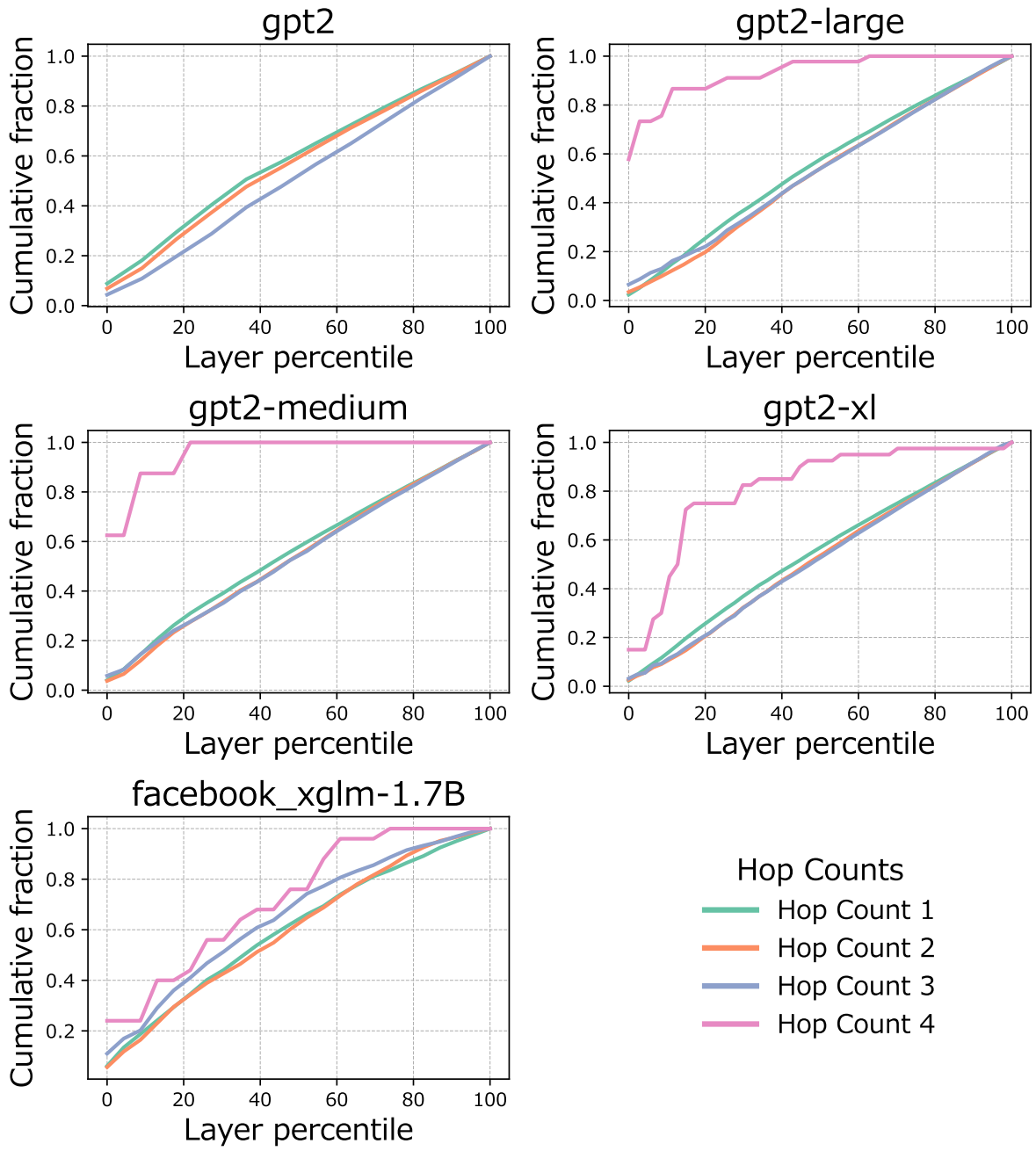


Figure 12: Cumulative distribution of attention heads exceeding a given UAS threshold, grouped by dependency hop count (1–4). Each curve shows the cumulative fraction of qualifying heads up to a given layer percentile. The last subplot (bottom right) shows the legend for hop counts. Results are shown separately for each model.

	advcl+mark	advcl	mark	xcomp+mark	xcomp	mark	ccomp+nsubj	ccomp	nsubj
Frequency	3506	3817	7774	1834	3070	7774	1925	2327	16270
GPT2	0.290	0.149	**0.419**	**0.903**	0.359	0.419	**0.524**	0.193	0.320
GPT2-Medium	0.432	0.121	**0.434**	**0.830**	0.514	0.434	**0.665**	0.304	0.307
GPT2-Large	0.339	0.163	**0.457**	**0.931**	0.575	0.457	**0.549**	0.427	0.304
GPT2-XL	0.411	0.206	**0.461**	**0.947**	0.675	0.461	**0.550**	0.495	0.365
XGLM-1.7B	0.340	0.205	**0.395**	**0.893**	0.536	0.395	**0.486**	0.376	0.344

Table 4: UAS scores for the moderately frequent multi-hop dependency path (MDP) sets (ranks 4–6) and their corresponding single-hop dependency paths (DPs) across all evaluated models.

	acl:relcl+nsubj	acl:relcl	nsubj	parataxis+nsubj	parataxis	nsubj	acl+mark	acl	mark
Frequency	1797	2005	16270	985	1562	16270	816	1493	7774
GPT2	**0.610**	0.325	0.320	0.133	0.156	**0.320**	**0.820**	0.421	0.419
GPT2-Medium	**0.616**	0.428	0.307	0.191	0.166	**0.307**	**0.838**	0.628	0.434
GPT2-Large	**0.631**	0.464	0.304	0.216	0.161	**0.304**	**0.800**	0.480	0.457
GPT2-XL	**0.610**	0.463	0.365	0.252	0.174	**0.365**	**0.820**	0.425	0.461
XGLM-1.7B	**0.538**	0.420	0.344	0.210	0.178	**0.344**	**0.812**	0.395	0.395

Table 5: UAS scores for the least frequent multi-hop dependency path (MDP) sets (ranks 7–9) and their corresponding single-hop dependency paths (DPs) across all evaluated models.

D Detailed Experimental Results

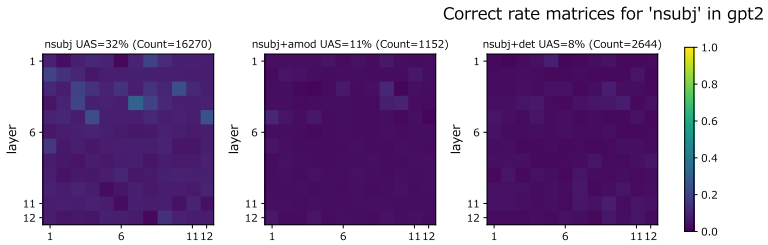


Figure 13: Correct rate matrices for the nsubj dependency in GPT-2. Left: single-hop dependency paths (DPs); Right: multi-hop dependency paths (MDPs). Rows correspond to transformer layers, columns to attention heads, and the color scale indicates the correct rate per head.

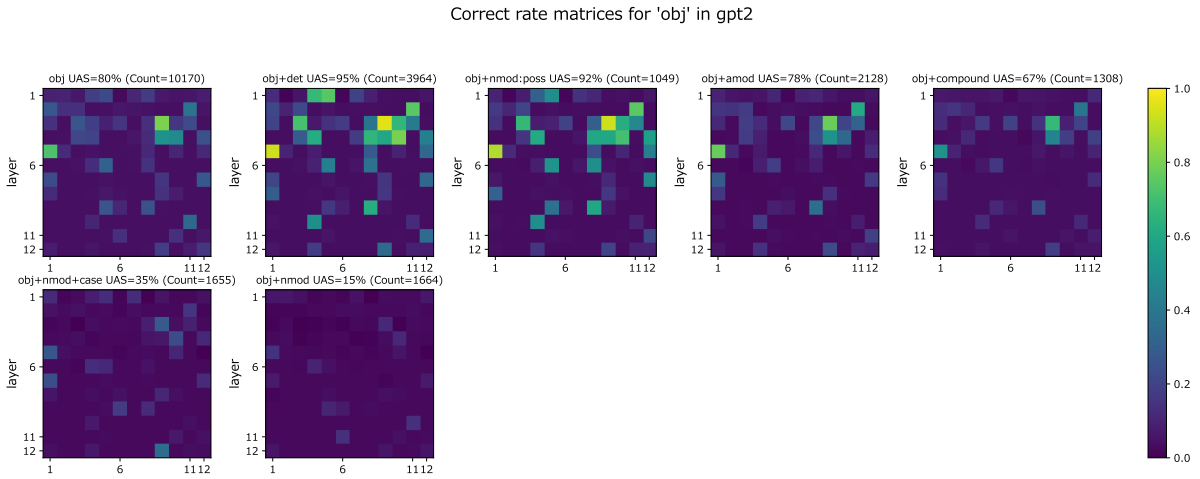


Figure 14: Correct rate matrices for the obj dependency in GPT-2. Left: single-hop dependency paths (DPs); Right: multi-hop dependency paths (MDPs). Rows correspond to transformer layers, columns to attention heads, and the color scale indicates the correct rate per head.

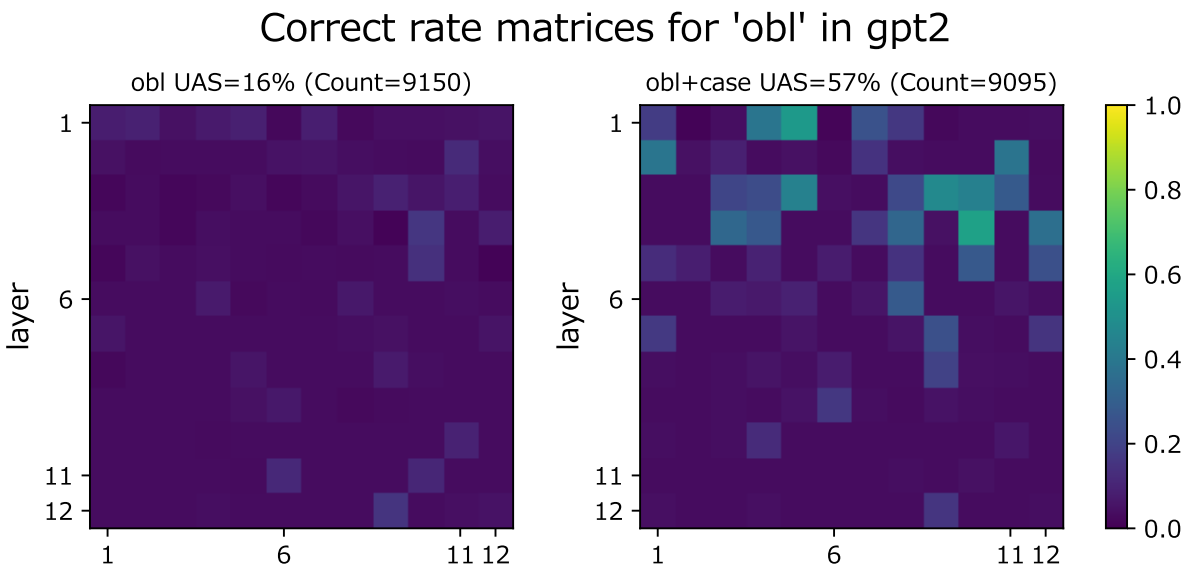


Figure 15: Correct rate matrices for the obl dependency in GPT-2. Left: single-hop dependency paths (DPs); Right: multi-hop dependency paths (MDPs). Rows correspond to transformer layers, columns to attention heads, and the color scale indicates the correct rate per head.

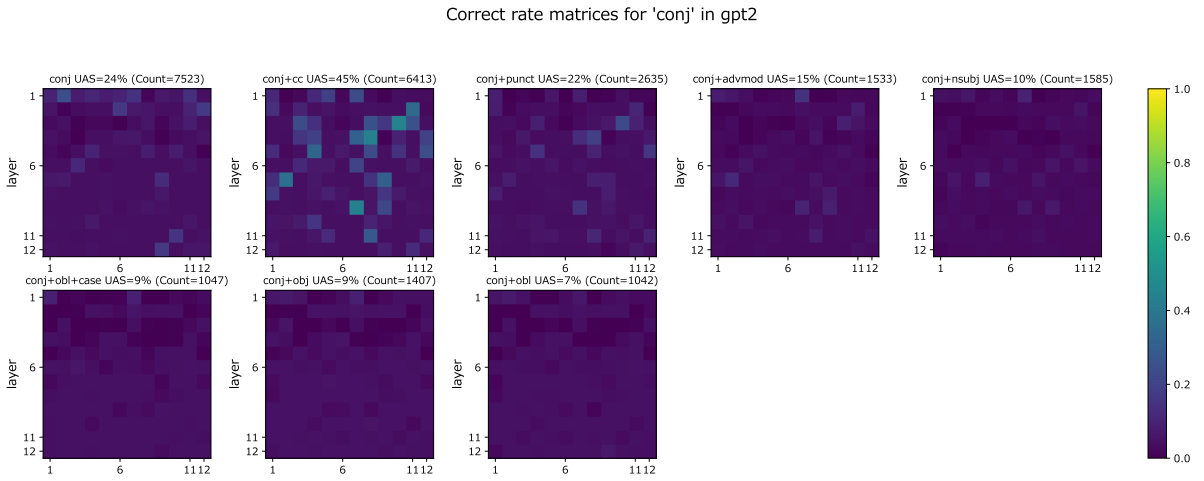


Figure 16: Correct rate matrices for the conj dependency in GPT-2. Left: single-hop dependency paths (DPs); Right: multi-hop dependency paths (MDPs). Rows correspond to transformer layers, columns to attention heads, and the color scale indicates the correct rate per head.

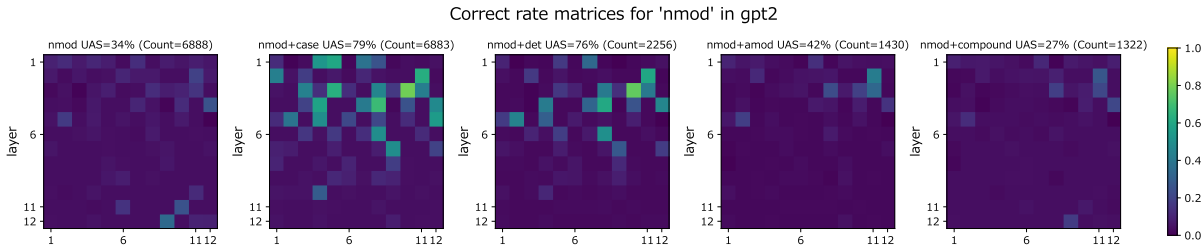
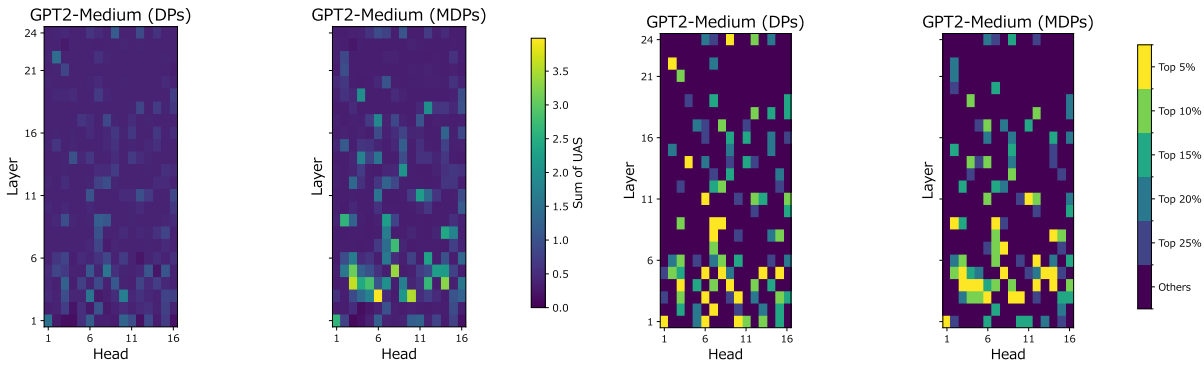


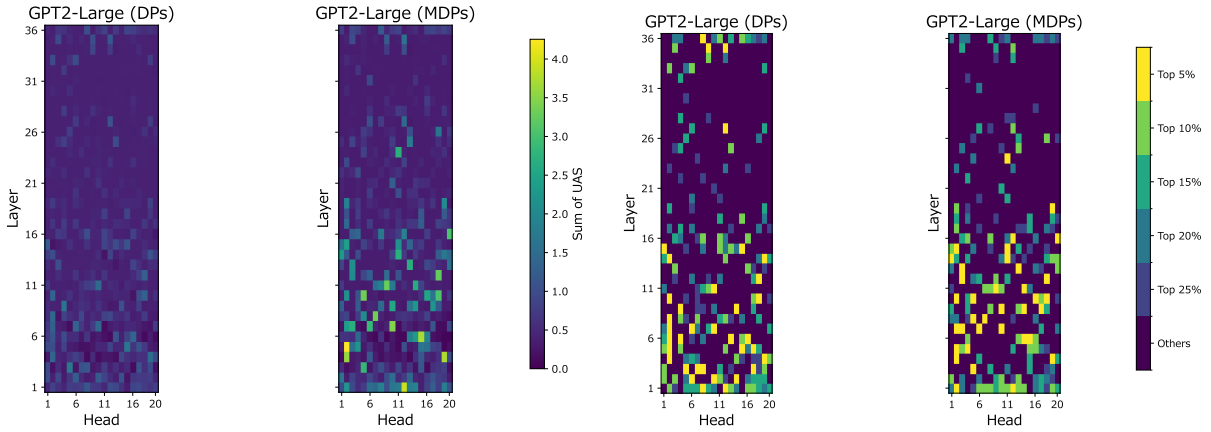
Figure 17: Correct rate matrices for the nmod dependency in GPT-2. Left: single-hop dependency paths (DPs); Right: multi-hop dependency paths (MDPs). Rows correspond to transformer layers, columns to attention heads, and the color scale indicates the correct rate per head.



(a) Cumulative correct-rate heatmap (DPs vs. MDPs).

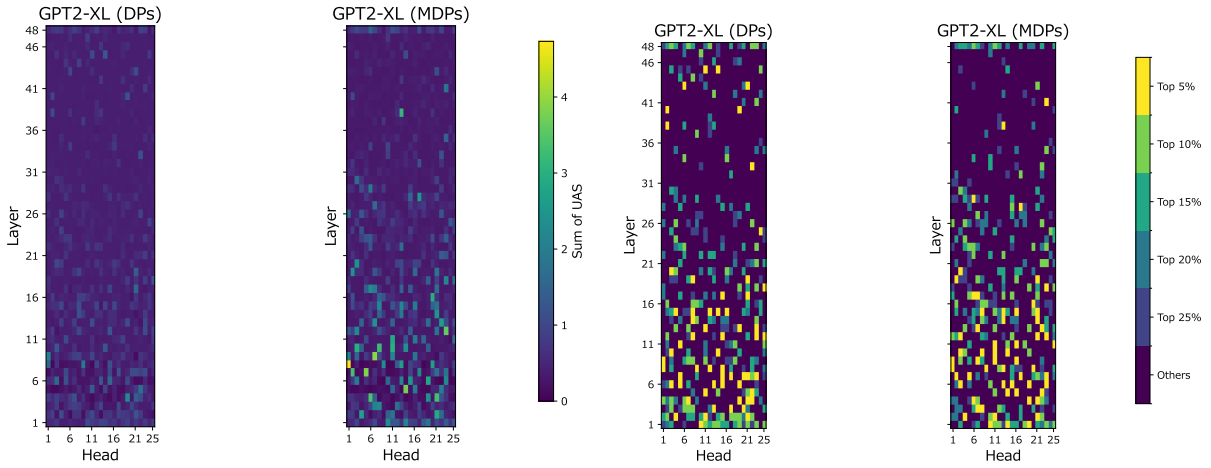
(b) Discrete percentile-based activation.

Figure 18: Head-selection heatmaps for GPT2-Medium (see main text Figure 5). Left: cumulative correct-rate; right: discrete percentiles.



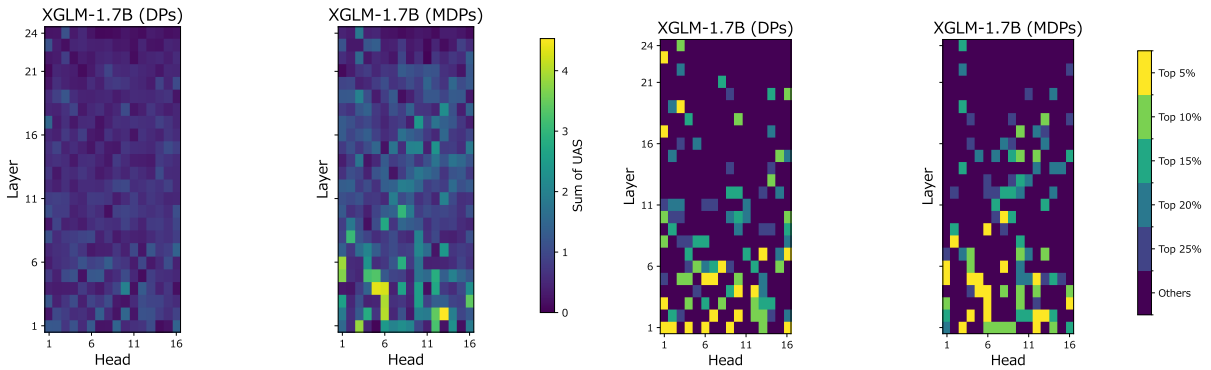
(a) Cumulative correct-rate heatmap (DPs vs. MDPs). (b) Discrete percentile-based activation.

Figure 19: Head-selection heatmaps for GPT2-Large (see main text Figure 5).



(a) Cumulative correct-rate heatmap (DPs vs. MDPs). (b) Discrete percentile-based activation.

Figure 20: Head-selection heatmaps for GPT2-XL (see main text Figure 5).



(a) Cumulative correct-rate heatmap (DPs vs. MDPs). (b) Discrete percentile-based activation.

Figure 21: Head-selection heatmaps for XGLM-1.7B (see main text Figure 5).

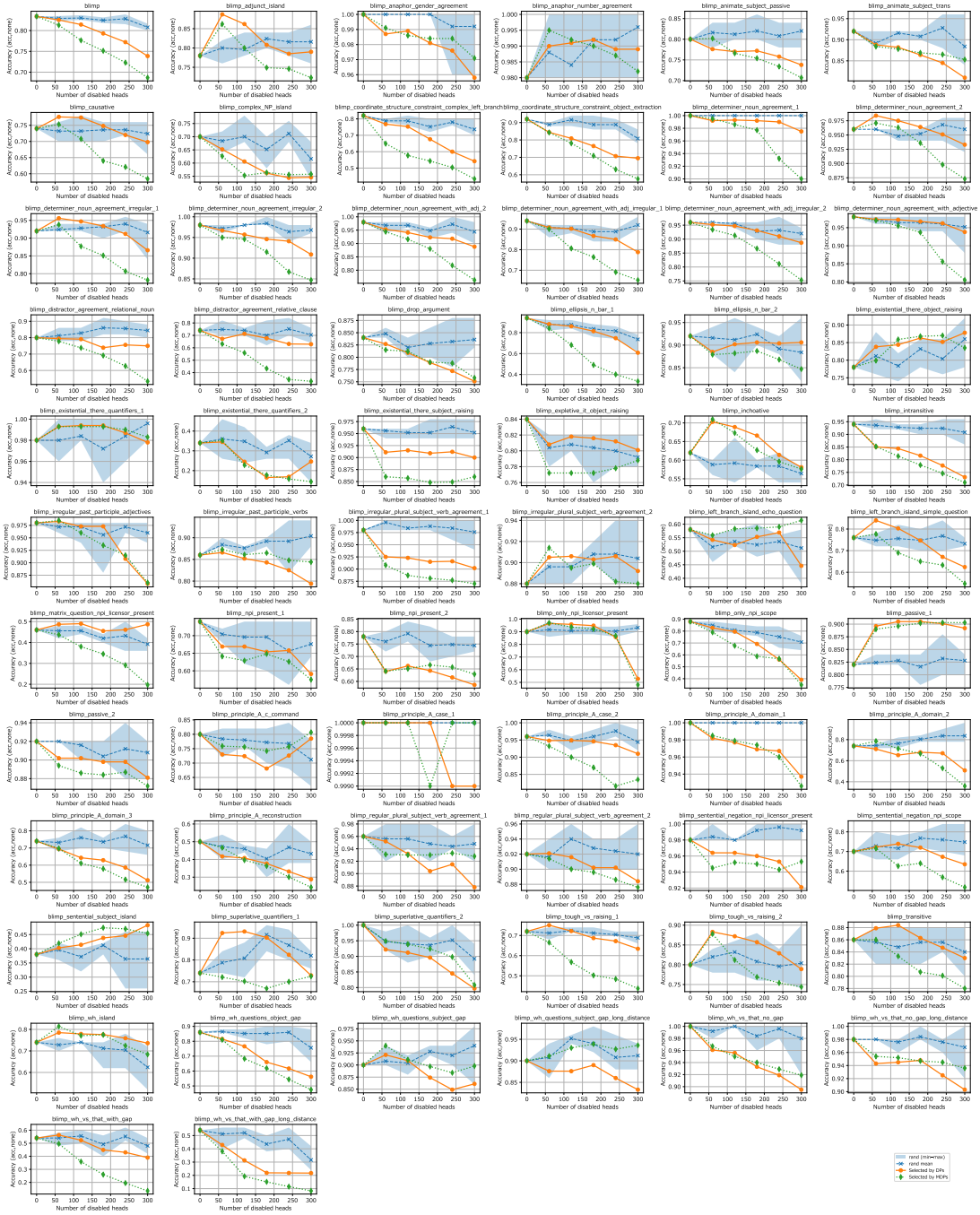


Figure 22: BLiMP grammaticality accuracy as a function of the number of disabled heads for GPT-2 XL. Each subplot shows performance when ablating heads selected by the nine most frequent single-hop dependency paths (Selected by DPs) and by the nine most frequent multi-hop dependency paths (Selected by MDPs). The shaded region (min–max) indicates performance when ablating the same number of heads at random. The horizontal axis is the number of ablated heads, and the vertical axis is BLiMP accuracy. Ablating heads chosen by MDPs produces a markedly larger drop in accuracy than ablating heads chosen by DPs, highlighting the stronger role of multi-hop paths in encoding grammatical constraints.

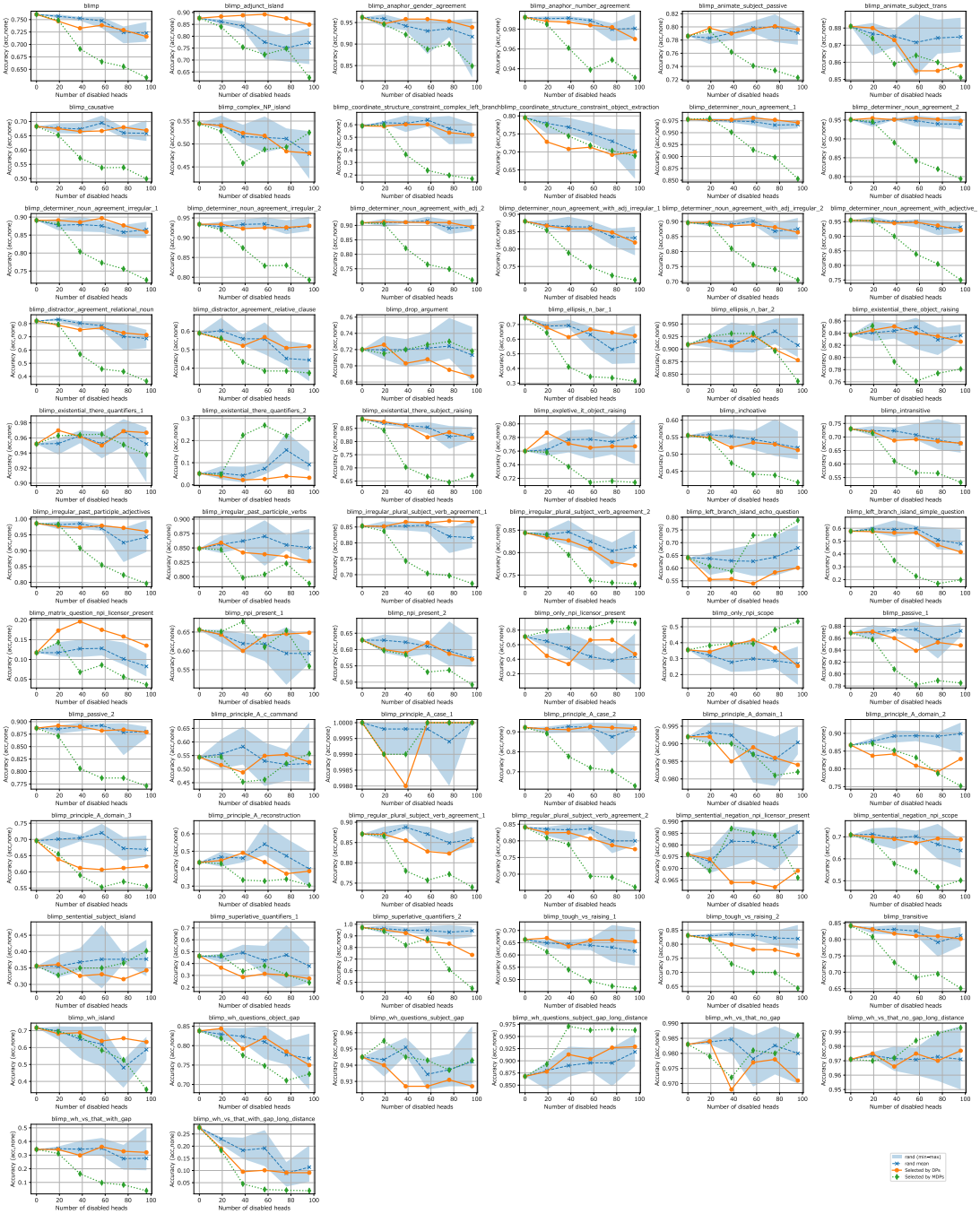


Figure 23: BLiMP grammaticality accuracy as a function of the number of disabled heads for XGLM-1.7B (see previous GPT-2 XL version for reference). Conventions are identical: “Selected by DPs” marks ablations on heads chosen via single-hop dependency paths, “Selected by MDPs” via multi-hop paths, and the shaded area (min–max) shows random ablations. The x-axis is disabled-head count and the y-axis is accuracy. Ablating MDP-selected heads again yields a substantially larger accuracy decline than DPs, reinforcing the importance of multi-hop dependency structure in model performance.