Investigating Internal Mechanisms of Long-distance Dependency Formation in Language Models

Anonymous ACL submission

Abstract

While previous studies reveal the presence of shared syntactic mechanisms by which dependencies are resolved in language models, they do not specify what procedure or strategy these mechanisms encode and whether they are differentiated by dependency types and model size. To give a deeper understanding of the internal mechanisms, we analyze four dependencies across model sizes using activation patching. Our finding is that (i) small models rely on similar attention-based heuristics, whereas larger models exhibit more differentiated attention patterns, and (ii) residual-stream-based predictive processing is robust across model sizes and is selectively associated with displacement dependencies. These results suggest that attention-based retrieval-like operations and residual-stream-based predictive processing contribute differently across dependency types and model sizes.

1 Introduction

There is an active debate as to how Transformer-based language models (LMs) (Vaswani et al., 2017) encode non-local syntactic dependencies, *long-distance dependencies* (Linzen and Baroni, 2021). More recently, mechanistic interpretability studies have shown that the mechanisms by which grammatical dependencies are resolved are supported by specific sub-networks or computational pathways (Ahuja et al., 2025; Boguraev et al., 2025; Kumon and Yanaka, 2026). Furthermore, previous studies observe that LMs internalize the mechanism similar to the human sentence processing strategies (Wilcox et al., 2021; Timkey and Linzen, 2023; Hanna and Mueller, 2025).

However, it remains unclear *what procedures* underlie the shared mechanisms and *which components* of the model encode them. More specifically, whether the shared mechanisms exhibit linguistically or cognitively motivated convergence in their internal activations has not been explored.

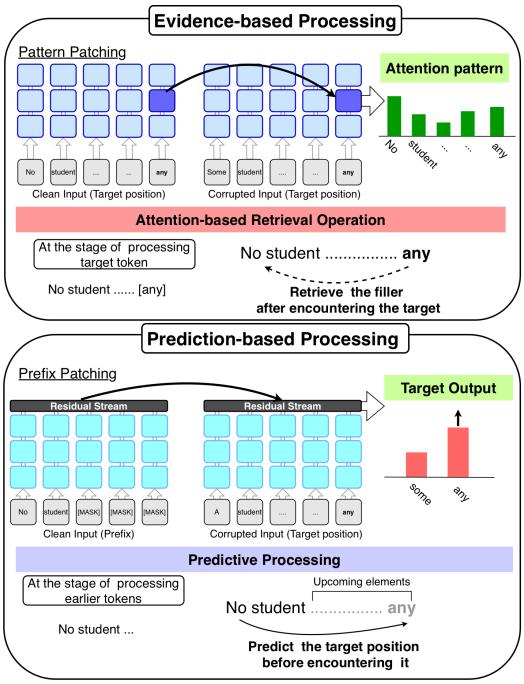

Figure 1: Overview of analytical focus. We investigate how LMs form long-distance dependencies, using two distinct patching methods: pattern patching and prefix patching. Each patching method is used to examine the two parsing strategies.

Given that, we aim to investigate how the internal mechanisms of dependency resolution in LMs are interpreted in terms of cognitively motivated strategies. This paper focuses on two well-known parsing strategies in human sentence processing: evidence-based processing (Lewis and Vasishth, 2005; Lewis et al., 2006) and prediction-based processing (Stowe, 1986; Crain and Fodor, 1985). The former is modeled as cue-based retrieval, where dependencies are resolved after encountering the gap. The latter is often instantiated as active filling strategy, where dependencies are formed before encountering the relevant elements (Frazier, 1987; Frazier and Flores d'Arcais, 1989). We adopt the view that each component in Transformer architec-

tures has a specific function that aligns with the sentence processing strategies: self-attention corresponds to processing based on retrieval (Merkx and Frank, 2021; Yoshida et al., 2025a) while residual stream corresponds to prediction-based processing (Yu and Ananiadou, 2024; Shai et al., 2024; Li et al., 2025).

To identify the processing strategies internalized in LMs, we employ *activation patching* (Nanda, 2023; Zhang and Nanda, 2024), using a distinct patching method for investigating each processing strategy: pattern patching and prefix patching. We examine the causal effect of attention pattern (pattern patching) and residual stream (prefix patching) upon long-distance dependency formation, as illustrated in Figure 1. We consider four dependency types: negative polarity items (NPI), reflexive binding, wh-dependencies, and clefts.

Since it has been argued that there exists a correlation between the model's performance and processing strategy (Timkey and Linzen, 2023; Kuribayashi et al., 2021), we analyze two GPT-2 models (small and large) to investigate how operational organization changes with scale.

Our analyses reveal that the processing strategies underlying dependency resolution are differentiated by dependency types and model sizes. Small models rely on similar attention-based operations across dependency types, suggesting limited operational specialization. In contrast, larger models exhibit differentiated operational pipelines: NPI and reflexive dependencies involve attention-based marking of structurally illicit positions, while displacement dependencies are resolved without such attention-based marking. By contrast, residual-stream-based predictive effects are robust across model sizes and appear to be selectively associated with filler-gap dependencies.

These findings indicate that LMs resolve long-distance dependencies through an interaction of two distinct processing strategies, but that the way these strategies are deployed is clearly ordered. In particular, attention-based operations appear to shift, with increasing model size, from a shared mechanism across constructions to more differentiated operations, whereas residual-stream-based operations remain comparatively stable across model sizes. This suggests that, as model size increases, the internal mechanisms instantiate linguistically meaningful distinctions, yielding an alignment with the human sentence processing strategies.[1]

2 Background

2.1 Parsing Strategies

2.1.1 Evidence-based Processing

Evidence-based processing hypothesizes that dependency resolution is based on retrieval operations. Its theoretical foundation lies in the interaction between the declarative memory and procedural memory system (Lewis and Vasishth, 2005). Under this view, the human parser begins to resolve long-distance dependencies only after encountering the relevant integration site (e.g., gap position). At that point, the parser retrieves the filler from memory with other relevant linguistic information (i.e., phrase structure, lexical semantics, formal features). Thus, the filler must be maintained in memory until the gap or other relevant position is reached. This approach has been applied to subject-verb agreement (Dillon et al., 2013), reflexive dependencies (Cunnings et al., 2014; Parker and Phillips, 2017; Jäger et al., 2020), VP-ellipsis (Parker, 2022; Cunnings and Sturt, 2018), and typical filler-gap dependencies (Saul et al., 2025).

2.1.2 Prediction-based Processing

Prediction-based processing hypothesizes that the human parser is *hypothesis-driven*. The theoretical foundation of this strategy is the interaction between the parser and grammatical principles: the parser is assumed to predict upcoming material based on grammatical information such as phrase structure, lexical semantics, and formal features (Stowe, 1986; Crocker, 1996). In other words, the human parser forms a hypothesis about upcoming strings to establish dependency as soon as possible. Active filling has been mostly associated with dependencies such as wh-questions (Stowe, 1986; Frazier and Flores d'Arcais, 1989), relative clauses (Omaki et al., 2015), parasitic gap constructions (Keshev and Meltzer-Asscher, 2017), NPI sentences (Pablos et al., 2019; Orth et al., 2021).

2.2 Dependency Formation in LMs

Along the line of mechanistic interpretability, several studies have attempted to show that LMs generalize mechanisms of filler-gap dependency resolution (Finlayson et al., 2021; Boguraev et al., 2025; Boguraev and Mahowald, 2026). Both supervised

[1] All datasets and analysis code will be released upon publication.

and unsupervised methods have shown that LMs internalize shared syntactic mechanisms to capture filler-gap dependencies (i.e., wh-dependencies, topicalization, or cleft) (Boguraev et al., 2025; Kumon and Yanaka, 2026). However, these studies do not address the actual internal procedures involved in dependency resolution.

Alternatively, several studies have shown that LM behavior in dependency formation aligns with human sentence processing strategies (Futrell et al., 2019; Kuribayashi et al., 2021; Wilcox et al., 2021). Timkey and Linzen (2023) observe that models with limited memory capacity exhibit processing error in agreement attraction, indicating the possibility that models utilize retrieval-based processing. Hanna and Mueller (2025) demonstrate that autoregressive LMs partially internalize incremental processing in ways similar to the human parser. These results indicate that LMs possibly internalize the filler-gap mechanism that aligns with the strategies in human sentence processing, depending on constructions. Since these studies do not treat a wide range of long-distance dependencies in their analyses, however, the distribution of such strategies across dependency types in LMs is not clear.

Given the discussion so far, there is no human-alignment study investigating fine-grained internal mechanisms or strategies by which long-distance dependencies are resolved. Therefore, our work lies at the intersection of mechanistic interpretability research on long-distance dependencies and work that relates human sentence-processing strategies to LLM behavior.

3 Method

3.1 Patching and Processing Strategies

Our patching experiment builds on the assumption that particular model components implement distinct modes of information processing (Davies and Khakzar, 2024). First, several studies assume that the attention mechanism functions as a retrieval operation in the psycholinguistic sense (Merkx and Frank, 2021; Yoshida et al., 2025a,b). Thus, we assume that attention-level patching identifies the causal effects of retrieval-like operations on dependency resolution.

In Experiment 1 (Section 5), we examine whether the attention patterns of high-contribution heads show convergence across constructions or across models of different sizes. If LMs resolve long-distance dependencies by means of retrieval-like operations, it is expected that the attention patterns of high-contribution heads converge in some linguistically meaningful way. Another possibility is that the attention patterns of high-contribution heads are differentiated by dependency type.

Second, there is work suggesting that the residual stream carries predictive representations, including belief-state-like information that supports future-oriented prediction (Shai et al., 2024; Li et al., 2025). We hypothesize here that the residual stream plays a predictive role in resolving linguistic dependencies as well. Experiment 2 (Section 6) examines whether residual-stream-patching, which we call prefix patching, identifies the causal effects of prediction-based processing. If LMs prefer prediction-based processing upon resolving dependencies, then it is expected that the residual streams at earlier positions make substantial contributions to prediction at the target token.

Boguraev et al. (2025) refer to the learned features as shared filler-gap mechanisms, but do not specify whether they instantiate linguistically or cognitively motivated operations. Likewise, Kumon and Yanaka (2026) identify high-contribution components as syntactic mechanisms, but do not discuss what kind of procedure those components encode. Our follow-up analyses in both experiments address this gap by characterizing what kinds of information these operations are sensitive to. In this sense, our analysis goes beyond coarse mechanism identification and instead aims to clarify the internal content of operations that previous work has treated as undifferentiated mechanisms.

3.2 Scaling

Our experiments also address the scale issue, asking whether there is a meaningful correlation between model performance and the internalized mechanisms (Kuribayashi et al., 2021; Timkey and Linzen, 2023; Mita et al., 2025). According to Kumon and Yanaka (2026), activation patching analyses reveal that the identified mechanisms are robust across models of different sizes. In particular, they focus on Pythia models (1B, 2.8B, and 6.9B) and Gemma 3 models (1B and 2B). Because their analysis is restricted to billion-parameter models, however, it remains unclear whether similar convergence of mechanisms is also observed in smaller million-parameter models. We therefore focus on smaller-scale GPT-2 models (Radford et al., 2019). We ask whether differences in model size affect the

processing strategies that models internalize.

4 Experiment Setting

4.1 Setup

Models We analyze two pretrained GPT-2 variants: GPT-2 small (12 layers and 12 heads) and GPT-2 large (36 layers and 20 heads). Although GPT-2 is relatively small by current standards, its non-trivial performance on linguistically relevant tasks suggests that it internalizes mechanisms for dependency formation (Wang et al., 2023; Wilcox et al., 2021; Warstadt et al., 2020).

Dataset The dataset for patching analysis is adapted from those in previous studies (Warstadt et al., 2020; Boguraev et al., 2025). Using predefined templates, we generate stimuli for four different types of dependency: NPI, reflexive resolution, wh-dependencies, and cleft. NPIs and reflexives instantiate typical long-distance dependencies without syntactic displacement, whereas wh-dependencies and clefts involve dependencies that require syntactic movement (i.e., filler-gap dependencies). These four constructions are therefore well suited to our purposes, since they are structurally comparable while differing in whether dependency formation involves syntactic displacement. This provides an appropriate basis for investigating both the generalization and the differentiation of mechanisms across dependency types.

4.2 Patching

We use `TransformerLens` package (Nanda, 2023) to perform the patching analyses.[2] This paper uses both *denoising* and *noising* patching (Heimersheim and Nanda, 2024). In denoising patching, activations in clean inputs are patched into the corresponding activations in the corrupted inputs. In noising patching, on the other hand, the order of patching is reversed, where activations in corrupted inputs are inserted into the ones in clean inputs. After patching a component, we evaluate the change in logit probability of patched output at the target tokens.

In the experiment, patching is applied to prediction tasks that require the model to generate a specific target token. We examine whether the next token prediction is recovered after patching each component in one input to activations at the target position in the other input.

[2] https://transformerlensorg.github.io/TransformerLens/

4.3 Basic Metrics

The attribution of each component is measured in terms of *Restoration Score* (RS).

$$R_\ell = \frac{M^{(\ell)}_{\text{patched}} - M_{\text{corr}}}{M_{\text{clean}} - M_{\text{corr}} + \varepsilon}. \quad (1)$$

It measures the degree to which the model's output in M_{corr} moves toward the clean outputs M_{clean} after patching. Positive RSs indicate the causal effects of the patched component for predicting the target token, while negative RSs indicate that the patched component suppresses the correct next-token prediction.

5 Experiment 1

Input Pairs Upon patching, we construct clean–corrupted input pairs as follows. Clean and corrupted inputs differ minimally, with the corrupted input lacking the grammatical feature necessary for dependency formation. We use 300 sentences for each dependency type. In order to assess the influence of intervening NPs on attention, all stimuli contain an object relative clause attached to the matrix subject. For example, in the NPI condition, the clean input places *no* in the matrix subject, *some* in the relative-clause subject, and *any* in the target position, where it forms a dependency with the matrix negation (see Table 1 for input pairs for all dependency types).

Role Tokens We assign a `role` label to each token based on its syntactic constituent. This abstraction is motivated by the fact that the number and type of tokens involved in forming a dependency vary across dependency types. The tokens that establish a dependency (e.g., licensors or binders) and those that intervene structurally differ between NPI, reflexive, wh-, and cleft dependencies.

In particular, tokens that are crucial for establishing dependencies are labeled as `role_key`, while dependency targets are labeled as `role_target`. All analyses in Experiment 1 are conducted over these role tokens. Examples of role assignments for each dependency type are provided in Appendix A.

Pattern Patching We perform patching at the level of head activations and attention patterns, identifying high-contribution heads by evaluating their attention patterns (Equation 2).

$$\text{pattern}_{\ell,h}(i,j) = \text{softmax}\left(\frac{Q_{\ell,h}(i) \cdot K_{\ell,h}(j)}{\sqrt{d}}\right). \quad (2)$$

A pattern of head h at a layer l is obtained by applying a softmax function to the dot product between its query and key vectors, excluding value vectors.

To characterize the content of the extracted mechanisms, we examine how attention patterns at the target position change before and after patching. We quantify attention-based effects using attention mass, defined as the sum of attention weights assigned to tokens associated with a given syntactic role at the target position.

$$\Delta\text{mass}(\text{role}) = \text{mass}_{patched} - \text{mass}_{corrupt}. \quad (3)$$

This metric measures the variation of attention mass to each token when each head is patched. For the analysis, we aggregate the attention mass of the top 10 heads ranked by RS over role tokens.

6 Experiment 2

Input Pairs In Experiment 2, clean–corrupted input pairs are constructed such that the corrupted input lacks the grammatical feature relevant to dependency formation. We use 50 sentences for each dependency type. Because the goal of this experiment is not to examine intervention effects mediated by attention, the input pairs do not have to contain relative clauses. For the same reason, we do not assign role tokens in this experiment. For example, in the wh-condition, the clean input contains a sentence-initial wh-expression, and the target position is set at the matrix verb, immediately preceding the expected gap site (for the example of the input pairs, see Table 6).

Prefix Patching We perform prefix patching, identifying prediction-based processing. The basic idea is to test whether residual-stream representations at earlier prefix positions causally support prediction at a target position. The residual-stream state at token position t in input x is defined as a d-dimensional vector, as in (4).

$$\mathbf{h}_t^{(\ell)}(x) \in \mathbb{R}^d. \quad (4)$$

This patching method replaces the residual stream at the target position t in the corrupted input with the residual stream from the source position s in the clean input.

$$\widetilde{\mathbf{h}}_j^{(\ell)}(x^{\text{corr}}; s \to t) = \begin{cases} \mathbf{h}_s^{(\ell)}(x^{\text{clean}}) & \text{if } j = t, \\ \mathbf{h}_j^{(\ell)}(x^{\text{corr}}) & \text{otherwise.} \end{cases} \quad (5)$$

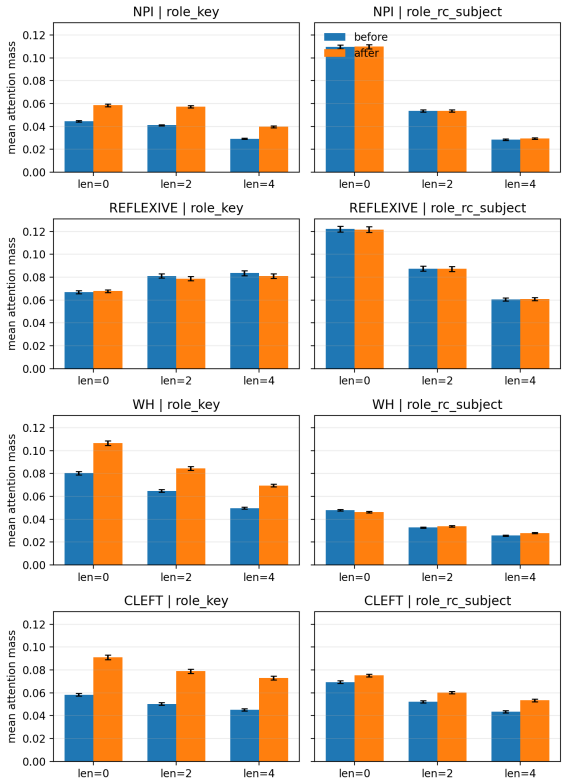

Figure 2: Effect of pattern patching on role-based attention in top-contribution heads (Top-10, GPT2-small).

This intervention tests whether predictive information encoded at an earlier prefix position in the clean input can restore target-token prediction in the corrupted input. A positive RS indicates that the patched source position carries causally relevant information for dependency formation.

7 Results

7.1 Experiment 1

7.1.1 Attention Pattern

We find that increasing model size leads to systematic changes in how different dependency types are supported by attention and other components. In GPT-2 small, attention mass was concentrated on `role_key` positions across all dependencies as shown in Figure 2 (mean attention mass: NPI, 0.057; reflexive, 0.067; Wh, 0.106; cleft, 0.090).

In GPT-2 large, however, a qualitatively different pattern emerged as in Figure 3. For NPI and reflexive dependencies, attention mass was concentrated on `role_rc_subject`, which is a structurally illicit position, with mean peak attention mass values of 0.274 (NPI) and 0.143 (reflexive). In contrast, wh- and cleft dependencies showed no such concentration, with attention mass not consistently aligned

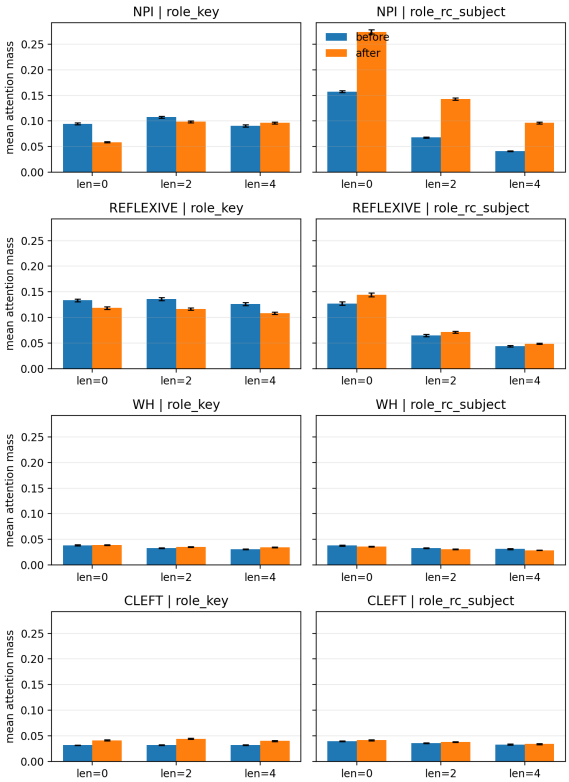

Figure 3: Effect of pattern patching on role-based attention in top-contribution heads (Top-10, GPT2-large).

with `role_key` and `role_rc_subject` tokens (approx. 0.03 for Wh and approx. 0.04 for cleft).

Taken together, these results suggest that attention-based contributions are broadly shared across dependency types in GPT-2 small, whereas GPT-2 large exhibits a clearer differentiation between displacement and non-displacement dependencies. At the same time, however, the interpretation of these patterns remains underdetermined: the broad effects in the small model may still reflect relatively shallow heuristics, and the large-model effects in NPI and reflexive dependencies do not by themselves establish that the same retrieval-like operation is involved in both constructions. We therefore turn to a follow-up analysis to clarify these possibilities.

7.1.2 Follow-up Analysis

In a follow-up analysis, we focused on NPI and reflexive dependencies. These analyses used controlled input pairs designed to isolate the source of the attention effect: for GPT-2 small, we contrasted sentence-initial NP and sentence-initial adjunct conditions (for the example of the input pairs, see Table 4), whereas for GPT-2 large we varied the structural position of the competing NP across several configurations (see Table 5).

For GPT-2 small, we tested whether the observed attention effects reflected a bias toward sentence-initial NPs, but found little evidence for such a reduction (Figures 6 and 7). For GPT-2 large, NPI and reflexive dependencies exhibited distinct attention patterns, suggesting that they may both involve retrieval-like processing while differing in the functional role of attention (Figures 8 and 9).

7.2 Experiment 2

7.2.1 Denoising Patching

Figures 4 and 5 reveal a qualitative difference across construction types but a consistent pattern across model sizes. In both models, displacement dependencies (wh, cleft) show localized positive RSs at relevant prefix positions, meaning that patching residual-stream values from those positions restores the clean condition at the target token. By contrast, non-displacement dependencies (NPI, reflexive) exhibit weaker and less systematic contribution of prefix residual streams.

These results suggest that prefix residual representations play a clearer predictive role in displacement dependencies than in non-displacement dependencies. However, denoising patching alone does not establish that the identified source positions are genuinely necessary for dependency formation, rather than merely correlated with it. To address this issue, we conduct a post hoc noising analysis focusing on the constructions that showed the clearest predictive profile.

7.2.2 Noising Patching

As a follow-up analysis, we additionally perform noising patching, in which the residual stream from the corrupted input is inserted into the clean input. We interpret noising patching as a diagnostic of necessity: source positions that support prediction in the denoising direction should impair prediction when corrupted information is inserted into the clean input (Heimersheim and Nanda, 2024). We quantify the effect of noising patching by measuring how much the patched input moves away from the clean prediction:

$$N^{(\ell)} = \frac{M_{\text{clean}} - M^{(\ell)}_{\text{patched}}}{M_{\text{clean}} - M_{\text{corr}} + \varepsilon}. \tag{6}$$

A larger positive value indicates that patching corrupted activation into the clean one degrades the clean prediction at the target position. Since the denoising analysis revealed a clear predictive profile

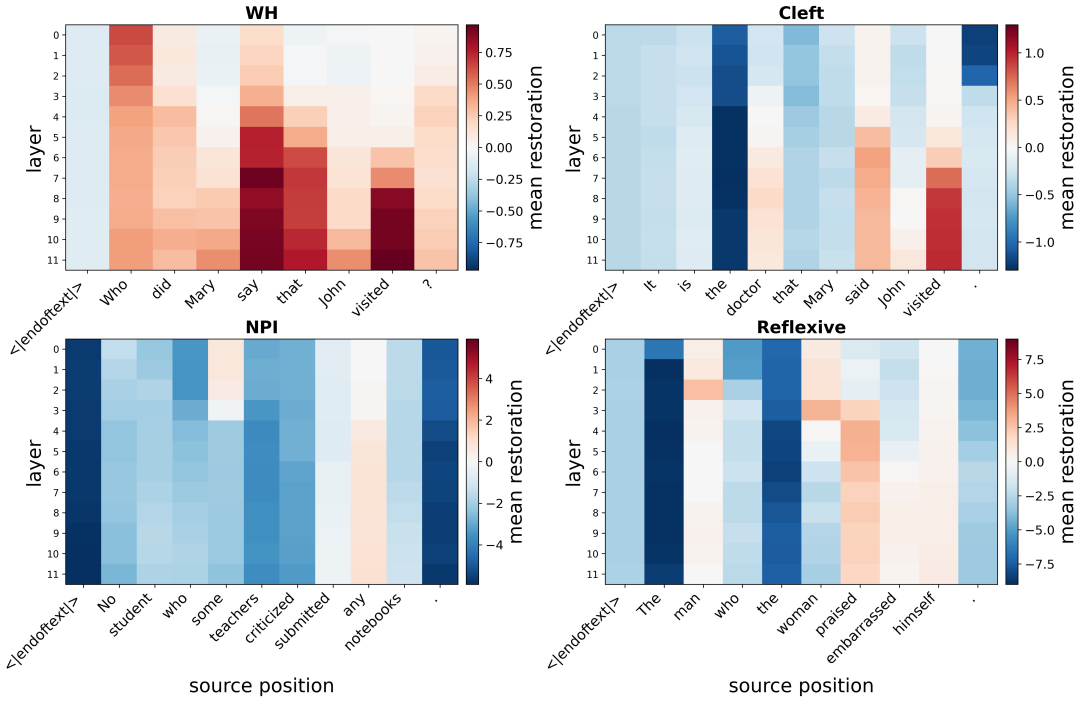

Figure 4: Layer-by-position heatmaps of mean restoration scores in prefix patching for dependency types (GPT-2-small).

only for wh-dependencies and clefts, we limit the noising analysis to these two constructions.

The results showed that injecting corrupted prefix representations systematically degraded the clean prediction in both constructions (Figures 10 and 11 in Appendix B). Importantly, this pattern was observed not only in GPT-2 small but also, in broad terms, in GPT-2 large. This supports the interpretation that the prefix positions identified in the denoising analysis carry causally relevant predictive information for filler-gap dependency formation, and that this predictive profile is relatively robust across model sizes.

8 General Discussion

Our results suggest that dependency resolution in LMs relies on distinct combinations of attention-based retrieval-like operations and residual-stream-based predictive processing. In the smaller model, attention appears to support a relatively shared operation across dependency types. Although this pattern is broad, our post hoc analyses suggest that it cannot be reduced to a trivial initial-NP bias. In the larger model, by contrast, attention becomes more differentiated: NPI and reflexive dependencies involve attention to noun phrases inside the relative clause, whereas wh-dependencies and clefts show little evidence of such attention-based marking. At the same time, NPI and reflexive dependencies diverge under post hoc manipulation, suggesting that they may share a retrieval-like strategy without instantiating the same operation (Merkx and Frank, 2021; Yoshida et al., 2025a,b).

The residual-stream results show a different pattern. Across both model sizes, prefix patching yields substantial positive restoration primarily for wh-dependencies and clefts, but not for NPIs or reflexives. This suggests that residual-stream-based predictive processing is relatively robust across model sizes and is selectively associated with filler-gap dependencies (Boguraev et al., 2025). In this sense, displacement and non-displacement dependencies differ not only in representation but also in the operations recruited for their resolution. More specifically, larger models appear to distinguish the two classes computationally: non-displacement dependencies recruit attention-based operations that track structurally competing noun phrases, whereas displacement dependencies are characterized instead by residual-stream-based predictive effects.

Taken together, these findings indicate that scaling does not simply strengthen a single mechanism. Rather, attention-based processes become more construction-specific with model size, whereas residual-stream-based predictive effects remain comparatively stable and appear to be tied to de-

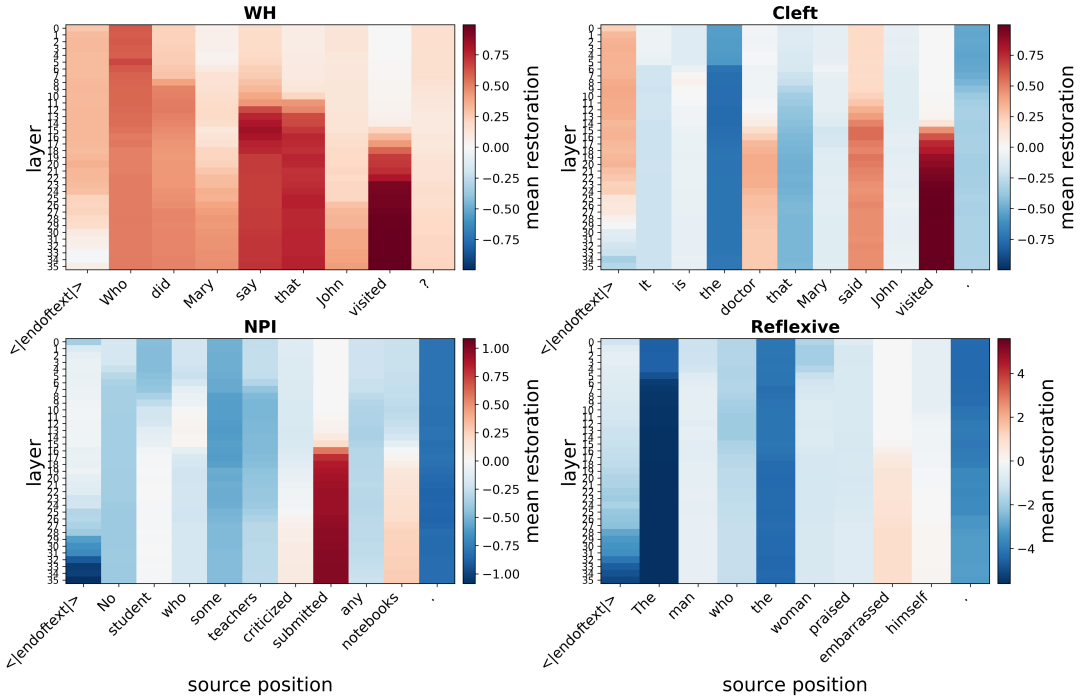

Figure 5: Layer-by-position heatmaps of mean restoration scores in prefix patching for dependency types (GPT-2-large).

pendencies involving syntactic displacement. This suggests that pretraining may lead models to distinguish displacement and non-displacement dependencies not only representationally but also operationally. Such a distinction may align with human sentence processing (Stowe, 1986; Frazier, 1987; Lewis and Vasishth, 2005).[3]

The present findings also bear on the issue of scaling. While Kumon and Yanaka (2026) suggest that the mechanisms revealed by activation patching remain qualitatively stable across model sizes, our results point to a different conclusion for attention-based operations. This suggests that mechanistic differentiation may arise within the million-parameter range represented by GPT-2 small and large, while remaining comparatively stable in larger, billion-parameter models. On this view, scaling does not simply preserve a fixed mechanism, but may involve a stage of reorganization in which attention-based operations become increasingly specific to dependency type.

This suggests that predictive processing in the residual stream reflects either an architecture-dependent property of Transformer models or a capacity that can be acquired relatively easily through pretraining even at smaller scales (Omaki et al., 2015; Atkinson et al., 2018). In this sense, the present study provides a new perspective on previous work on model size and grammatical competence by suggesting that scaling may affect different components of dependency resolution in different ways, rather than strengthening a shared mechanism (Kuribayashi et al., 2021; Timkey and Linzen, 2023; Kuribayashi et al., 2025).

9 Conclusion

This paper investigates how different syntactic dependencies are implemented in transformer LMs at the level of internal computation. Using activation patching, we show that the internal mechanisms of dependency formation in LMs are guided by processing strategies that are linguistically and cognitively convergent. However, the degree of generalization and the construction-specific realization differ across model sizes, at least in the million-parameter range. These findings indicate that dependency resolution in LMs is convergent but becomes increasingly differentiated with scale. More generally, one implication of the present causal intervention analysis is that the two cognitive processing strategies may constitute *linguistic theory* in the sense of Futrell and Mahowald (2025), as they accommodate aspects of the linguistic behaviors of both humans and LMs.

[3]This does not imply that language models implement the same algorithms as human parsers.

Limitations

We restrict our investigation to four dependency types: NPI, reflexive binding, wh-dependencies, and cleft sentences. While these dependencies differ along several theoretically relevant dimensions, they do not exhaust the space of syntactic dependencies. In particular, dependencies such as quantifier-pronoun binding or scope interactions may reveal additional modes of operational specialization, and testing these cases constitutes an important direction for future work.

In addition, our scaling analysis is limited to GPT-2 small and GPT-2 large, and does not extend to billion-parameter models such as GPT-2 XL. It therefore remains unclear whether the distribution of operations observed in GPT-2 large would continue to change at larger scales or instead remain comparatively stable. A further limitation concerns the relation to BabyLM-scale training. Chang et al. (2025) suggest that small-data models acquire filler-gap dependencies only partially and with limited generalization. Since the present study examines million-parameter GPT-2 models trained under standard conditions, it remains to be determined whether comparable mechanisms emerge under BabyLM-scale input, or whether such models rely on heuristic procedures, as GPT-2 small appears to do (Experiment 1).

References

Kabir Ahuja, Vidhisha Balachandran, Madhur Panwar, Tianxing He, Noah A. Smith, Navin Goyal, and Yulia Tsvetkov. 2025. Learning syntax without planting trees: Understanding hierarchical generalization in transformers. *Transactions of the Association for Computational Linguistics*, 13:121–141.

Emily Atkinson, Matthew W. Wagers, Jeffrey Lidz, Colin Phillips, and Akira Omaki. 2018. Developing incrementality in filler-gap dependency processing. *Cognition*, 179:132–149.

Sasha Boguraev and Kyle Mahowald. 2026. Causal drawbridges: Characterizing gradient blocking of syntactic islands in transformer lms. *Preprint*, arXiv:2604.13950.

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 15049–15065, Suzhou, China. Association for Computational Linguistics.

Stephen Crain and Janet Dean Fodor. 1985. Rules and constraints in sentence processing. In David R. Dowty, Lauri Karttunen, and Arnold M. Zwicky, editors, *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, pages 38–52. Cambridge University Press, Cambridge.

Matthew W. Crocker. 1996. *Computational Psycholinguistics: An Interdisciplinary Approach to the Study of Language*, volume 20 of *Studies in Theoretical Psycholinguistics*. Kluwer Academic Publishers, Dordrecht; Boston.

Ian Cunnings, Clare Patterson, and Claudia Felser. 2014. Variable binding and coreference in sentence comprehension: Evidence from eye movements. *Journal of Memory and Language*, 71:39–56.

Ian Cunnings and Patrick Sturt. 2018. Retrieval interference and semantic interpretation. *Journal of Memory and Language*, 102:16–27.

Adam Davies and Ashkan Khakzar. 2024. The cognitive revolution in interpretability: From explaining behavior to interpreting representations and algorithms. *Preprint*, arXiv:2408.05859.

Brian Dillon, Alan Mishler, Shayne Sloggett, and Colin Phillips. 2013. Contrasting intrusion profiles for agreement and anaphora: Experimental and modeling evidence. *Journal of Memory and Language*, 69(2):85–103.

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.

Lyn Frazier. 1987. Sentence processing: A tutorial review. In Max Coltheart, editor, *Attention and Performance XII: The Psychology of Reading*, pages 559–586. Lawrence Erlbaum Associates, Hillsdale, NJ.

Lyn Frazier and Giovanni B. Flores d'Arcais. 1989. Filler driven parsing: A study of gap filling in dutch. *Journal of Memory and Language*, 28(3):331–344.

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

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Michael Hanna and Aaron Mueller. 2025. Incremental sentence processing mechanisms in autoregressive transformer 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 3181–3203, Albuquerque, New Mexico. Association for Computational Linguistics.

Stefan Heimersheim and Neel Nanda. 2024. How to use and interpret activation patching. *Preprint*, arXiv:2404.15255.

Lena A. Jäger, Daniela Mertzen, Julie A. Van Dyke, and Shravan Vasishth. 2020. Interference patterns in subject-verb agreement and reflexives revisited: A large-sample study. *Journal of Memory and Language*, 111:104063.

Maayan Keshev and Aya Meltzer-Asscher. 2017. Active dependency formation in islands: How grammatical resumption affects sentence processing. *Language*, 93(3):549–568.

Ryoma Kumon and Hitomi Yanaka. 2026. Fine-grained analysis of shared syntactic mechanisms in language models. *Preprint*, arXiv:2604.22166.

Tatsuki Kuribayashi, Yohei Oseki, Takumi Ito, Ryo Yoshida, Masayuki Asahara, and Kentaro Inui. 2021. Lower perplexity is not always human-like. 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 5203–5217, Online. Association for Computational Linguistics.

Tatsuki Kuribayashi, Yohei Oseki, Souhaib Ben Taieb, Kentaro Inui, and Timothy Baldwin. 2025. Large language models are human-like internally. *Transactions of the Association for Computational Linguistics*, 13:1743–1766.

Richard L. Lewis and Shravan Vasishth. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29(3):375–419.

Richard L. Lewis, Shravan Vasishth, and Julie A. Van Dyke. 2006. Computational principles of working memory in sentence comprehension. *Trends in Cognitive Sciences*, 10(10):447–454.

Belinda Z. Li, Zifan Carl Guo, and Jacob Andreas. 2025. (how) do language models track state? *Preprint*, arXiv:2503.02854.

Tal Linzen and Marco Baroni. 2021. Syntactic structure from deep learning. *Annual Review of Linguistics*, 7(1):195–212.

Danny Merkx and Stefan L. Frank. 2021. Human sentence processing: Recurrence or attention? In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, page 12–22. Association for Computational Linguistics.

Masato Mita, Ryo Yoshida, and Yohei Oseki. 2025. Developmentally-plausible working memory shapes a critical period for language acquisition. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9386–9399, Vienna, Austria. Association for Computational Linguistics.

Neel Nanda. 2023. Attribution patching: Activation patching at industrial scale.

Akira Omaki, Ellen Lau, Imogen Davidson White, Myles L. Dakan, Aaron Apple, and Colin Phillips. 2015. Hyper-active gap filling. *Frontiers in Psychology*, 6:384.

Wesley Orth, Masaya Yoshida, and Shayne Sloggett. 2021. Negative polarity item (NPI) illusion is a quantification phenomenon. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 47(6):906–947.

Robles-Leticia Pablos, Jenny Doetjes, Brigit Ruijgrok, and Lisa Lai-Shen Cheng. 2019. Backward licensing of negative polarity items in dutch: An erp investigation. *Journal of Neurolinguistics*, 51:96–115.

Dan Parker. 2022. Ellipsis interference revisited: New evidence for feature markedness effects in retrieval. *Journal of Memory and Language*, 124:104314.

Dan Parker and Colin Phillips. 2017. Reflexive attraction in comprehension is selective. *Journal of Memory and Language*, 94:272–290.

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

Niki Saul, Maayan Keshev, and Aya Meltzer-Asscher. 2025. Interference in the formation of filler-gap dependencies: Evidence from hebrew relative clauses. *Journal of Memory and Language*, 143:104626.

Adam S. Shai, Sarah E. Marzen, Lucas Teixeira, Alexander Gietelink Oldenziel, and Paul M. Riechers. 2024. Transformers represent belief state geometry in their residual stream. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Laurie A. Stowe. 1986. Parsing WH-constructions: Evidence for on-line gap location. *Language and Cognitive Processes*, 1(3):227–245.

William Timkey and Tal Linzen. 2023. A language model with limited memory capacity captures interference in human sentence processing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8705–8720, Singapore. Association for Computational Linguistics.

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

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

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, Pranali Vani, and Roger Levy. 2021. A targeted assessment of incremental processing in neural language models and humans. 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 939–952, Online. Association for Computational Linguistics.

Ryo Yoshida, Shinnosuke Isono, Kohei Kajikawa, Taiga Someya, Yushi Sugimoto, and Yohei Oseki. 2025a. If attention serves as a cognitive model of human memory retrieval, what is the plausible memory representation? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9795–9812, Vienna, Austria. Association for Computational Linguistics.

Ryo Yoshida, Yushi Sugimoto, and Yohei Oseki. 2025b. Investigating psychometric predictive power of syntactic attention. In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 407–418, Vienna, Austria. Association for Computational Linguistics.

Zeping Yu and Sophia Ananiadou. 2024. Neuron-level knowledge attribution in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3267–3280, Miami, Florida, USA. Association for Computational Linguistics.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*.

A Experiment 1

Role Tokens To analyze how attention is distributed over linguistically relevant positions, we annotate each stimulus with role tokens corresponding to syntactic and semantic functions within the dependency. The `role_key` corresponds to the element that determines the well-formedness of the dependency (e.g., the NPI licensor or the antecedent of a reflexive or displaced element).

Follow-up Analysis In a post hoc analysis for Experiment 1, we further examined NPI and reflexive dependencies using controlled input-pair manipulations. For GPT-2 small, we constructed sentence-initial NP and sentence-initial adjunct conditions to test whether the observed attention effects reduced to a trivial initial-NP bias. For GPT-2 large, we compared three structural families for each dependency type in order to assess whether the attention profile was stable across intervention configurations. The results suggest that the small-model effect is not reducible to a simple initial-NP bias, while NPI and reflexive dependencies in the large model exhibit distinct attention profiles. We also found that NPI patterns were unstable under fixed top-k head selection, indicating that threshold-based or proportion-based criteria may be more appropriate.

B Experiment 2

Experiment 2 provides concrete examples of the clean–corrupted input pairs and the results of the follow-up noising analysis. Table 6 gives representative input pairs for each dependency type, and Figures 10 and 11 show the noising-patching results for wh-dependencies and clefts in GPT-2 small and GPT-2 large.

	NPI
Clean	**No** student who some teachers scolded [any] notebook.
Corrupted	**Some** student who some teachers scolded [any] notebook.
	Reflexive
Clean	**The man** who the woman predominantly dislikes admired [himself] yesterday.
Corrupted	**The woman** who the woman predominantly dislikes admired [himself] yesterday.
	Wh
Clean	**Which man** did the woman who wrote the paper [visited] yesterday ?
Corrupted	**Which paper** did the woman who praised the man [visited] yesterday ?
	Cleft
Clean	It is **the man** that the woman who wrote the paper [visited] yesterday.
Corrupted	It is **the paper** that the woman who praised the man [visited] yesterday.

Table 1: Pairs of clean-corrupted input per dependency types in Experiment 1

	$role_{key}$	$role_{rel}$	$role_{rel-subj}$	$role_{rel-verb}$	$role_{adj}$	$role_{main-verb}$	$role_{target}$
NPI	[No NP]	who	[some NP]	[verb]	[adverb]	[verb]	[any]
Reflexive	[The NP_m]	who	[the NP_f]	[verb]	[adverb]	[verb]	[himself]

Table 2: Examples of role assignment for non-displacement dependency types: NP$_m$ and NP$_f$ describe male nouns (e.g., "son", "boy") and female nouns ("daughter", "girl"), respectively.

	$role_{key}$	$role_{pro}$	$role_{subj}$	$role_{rel}$	$role_{rel-subj}$	$role_{rel-verb}$	$role_{target}$
Wh-dependency	[Wh NP_a]	did	[the NP_i]	which	[verb]	[adv]	[verb]
Cleft	[It was the NP_a]	who	[the NP_i]	[which]	[verb]	[adv]	[verb]

Table 3: Examples of role assignment for displacement dependency types: NP$_a$ and NP$_i$ describe animate and inanimate nouns, respectively.

	Condition	Initial element / competitor	Clean key element	Corrupted key element
NPI	NP-initial	The report says that	no	some
NPI	Adjunct-initial	Yesterday,	no	some
Reflexive	NP-initial	The report says that	the man	the woman
Reflexive	Adjunct-initial	Yesterday,	the man	the woman

Table 4: Summary of the controlled input pairs used in the post hoc analysis for GPT-2 small.

	Condition	Structural competitor	Clean key element	Corrupted key element
NPI	RC-subject	relative-clause subject	no	some
NPI	RC-object	relative-clause object	no	some
NPI	PP	PP noun phrase	no	some
NPI	Post-verbal PP	post-verbal PP noun phrase	no	some
Reflexive	RC-subject	relative-clause subject	the man	the woman
Reflexive	RC-object	relative-clause object	the man	the woman
Reflexive	PP	PP noun phrase	the man	the woman
Reflexive	Post-verbal PP	post-verbal PP noun phrase	the man	the woman

Table 5: Summary of the controlled input pairs used in the post hoc analysis for GPT-2 large. The structural competitor was varied across conditions in order to test whether the observed attention effects remained stable across different intervention configurations.

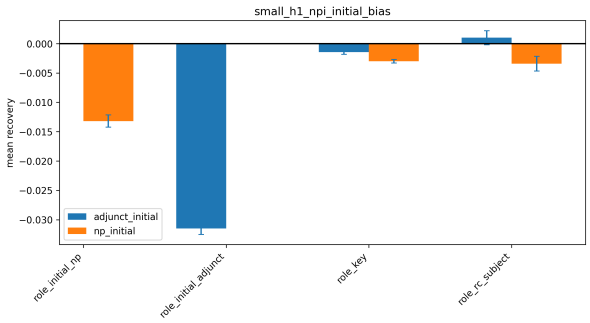

Figure 6: Follow-up control for GPT-2 small: NPI.

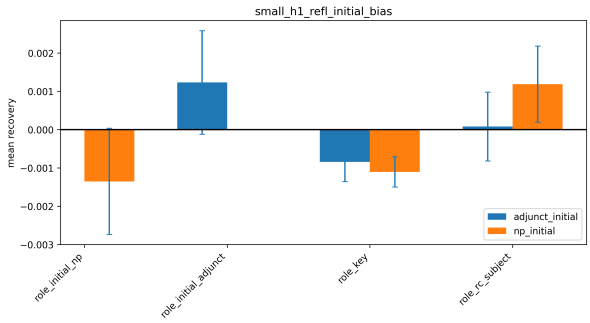

Figure 7: Follow-up control for GPT-2 small: Reflexive.

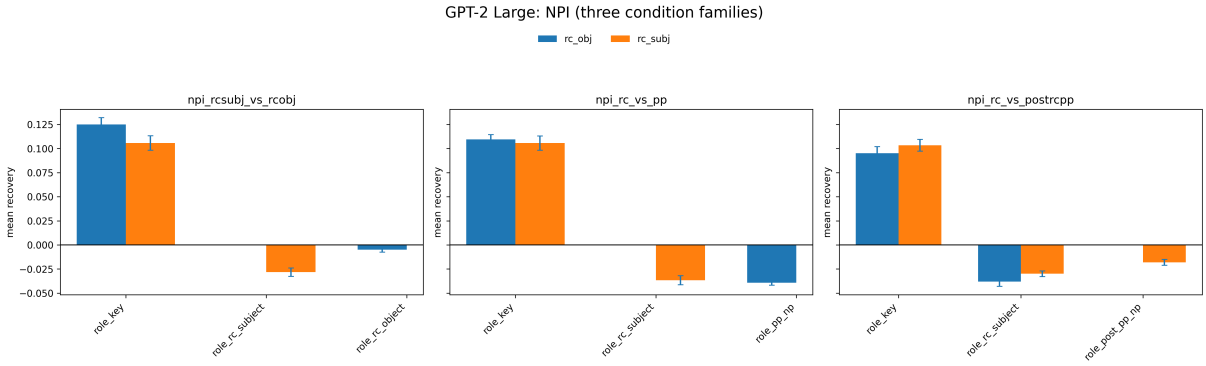

Figure 8: Follow-up analysis for Experiment 1 in GPT-2 large: NPI. The three panels compare controlled condition families (*rc_subj* vs. *rc_obj*, *rc* vs. *pp*, and *rc* vs. *postrcpp*) in terms of mean attention recovery over role tokens. The pattern indicates that the attention effect in NPI dependencies is sensitive to the structural configuration of the competitor.

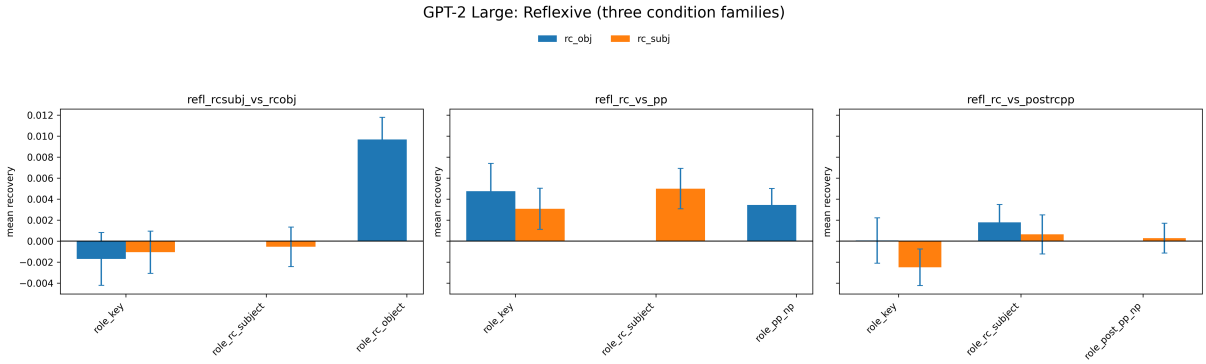

Figure 9: Follow-up analysis for Experiment 1 in GPT-2 large: Reflexive. The three panels compare controlled condition families (*rc_subj* vs. *rc_obj*, *rc* vs. *pp*, and *rc* vs. *postrcpp*) in terms of mean attention recovery over role tokens. The resulting profile differs from that of NPI, suggesting that although both constructions may involve retrieval-like processing, the underlying attention-based operations are not identical.

	Condition	Structural manipulation	Clean example	Corrupted example
WH	filler-gap	sentence-initial wh vs. overt object	Who did Mary say that John visited ?	Did Mary say that John visited the man ?
Cleft	cleft dependency	cleft pivot vs. overt object	It was the man that Mary said John visited .	Mary said that John visited the man .
NPI	licensing	matrix licensor	No student who some teachers scolded submitted any notebooks .	Some student who some teachers scolded submitted any notebooks .
Reflexive	binding	matrix antecedent gender	The man who teachers criticized embarrassed himself .	The woman who teachers criticized embarrassed himself .

Table 6: Concrete examples of the clean–corrupted input pairs used in Experiment 2. For WH and cleft dependencies, the clean input contains a displacement dependency, whereas the corrupted input replaces it with an overt object. For NPI and reflexive dependencies, the clean and corrupted inputs differ in the grammatical feature required for licensing or binding, while preserving the relative-clause structure.

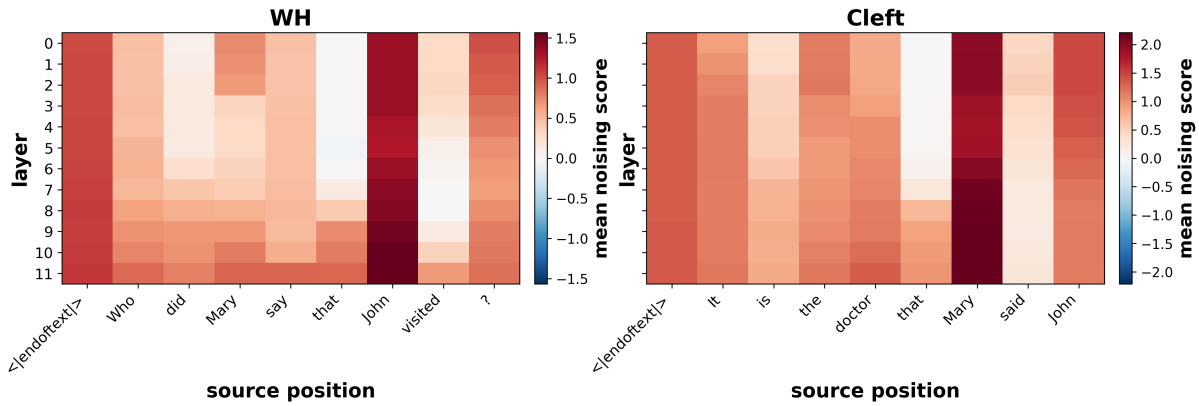

Figure 10: Layer-by-position heatmaps of mean noising scores for wh-dependencies and clefts in GPT-2 small.

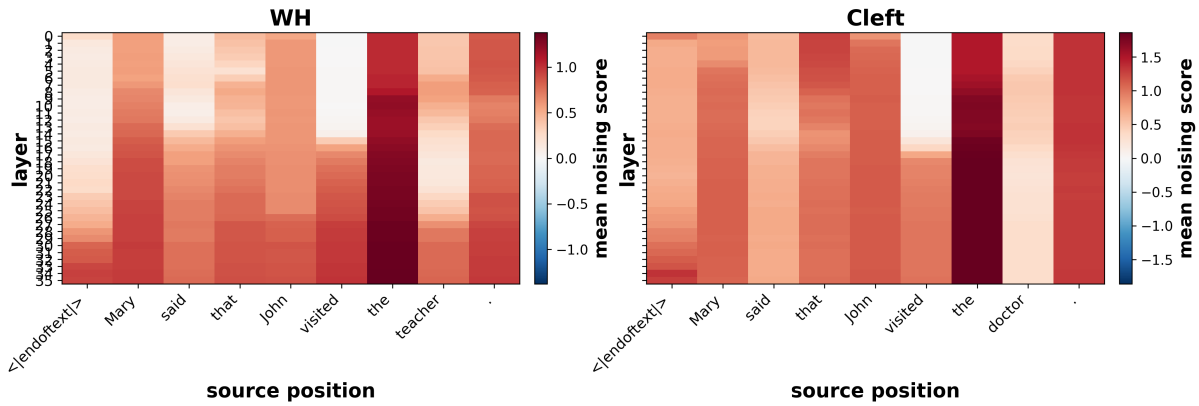

Figure 11: Layer-by-position heatmaps of mean noising scores for wh-dependencies and clefts in GPT-2 large.