Causal Interventions Reveal Shared Structure Across English Filler–Gap Constructions

Sasha Boguraev[1] **Christopher Potts**[2] **Kyle Mahowald**[1]
[1]The University of Texas at Austin [2]Stanford University
{sasha.boguraev,kyle}@utexas.edu cgpotts@stanford.edu

Abstract

Large Language Models (LLMs) have emerged as powerful sources of evidence for linguists seeking to develop theories of syntax. In this paper, we argue that causal interpretability methods, applied to LLMs, can greatly enhance the value of such evidence by helping us characterize the abstract mechanisms that LLMs learn to use. Our empirical focus is a set of English filler–gap dependency constructions (e.g., questions, relative clauses). Linguistic theories largely agree that these constructions share many properties. Using experiments based in Distributed Interchange Interventions, we show that LLMs converge on similar abstract analyses of these constructions. These analyses also reveal previously overlooked factors – relating to frequency, filler type, and surrounding context – that could motivate changes to standard linguistic theory. Overall, these results suggest that mechanistic, internal analyses of LLMs can push linguistic theory forward.

1 Introduction

Language models can generate and process utterances typically thought to require rich linguistic grammatical structure (Futrell et al., 2019; Wilcox et al., 2018; Manning et al., 2020; Hu et al., 2020), including much-studied syntactic constructions like long-distance filler–gap constructions (Wilcox et al., 2024). These results have been taken to challenge claims that these phenomena can be learned only with strong innate priors (Piantadosi, 2023; Futrell and Mahowald, 2025).

Despite the strong performance, questions remain as to whether models acquire syntax in ways that are posited by linguists to be human-like (e.g., acquiring rich grammatical abstraction and syntactic structure). Causal interpretability methods now make it possible to characterize the abstract mechanisms underlying neural networks (Vig et al., 2020; Finlayson et al., 2021; Geiger et al., 2021; Meng

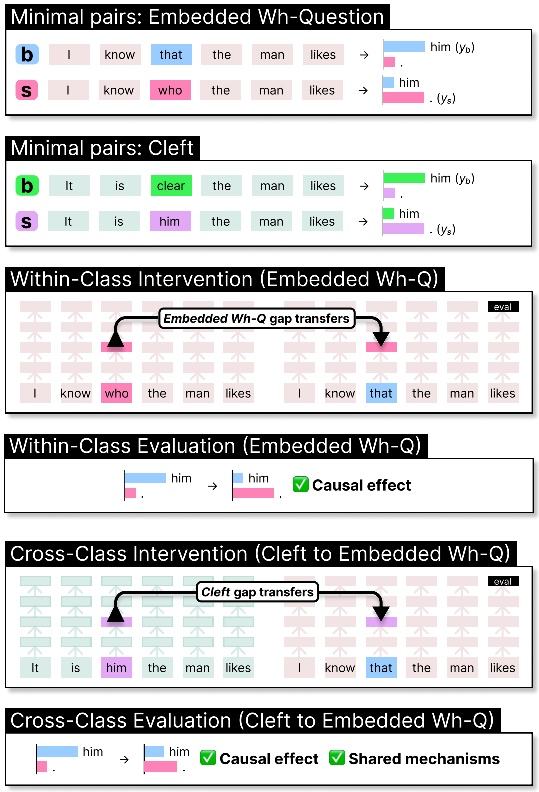


Figure 1: **Causal intervention overview.** Here, we illustrate our methodology when we intervene within a class, transferring an embedded wh- filler–gap structure into a corresponding minimal pair that didn't previously have one. We then show intervening across classes, inserting a wh- filler–gap into a gap-less cleft sentence

et al., 2022; Geiger et al., 2023; Wang et al., 2023). These methods have revealed non-trivial linguistic syntactic structure is learned by models (Arora et al., 2024; Lasri et al., 2022; Finlayson et al., 2021; Mueller et al., 2022; Lakretz et al., 2019). But a key hypothesis in the history of linguistics is that seemingly different linguistic constructions can share underlying structure. For instance, compare "I wonder what the lion ate." to "It was the gazelle that the lion ate." The former is an embedded wh- clause and the latter is a cleft construction.

These are distinct constructions but share something in common: both have a long-distance dependency with an extracted element, often specified with a linguistic trace: "I wonder what$_t$ the lion ate ___$_t$." and "It was the gazelle$_t$ that the lion ate ___$_t$." Thus, many linguistic theories predict common processing characteristics between these sentences (Fodor, 1989). On the other hand, there is also reason to expect wh- sentences to be quite different from clefts since both wh- elements and clefts have idiosyncratic properties (Ross, 1967; Culicover, 1999; Ginzburg and Sag, 2001).

To tackle these questions, we take advantage of advances in large open source models as well as advances in mechanistic interpretability, specifically the Causal Abstraction framework (Geiger et al., 2023) and Distributed Alignment Search (DAS; Geiger et al. 2024). Our resulting methodology gives us direct access to the abstract causal mechanisms learned by these models. By accessing these causal mechanisms, we can take a filler-gap mechanism learned on Construction A (e.g., wh- sentences), transfer it to Construction B (e.g., clefts), and see if we get predictable filler–gap behavior (see Figure 1). If we do, this would be strong evidence of underlying shared structure learned by the model.

Importantly, this method gives us a gradient measure of transfer. As such, we explore whether more similar constructions transfer more readily to each other; whether some constructions in general tend to serve as sources of transfer; whether mechanisms transfer across clauses; and whether transfer is greater when lexical items are shared across constructions (an effect predicted by the "lexical boost" in syntactic priming, whereby syntactic structures are primed more strongly when there is lexical overlap Pickering and Branigan, 1998).

Ultimately, we find strong generalization in LMs across a range of filler–gap constructions, with effects observed at all positions within constructions. We observe lexical boost: effects are stronger when lexical items match (e.g., the same animacy). Moreover, we identify *source* constructions whose underlying mechanisms generalize broadly, as well as *sink* constructions that consistently benefit from such transferred mechanisms. Finally, we provide evidence that such generalization does not seem to extend across clausal boundaries.

We claim these experiments make good on the promise that studying LMs can help us better understand linguistic structure and language learning in general by not just serving as proxies for data-driven learners, but by helping us develop linguistically interesting hypotheses (Futrell and Mahowald, 2025; Portelance and Jasbi, 2024; Potts, 2023).[1]

2 Filler–Gaps and Neural Models

Consider the following sentence:

(1) [The bagel]$_t$, I liked __$_t$.

The embedded clause, *I liked*, seems incomplete, lacking an object. However, the sentence is grammatical, as the fronted entity *the bagel* is understood to be the object of the anteceding clause.

Grammatical constructions of this nature are termed **filler–gaps**, due to constituents appearing as 'fillers' in non-canonical positions, colloquially being said to leave a 'gap' at its canonical position. This grammatical family encompasses a wide range of common constructions including *wh*-questions, relative clauses, clefts, and more.

Filler–gap dependencies have long been a target of linguistic inquiry. They are believed to require sophisticated syntactic machinery, beyond simple surface statistics, since a word might appear quite linearly far from a word that it depends on for its meaning (Chomsky, 1957; Ross, 1967). They have been of interest in computational linguistics for the same reason: earlier models like n-gram models were fundamentally unable to handle structures over long distances.

Hence, filler–gaps have served as a common testbed for LMs' grammatical capacities. Wilcox et al. (2018) provided early positive evidence of RNN's grammatical competence in English by comparing LMs' surprisals for gap and gapless continuations in the presence and absence of fillers. More recently Ozaki et al. (2022) and Wilcox et al. (2024) have demonstrated LM sensitivities to linguistic constraints on these constructions. Kobzeva et al. (2023) found mixed results in Norwegian, a language known to have very different filler–gap structures and constraints than English.

There has been further work to measure the generalization capacities of LMs across filler–gap constructions. Lan et al. (2024) test models' knowledge of parasitic gaps and across-the-board movement, finding that unless the training data is supplemented with adequate examples, LMs struggle to learn these constructions from small corpora. Howitt et al. (2024) build on the methodology of

[1] Code will be released with final version.

Construction	Prefix	Filler	NC	Article	NP	Verb	Label
Emb. Wh-Question (*Know*-Class)	I know	who/that		the	man	liked	./him
Emb. Wh-Question (*Wonder*-Class)	I wonder	who/if		the	man	liked	./him
Matrix Wh-Question		Who/""	did	the	man	like	?/him
Restrictive Relative Clause	The boy	who/and		the	man	liked	was/him
Cleft	It was	the boy/clear	that	the	man	liked	./the boy
Pseudo-Cleft		Who/That		the	man	liked	was/it
Topicalization	Actually,	the boy/""		the	man	liked	./the boy
Subject-Verb Agreement	The	boy/boys	that	the	man	liked	is/are
Transitive/Intransitive Verbs		Last night/Yesterday		some/that	man/boy	ran/liked	./him

Table 1: **Exemplar minimal pairs** for each evaluated construction's single-clause, animate extraction variant. The filler/label combinations are used to evaluate whether the model is processing the construction correctly and whether our causal interventions are successful. The final two constructions are used as control conditions. For a full set of examples, including the multi-clause and inanimate extraction variants, see Appendix A.

Lan et al. (2024), training LSTMs on specific filler–gap constructions and evaluating LM performance on others, with results suggesting little generalization in LMs. Prasad et al. (2019) and Bhattacharya and van Schijndel (2020) further use a methodology based on psycholinguistic priming to explore filler–gap generalization in LMs, with the former finding evidence suggesting that LMs hierarchically organize relative clauses in representation space, and the latter finding general representations for filler–gaps which are shared across various constructions.

These previous works show LMs can learn to process filler–gap constructions, but show more mixed results as to whether this processing is shared across constructions. But most of this work has been behavioral, without exploring the model's underlying causal mechanisms. Our work fills this gap. We first uncover the causal mechanisms LMs learn to process various filler–gap dependencies, and then we measure to what extent these mechanisms generalize across different filler–gaps.

3 Methods

3.1 Data

Evaluated Constructions We focus our investigation on seven filler–gap constructions: embedded wh-questions with a finite complementizer (denoted as the *know*-class), embedded wh-questions with a non-finite complementizer (*wonder*-class), matrix-level wh-questions, restrictive relative clauses, clefts, pseudo-clefts, and topicalization. For each construction, we design sentential templates in the style of Arora et al. (2024), allowing us to sample a large number of minimal pairs differing in our targeted grammatical phenomenon.

We design four templates per construction, differing in the extracted object's animacy and by the number of clausal boundaries between the filler and the gap left by its extraction (one or two clauses). We manipulated animacy since changing animacy requires changing the key wh- element ("who" vs. "what"), but is not hypothesized to affect the sentence's structure. All our templates involve the extraction of a direct object from a verb phrase and all follow a general template, allowing cross-construction alignment by position. Our general template, as well as examples of animate extraction from a single-clause variant of each construction, are in Table 1.

Controls Our first control is the task of subject–verb number agreement (e.g., "The boy is", not "The boy are"). This task was selected because, relative to our constructions of interest, there is a similar distance between the subject and the verb. However, while subject–verb agreement can operate over long linear distances, it does not have the filler–gap property of our target constructions (as agreement is always between clausemate elements) and thus we hypothesize that it should *not* rely on the same mechanism.

The second control is the task of predicting a continuation after transitive or intransitive verbs. This task controls for the predicted label, ensuring that any generalization we find is meaningful, not merely due to heuristics related to the predicted labels. In order to maintain the distance between minimal contrast and prediction location, we have lexical items in faux-contrast at the FILLER, ARTICLE, NP positions, such that there is no meaningful difference in the sampled items at those positions.

3.2 Distributed Alignment Search

To localize internal mechanisms used by LMs to process our constructions of interest, we use Distributed Alignment Search (DAS; Geiger et al. 2024; Wu et al. 2023). DAS is a supervised interpretability method that can be used to assess whether a given feature is encoded in a particular set of neural activations. We rely on the 1-dimensional variant of DAS used by Arora et al. (2024). The core intervention performed is

$$\mathbf{b} + (\mathbf{s}\mathbf{a}^\top - \mathbf{b}\mathbf{a}^\top)\mathbf{a}$$

where $\mathbf{b} \in \mathbb{R}^n$ is a representation formed by the model when it processes a base example (right sides in Figure 1), and $\mathbf{s} \in \mathbb{R}^n$ is the corresponding representation formed when the model processes a source example (left sides in Figure 1). In our experiments, $\mathbf{b}$ and $\mathbf{s}$ are always the outputs of a Transformer block. Intuitively, this intervention defines a direction in the rotated feature space defined by the learned vector $\mathbf{a} \in \mathbb{R}^n$. This is a soft intervention targeting only the learned feature and preserving orthogonal dimensions of $\mathbf{b}$. In DAS, all LM parameters are kept frozen, and $\mathbf{a}$ is learned via a standard cross-entropy loss trained on interventions of the sort depicted in Figure 1. The goal of learning is to make the correct predictions under the intervention. For example, in the within-class intervention in Figure 1, we seek to learn an intervention that predicts a gap site (signaled by a period) even though the inputs correspond to a non-filler–gap case. The extent to which we can learn such an intervention provides the basis for assessing the hypothesis that the filler–gap dependency itself can be localized to the intervention site.

We chose to use DAS for two main reasons. First, Arora et al. (2024) demonstrate that, in a comparison among several interpretability methods, DAS consistently performed the best in finding causally efficacious features in syntactic tasks. Second, Wu et al. (2023) show that the feature-alignments learned by DAS are robust and generalize strongly.

Training We train interventions at each position from the FILLER onwards, and across every layer of our given LM. We use the `pythia` series of models (Biderman et al., 2023), a series of open-source, open-data LLMs. We run our experiments on the 1.4, 2.8 and 6.9 billion parameter models. We find qualitatively similar results for all sizes, reporting those of the 1.4b variant in the main text (results for 2.8b and 6.9b variants in Appendix H).

We evaluate two distinct categories of interventions: (1) single-source interventions, where for each of the n constructions, $c_{i<n}$, the training dataset for DAS contains sentences sampled from the templates of c_i, and (2) leave-one-out construction interventions, where for each of the n constructions, $c_{i<n}$, the training dataset contains sentences sampled from the templates of $c_{j\neq i}$ – that is, all constructions that are not 'left-out'.

Evaluation For evaluation, we use the **ODDS** metric from Arora et al. (2024). This metric measures how much more likely a counterfactual label is after performing an intervention, with higher **ODDS** denoting larger causal effect from the given intervention. Intuitively, it tells us: after intervention, how much more likely is the continuation expected based on the "source sentence" than the one naively expected based on the "base sentence". We measure the average **ODDS** at each position-layer pair across 400 sentences, sampled from the templates of each individual construction.

In cases of aggregation, we max-pool the average **ODDS** value across layers at each sentential position (we refer to this metric as **MAX ODDS** hereafter). We also normalize the **MAX ODDS** by the corresponding average **MAX ODDS** for the items present in the training set, with this normalization giving us a measure of how much the mechanisms used by a given set of constructions generalize to an evaluated construction, relative to how much they generalize to those they were trained on. We aggregate across layers by max-pooling **ODDS** because our causal methodology aims to localize syntactic features in the model, with the maximum value representing the most causally efficacious localization of the given features.

4 Exp. 1: Do LMs Share Filler–Gap Mechanisms Across Constructions?

Our first experiment investigates the extent to which language models employ common mechanisms for processing different filler–gaps.

Setup We measure the **MAX ODDS** for all trained interventions evaluated on every construction of the same clausal category (for a discussion on cross-clause generalization see §6). We then group these values into six categories, depending on the relation between the set of constructions the interventions were trained on and those used to generate the evaluation set. These groups comprise (1) the same set of constructions in the training set and the evaluation set, with the same animacy – this is our reference group as training and evaluation sentences are drawn from the same distribution; (2)

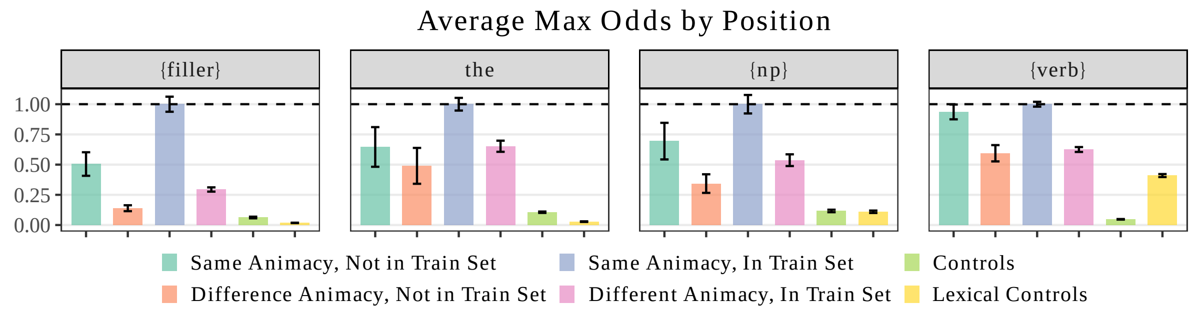


Figure 2: Average normalized **MAX ODDS** across positions, ±1 standard error. Corresponding multi-clause plots can be found in Appendix E. Note that normalization fixes the “Same Animacy, In Train Set” condition at 1.00.

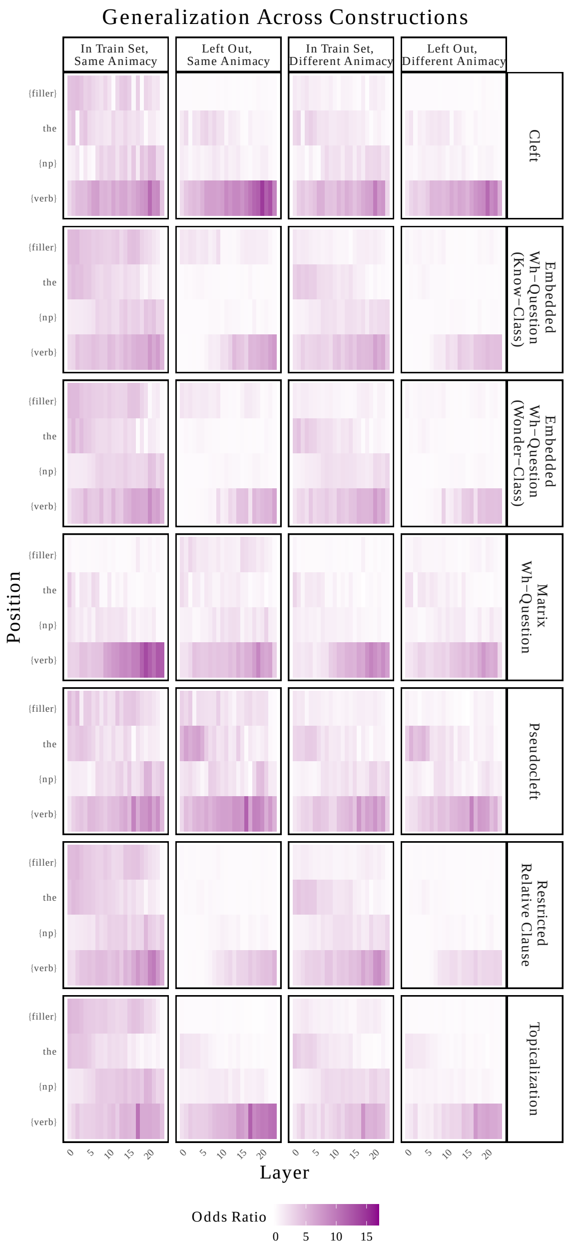


Figure 3: For each source construction, we measure the **ODDS** at each position layer pair, aggregating the values by evaluation group. Corresponding plots with control values and multi-clause variants are in Appendix E.

the same set of constructions in training set and evaluation set, with different animacy; (3) evaluation on the held-out constructions, but with the same animacy as the training set; (4) evaluation on the held-out constructions, and differing animacy from the training set; and (5–6) the two controls.

Hypothesis We hypothesize that the **MAX ODDS** for all our targeted evaluation groups will be greater than that of the controls. We further expect **MAX ODDS** to be higher when the evaluated constructions are in training or match in animacy.

Results Figure 2 shows the average **MAX ODDS** of the aforementioned groups at each position in our single-clause templates. In both these single-clause variants and the embedded-clause variants of our constructions (corresponding figure in Appendix E), we find consistently high **MAX ODDS** values for each of the aforementioned non-control groups. The controls show significantly less transfer. We run pairwise t-tests with a Holm-Bonferroni correction, finding the **MAX ODDS** of each of our test groups is significantly higher than both controls at every position in the single clause templates and nearly every position in the multi-clause ones. These results strongly suggest **shared internal representations across filler–gap constructions in the evaluated models.**

To test our hypotheses regarding the effect of training and evaluation set overlap and matching animacy, we fit a linear mixed-effects regression model to our **MAX ODDS** data at each position. Our random effects comprise intervention training set, and evaluation construction, and our fixed effects take the form of binary indicator variables for (1) whether the evaluated construction was in the training set and (2) whether the animacy condition of the evaluated construction matches that of the training set. We find significant, positive, effects for overlap, matching animacy, and their interaction at the FILLER, THE, and NP positions, and for matching animacy at the VERB position. See Appendix C.1 for regression details. Thus, across positions, **LM internal processing is sensitive to linguistically meaningful features, such as animacy of the extracted item** (possible evidence of "lexical boost").

While we broadly see generalization as fitting into held-out constructions (Figure 3), embedded wh- questions and restrictive relative clauses show noticeably less generalization than other constructions. We briefly offer up two accounts for these peculiarities: (1) there is asymmetry in LM generalization between different filler–gap dependencies or (2) these constructions are processed by largely different mechanisms than the other constructions. Clarifying which of these applies to each construction helps motivate our next experiment.

5 Exp. 2: What Factors Drive Filler–Gap Generalization in LMs?

Our previous experiment demonstrated significant overlap between the LM's abstract representations of various filler–gap constructions. However, we also observed notable variation in the strength of this generalization across positions and constructions. Here, we attempt to characterize the nature of this cross-construction generalization. In particular, we attempt to identify whether there exist constructions which serve as *sources* (their filler–gap properties transfer well to other constructions) or *sinks* (filler–gap properties from other constructions transfer well to them). We further investigate which features of natural language (e.g. distributional properties construction frequency, or linguistic properties like the nature of the filler item) may drive this generalization.

Setup To characterize the degree to which a given construction is a source or sink, we perform the following procedure. First, we evaluate all single-source interventions on all constructions of the same clausal length, averaging the normalized **MAX ODDS** across the animacy-conditions at each position, training construction, and evaluation construction triple. We take the resulting $n \times n$ matrix to be an adjacency matrix for a weighted, directed graph $G = (V, E)$ in which vertex V is a construction and each directed edge $E_{i,j}$ is the transfer from construction i to construction j. We then calculate the *out-degree centrality* – the fraction of a graph's

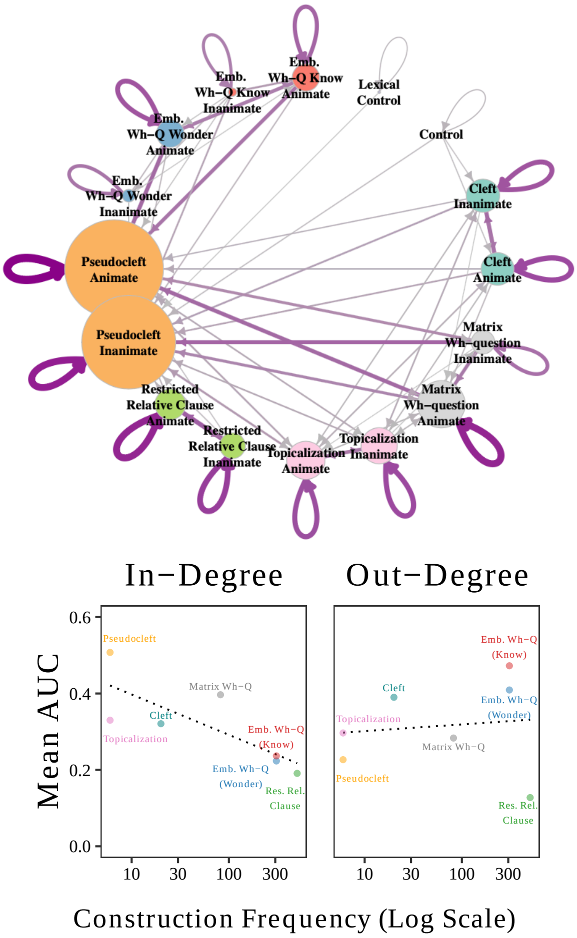


Figure 4: **Top**: Generalization network at single-clause THE position with edge-threshold of 1. Nodes size proportional to in-degree, edge size and color proportional to **ODDS** of the source construction's interventions measured on the target construction. **Bottom:** In- and out-degree centrality AUCs against construction frequency.

total a given node's outgoing edges are connected to – and *in-degree centrality* – the fraction of nodes its incoming edges come from. We do this for nodes (constructions) across a range of edge thresholds – that is, the minimum edge weight retained in the graph. We measure each construction's area under the threshold-centrality curves (AUC). The resulting out- and in-degree AUCs serve as proxies for the degree to which a given construction is a source or sink respectively. We provide a representative generalization network (for the THE position) in Figure 4. That figure shows particularly strong transfer into pseudoclefts, very little transfer into either control, strong within-construction transfer (dark recurrent arrows), and some non-random structure of transfer across constructions.

We also analyzed the effect of construction frequency on generalization capacity. We extracted estimates of each construction's prevalence in the English-EWT Universal Dependencies dataset (De Marneffe et al., 2021; Nivre et al., 2020; Silveira et al., 2014). See Appendix D for details.

We further investigate the effects of four parameters of linguistic variation across filler–gap constructions: the nature of the filler, whether the head daughter is inverted, the syntactic category of the mother (the word under which a construction is embedded), and the semantic/pragmatic nature of the construction (whether the fronted element is fronted by necessity or for discourse reasons). At each position, we fit a linear mixed-effects model with binary indicator variables denoting whether the source and evaluated construction match for each of the above posited parameters of variation as fixed effects, with random effects for training-source construction, and evaluated construction. For regression details, see Appendix C.2.

Hypothesis We expect to see specific constructions serving as strong sources and others as strong sinks in the generalization network. We further expect a positive relationship between a given construction's frequency and the degree to which it is a source, and conversely, a negative relationship between its frequency and its sink-ness. Finally, we anticipate stronger generalization between linguistically similar constructions than dissimilar ones.

Results Figure 4 shows construction frequency against in-degree and out-degree AUCs, mean-pooled across sentence positions. Constructions are spread across the AUC axis, suggesting varying levels of generalization. These AUCs are consistent across both sentence position and clausal variant (single and multi-clause AUCs, faceted by position are available in Appendix F).

Figure 4 also shows a negative relationship between construction frequency and in-degree AUC and a (weak) positive relationship between construction frequency and out-degree AUC. There are some notable exceptions to these trends, such as the low-frequency topicalization construction having a surprisingly low in-degree AUC and the most frequent construction, restrictive relative clauses, having a low out-degree AUC. Below, we argue that these anomalies are linguistically explainable.

We further find evidence supporting our hypothesis that linguistic similarity aids generalization between constructions. Our regression reveals significant, positive effects for filler type at the FILLER and THE positions, inversion of the head daughter and nature of the fronted element at the FILLER, THE, and NP positions, and syntactic category of the mother at all positions.

Discussion These results paint a clear picture of filler–gap generalization in LMs. Frequent constructions are encountered at a high-enough rate during training so as to drive the development of robust mechanisms to process them. Less frequent constructions are not encountered enough for stand-alone, robust, processing mechanisms to form. Instead, their processing relies on the mechanisms of more frequent, linguistically similar constructions.

These analyses shed light on the anomalous results. For instance, we observed a low in-degree AUC for the low-frequency construction topicalization. Topicalization is linguistically dissimilar to higher-frequency constructions, being the only construction with a phrasal element at its filler site, and it generally shares very few linguistic features with more frequent constructions. In this light, its low in-degree AUC is not surprising, especially when compared to pseudoclefts, which much more closely resemble higher-frequency constructions (especially wh-questions).

Similarly, restrictive relative clauses are the only constructions which are embedded under a noun phrase, possess a wh-item at the filler position, and have their filler item fronted out of syntactic necessity, not for discourse purposes. This makes them linguistically dissimilar to many of the lower frequency constructions along the features found important by our mixed-effects model. As such, despite their high frequency, their mechanisms do not transfer broadly to these constructions, leading to a relatively low out-degree.

These results also answer the questions posed at the end of Experiment 1. Namely, embedded wh-questions and restrictive relative clauses show little generalization in the leave-one-out setting, as they are frequent enough to largely not rely on the processing mechanisms of other constructions. However, embedded wh-questions possess enough linguistic overlap with less frequent constructions to aid in their processing, whereas restrictive relative clauses are more isolated in the generalization network due to their linguistic dissimilarities.

6 Exp. 3: Do Language Models Generalize Across Clausal Boundaries?

Our first two experiments demonstrate that LMs share processing mechanisms across various filler–

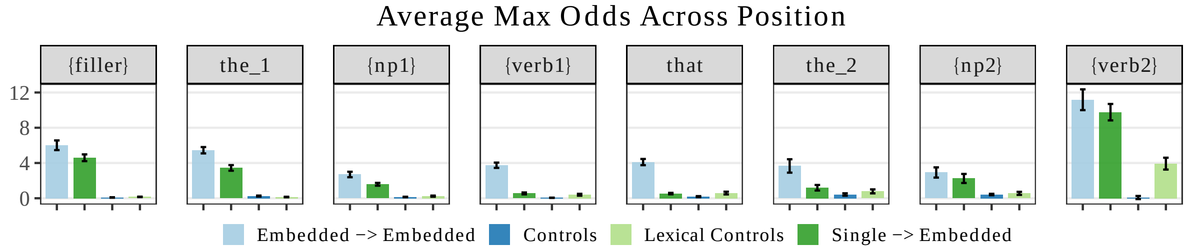


Figure 5: **MAX ODDS** ± 1 standard error, by position for interventions (1) trained and evaluated on multi-clause variants, (2) trained on single-clause variants and evaluated on multi-clause variants, and (3-4) controls.

gap constructions of the same clausal length. In this section, we analyze whether our constructions' single-clause processing mechanisms are used to process both clauses in the multi-clause variant.

Set-Up We evaluate the interventions trained at each position of the single-source variants on the corresponding positions in the matrix and embedded clause of the same construction's multi-clause template. We compare the by-position **MAX ODDS** values to the corresponding values of interventions trained and evaluated on the multi-clause variants.

Hypothesis Under a purely modular account of syntactic structure, we expect to see generalization across clausal boundaries. That is, we expect the single-clause interventions to show above-chance **MAX ODDS** when evaluated on both the matrix and embedded clause of our multi-clause variants.

Results Our results are displayed in Figure 5. Our single-clause mechanisms show above-chance **MAX ODDS** at the FILLER through NP_1 positions of the matrix clause, before dropping off at the VERB_1 through THE_2 positions, and then slowly rebounding as we move towards the final VERB_2.

These results make sense when we consider the relative sentential structures of single-clause and multi-clause sentences, and the auto-regressive nature of the LMs we study. Primarily, the first three positions of a multi-clause sentence – that is, FILLER, THE_1, and NP_1 – are indistinguishable from the first three positions of a singular-clause sentence. As such, we would expect an auto-regressive LM, processing from left-to-right, to not be aware that it is processing an embedded clause until it reached the VERB_1 position. Until then, it will use the same mechanisms it would to process a sentence with a single clause. This is reflected in the strong generalization through these first three positions.

In the VERB_1 position, however, single-clause and multi-clause sentences have verbs that sharply diverge in their semantic character and syntactic properties. Specifically, the verbs at this position in a multi-clause sentence must be ones which can embed a clause (e.g. *say*, *know*, and *wonder*, among others), whereas in a single-clause sentence this is not necessary. As such, upon encountering this position, the LM encounters a different set of verbs than it was trained on, leading to a drop in the single-clause intervention's **MAX ODDS**.

As the LM processes the next couple of positions (THAT, THE_2, and NP_2), we see the single-clause intervention's **MAX ODDS** steadily increasing, as the LM gets closer to a position where it can potentially discharge its filler. This process culminates at the VERB_2 where we see clear, above-chance, generalization from the single-clause mechanisms to the embedded-clause.

7 Conclusion

Long-held views in linguistics suggest that there should be common processing characteristics across diverse filler–gap constructions. We found this to be the case: we were able to transfer the filler–gap property across neural representations of different filler–gap constructions, suggesting that neural models rely on similar representations across distinct constructions.

This transfer is not entirely uncomplicated, however: transfer was stronger when animacy matched and less strong when animacy did not match *even within a construction*. This was true even though animacy is not a key part of the usual account of English filler–gap constructions. We also found that some constructions were stronger sources of filler–gap transfer than others — and others were stronger sinks. Finally, we found that transfer across main and embedded clauses was not strong. Taken together, these results show how mechanistic analysis of LLMs can provide novel insights into the nature of syntactic structures.

8 Limitations

Our work is primarily an attempt to show that LLMs can be useful tools for pushing linguistic theory forward. This brings with it specific theoretical presuppositions that are worth articulating to avoid a suggestion that there is scientific consensus where there is not.

Our investigation is oriented toward finding evidence of modular structure in LLMs. However, it is not a settled question what constitutes rule-like or systematic linguistic behavior in neural systems (Geiger et al., 2024; Nefdt, 2023; Buckner, 2024; Futrell and Mahowald, 2025). How causally systematic should a syntactic behavior be for it to be rule-like? One reading of our results would be that our causal interventions capture human filler–gap behavior but noisily (e.g., imperfect transfer across constructions, less transfer when animacy differs).

This is possible, but another reasonable interpretation is that the relevant constructs are also fuzzy in humans. Despite a historical proclivity for rules, nearly all syntactic theories allow for numerous exceptions, and human behavior itself is variable and subject to errors. As such, the questions we ask regarding the rule-like nature of LLMs extend beyond such models, becoming broader questions about human processing and behavior. Our findings alone cannot adjudicate these questions, though.

We also note that our results are only in English. It would be valuable to extend to other languages, particularly those with typologically different filler-gap patterns.

We relied here on templatically generated sentences, which are known to differ in systematic ways from naturally occuring sentences. We would like to extend this work to naturalistic sentences, but doing so is challenging because of the strong constraint that we have matched pairs.

Acknowledgments

We would like to thank Qing Yao and, more broadly, the computational linguistics research group at UT Austin for their helpful conversations regarding this project. We further acknowledge funding from NSF CAREER grant 2339729 (to Kyle Mahowald).

References

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

Dale J. Barr, Roger Levy, Christoph Scheepers, and Harry J. Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68(3):255–278.

Debasmita Bhattacharya and Marten van Schijndel. 2020. Filler-gaps that neural networks fail to generalize. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 486–495, Online. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *Preprint*, arXiv:2304.01373.

Cameron J Buckner. 2024. *From deep learning to rational machines: What the history of philosophy can teach us about the future of artificial intelligence*. Oxford University Press.

Noam Chomsky. 1957. *Syntactic Structures*. Walter de Gruyter.

Peter W. Culicover. 1999. *Syntactic Nuts: Hard Cases, Syntactic Theory, and Language Acquisition*. Oxford University Press, Oxford.

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

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.

Janet Dean Fodor. 1989. Empty categories in sentence processing. *Language and Cognitive Processes*, 4(3-4):SI155–SI209.

Richard Futrell and Kyle Mahowald. 2025. How linguistics learned to stop worrying and love the language models. *arXiv preprint arXiv:2501.17047*.

Richard Futrell, Peng Qian, Edward Gibson, Evelina Fedorenko, and Idan Blank. 2019. Syntactic dependencies correspond to word pairs with high mutual information. In *Proceedings of the Fifth International*

Conference on Dependency Linguistics (Depling, SyntaxFest 2019), pages 3–13, Paris, France. Association for Computational Linguistics.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, et al. 2023. Causal abstraction: A theoretical foundation for mechanistic interpretability. *arXiv preprint arXiv:2301.04709*.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. 2024. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pages 160–187. PMLR.

Jonathan Ginzburg and Ivan A. Sag. 2001. *Interrogative Investigations: The Form, Meaning, and Use of English Interrogatives*. CSLI, Stanford, CA.

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

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

Anastasia Kobzeva, Suhas Arehalli, Tal Linzen, and Dave Kush. 2023. Neural networks can learn patterns of island-insensitivity in Norwegian. In *Proceedings of the Society for Computation in Linguistics 2023*, pages 175–185, Amherst, MA. Association for Computational Linguistics.

Alexandra Kuznetsova, Per B Brockhoff, and Rune HB Christensen. 2017. lmertest package: tests in linear mixed effects models. *Journal of statistical software*, 82:1–26.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. 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 11–20, Minneapolis, Minnesota. Association for Computational Linguistics.

Nur Lan, Emmanuel Chemla, and Roni Katzir. 2024. Large language models and the argument from the poverty of the stimulus. *Linguistic Inquiry*, pages 1–56.

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

Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Aaron Mueller, Yu Xia, and Tal Linzen. 2022. Causal analysis of syntactic agreement neurons in multilingual language models. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 95–109, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ryan M Nefdt. 2023. *Language, Science, and Structure: A Journey into the Philosophy of Linguistics*. Oxford University Press.

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

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

Steven Piantadosi. 2023. Modern language models refute Chomsky's approach to language. *Lingbuzz Preprint, lingbuzz*, 7180.

Martin J Pickering and Holly P Branigan. 1998. The representation of verbs: Evidence from syntactic priming in language production. *Journal of Memory and language*, 39(4):633–651.

Eva Portelance and Masoud Jasbi. 2024. The roles of neural networks in language acquisition. *Language and Linguistics Compass*, 18(6):e70001.

Christopher Potts. 2023. Characterizing english preposing in pp constructions. *Journal of Linguistics*, pages 1–39.

Grusha Prasad, Marten van Schijndel, and Tal Linzen. 2019. Using priming to uncover the organization of syntactic representations in neural language models. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

John Robert Ross. 1967. *Constraints on Variables in Syntax*. Ph.D. thesis, MIT, Cambridge, MA.

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

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural nlp: The case of gender bias. *Preprint*, arXiv:2004.12265.

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*.

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

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

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

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. 2023. Interpretability at scale: Identifying causal mechanisms in Alpaca. In *Advances in Neural Information Processing Systems*, volume 36, pages 78205–78226. Curran Associates, Inc.

A Construction Templates

We provide templates and examples for our single-clause inanimate extraction (Table 3), multi-clause animate extraction (Table 4), and multi-clause inanimate extractions (Table 5). In these tables, we use the shorthand demonstrated in Table 2 to refer to our constructions.

Full Construction	**Shorthand**
Emb. Wh-Question (*Know*-Class)	Emb. Wh-Q (*K*)
Emb. Wh-Question (*Wonder*-Class)	Emb. Wh-Q (*W*)
Matrix Wh-Question	Matrix Wh-Q
Restrictive Relative Clause	RRC
Pseudo-Cleft	PC
Topicalization	Topic
Subject-Verb Agreement	SVA
Transitive/Intransitive Verbs	T/I Verbs

Table 2: Abbreviations for syntactic constructions in Tables 3 to 5.

B Training and Evaluation Details

We access the `pythia` models used in this study through the `transformers` python package (Wolf et al., 2020). For each construction, we build out training sets as described by Arora et al. (2024), sampling 200 sentences to form the basis of our training set, before adding each sentence's minimal pair, resulting in perfectly balanced training sets. To train DAS, we use the `pyvene` library (Wu et al., 2024) and follow the hyperparameters used by Arora et al. (2024).

Our evaluation sets for the `pythia-1.4b` models used in the main text consist of 400 sentences, with **ODDS** at each position-layer pair averaged across all evaluation sentences. For the other model variants evaluated (`pythia-2.8b` and `pythia-6.9b`) we use evaluation sets of 96 sentences due to computational constraints, noting that this is still larger than the prescribed size of 50 evaluation sentences from Arora et al. (2024). We also ensure that the intersection of train sets and evaluation sets is empty, so as to not bias our evaluations. Our training and evaluation ran on 2 NVIDIA A40 GPUs. For one model size, training totaled $\sim$12 hours, and evaluation $\sim$250 hours.

Construction	Prefix	Filler	NC	Article	NP	Verb	Label
Emb. Wh-Q (K)	I know	what/that		the	man	built	./it
Emb. Wh-Q (W)	I wonder	what/if		the	man	built	./it
Matrix Wh-Q		What/""	did	the	man	build	?/it
RRC	The chair	which/and		the	man	built	was/it
Cleft	It was	the chair/clear	that	the	man	built	./the chair
PC		What/That		the	man	built	was/it
Topic.	Actually,	the chair/""		the	man	built	./the chair
SVA	The	boy/boys	that	the	man	liked	is/are
T/I Verbs		Last night/Yesterday		some/that	man/boy	ran/built	./it

Table 3: Template and exemplar sentences for inanimate extraction from our single-clause construction variants.

Construction	Prefix	Filler	NC	Article_1	NP_1	Verb_1	that	Article_2	NP_2	Verb_2	Label
Emb. Wh-Q (K)	I know	who/that		the	nurse	said	that	the	man	liked	./it
Emb. Wh-Q (W)	I wonder	who/if		the	nurse	said	that	the	man	liked	./it
Matrix Wh-Q		Who/""	did	the	nurse	say	that	the	man	liked	?/it
RRC	The boy	who/and		the	nurse	said	that	the	man	liked	was/it
Cleft	It was	the boy/clear	that	the	nurse	said	that	the	man	liked	./the chair
PC		Who/That		the	nurse	said	that	the	man	liked	was/it
Topic.	Actually,	the boy/""		the	nurse	said	that	the	man	liked	./the chair
SVA	The	boy/boys	that	the	nurse	said	that	the	man	liked	is/are
T/I Verbs		Last night/Yesterday		the	nurse	said	that	some/that	man/boy	ran/liked	./it

Table 4: Template and exemplar sentences for animate extraction from our multi-clause construction variants.

Construction	Prefix	Filler	NC	Article_1	NP_1	Verb_1	that	Article_2	NP_2	Verb_2	Label
Emb. Wh-Q (K)	I know	what/that		the	nurse	said	that	the	man	built	./it
Emb. Wh-Q (W)	I wonder	what/if		the	nurse	said	that	the	man	built	./it
Matrix Wh-Q		What/""	did	the	nurse	say	that	the	man	built	?/it
RRC	The chair	which/and		the	nurse	said	that	the	man	built	was/it
Cleft	It was	the chair/clear	that	the	nurse	said	that	the	man	built	./the chair
PC		What/That		the	nurse	said	that	the	man	built	was/it
Topic.	Actually,	the chair/""		the	nurse	said	that	the	man	built	./the chair
SVA	The	boy/boys	that	the	nurse	said	that	the	man	liked	is/are
T/I Verbs		Last night/Yesterday		the	nurse	said	that	some/that	man/boy	ran/built	./it

Table 5: Template and exemplar sentences for inanimate extraction from our multi-clause construction variants.

```
model <- lmer(max_odds) ~(1 + in_train_set * same_animacy | from) +
                         (1 + in_train_set * same_animacy | to) +
                         in_train_set * same_animacy
```

Figure 6: Model formula used at each position for the linear mixed-effects regressions in Experiment 1.

```
model <- lmer(max_odds) ~(1 + match_filler_class + match_inversion +
                         match_embedded_under + match_discourse_fronted || from ) +
                         (1 + match_filler_class + match_inversion +
                         match_embedded_under + match_discourse_fronted || to ) +
                         match_filler_class + match_inversion +
                         match_embedded_under + match_discourse_fronted
```

Figure 7: Model formula used at each position for the linear mixed-effects regressions in Experiment 2.

Construction	Filler Class	Inverted Clause	Embedding Item	Fronting for Discourse
Embedded Wh-Q	Wh-Item	False	Verb Phrase	False
Matrix Wh-Q	Wh-Item	True	N/A	False
RRC	Wh-Item	False	Noun Phrase	False
Cleft	Null-Element	False	Verb Phrase	True
Pseudocleft	Wh-Item	False	N/A	True
Topicalization	Phrasal Element	False	N/A	True

Table 6: Parameter values for each filler–gap construction.

C Regression Details

We perform all regressions with the `lmerTest` package in R (Kuznetsova et al., 2017).

C.1 Experiment 1 Regression

In the leave-one-out setting, we fit a linear mixed effects model at each position with our dependent variable as the **MAX ODDS** at each training-set, and evaluation-set pair. We treat the training-set and evaluation-set as random effects, and indicator variables for whether the evaluation-set comprises a construction in the training-set and whether the evaluation-set has the same animacy as the training-set as fixed effects. We also include a term to investigate their interaction. As per Barr et al. (2013), we include maximal random effect slope structures. Our full regression model is reported in Figure 6, which we fit to obtain the reported β coefficients, and corresponding p-values.

Indicator variables are codified such that if the evaluated construction is in the training-set, `in_train_set = 1` with `in_train_set = -1` otherwise. Similarly, if the evaluated construction's animacy matches that of the training conditions, `same_animacy = 1` with `same_animacy = -1` otherwise. The full results of this regression can be found in Table 7. *Note: In this setting, the* `construction_from` *variable denotes the held-out construction.*

C.2 Experiment 2 Regression

In the single-construction setting, we fit a linear mixed effects model at each position with our dependent variable as the **MAX ODDS** at each training-set and evaluation-set pair. We treat the training-set and evaluation-set as random effects. Our mixed effects comprise indicator variable denoting whether the training construction and the evaluation construction match in our proposed filler–gap parameters of variation. A full breakdown of these parameters of variation and how they apply to our constructions of interest can be seen in Table 6. The resulting indicator variables take a value of 1 if the construction in the trainset and the construction in the evaluation set match for that given parameter, and -1 otherwise. We include maximal random effect slope structures, excluding correlations to help convergence, as per Barr et al. (2013).

Our resulting regression model is reported in in Figure 7, which we fit to obtain the reported β coefficients, and corresponding p-values (Table 8).

D Frequencies

To calculate frequencies, we use the English-EWT Universal Dependencies dataset (De Marneffe et al., 2021; Nivre et al., 2020; Silveira et al., 2014). It is sourced from the English Web Treebank, a corpus which totals 16,622 sentences scraped from the web. We parse the train, test, and dev `connlu` associated files searching for dependency relations denoting each of our given constructions. We do not differentiate between our two classes of embedded wh-questions, as the lexically defined constraint would have likely yielded a non-exhaustive extraction of all possible sentences. Instead we calculate a generic total for embedded wh-questions, and share this count among both of them. We present the final counts in Table 9.

Construction Type	Total Count
Restrictive Relative Clauses	504
Embedded Wh-Questions	308
Matrix Wh-Questions	82
Clefts	20
Pseudocleft	6
Topicalization	6
Total Sentences	**16622**

Table 9: Construction Type Counts

Term	β_{FILLER}	β_{THE}	β_{NP}	β_{VERB}
(Intercept)	**1.93*** **	**2.70*** **	**1.87*** **	**9.06*** **
`in_train_set`	**0.67*** **	**0.56*** **	**0.42** **	0.26
`same_animacy`	**1.08*** **	**0.51*** **	**0.60*** **	**2.13*** **
`in_train_set:same_animacy`	**0.36** **	**0.20** **	**0.10* **	0.10

Table 7: Experiment 1 Regression Results. * denotes $p < .05$, ** denotes $p < .01$, and *** denotes $p < .001$.

Term	β_{FILLER}	β_{THE}	β_{NP}	β_{VERB}
(Intercept)	**1.15*** **	**1.96*** **	**1.32*** **	**7.12*** **
`match_filler_class`	**0.75*** **	**1.06** **	0.28	0.53
`match_inversion`	**0.38** **	**0.51** **	**0.40** **	0.06
`match_embedded_under`	**0.85*** **	**1.05** **	**0.54** **	**2.06** **
`match_discourse_fronted`	**0.30** **	**0.36* **	**0.34*** **	0.32

Table 8: Experiment 2 Regression Results. * denotes $p < .05$, ** denotes $p < .01$, and *** denotes $p < .001$

.

E Experiment 1: Supplementary Information

A by-position aggregation figure for the multi-clause variant is in Figure 8, complementing Figure 2. An extended version of the mechanistic plots in Figure 3, including controls, appears in Figure 9, with a multi-clause counterpart shown in Figure 10.

F Experiment 2: Supplementary Information

We report raw bar charts for AUCs of in-degree and out-degree centrality across single- and multi-clause settings (Figures 11 to 14).

G Experiment 3: Supplementary Information

We also provide mechanistic heatmaps for our cross-clausal generalization experiments. They can be found in Figure 15.

H Duplication with Other Model Sizes

We duplicate these experiments with other model sizes, namely `pythia-2.8b` and `pythia-6.9b`. Below, we report these results.

H.1 Experiment 1

We provide the aggregation figures across positions – single (Figure 16) and multi-clause (Figure 17) variants. We note that we find significant differences in the same positions as with the `pythia-1.4b` models. We provide regression results in Table 10.

Term	β_{FILLER}	β_{THE}	β_{NP}	β_{VERB}
`pythia-2.8b`				
(Intercept)	**1.95*** **	**2.74*** **	**1.83*** **	**7.68*** **
`in_train_set`	**0.67*** **	**0.50*** **	**0.37** **	**0.48* **
`same_animacy`	**1.08*** **	**0.51*** **	**0.51*** **	**2.18*** **
`in_train_set:same_animacy`	**0.45** **	**0.19** **	0.09	0.13
`pythia-6.9b`				
(Intercept)	**1.78*** **	**2.59*** **	**1.48*** **	**9.15*** **
`in_train_set`	**0.76*** **	**0.59*** **	**0.36** **	0.20
`same_animacy`	**1.05*** **	**0.47*** **	**0.46*** **	**2.45*** **
`in_train_set:same_animacy`	**0.42** **	**0.18** **	0.07	0.00

Table 10: Experiment 1 Regression Results for `pythia-2.8b` and `pythia-6.9b`. * denotes $p < .05$, ** denotes $p < .01$, and *** denotes $p < .001$.

H.2 Experiment 2

For experiment 2, we provide scatter plots in Figure 18 and regression results in Table 11.

Term	β_{FILLER}	β_{THE}	β_{NP}	β_{VERB}
`pythia-2.8b`				
(Intercept)	**1.05*** **	**1.99*** **	**1.20*** **	**6.20*** **
`match_filler_class`	**0.68*** **	**1.16** **	0.27	**0.78** **
`match_inversion`	**0.42** **	**0.46** **	**0.48*** **	0.29
`match_embedded_under`	**0.82*** **	**1.03** **	**0.35*** **	**1.95** **
`match_discourse_fronted`	**0.30* **	0.37	**0.58** **	0.32
`pythia-6.9b`				
(Intercept)	**1.10*** **	**1.84*** **	**1.08*** **	**7.61*** **
`match_filler_class`	**0.62** **	**1.10** **	0.31	0.28
`match_inversion`	**0.36* **	**0.60** **	**0.48*** **	0.01
`match_embedded_under`	**0.82** **	**1.02** **	**0.53** **	**2.05** **
`match_discourse_fronted`	**0.35* **	0.31	**0.39* **	0.14

Table 11: Experiment 2 Regression Results for `pythia-2.8b` and `pythia-6.9b`. * denotes $p < .05$, ** denotes $p < .01$, and *** denotes $p < .001$.

H.3 Experiment 3

For experiment 3, we provide corollary figures to Figure 5 in Figure 19.

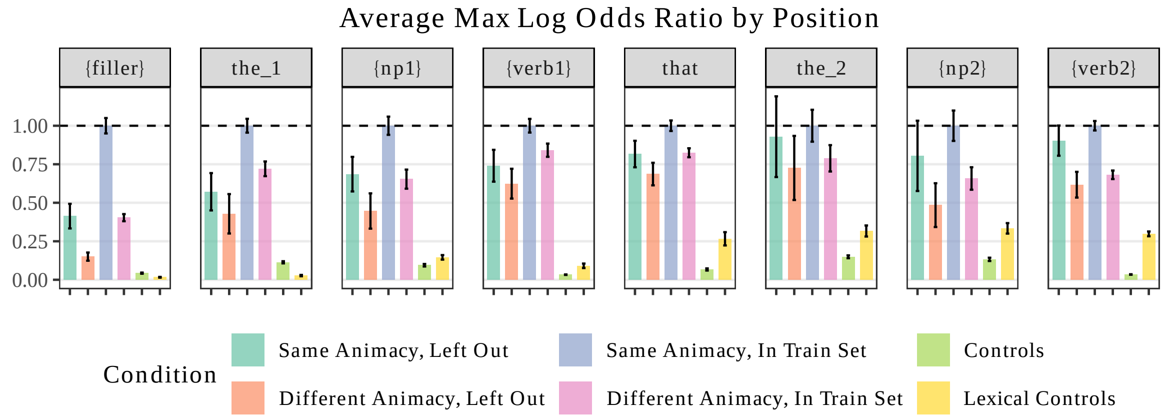


Figure 8: Multi-Clause Aggregation Values by Evaluation Group

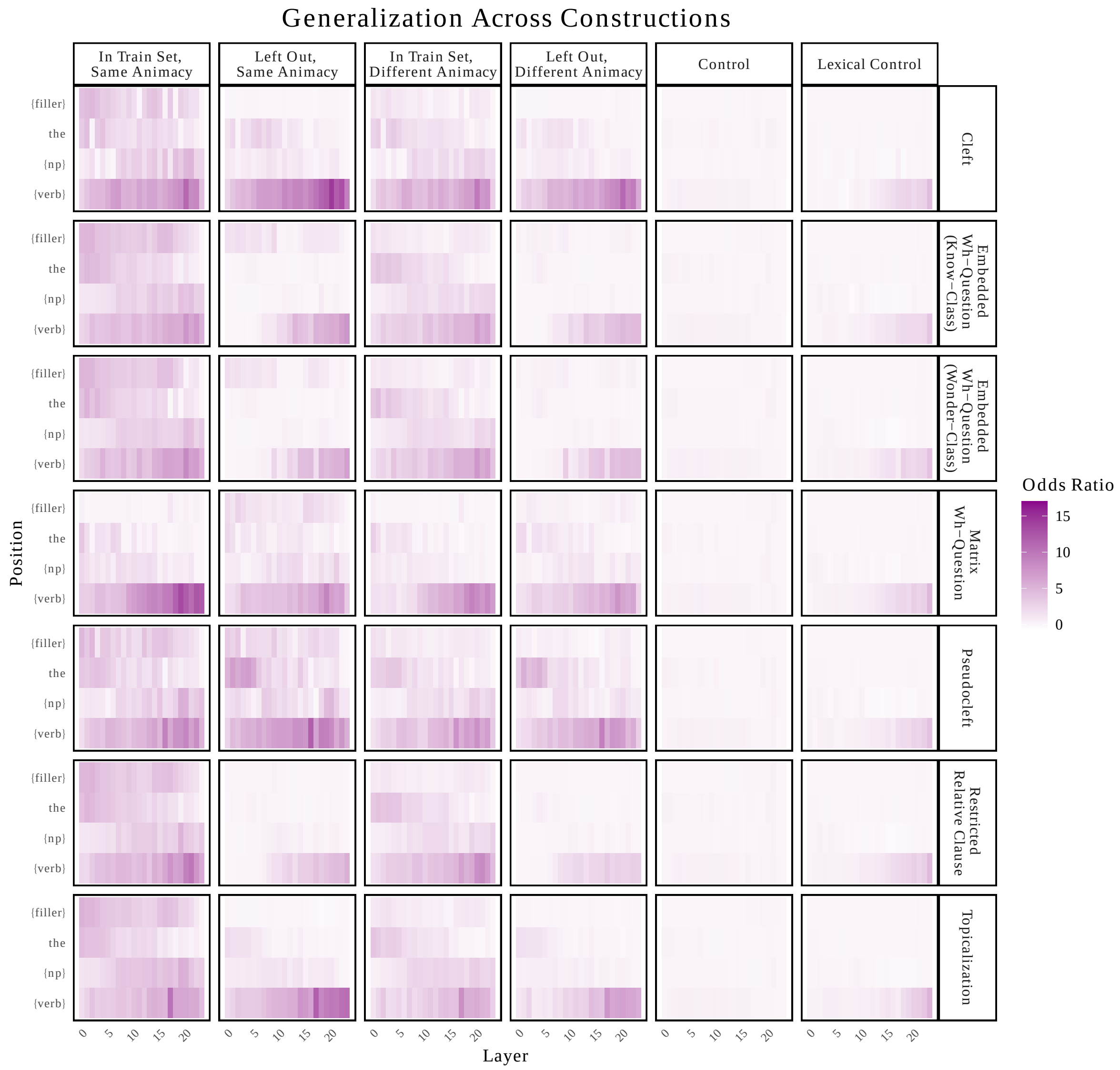


Figure 9: Single Clause **ODDS** at each position-layer pair for each construction. Averaged across animacy conditions.

Figure 10: Multi-Clause **ODDS** at each position-layer pair for each construction. Averaged across animacy conditions.

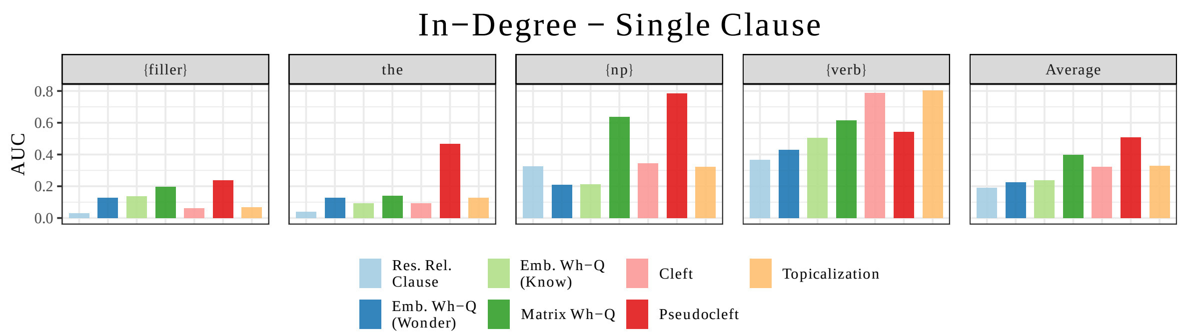


Figure 11: In-Degree AUC by position, with the final facet denoting the average across positions.

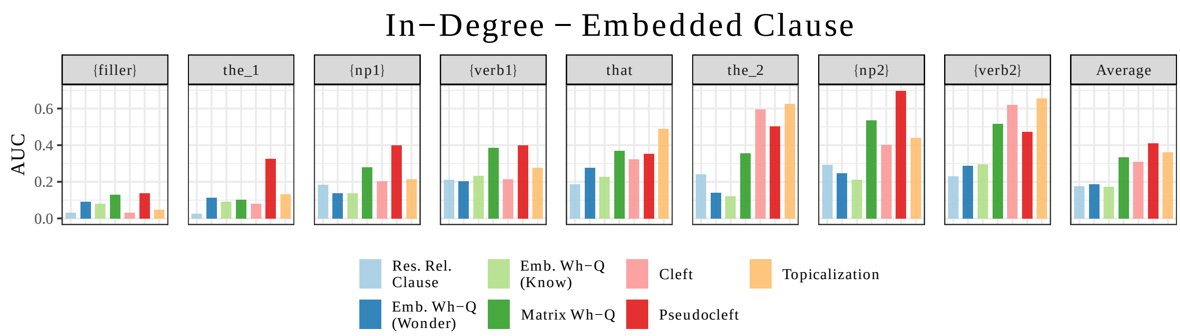


Figure 12: In-Degree AUC by position, with the final facet denoting the average across positions.

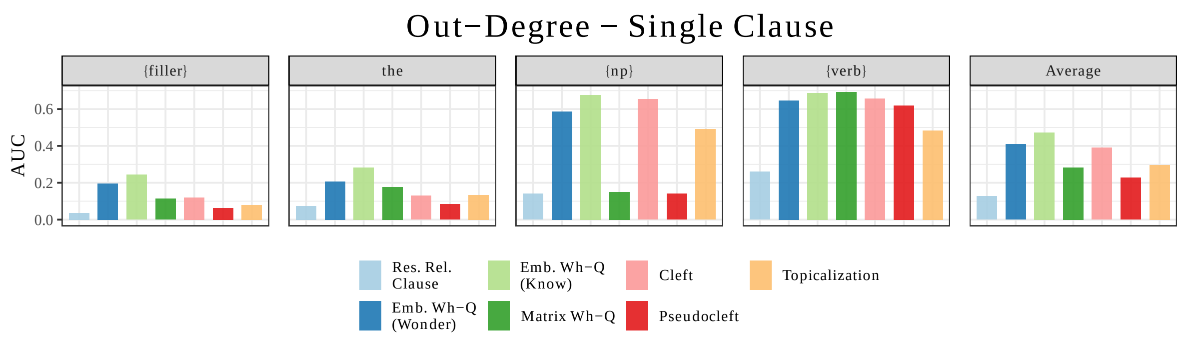


Figure 13: Out-Degree AUC by position, with the final facet denoting the average across positions.

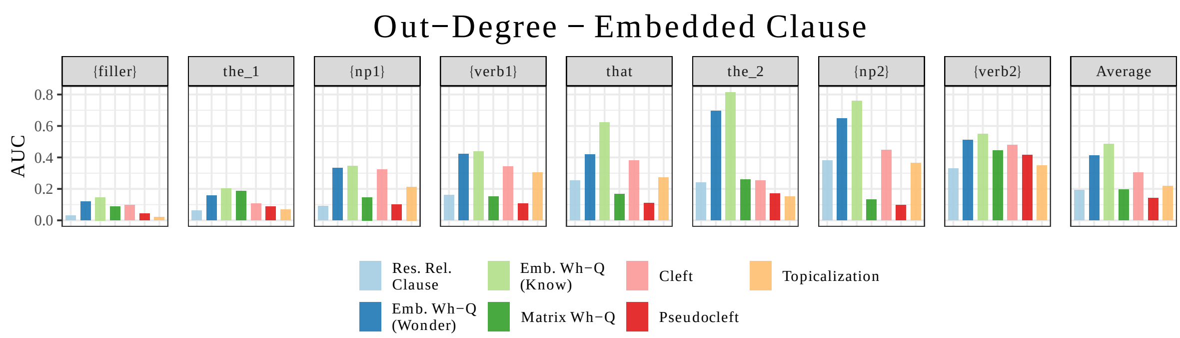


Figure 14: Out-Degree AUC by position, with the final facet denoting the average across positions.

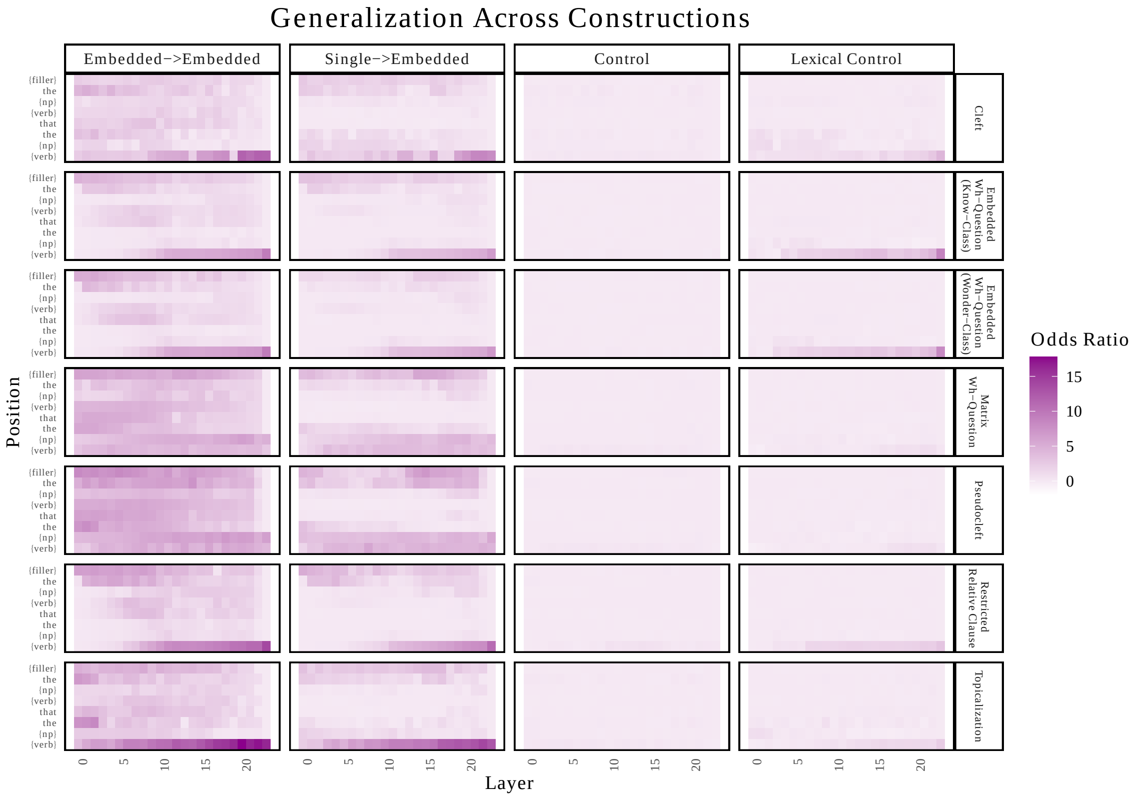
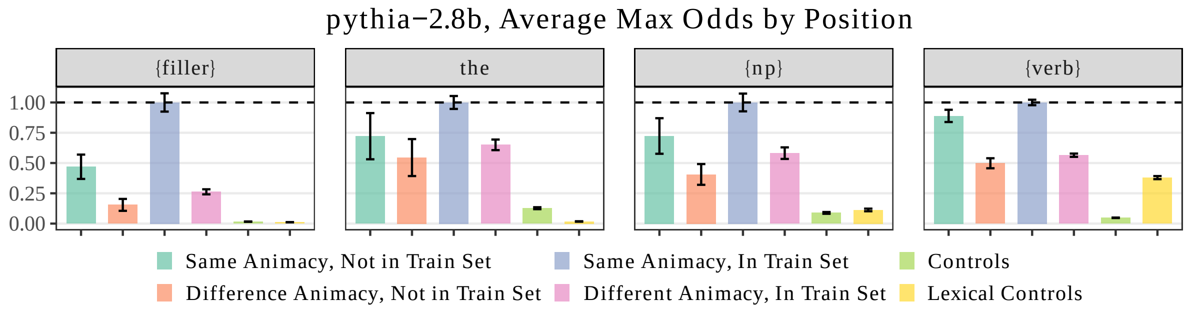
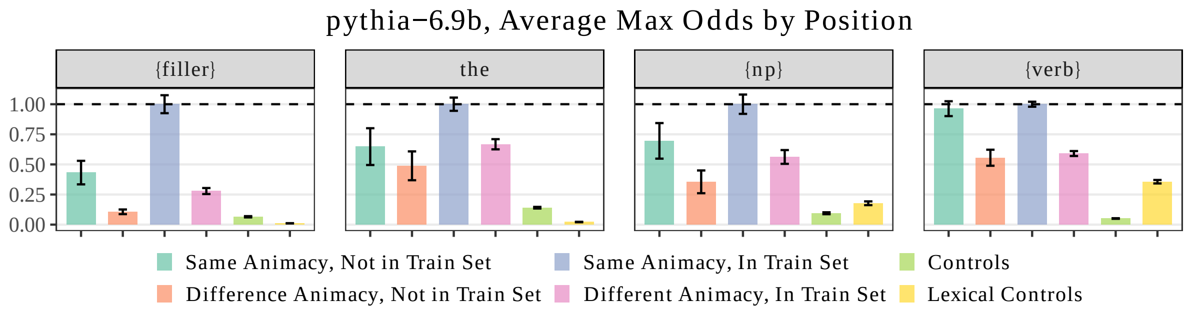


Figure 15: **ODDS** at each position-layer pair for each construction in the cross-clausal generalization experiment. Averaged across animacy conditions and items in a given group.

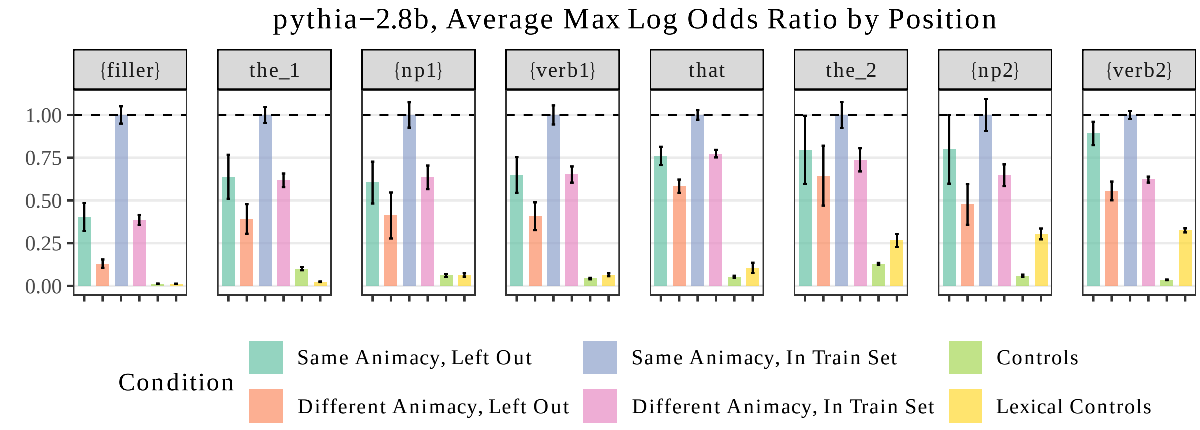


(a) `pythia 2.8b` average normalized **MAX ODDS**.

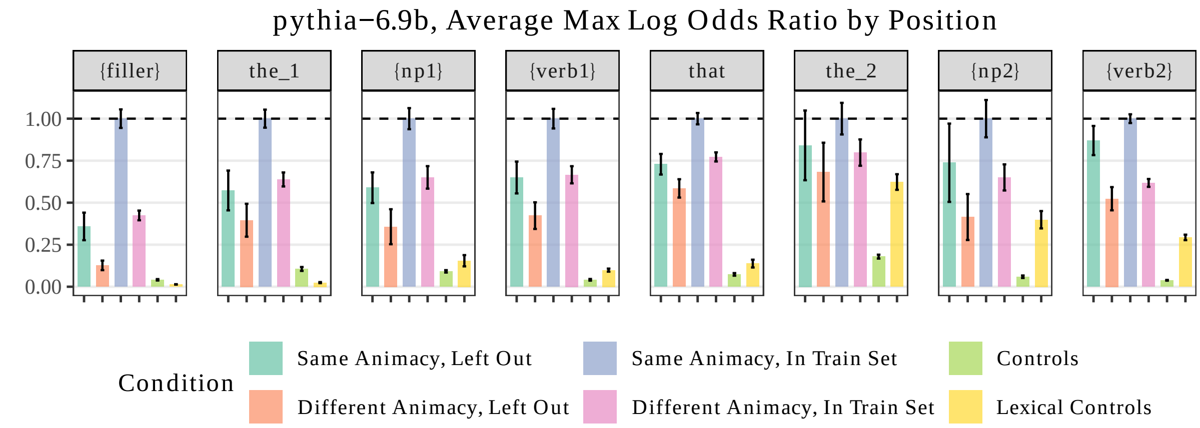


(b) `pythia 6.9b` average normalized **MAX ODDS**.

Figure 16: **Top:** `pythia-2.8b` and **bottom:** `pythia-6.9b` average normalized **MAX ODDS** across positions in the single-clause variants, ± 1 standard error. Normalization fixes the "Same Animacy, In Train Set" condition at 1.00.

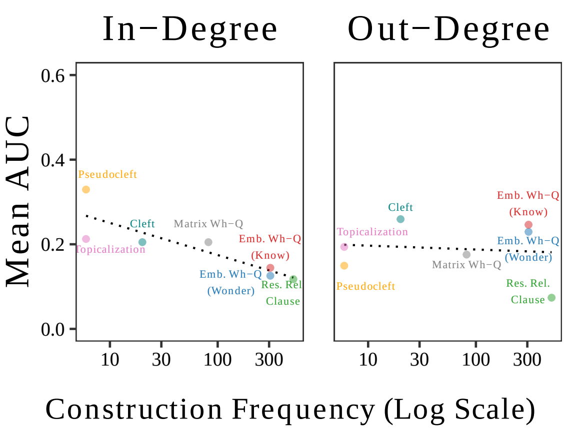


(a) `pythia 2.8b`

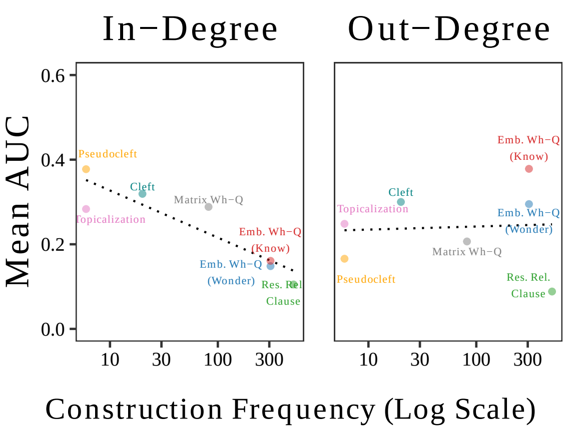


(b) `pythia 6.9b`

Figure 17: **Top:** `pythia-2.8b` and **bottom:** `pythia-6.9b` average normalized **MAX ODDS** across positions in the single-clause variants, ± 1 standard error. Normalization fixes the "Same Animacy, In Train Set" condition at 1.00.

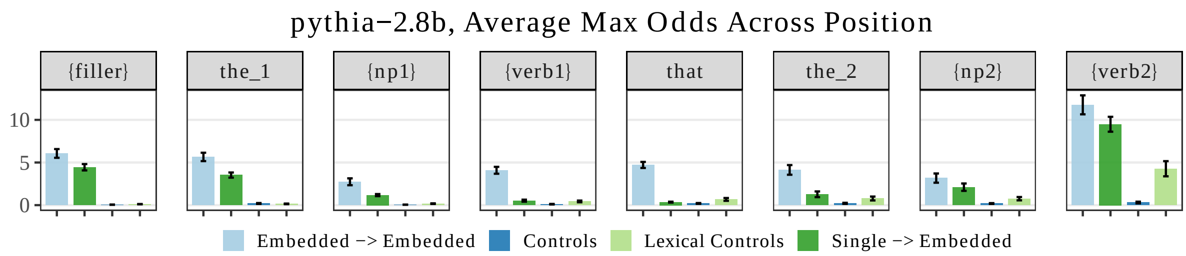


(a) `pythia 2.8b`

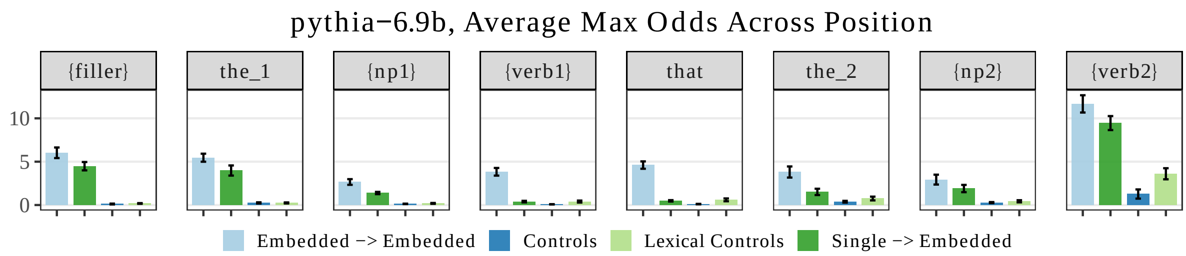


(b) `pythia 6.9b`

Figure 18: Average in-degree centrality AUC and out-degree centrality AUC plotted against construction frequency.



(a) `pythia 2.8b`



(b) `pythia 6.9b`

Figure 19: **MAX ODDS** ± 1 standard error, by position for interventions (1) trained and evaluated on multi-clause variants, (2) trained on single-clause variants and evaluated on multi-clause variants, and (3-4) controls. Evaluations are performed on sentences matching training conditions (i.e. same construction and same animacy).