Mission: Impossible Language Models

Julie Kallini[1], **Isabel Papadimitriou**[1], **Richard Futrell**[2],
Kyle Mahowald[3], **Christopher Potts**[1]
[1]Stanford University; [2]University of California, Irvine; [3]University of Texas, Austin
kallini@stanford.edu

Abstract

Chomsky and others have very directly claimed that large language models (LLMs) are equally capable of learning languages that are possible and impossible for humans to learn. However, there is very little published experimental evidence to support such a claim. Here, we develop a set of synthetic *impossible languages* of differing complexity, each designed by systematically altering English data with unnatural word orders and grammar rules. These languages lie on an impossibility continuum: at one end are languages that are inherently impossible, such as random and irreversible shuffles of English words, and on the other, languages that may not be intuitively impossible but are often considered so in linguistics, particularly those with rules based on counting word positions. We report on a wide range of evaluations to assess the capacity of GPT-2 small models to learn these uncontroversially impossible languages, and crucially, we perform these assessments at various stages throughout training to compare the learning process for each language. Our core finding is that GPT-2 struggles to learn impossible languages when compared to English as a control, challenging the core claim. More importantly, we hope our approach opens up a productive line of inquiry in which different LLM architectures are tested on a variety of impossible languages in an effort to learn more about how LLMs can be used as tools for these cognitive and typological investigations.

1 Introduction

Chomsky (2023), Chomsky et al. (2023), Moro et al. (2023), and Bolhuis et al. (2024) make very broad claims to the effect that large language models (LLMs) are equally capable of learning possible and impossible human languages. For these authors, it follows from this claim that LLMs cannot teach us anything about language, and so the claim (if true) would have significant consequences for linguistic methodology and potentially also for the viability of LLMs as the basis for robust language capabilities.

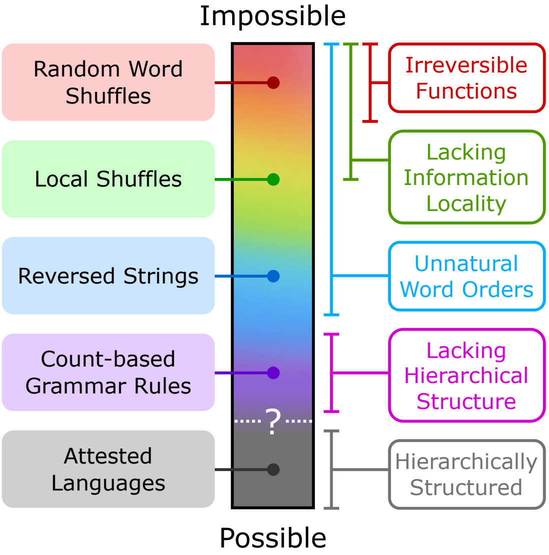


Figure 1: Partial impossibility continuum of languages based on complexity. We assess the learnability of languages at different points in the continuum and push the (currently unclear) boundary between possible and impossible.

These authors state this claim in absolute terms. For example, Chomsky et al. (2023) flatly assert that LLMs "are incapable of distinguishing the possible from the impossible," Chomsky (2023) says this property "can't be modified," and Moro et al. (2023) write that "the distinction between possible versus impossible languages cannot be formulated by definition for LLM." Bolhuis et al. (2024) go so far as to claim that "LLMs can produce 'impossible' languages [...] just as well as (if not better than) natural language output." One might expect such strong claims to be supported by extensive formal analysis and/or experimental evidence. However, as far as we are aware, this is not the case. The sole experimental paper cited by the above authors is Mitchell and Bowers 2020—an important and

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14691–14714
August 11-16, 2024 ©2024 Association for Computational Linguistics

inspiring paper but not one that can resolve these questions on its own. In addition, linguists themselves do not even have an agreed upon notion of what defines the possible or the impossible languages, to say nothing of having formal results with respect to LLMs.

Here we provide extensive new experimental evidence to inform the claim that LLMs are equally capable of learning possible and impossible languages in the human sense. Arguably, the central challenge for such work is the fact that there is no agreed-upon way of distinguishing these two groups. We do not feel positioned ourselves to assert such a definition, so we instead offer some examples of impossible languages on a continuum of intuitive complexity (Figure 1). Some of these examples seem intuitively impossible, such as random sentence-level shuffling of English words. Others operationalize less obvious but common claims in the linguistics literature about rules that are impossible, like those that depend on counting words.

All of our examples are, we take it, uncontroversial instances of impossible languages. Thus, our experiments can inform the core hypotheses as follows: if LLMs learn these languages as well as they learn natural languages, then the claims of Chomsky and others are supported (for the specific class of LLMs tested). Conversely, if LLMs do not learn these languages as well as the possible ones, it would call into question those assertions. In that case, proponents of those claims ought to provide examples of impossible languages that they find more informative, which we can then evaluate using our approach to further advance the discussion.

Our experiments use GPT-2 small models (Radford et al., 2018, 2019), and our base training corpus is the BabyLM dataset (Warstadt et al., 2023), which we modify in various ways to implement our impossible languages. What we find is that these models indeed struggle to learn impossible languages, shown through three core experiments:

- In **Experiment 1**, we train GPT-2 models on our set of defined possible and impossible languages, measuring their learning efficiency through test set perplexities. We find that *models trained on possible languages learn more efficiently*, evident from lower perplexities achieved in fewer training steps.
- In **Experiment 2**, we more closely examine a set of languages that exhibit count-based verb marking rules, using surprisal comparisons to target the relevant patterns. We find that GPT-2s trained on possible languages are more surprised by ungrammatical constructions, indicating that *models disprefer agreement rules involving counting*.
- In **Experiment 3**, we dive deeper into the internal mechanisms that models may develop to learn such count-based grammar rules using causal abstraction analysis. We find that *models develop natural, modular solutions to unnatural grammatical patterns*.

Overall, our experimental results strongly challenge the claims of Chomsky and others given above, and we believe they pave the way for even deeper discussions of LLMs as models of language learning. At the same time, we recognize that models and humans exhibit fundamental differences, but the extent to which models favor or disfavor natural languages can be influenced by specific architectural decisions (as demonstrated by our findings on tokenization and positional encodings). We hope this paper initiates a new line of work that explores how different model architectures can distinguish between the possible and impossible languages.[1]

2 Background and Related Work

2.1 Impossible Human Languages and Language Universals

The notion of an impossible human language is elusive and difficult to define, in part due to a lack of consensus on which properties are universal in human language and which properties are "impossible" (Comrie, 1989; Evans and Levinson, 2009; Nefdt, 2024). For instance, *recursion*, or the principle that all languages produce hierarchical syntactic structures via recursive procedures, has been claimed to be a universal property of human language (Chomsky, 1957, 1965, 2002; Hauser et al., 2002). However, the motivations for recursion have been questioned, with empirical limits on the maximum depth of nested phrases (Karlsson, 2007; Jin et al., 2018) and counterevidence from at least one natural language that seems to lack embedded structures (Everett, 2012). Still, if we grant that possible languages are defined by hierarchical, recursive rules, what defines the impossible

[1] The code for this paper is available at `https://github.com/jkallini/mission-impossible-language-models`.

languages? Moro et al. (2023) claim that the class of impossible languages would use the "opposite" type of rules: those based on the linear order of words. Musso et al. (2003) provide a few concrete examples that involve counting word positions to mark features like negation and agreement, and we include languages with similar rules in our set of tested impossible languages.

It is important to also distinguish what is impossible from what is merely typologically marked, such as the word order patterns listed in Greenberg's (1963) language universals. Previous work has shown that such word order universals can arise through a language's optimization of communication efficiency, achieved by balancing complexity and ambiguity (Hahn et al., 2020; Futrell and Hahn, 2022). While our current exploration does not encompass attested languages, various impossible languages can similarly differ in their information-theoretic complexity, informing the patterns that lie at the boundary between possible and impossible.

2.2 Training Language Models with Unnatural Word Orders

The only work cited by Chomsky that investigates neural language models' ability to learn impossible languages is Mitchell and Bowers 2020, which finds that recurrent neural networks (RNNs; Elman, 1990) trained on various unnatural language constructs, such as reversed sentences and randomized vocabularies, achieve high accuracy on a subject–verb number agreement task. Other work turns to more recent Transformer-based language models (Vaswani et al., 2017), observing their sensitivity to word order and phrase structure (Alleman et al., 2021; Galke et al., 2023) as well as their surprising ability to learn from syntactic information alone (Huang et al., 2023). Studies by Sinha et al. (2021) and Abdou et al. (2022) debate the impact of tokenization, pretraining adjustments, and positional encodings in recovering word order information from shuffled languages. Further investigations into BERT's (Devlin et al., 2019) reliance on word order for grammatical role classification suggest that lexical cues alone may not always be sufficient for good performance (Papadimitriou et al., 2022; see also Hessel and Schofield, 2021; Pham et al., 2021).

2.3 Language Models and Formal Languages

A related line of research examines the abilities of neural language models to express formal languages, as defined by the *Chomsky hierarchy* (Chomsky, 1956, 1959). Human language is considered to be slightly more expressive than context-free languages due to certain syntactic phenomena that interleave constituents (Shieber, 1985; Joshi, 1985). Previous work has shown that RNNs or related models can represent variants of counter and DYCK languages, which are context-free (Weiss et al., 2018; Merrill, 2019; Merrill et al., 2020; Hewitt et al., 2020).[2] Similar work on Transformer architectures has shown that, while they are theoretically Turing-complete provided arbitrary precision and decoder steps (Pérez et al., 2021), they cannot empirically model many regular and non-regular languages (Hahn, 2020; Ebrahimi et al., 2020; Deletang et al., 2023).

The inability of Transformer-based language models to learn more complex languages in the Chomsky hierarchy seems surprising, given their impressive performance on natural language. This could be interpreted as evidence that theoretically weak computational models are sufficient for expressing human language. Alternatively, Transformer-based models can be augmented to have inductive biases for nested, hierarchical structures through architecture changes, like the addition of a stack component (Hao et al., 2018; Murty et al., 2023), or data-centered approaches, like structural pretraining (Papadimitriou and Jurafsky, 2023).

3 Impossible Languages

Core to our experiments are the set of *impossible languages* we synthesize. In constructing these artificial counterfactual languages, we consider their information-theoretic attributes relevant to machine learning, such as entropy rate, as well as their formal linguistic characteristics, such as adherence to hierarchical grammatical structures. We believe that our choice of languages broadly spans the impossibility continuum hypothesized in Figure 1.

Concretely, we specify impossible languages by defining *perturbation functions* of English sentences. These perturbation functions map English input sentences to sequences of tokens. We categorize our languages into three classes: *SHUFFLE, *REVERSE, and *HOP, defined in the next subsections. Each class has one control language that represents unaltered English, or a pattern that is very similar to English. Table 1 provides examples

[2] Though counter and DYCK languages are context-free, some of the variants in the cited work are regular.

Class	Language	Example 1	Example 2
*SHUFFLE	NOSHUFFLE	He cleans his very messy books he lf .	They clean his very messy books he lf .
	NONDETERMINISTICSHUFFLE	messy books his he very . lf He cleans	his . very he They messy lf books clean
	DETERMINISTICSHUFFLE($s = 21$)	cleans He messy books he lf very . his	clean They messy books he lf very . his
	DETERMINISTICSHUFFLE($s = 57$)	cleans his He messy . he very lf books	clean his They messy . he very lf books
	DETERMINISTICSHUFFLE($s = 84$)	He messy . lf his very books cleans he	They messy . lf his very books clean he
	LOCALSHUFFLE($w = 3$)	his He cleans books very messy . he lf	his They clean books very messy . he lf
	LOCALSHUFFLE($w = 5$)	his messy very He cleans lf books he .	his messy very They clean lf books he .
	LOCALSHUFFLE($w = 10$)	messy books his he very . lf He cleans	messy books his he very . lf They clean
	EVENODDSHUFFLE	He his messy he . cleans very books lf	They his messy he . clean very books lf
*REVERSE	NOREVERSE	He cleans his very messy books R he lf .	They clean his R very messy books he lf .
	PARTIALREVERSE	He cleans his very messy books R . lf he	They clean his R . lf he books messy very
	FULLREVERSE	. lf he R books messy very his cleans He	. lf he books messy very R his clean They
*HOP	NOHOP	He clean S his very messy books he lf .	They clean P his very messy books he lf .
	TOKENHOP	He clean his very messy books S he lf .	They clean his very messy books P he lf .
	WORDHOP	He clean his very messy books he lf S .	They clean his very messy books he lf P .

Table 1: List of impossible languages with examples. Control (‘NO*’) languages have patterns that resemble English. Differently colored blocks represent different GPT-2 tokens.

of perturbed sentences in each language.

3.1 *SHUFFLE Languages.

The first set of impossible languages, which we call the *SHUFFLE languages, involve different shuffles of tokenized English sentences.

1. **NOSHUFFLE**: The input sentence is tokenized, and the token sequence is unaltered. This language is simply English, used for comparison with other *SHUFFLE languages.

2. **NONDETERMINISTICSHUFFLE**: The tokenized input sentence is randomly shuffled. A different random shuffle is used for each input sentence, with no consistency across inputs.

3. **DETERMINISTICSHUFFLE(s)**: The tokenized input sentence is deterministically shuffled based on the length of the token sequence. For example, all token sequences of length 5 are shuffled in the same order. We create several languages by varying the random seed s that produces the shuffle.

4. **LOCALSHUFFLE(w)**: The tokenized input sentence is deterministically shuffled in local windows of a fixed size w. We create several languages by varying w.

5. **EVENODDSHUFFLE**: The tokenized input sentence is reordered such that all even-indexed tokens appear first, followed by all odd-indexed tokens.

The random shuffling function that generates the NONDETERMINISTICSHUFFLE language is irreversible, resulting in sentences that are purely bags of words—any structural information in the original linguistic signal is irretrievable. While the DETERMINISTICSHUFFLE languages are created using a reversible perturbation function, this function operates in an entirely non-linguistic manner; words are ordered based solely on the random seed and sentence length, without considerations for linguistic features or *information locality*—the property that, when parts of text predict each other, they are often close together (Futrell, 2019; Mansfield and Kemp, 2023). This method is arguably even less humanly feasible than NONDETERMINISTICSHUFFLE, as it relies on an arbitrarily complex yet consistent rule to determine word order.[3] The question of ranking these two families of languages in the impossibility continuum probes at the definition of impossibility and whether reversibility to an attested language like English is a relevant quantity.

The LOCALSHUFFLE languages offer a finer-grained testbed for the importance of information locality, since we can observe the effects of different window sizes. Finally, EVENODDSHUFFLE also manipulates locality, but interestingly preserves part of the linear word order of English while

[3] Even in the imaginable case of a language with completely free word order, it seems extremely unlikely that this freedom would be totally insensitive to any clause boundaries while the language otherwise looks morphologically like English does. It thus seems very safe to assume that our NONDETERMINISTICSHUFFLE language counts as impossible.

introducing new long-distance dependencies.

3.2 *Reverse Languages.

The *Reverse impossible languages involve reversals of all or part of input sentences.

1. **NoReverse**: The input sentence is tokenized, and a special marker token $\boxed{R}$ is inserted at a random position in the token list. Like NoShuffle, this language is most similar to English. We use it for comparison with other *Reverse languages.

2. **PartialReverse**: The input sentence is tokenized, a special marker token $\boxed{R}$ is inserted at a random position in the list of tokens, and the following tokens are reversed.

3. **FullReverse**: The input sentence is tokenized, a special marker token $\boxed{R}$ is inserted at a random position in the token list, and *all* tokens are reversed.

The PartialReverse language is inspired by the experiments of Mitchell and Bowers (2020) on partially reversed English data, though our experiments are not a direct replication, since we use a different model architecture and dataset. FullReverse may seem like a plausible language syntactically, but higher-level linguistic concepts like anaphora would be highly disrupted. The $\boxed{R}$ tokens are placed at the same positions across the data in all *Reverse languages to control for the entropy introduced by their random placement.

3.3 *Hop Languages.

The *Hop languages perturb verb inflection with counting rules.

1. **NoHop**: All 3rd-person present tense verbs in the input sentence are lemmatized, and the sentence is tokenized. For each 3rd-person present tense verb, a special marker representing the verb's number and tense is placed right after the lemmatized verb. Singular verbs are marked with a special token $\boxed{S}$, and plural verbs are marked with $\boxed{P}$. Like the other control languages, NoHop has a pattern that is most similar to English.

2. **TokenHop**: Identical transformation to NoHop, but the special number/tense markers are placed 4 tokens after the verb.

3. **WordHop**: Identical transformation to NoHop and TokenHop, but the special number/tense markers are placed 4 *words* after the verb, skipping punctuation.

These languages specifically investigate GPT-2's ability to learn grammar rules that involve counting the positions of words or tokens.

4 Experiments

We run several experiments to assess GPT-2's learning of our impossible languages. Our first experiment (Section 4.2) uses perplexities as a general evaluation to compare how well each impossible language model has learned its own perturbed language and see whether this reflects the hypothesized impossibility continuum. In our second and third experiments, we conduct a closer examination of the *Hop languages. Given that their count-based verb marking rules appear to be the least clearly implausible among our proposed languages, we focus on examining these rules specifically through targeted assessments using surprisal theory (Section 4.3). Finally, we dive deeper into the mechanisms each *Hop model uses to predict their respective verb marking rules using causal abstraction analysis (Section 4.4). For all evaluations, we run tests on several model checkpoints to observe the learning process over intervals of training steps.[4]

4.1 Implementation Details

For each impossible language, we apply its perturbation function to each sentence of the BabyLM dataset (Warstadt et al., 2023) to create a transformed dataset. Appendix A provides details on preprocessing and formatting, and describes the language-specific filtering needed to achieve the criteria that define each language.

We train standard GPT-2 small models (Radford et al., 2018, 2019) on each impossible language. To produce confidence intervals for our experiments, we train 5 sets of models for each language using different random seeds, which affect the model parameter initialization and dataset shuffling during training. Training and model hyperparameter choices are detailed in Appendix B. The primary set of GPT-2 models we train have absolute positional encodings. We also train a set of GPT-2 small

[4] We also conduct a constituency probing experiment to test effects on GPT-2's implicit understanding of syntax, with minimal observed differences among models (see Appendix D).

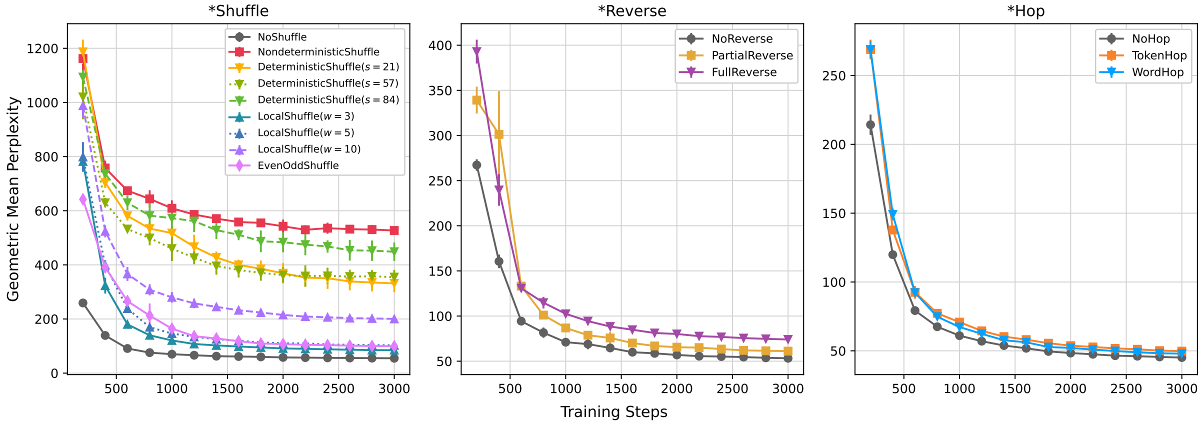


Figure 2: Perplexities on a sample of 10K test sentences for each impossible language model over training steps. Error bars indicate 95% confidence intervals across 5 training runs initialized with different random seeds and evaluated on different test samples.

models with an architecture in which the positional encodings are removed, so that the models' only notion of word order is derived from GPT-2's causal language modeling learning objective (Kazemnejad et al., 2023). Results for these additional experiments supported our main findings on the unaltered GPT-2 architecture. These results are provided in Appendix C.

4.2 Experiment 1: Language Models Reflect the Impossibility Continuum

We train GPT-2 models on all of the languages described in Table 1, and evaluate each model's perplexities on a test set over the course of training. Test perplexities provide a general metric for the extent to which a model has learned a language.

Setup. We sample 10K sentences from the BabyLM test set and perturb this sample for each impossible language. For a given impossible language model, we report the geometric mean of the individual sentence perplexities in the corresponding test sample.

Hypothesis. Models trained on possible languages will achieve lower average perplexities more quickly (as measured in training steps) than those trained on impossible languages.

Results. Our results are in Figure 2. There are clear distinctions between model perplexities after about 500 training steps. First considering the *Shuffle models, the NondeterministicShuffle model has the highest perplexities, followed by the three DeterministicShuffle models, indicating that GPT-2 is better at learning shuffling patterns when they are deterministic, invertible functions.[5] The prevalence of certain sentence lengths in the corpus could also limit the variety of sentence shuffles in the DeterministicShuffle languages, potentially resulting in similarly functioning words frequently occupying the same token positions, thus increasing their predictability.

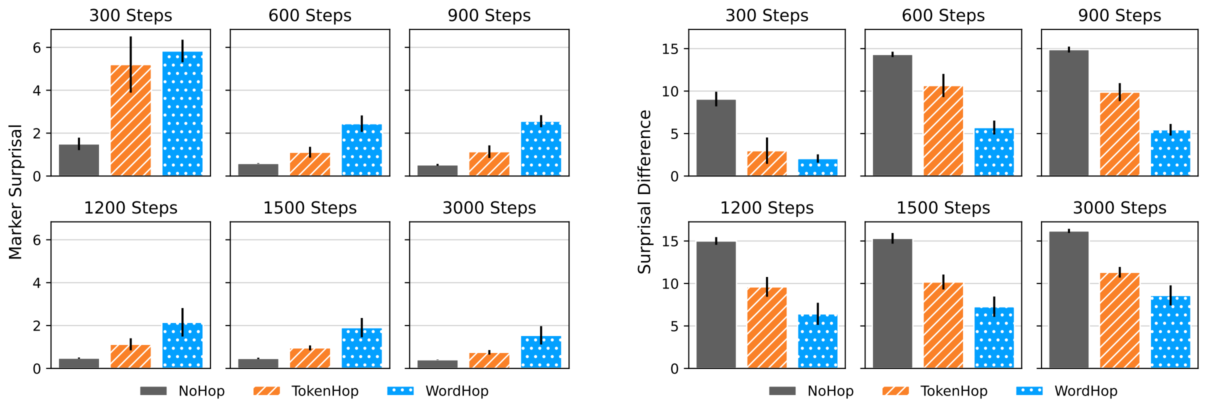
Following the sentence-level shuffles, the next models in the order of decreasing perplexity are the three LocalShuffle models, with smaller window sizes having lower perplexities. LocalShuffle($w = 3$) and EvenOddShuffle have perplexities closest to the NoShuffle model (which represents unaltered English), but NoShuffle consistently has the lowest perplexities throughout the training process.

Compared to the *Shuffle models, the experimental *Reverse models have perplexities that are much closer to the NoReverse model, and PartialReverse is slightly better than FullReverse. For the *Hop languages, their respective control model again has the lowest perplexities, although differences among the models are quite minimal. This warrants our deep-dive into the particular verb marking patterns for this set of models.

4.3 Experiment 2: Language Models Disprefer Counting Rules

In Experiment 1, we show that impossible languages are harder for GPT-2 to learn. However, perplexity is a coarse-grained metric of language learning, and the question remains: do language

[5] This result is also supported by separate evaluations of each DeterministicShuffle model on test data from other shuffles (see Appendix E). Each model has lower perplexities on its own deterministic shuffle.



(a) Test 1: mean surprisals of the verb marker token ($\boxed{\text{S}}$ or $\boxed{\text{P}}$) for each *Hop model.

(b) Test 2: mean surprisal difference between the verb marker token ($\boxed{\text{S}}$ or $\boxed{\text{P}}$) and the following token for each *Hop model.

Figure 3: Surprisal tests for each *Hop model over training steps. Error bars indicate 95% confidence intervals across 5 training runs initialized with different random seeds and evaluated on different test samples.

models learn natural grammatical structures better than impossible grammars?

The structure of the *Hop languages invites a finer-grained evaluation of their verb marking rules. We use *surprisals* to measure how well each *Hop model can predict the placement of its verb marker tokens, $\boxed{\text{S}}$ and $\boxed{\text{P}}$. The surprisal $S(w_i)$ of a word w_i is the negative log probability of w_i given the context words $w_1, \ldots, w_{i-1}$ that precede it: $S(w_i) = -\log_2 p(w_i|w_1, \ldots, w_{i-1})$. Surprisals have been used as acceptability judgments from neural language models to probe for their processing of syntactic information (Wilcox et al., 2018; Futrell et al., 2019; Hu et al., 2020; Wilcox et al., 2023) and have been shown to correlate with human sentence processing difficulty (Hale, 2001; Levy, 2008).

Setup. To test the *Hop models' sensitivity to marker placement, we conduct two tests on a sample of 10K sentences extracted from the BabyLM dataset containing the verb marker tokens ($\boxed{\text{S}}$ or $\boxed{\text{P}}$). As an example, consider the following pair of sentences for the NoHop language shown in (1).

(1) a. He clean$\boxed{\text{S}}$ his very messy bookshelf.
 b. *He clean__ his very messy bookshelf.

Sentence (1-a) is an example in the NoHop language, and (1-b) is an ungrammatical counterfactual in which the marker token does not appear.

In the first test, we compare the average surprisals of the marker tokens across the three *Hop languages, using grammatical examples like (1-a). In the case of (1-a), the marker is singular, and its surprisal $S(\boxed{\text{S}})$ is defined as:

$$S(\boxed{\text{S}}) = -\log_2 p(\boxed{\text{S}} \,|\, \text{He clean})$$

We average this surprisal value for instances of $\boxed{\text{S}}$ or $\boxed{\text{P}}$ in the test sample.

In the second test, we construct minimal pairs from the example sentences in which the marker token appears and does not appear, and then compare the surprisal of the marker token to the surprisal of the token that follows it, both conditioned on the same context. In example (1-b), the surprisal of the following token $S(\text{ his})$ is defined as:

$$S(\text{ his}) = -\log_2 p(\text{ his} \,|\, \text{He clean})$$

We expect $S(\text{ his}) - S(\boxed{\text{S}})$ to be a large positive value. We average such surprisal differences over instances of the marker tokens in the test sample and similarly define marker surprisals and minimal pair configurations for the other *Hop languages.

Hypothesis. For the first surprisal test, our hypothesis is that the mean surprisal of the marker tokens across test examples will be smaller for the control language than for the impossible languages. For the second test, our hypothesis is that the mean surprisal difference across all test pairs will be larger for possible languages than for impossible ones.

Results. Our results are presented in Figure 3. The NoHop model, which has the verb marking pattern most similar to English, consistently has the lowest mean marker surprisal across training steps in test 1 (Figure 3a). The NoHop model also has the highest mean surprisal difference across training steps in test 2 (Figure 3b). Both of these results indicate that GPT-2 has learned to expect the marker tokens when they follow a more natural grammatical pattern and was very surprised when they did not appear at the correct positions.

GPT-2 learns to expect marker tokens at the right locations in the other *Hop models, just not as well as the control. TokenHop tends to have a lower marker surprisal and a higher mean surprisal difference compared to WordHop across training steps, indicating that GPT-2 is better at learning the verb marking rule when the units being counted are tokens instead of words.

4.4 Experiment 3: Language Models Develop Natural Solutions to Unnatural Patterns

Experiment 2 demonstrates that, while GPT-2 favors natural grammar rules, it is also capable of acquiring count-based grammar rules like those seen in the verb marking patterns of our *Hop languages. But what sorts of internal mechanisms does it implement to learn such grammar rules, and how do these mechanisms compare to the more natural control? To address this, we conduct a final experiment using *causal abstraction analysis*, which offers an interpretability framework for identifying and examining causal mechanisms within neural models (Geiger et al., 2020, 2021; Wu et al., 2022, 2023a,b; Geiger et al., 2023). We employ the *interchange intervention* technique on our *Hop models. To perform a basic interchange intervention on a neural model M, we create two instances of M that are provided two different inputs, the base input b and the source input s. Then, we interchange representations created while processing b with representations created while processing s and observe the effect on the output of M. Such interventions allow us to piece together a causal understanding of how the model processes inputs.

Setup. We use interchange interventions to identify representations in our *Hop models that have causal effects on their output behaviors on a subject–verb agreement task. In our experimental setup, b is a sentence prefix with a singular subject and s is an identical prefix with the plural form of the subject. These prefixes include all tokens up to but not including the markers ($\boxed{S}$ and $\boxed{P}$). We interchange the GPT-2 block outputs from processing b with GPT-2 block outputs from processing s and observe whether the probability of plural marker $\boxed{P}$ is higher than the probability of singular marker $\boxed{S}$ after the intervention. This is shown more concretely in Figure 4.

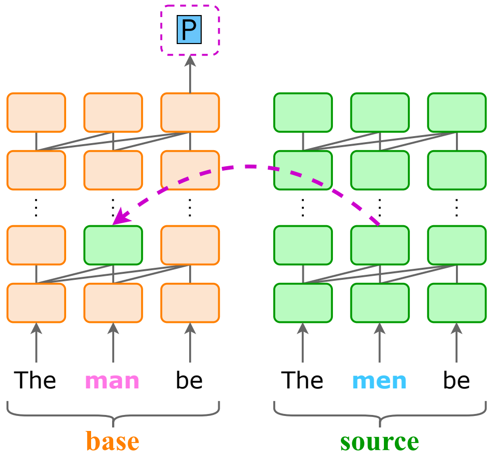


Figure 4: An *interchange intervention* on the NoHop model with base input b = `The man be` and source input s = `The men be`. The intervention is performed at the second layer and second token position, causing a change in prediction from $\boxed{S}$ to $\boxed{P}$.

We run such interventions at each GPT-2 layer and token position to see which parts of the model cause a change in the marker prediction. We run all of these interventions over several test examples and report the *interchange intervention accuracy* (IIA), a metric that represents the subject–verb agreement accuracy if the counterfactual (i.e. plural) were the ground truth. The test examples for each *Hop model are extracted from their respective versions of the BabyLM test set, and minimally-different counterfactual examples are created by changing the singular subjects to plural subjects. To ensure that interventions on different examples are analogous, we use regular expressions to locate examples that follow the same structure (i.e. subjects and verbs at the same positions).

Results. Our results are presented in Figure 5. The IIA graphs demonstrate how information about the marker tokens flows through the models. We can see that, in all three *Hop models, IIA is high at the token position of the subject up until about layer 3; then there is a transition to the position of the last token in the prefix, preceding the location where the marker should be predicted. All models develop the same modular solution to the task by tracking agreement through the representations at the relevant positions, but the NoHop model obtains nearly 100% IIA earlier during training, at about 1,500 training steps, supporting the previous surprisal results.

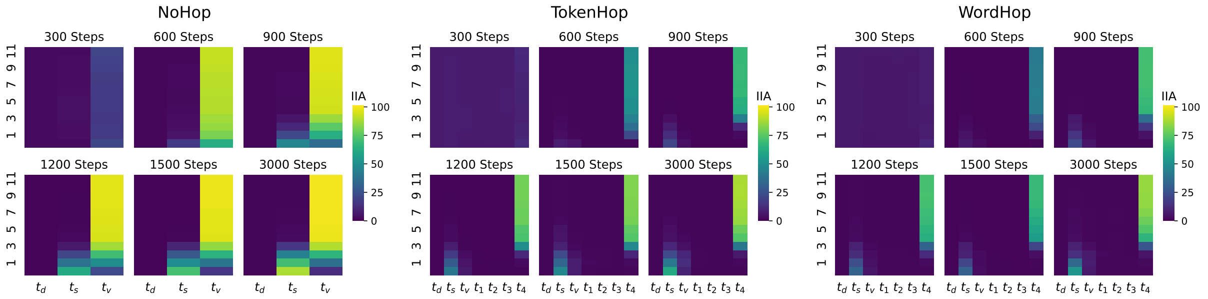


Figure 5: Subject–verb agreement interchange intervention accuracies (IIA) for each *HOP model over training steps. Vertical axes denote the GPT-2 layer of the intervention, and horizontal axes denote the token position of the intervention. t_d, t_s, and t_v represent the tokens for the determiner, subject, and verb, respectively. $t_1 \ldots t_4$ represent the four tokens/words between the verb and its marker for TOKENHOP and WORDHOP. IIA values are averaged over results from 5 models initialized on different random seeds. See Appendix F for confidence intervals.

5 Discussion and Conclusion

Contra claims by Chomsky and others that LLMs cannot possibly inform our understanding of human language, we argue there is great value in treating LLMs as a comparative system for human language and in understanding what systems like LLMs can and cannot learn. Prior explorations of neural language models have already been fruitful for understanding the generalization of syntactic principles from data (Wilcox et al., 2018; Marvin and Linzen, 2018; Futrell et al., 2019; Prasad et al., 2019; Hu et al., 2020). Our paper complements this line of work. We have shown that GPT-2 models do not master our set of synthetic impossible languages as well as natural ones, challenging the unfounded assertions stated previously.

Even in the absence of a clear definition of what constitutes a possible or impossible language, we believe that our investigations advance this debate regarding LLMs. The lack of a definition does not hinder inquiry into this topic; in fact, it beckons further explorations of the boundary between the possible and impossible languages, as shown in our hypothesized continuum in Figure 1. We believe that the *HOP languages we propose closely approach this boundary.

At the same time, conclusions about LLMs' linguistic competence and preferences for natural languages should be informed by an understanding of the ways that models fundamentally differ from humans. For instance, we saw that models can perform operations that involve counting tokens because LLMs rely on tokens as basic units. While humans are sensitive to morpheme boundaries and word boundaries, it is unlikely humans rely on atomic tokens in the way that LLMs do. This does not mean that LLMs can fundamentally tell us nothing about human language. Rather, as we did here, it is valuable to consider and control for this difference before making generalizations.

Since at least the 1950s, a major line of linguistic inquiry has focused on what aspects of syntactic structure can be learned just from data, without domain-specific innate priors (e.g. a *Universal Grammar*). LLMs lack strong in-built linguistic priors, yet they can learn complex syntactic structures. While many LLMs are trained with vastly more data than children see, there is increasing evidence that even systems trained on smaller amounts of data can learn interesting linguistic information (Warstadt et al., 2023). The current paper raises further questions along similar lines. Since we do find that real languages are more learnable by GPT-2, this leads us to wonder what inductive bias of GPT language models matches natural language. We believe that this inductive bias is related to information locality, the tendency for statistical correlations in text to be short range. Information locality arises in GPTs due to their autoregressive training objective and has been argued to arise in humans due to the incremental nature of real-time language processing (Futrell, 2019; Hahn et al., 2021).

Since LLMs have been shown to learn the complex structures of human language and have a preference for learning such structures over unnatural counterfactuals, it follows that they are clearly relevant to investigations and claims about the necessary innate priors for language learning. Arguments that they are "by design, unlimited in what they can 'learn'" and "incapable of distinguishing the possible from the impossible" (Chomsky et al., 2023) do not offer convincing evidence otherwise.

6 Acknowledgments

The authors would like to thank Aryaman Arora, Christiane Fellbaum, Roger Levy, Tristan Thrush, and Diyi Yang for helpful comments on the project. We would also like to thank the members of the Stanford NLP Group, the MIT Computational Psycholinguistics Lab, and the anonymous reviewers for useful discussions. Julie Kallini is supported by a National Science Foundation Graduate Research Fellowship under grant number DGE-2146755.

7 Limitations

Due to resource constraints, we exclusively use the GPT-2 architecture to train models on our various synthetic impossible languages. Each of our experiments involves training a GPT-2 model from scratch on a different language dataset, and for every such language, we train multiple GPT-2 models to establish confidence intervals for our evaluation metrics. Applying this approach to several different model architectures would be quite resource-intensive, so we opted to choose a single architecture in this paper. Future work could apply our methodology to models trained with different architectures or training objectives.

Our impossible languages are derived by manipulating an English dataset. While we do not conduct experiments that use other natural languages as a starting point, our experimental choices (i.e. the synthetic languages we design) are informed by linguistic diversity and typology, distinguishing our impossible languages from those that are rare but attested. However, future work might involve deriving impossible languages from base languages other than English and include more morphological manipulations.

8 Ethics Statement

While this work makes the case for language models as useful tools for cognitive science and linguistics research, these models learn and generate language through processes that are fundamentally different from those employed by humans. Making direct claims about human language learning based on the results of this paper could pose potential risks and harms. This research merely aims to explore the learnability of different languages (specifically, those languages that *cannot* be acquired by humans and are not representative of any known human language) through the lens of neural models.

References

Mostafa Abdou, Vinit Ravishankar, Artur Kulmizev, and Anders Søgaard. 2022. Word order does matter and shuffled language models know it. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6907–6919, Dublin, Ireland. Association for Computational Linguistics.

Matteo Alleman, Jonathan Mamou, Miguel A Del Rio, Hanlin Tang, Yoon Kim, and SueYeon Chung. 2021. Syntactic perturbations reveal representational correlates of hierarchical phrase structure in pretrained language models. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 263–276, Online. Association for Computational Linguistics.

Johan J. Bolhuis, Stephen Crain, Sandiway Fong, and Andrea Moro. 2024. Three reasons why AI doesn't model human language. *Nature*, 627(8004):489–489.

Noam Chomsky. 1956. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124.

Noam Chomsky. 1957. *Syntactic Structures*. De Gruyter Mouton, Berlin, Boston.

Noam Chomsky. 1959. On certain formal properties of grammars. *Information and Control*, 2(2):137–167.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. The MIT Press.

Noam Chomsky. 2002. *On Nature and Language*. Cambridge University Press.

Noam Chomsky. 2023. Conversations with Tyler: Noam Chomsky. Conversations with Tyler Podcast.

Noam Chomsky, Ian Roberts, and Jeffrey Watumull. 2023. Noam Chomsky: The false promise of ChatGPT. *The New York Times*.

Bernard Comrie. 1989. *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press.

Gregoire Deletang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A Ortega. 2023. Neural networks and the Chomsky hierarchy. In *The Eleventh International Conference on Learning Representations*.

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

Javid Ebrahimi, Dhruv Gelda, and Wei Zhang. 2020. How can self-attention networks recognize Dyck-n languages? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4301–4306, Online. Association for Computational Linguistics.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Nicholas Evans and Stephen C Levinson. 2009. The myth of language universals: Language diversity and its importance for cognitive science. *Behavioral and brain sciences*, 32(5):429–448.

Daniel L. Everett. 2012. What does Pirahã grammar have to teach us about human language and the mind? *WIREs Cognitive Science*, 3(6):555–563.

Richard Futrell. 2019. Information-theoretic locality properties of natural language. In *Proceedings of the First Workshop on Quantitative Syntax (Quasy, SyntaxFest 2019)*, pages 2–15, Paris, France. Association for Computational Linguistics.

Richard Futrell and Michael Hahn. 2022. Information theory as a bridge between language function and language form. *Frontiers in Communication*, 7.

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.

Lukas Galke, Yoav Ram, and Limor Raviv. 2023. What makes a language easy to deep-learn?

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. Curran Associates, Inc.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. 2023. Finding alignments between interpretable causal variables and distributed neural representations. In *Proceedings of Causal Learning and Reasoning 2024*.

Joseph Greenberg. 1963. Some universals of grammar with particular reference to the order of meaningful elements. *Universals of Language*, pages 73–113.

Michael Hahn. 2020. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171.

Michael Hahn, Judith Degen, and Richard Futrell. 2021. Modeling word and morpheme order in natural language as an efficient trade-off of memory and surprisal. *Psychological Review*, 128(4):726–756.

Michael Hahn, Dan Jurafsky, and Richard Futrell. 2020. Universals of word order reflect optimization of grammars for efficient communication. *Proceedings of the National Academy of Sciences*, 117(5):2347–2353.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Yiding Hao, William Merrill, Dana Angluin, Robert Frank, Noah Amsel, Andrew Benz, and Simon Mendelsohn. 2018. Context-free transductions with neural stacks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 306–315, Brussels, Belgium. Association for Computational Linguistics.

Marc D. Hauser, Noam Chomsky, and W. Tecumseh Fitch. 2002. The faculty of language: What is it, who has it, and how did it evolve? *Science*, 298(5598):1569–1579.

Jack Hessel and Alexandra Schofield. 2021. How effective is BERT without word ordering? implications for language understanding and data privacy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 204–211, Online. Association for Computational Linguistics.

John Hewitt, Michael Hahn, Surya Ganguli, Percy Liang, and Christopher D. Manning. 2020. RNNs can generate bounded hierarchical languages with optimal memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1978–2010, Online. 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.

Qian Huang, Eric Zelikman, Sarah Li Chen, Yuhuai Wu, Gregory Valiant, and Percy Liang. 2023. Lexinvariant language models.

Lifeng Jin, Finale Doshi-Velez, Timothy Miller, William Schuler, and Lane Schwartz. 2018. Depth-bounding is effective: Improvements and evaluation of unsupervised PCFG induction. In *Proceedings of the*

2018 Conference on Empirical Methods in Natural Language Processing, pages 2721–2731, Brussels, Belgium. Association for Computational Linguistics.

Aravind K. Joshi. 1985. *Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions?*, Studies in Natural Language Processing, page 206–250. Cambridge University Press.

Siddharth* Karamcheti, Laurel* Orr, Jason Bolton, Tianyi Zhang, Karan Goel, Avanika Narayan, Rishi Bommasani, Deepak Narayanan, Tatsunori Hashimoto, Dan Jurafsky, Christopher D. Manning, Christopher Potts, Christopher Ré, and Percy Liang. 2021. Mistral - a journey towards reproducible language model training.

Fred Karlsson. 2007. Constraints on multiple center-embedding of clauses. *Journal of Linguistics*, 43(2):365–392.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. *arXiv preprint arXiv:2305.19466*.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

John Mansfield and Charles Kemp. 2023. The emergence of grammatical structure from inter-predictability.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

William Merrill. 2019. Sequential neural networks as automata. In *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges*, pages 1–13, Florence. Association for Computational Linguistics.

William Merrill, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A. Smith, and Eran Yahav. 2020. A formal hierarchy of RNN architectures. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 443–459, Online. Association for Computational Linguistics.

Jeff Mitchell and Jeffrey Bowers. 2020. Priorless recurrent networks learn curiously. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5147–5158, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Andrea Moro, Matteo Greco, and Stefano F. Cappa. 2023. Large languages, impossible languages and human brains. *Cortex*, 167:82–85.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D. Manning. 2023. Pushdown layers: Encoding recursive structure in transformer language models.

Mariacristina Musso, Andrea Moro, Volkmar Glauche, Michel Rijntjes, Jürgen Reichenbach, Christian Büchel, and Cornelius Weiller. 2003. Broca's area and the language instinct. *Nature Neuroscience*, 6(7):774–781.

Ryan M. Nefdt. 2024. *The Philosophy of Theoretical Linguistics: A Contemporary Outlook*. Cambridge University Press.

Isabel Papadimitriou, Richard Futrell, and Kyle Mahowald. 2022. When classifying grammatical role, BERT doesn't care about word order... except when it matters. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 636–643, Dublin, Ireland. Association for Computational Linguistics.

Isabel Papadimitriou and Dan Jurafsky. 2023. Injecting structural hints: Using language models to study inductive biases in language learning.

Thang Pham, Trung Bui, Long Mai, and Anh Nguyen. 2021. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1145–1160, Online. Association for Computational Linguistics.

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.

Jorge Pérez, Pablo Barceló, and Javier Marinkovic. 2021. Attention is Turing-complete. *Journal of Machine Learning Research*, 22(75):1–35.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Ms, OpenAI.

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

Stuart M. Shieber. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8(3):333–343.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

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.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023. Call for papers – the BabyLM challenge: Sample-efficient pretraining on a developmentally plausible corpus.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision RNNs for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745, Melbourne, Australia. Association for Computational Linguistics.

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. 2023. Using computational models to test syntactic learnability. *Linguistic Inquiry*, pages 1–44.

Zhengxuan Wu, Karel D'Oosterlinck, Atticus Geiger, Amir Zur, and Christopher Potts. 2023a. Causal proxy models for concept-based model explanations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 37313–37334. PMLR.

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

Zhengxuan Wu, Atticus Geiger, Joshua Rozner, Elisa Kreiss, Hanson Lu, Thomas Icard, Christopher Potts, and Noah Goodman. 2022. Causal distillation for language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4295, Seattle, United States. Association for Computational Linguistics.

Supplementary Materials

A Dataset Filters

The BabyLM dataset (Warstadt et al., 2023) is an English-language dataset of about 100 million words intended to approximate the amount of linguistic data available to an English-speaking child. To create a dataset for an impossible language, we first pre-process the BabyLM dataset using Stanza (Qi et al., 2020). We perform sentence segmentation on each dataset file and then extract part-of-speech (POS) and morphological feature tags for all the sentences, which are required for the *Hop transformations. We transform each tagged sentence in the original BabyLM dataset using the impossible language's rule-based perturbation function, as described in Section 3. Depending on the class of the impossible language and the specific features of the input sentence, perturbed sentences may be included or excluded from the final dataset used for model training (see below for details on this filtering). Since we apply these filters, the language classes have datasets of slightly different sizes. The *Shuffle and *Reverse languages have training sets of about 9.69 million sentences, and the *Hop languages have training sets of about 8.43 million sentences.

***Shuffle Filters** For the *Shuffle languages, we filter sentences from the BabyLM dataset such that the set of token sequence lengths seen in the validation and test sets are also seen in the training set. This ensures that any shuffles for the DeterministicShuffle perturbation (which are determined by the token sequence length) in the test set have also occurred at least once in the training set. We apply these filters for all *Shuffle languages such that their datasets are comprised of the same subset of original sentences.

***Reverse Filters** For the *Reverse languages, we do not apply any sentence filtering, so their models are trained on the entire BabyLM dataset.

***Hop Filters** For the *Hop languages, we filter out sentences from the BabyLM dataset that would not allow the special markers to fully complete 4 hops in the TokenHop or WordHop perturbations, i.e. sentences in which a 3rd-person present tense verb is too close to the end of the sentence. We again filter out these sentences from all perturbations, so TokenHop, WordHop, and NoHop are comprised of the same subset of original sentences from the BabyLM dataset.

B GPT-2 Training Details and Hyperparameters

We train GPT-2 small models with a standard training regime (Radford et al., 2018, 2019) using the library of Karamcheti et al. (2021). We mostly use the default GPT-2 small hyperparameters to train our models (context length of 1024, batch size of 512, etc.). We only change the total number of training steps and the number of warm-up steps. We train with a learning rate that linearly warms up from 0 to 6e-4 over 300 steps. While 10% of steps for warm-up is typical for LLM training, we acknowledge that the best warm-up may be different when using a small pretraining dataset, so we also tried 1,000 warm-up steps and 4,000 warm-up steps. (4,000 steps is the GPT-2 default. Since we only train for 3,000 steps, this effectively means we have a learning-rate that linearly warms up from 0 to 4.5e-4.) Using a different warm-up did not change the ranking of impossible language model perplexities.

We train the models for 3,000 training steps, which equates to about 11.03 epochs for the *Shuffle languages, 10.05 epochs for the *Reverse languages, and 12.04 epochs for the *Hop languages. The vocabulary set also varies based on the language. The *Shuffle languages use the standard GPT-2 vocabulary containing 50,257 tokens; the *Reverse languages add one special token $\boxed{\text{R}}$, for a vocabulary size of 50,258; and the *Hop languages add two special tokens $\boxed{\text{S}}$ and $\boxed{\text{P}}$ for verb inflection, for a vocabulary size of 50,259. We train on NVIDIA RTX 3090 (24GB) GPUs and NVIDIA RTX A6000 (48GB) GPUs. The runtime for each pretraining experiment was ~24 hours (for one language and one random seed), for a total experiment runtime of ~1800 hours.

C Results for Models without Positional Encodings

Here, we present results for each of our experiments using GPT-2 models we trained without positional encodings. All other aspects of the experiments are the same, including the impossible language datasets and training hyperparameters. We again train 5 sets of models initialized using different random seeds. Figure 6 presents the perplexity results; Figure 7 presents the surprisal results; and

Figure 8 presents the causal intervention results.

D Constituency Probing Evaluation

We also test how perturbations might influence latent linguistic properties in sentences that are seemingly *unaffected* by the perturbations. For this, we develop a constituency probing experiment to examine whether the contextual representations generated by different models are effective in classifying a sequence of tokens with an appropriate constituent label, similar to the edge probing experiments of Tenney et al. 2019. For example, if the input sentence is "I enjoy strawberry ice cream" and the span of tokens in question represents the constituent "strawberry ice cream," the span should be labeled as a noun phrase (NP).

Setup. We conduct these experiments for *Reverse and *Hop languages, since these languages have constituents in contiguous token sequences. For NoReverse and PartialReverse, we take a sample of unaltered BabyLM test sentences and omit the reversal token $\boxed{R}$. For FullReverse, we use the same sample sentences, but reverse the tokens. For the *Hop languages, we use a sample of BabyLM test sentences that are unaffected by the perturbation, which are sentences that do not contain 3rd-person present tense verbs. To extract constituents for testing, we parse the sample sentences using Stanza's BERT-based consituency parser. We include noun phrases (NP), verb phrases (VP), adjective phrases (ADJP), adverb phrases (ADVP), and prepositional phrases (PP), and we stratify the samples so that there are equal numbers of example constituents for each phrasal category. We obtain a total of 10K examples for probe training and testing for each language class, where an example is comprised of a tokenized sentence, indices of the constituent span, and the constituent label.

Our probes are L2-regularized logistic regression classifiers trained on the span representations of the tokens corresponding to constituents in the examples. To obtain span representations for training the probes, we mean-pool the representations of the tokens within the span. We try extracting representations from GPT-2 by averaging the last four hidden layers of the model or using different layers individually. We train each probe for a maximum of 10 iterations and hold out 20% of constituent examples for testing.

Hypothesis. Constituency probes will achieve higher accuracy for possible languages than impossible ones, in virtue of the fact that the impossible languages are defined by some rules that do not respect constituency boundaries.

Results. The results of the probing experiment using the average of the last four GPT-2 layers are presented in Figure 9. Across *Reverse and *Hop models trained *with* positional encodings, there are not any clear trends indicating that certain models have better representations of constituents than others, as differences among probe accuracies are minimal and unstable across training steps. However, looking closely at the *Reverse models *without* positional encodings, we can see that PartialReverse has significantly lower probe accuracy than the other models up until 2K training steps. We found similar results when using different layers for span representations, as shown in Figure 10. These results might indicate that the *Hop perturbations were too weak to fundamentally affect the models' representations of latent linguistic structure, but quite unnatural reversal rule of the PartialReverse language disturbed consituency boundaries in a way that could not be recovered by GPT-2 models without positional encodings.

E Additional DeterministicShuffle Results

In addition to perplexities of each impossible language model on its own test data, we also obtain perplexities for each DeterministicShuffle model on the NondeterministicShuffle test sample and all other DeterministicShuffle test samples. This measures whether these models have learned to distinguish their own shuffles from other shuffles. We found that this was indeed the case, as shown in the results in Figure 11.

F Confidence Intervals for Interchange Intervention Accuracies

We present the same results of our causal abstraction experiments from Section 4.4, but include confidence intervals for results across models initialized on different random seeds. Figure 12 presents the results for NoHop; Figure 13 presents the results for TokenHop; and Figure 14 presents the results for WordHop. Figures 15, 16, and 17 show the same plots for each *Hop model trained without positional encodings, respectively.

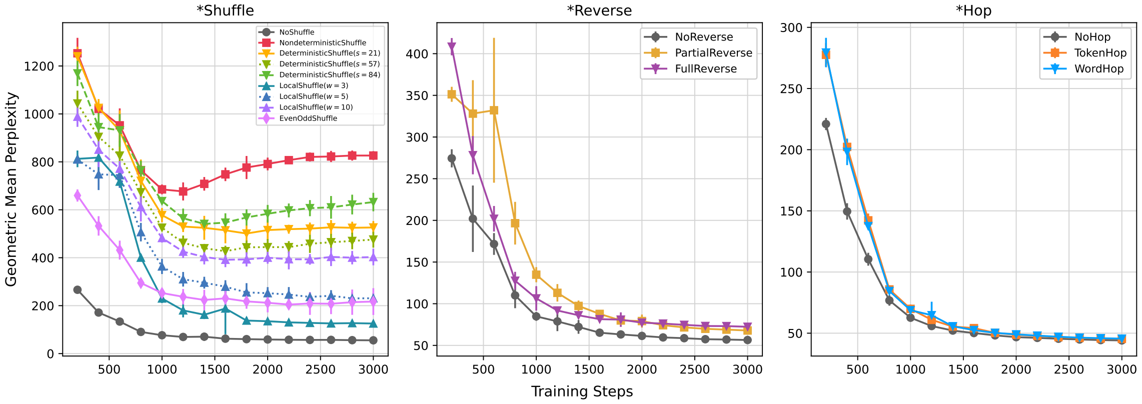
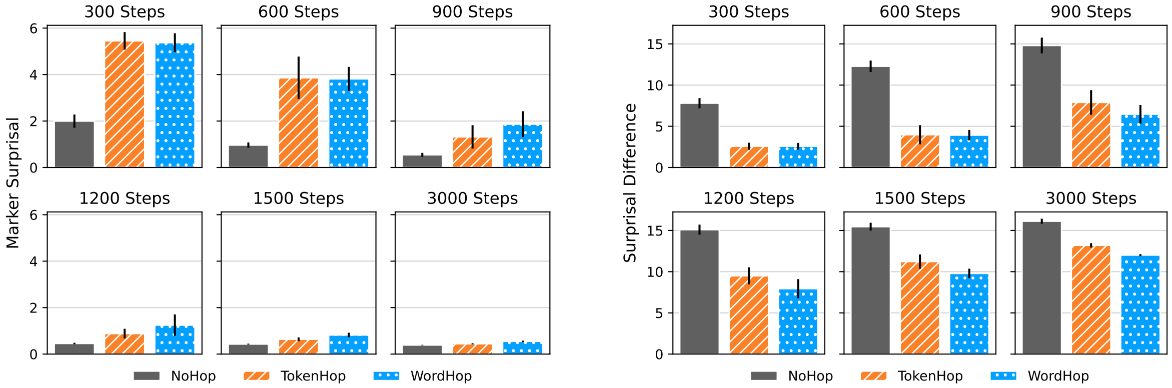


Figure 6: Perplexities on a sample of 10K test sentences for each impossible language model trained *without positional encodings*. Error bars indicate 95% confidence intervals across 5 training runs initialized with different random seeds and evaluated on different test samples.



(a) Mean surprisals of the verb marker token ($\boxed{\text{S}}$ or $\boxed{\text{P}}$) for each *Hop model.

(b) Mean surprisal difference between the verb marker token ($\boxed{\text{S}}$ or $\boxed{\text{P}}$) and the following token for each *Hop model.

Figure 7: Surprisal tests for each *Hop model over training steps (trained *without positional encodings*). Error bars indicate 95% confidence intervals across 5 training runs initialized with different random seeds and evaluated on different test samples.

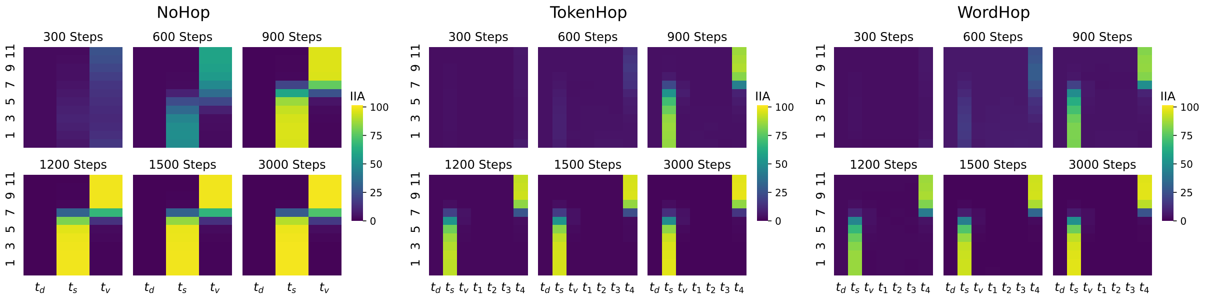
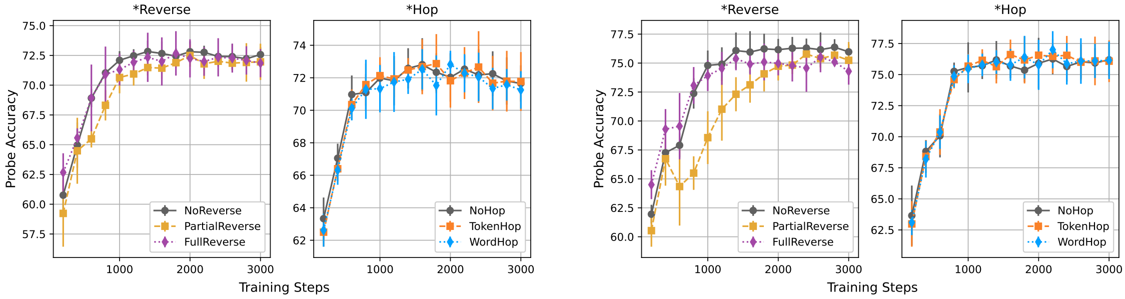


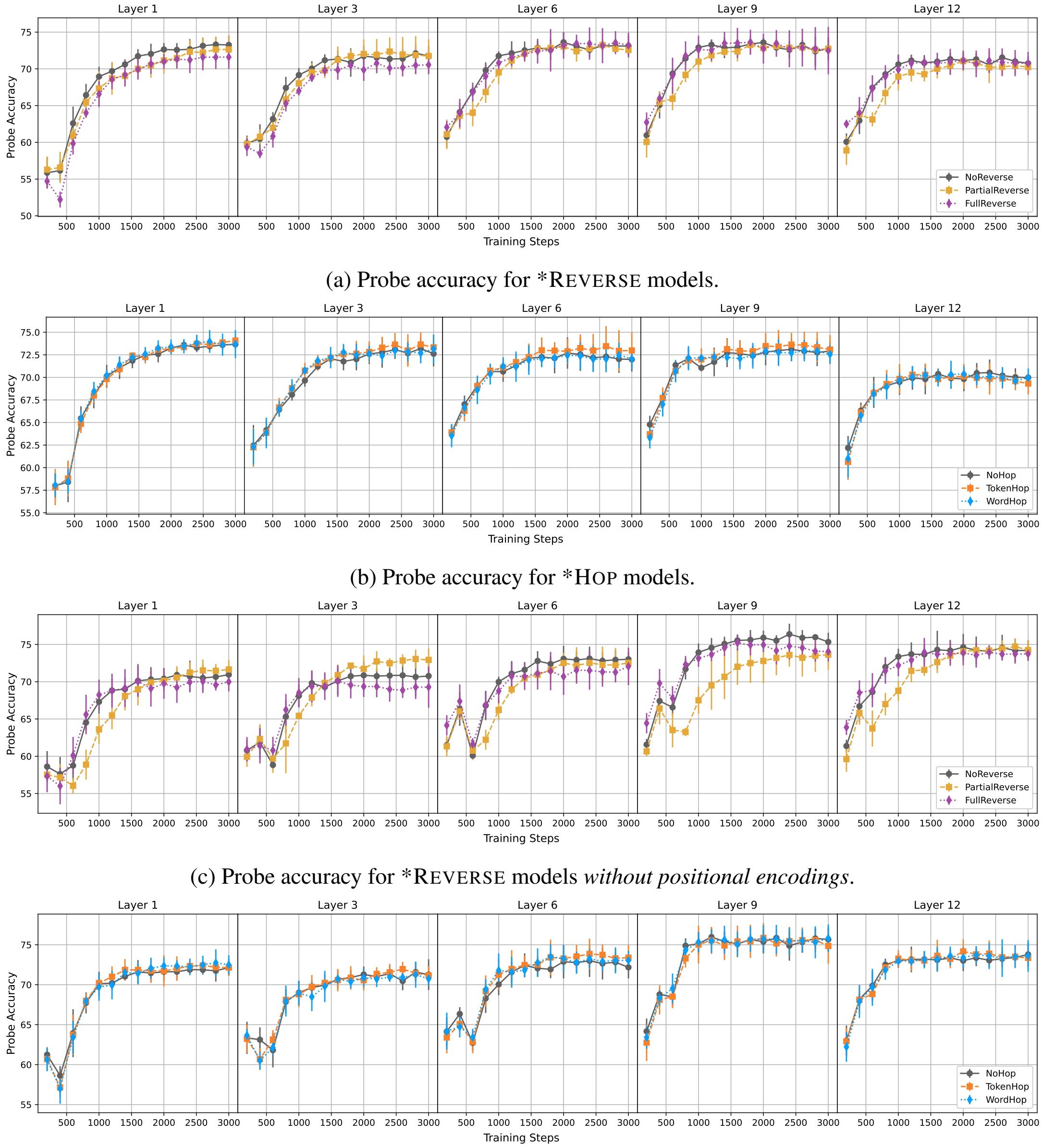
Figure 8: Subject–verb agreement interchange intervention accuracies (IIA) for each *Hop model trained *without positional encodings*. Vertical axes denote the GPT-2 layer of the intervention, and horizontal axes denote the token position of the intervention. t_d, t_s, and t_v represent the tokens for the determiner, subject, and verb, respectively. $t_1 \dots t_4$ represent the four tokens/words between the verb and its marker for TokenHop and WordHop. IIA values are averaged over results from 5 models initialized on different random seeds. See Figures 15, 16, and 17 for confidence intervals.



(a) Probe accuracy for *REVERSE and *HOP models.

(b) Probe accuracy *without positional encodings*.

Figure 9: Constituency probe accuracy for *REVERSE and *HOP models over training steps. Span representations were extracted by averaging the last four hidden layers of GPT-2. Error bars indicate 95% confidence intervals across 5 training runs initialized with different random seeds and evaluated on different test samples.



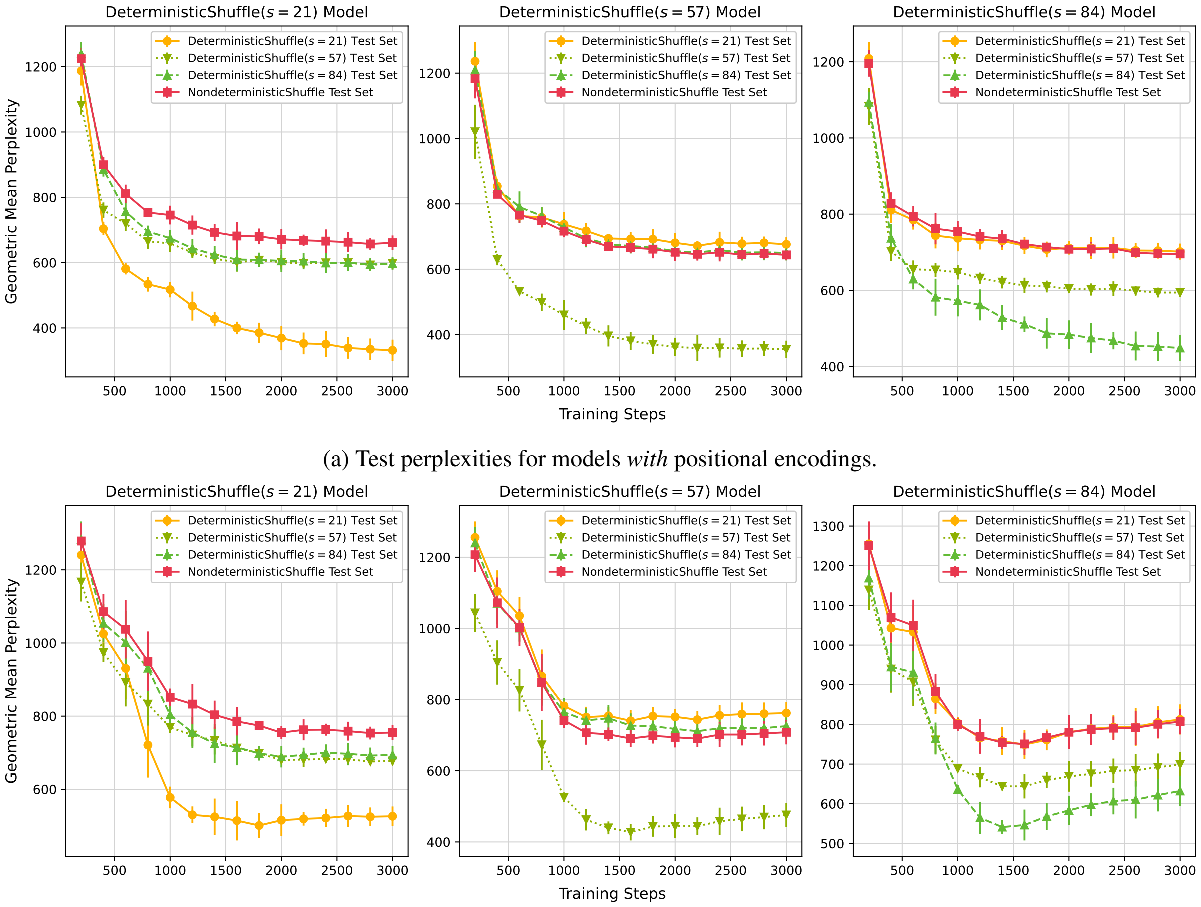
(a) Probe accuracy for *REVERSE models.

(b) Probe accuracy for *HOP models.

(c) Probe accuracy for *REVERSE models *without positional encodings*.

(d) Probe accuracy for *HOP models *without positional encodings*.

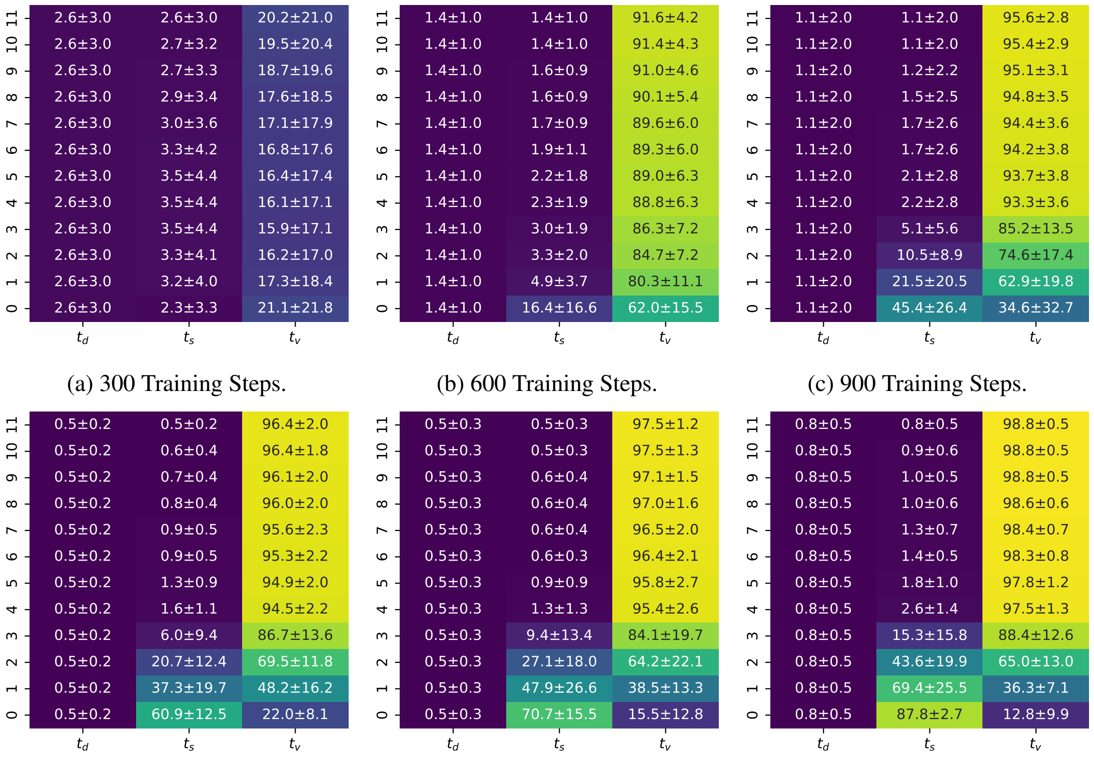
Figure 10: Constituency probe accuracy for *REVERSE and *HOP models using span representations extracted from different GPT-2 layers (1, 3, 6, 9, 12) over training steps. Error bars indicate 95% confidence intervals across 5 training runs initialized with different random seeds and evaluated on different test samples.



(a) Test perplexities for models *with* positional encodings.

(b) Test perplexities for models *without* positional encodings.

Figure 11: Test perplexities for each DETERMINISTICSHUFFLE model ($s = 21$ left, $s = 57$ middle, $s = 84$ right) on the NONDETERMINISTICSHUFFLE test sample and all other DETERMINISTICSHUFFLE test samples. Perplexities were taken on a sample of 10K test sentences from each shuffled test set. Error bars indicate 95% confidence intervals across 5 training runs initialized with different random seeds and evaluated on different test samples.



(a) 300 Training Steps.

(b) 600 Training Steps.

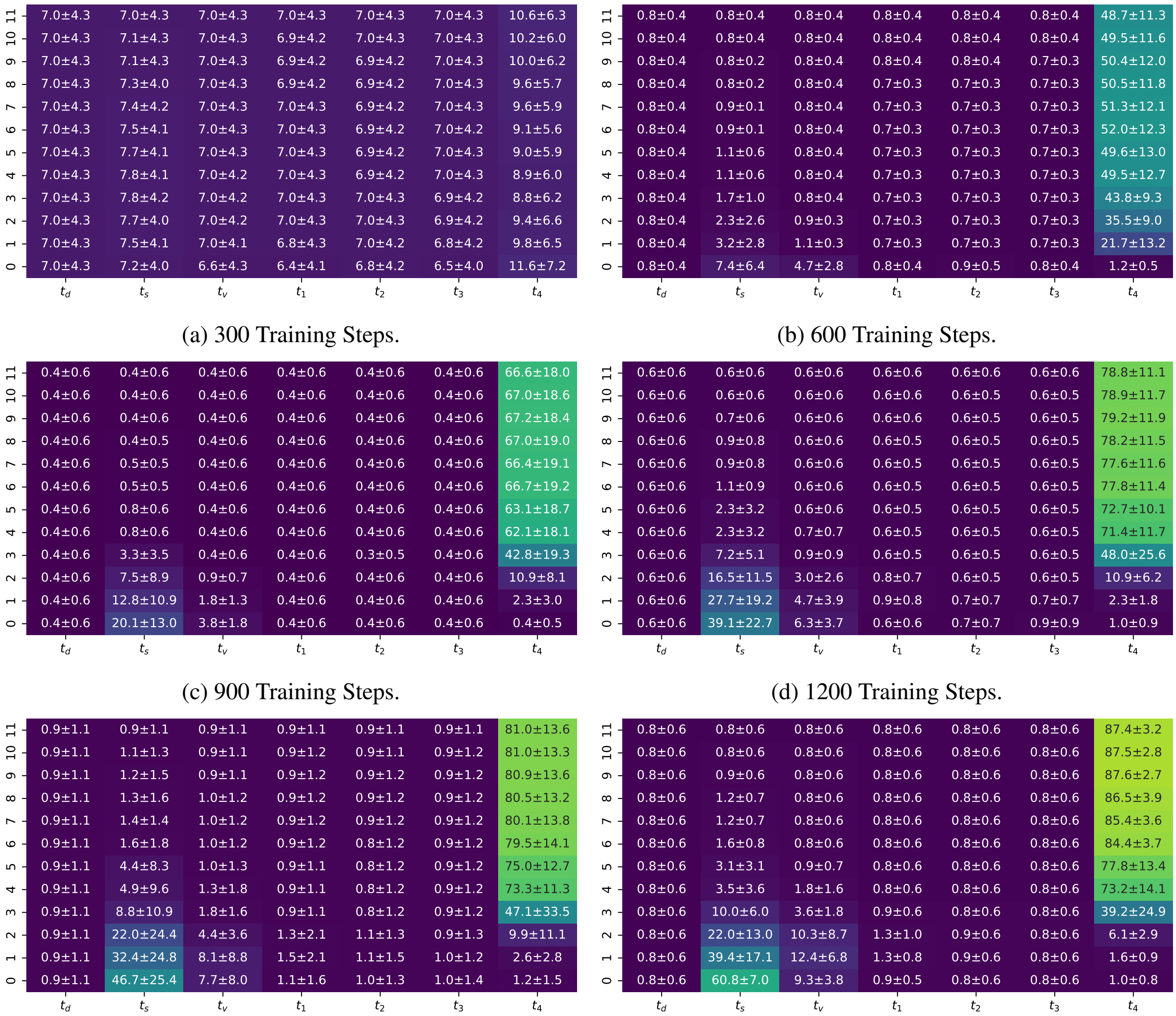
(c) 900 Training Steps.

(d) 1200 Training Steps.

(e) 1500 Training Steps.

(f) 3000 Training Steps.

Figure 12: Subject–verb agreement interchange intervention accuracies (IIA) for NoHop, with confidence intervals across models trained on 5 different random seeds. Vertical axes denote the GPT-2 layer of the intervention, and horizontal axes denote the token position of the intervention. t_d, t_s, and t_v represent the tokens for the determiner, subject, and verb, respectively.



(a) 300 Training Steps.

(b) 600 Training Steps.

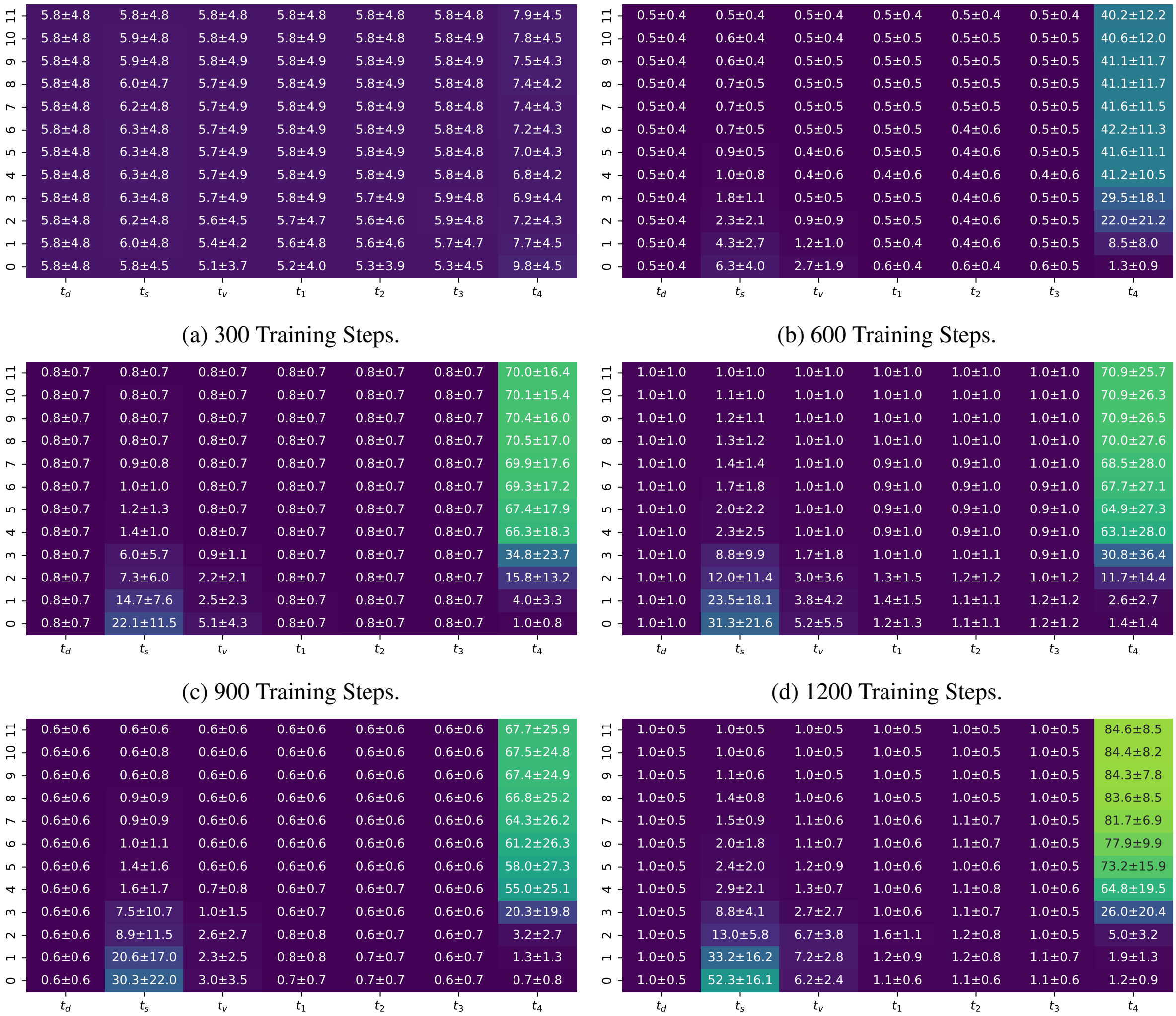
(c) 900 Training Steps.

(d) 1200 Training Steps.

(e) 1500 Training Steps.

(f) 3000 Training Steps.

Figure 13: Subject–verb agreement interchange intervention accuracies (IIA) for TOKENHOP, with confidence intervals across models trained on 5 different random seeds. Vertical axes denote the GPT-2 layer of the intervention, and horizontal axes denote the token position of the intervention. t_d, t_s, and t_v represent the tokens for the determiner, subject, and verb. $t_1 \ldots t_4$ represent the four tokens/words between the verb.



	t_d	t_s	t_v	t_1	t_2	t_3	t_4
11	5.8±4.8	5.8±4.8	5.8±4.8	5.8±4.8	5.8±4.8	5.8±4.8	7.9±4.5
10	5.8±4.8	5.9±4.8	5.8±4.9	5.8±4.9	5.8±4.8	5.8±4.9	7.8±4.5
9	5.8±4.8	5.9±4.8	5.8±4.9	5.8±4.9	5.8±4.9	5.8±4.9	7.5±4.3
8	5.8±4.8	6.0±4.7	5.7±4.9	5.8±4.9	5.8±4.9	5.8±4.8	7.4±4.2
7	5.8±4.8	6.2±4.8	5.7±4.9	5.8±4.9	5.8±4.9	5.8±4.8	7.4±4.3
6	5.8±4.8	6.3±4.8	5.7±4.9	5.8±4.9	5.8±4.9	5.8±4.8	7.2±4.3
5	5.8±4.8	6.3±4.8	5.7±4.9	5.8±4.9	5.8±4.9	5.8±4.8	7.0±4.3
4	5.8±4.8	6.3±4.8	5.7±4.9	5.8±4.9	5.8±4.9	5.8±4.8	6.8±4.2
3	5.8±4.8	6.3±4.8	5.7±4.9	5.8±4.9	5.7±4.9	5.9±4.8	6.9±4.4
2	5.8±4.8	6.2±4.8	5.6±4.5	5.7±4.7	5.6±4.6	5.9±4.8	7.2±4.3
1	5.8±4.8	6.0±4.8	5.4±4.2	5.6±4.8	5.6±4.6	5.7±4.7	7.7±4.5
0	5.8±4.8	5.8±4.5	5.1±3.7	5.2±4.0	5.3±3.9	5.3±4.5	9.8±4.5

(a) 300 Training Steps.

	t_d	t_s	t_v	t_1	t_2	t_3	t_4
11	0.5±0.4	0.5±0.4	0.5±0.4	0.5±0.4	0.5±0.4	0.5±0.4	40.2±12.2
10	0.5±0.4	0.6±0.4	0.5±0.4	0.5±0.5	0.5±0.5	0.5±0.5	40.6±12.0
9	0.5±0.4	0.6±0.4	0.5±0.5	0.5±0.5	0.5±0.5	0.5±0.5	41.1±11.7
8	0.5±0.4	0.7±0.5	0.5±0.5	0.5±0.5	0.5±0.5	0.5±0.5	41.1±11.7
7	0.5±0.4	0.7±0.5	0.5±0.5	0.5±0.5	0.5±0.5	0.5±0.5	41.6±11.5
6	0.5±0.4	0.7±0.5	0.5±0.5	0.5±0.5	0.4±0.6	0.5±0.5	42.2±11.3
5	0.5±0.4	0.9±0.5	0.4±0.6	0.5±0.5	0.4±0.6	0.5±0.5	41.6±11.1
4	0.5±0.4	1.0±0.8	0.4±0.6	0.4±0.6	0.4±0.6	0.4±0.6	41.2±10.5
3	0.5±0.4	1.8±1.1	0.5±0.5	0.5±0.5	0.4±0.6	0.5±0.5	29.5±18.1
2	0.5±0.4	2.3±2.1	0.9±0.9	0.5±0.5	0.4±0.6	0.5±0.5	22.0±21.2
1	0.5±0.4	4.3±2.7	1.2±1.0	0.5±0.4	0.4±0.6	0.5±0.5	8.5±8.0
0	0.5±0.4	6.3±4.0	2.7±1.9	0.6±0.4	0.6±0.4	0.6±0.5	1.3±0.9

(b) 600 Training Steps.

	t_d	t_s	t_v	t_1	t_2	t_3	t_4
11	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	70.0±16.4
10	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	70.1±15.4
9	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	70.4±16.0
8	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	70.5±17.0
7	0.8±0.7	0.9±0.8	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	69.9±17.6
6	0.8±0.7	1.0±1.0	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	69.3±17.2
5	0.8±0.7	1.2±1.3	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	67.4±17.9
4	0.8±0.7	1.4±1.0	0.8±0.7	0.8±0.7	0.8±0.7	0.8±0.7	66.3±18.3
3	0.8±0.7	6.0±5.7	0.9±1.1	0.8±0.7	0.8±0.7	0.8±0.7	34.8±23.7
2	0.8±0.7	7.3±6.0	2.2±2.1	0.8±0.7	0.8±0.7	0.8±0.7	15.8±13.2
1	0.8±0.7	14.7±7.6	2.5±2.3	0.8±0.7	0.8±0.7	0.8±0.7	4.0±3.3
0	0.8±0.7	22.1±11.5	5.1±4.3	0.8±0.7	0.8±0.7	0.8±0.7	1.0±0.8

(c) 900 Training Steps.

	t_d	t_s	t_v	t_1	t_2	t_3	t_4
11	1.0±1.0	1.0±1.0	1.0±1.0	1.0±1.0	1.0±1.0	1.0±1.0	70.9±25.7
10	1.0±1.0	1.1±1.0	1.0±1.0	1.0±1.0	1.0±1.0	1.0±1.0	70.9±26.3
9	1.0±1.0	1.2±1.1	1.0±1.0	1.0±1.0	1.0±1.0	1.0±1.0	70.9±26.5
8	1.0±1.0	1.3±1.2	1.0±1.0	1.0±1.0	1.0±1.0	1.0±1.0	70.0±27.6
7	1.0±1.0	1.4±1.4	1.0±1.0	0.9±1.0	0.9±1.0	1.0±1.0	68.5±28.0
6	1.0±1.0	1.7±1.8	1.0±1.0	0.9±1.0	0.9±1.0	0.9±1.0	67.7±27.1
5	1.0±1.0	2.0±2.2	1.0±1.0	0.9±1.0	0.9±1.0	0.9±1.0	64.9±27.3
4	1.0±1.0	2.3±2.5	1.0±1.0	0.9±1.0	0.9±1.0	0.9±1.0	63.1±28.0
3	1.0±1.0	8.8±9.9	1.7±1.8	1.0±1.0	1.0±1.1	0.9±1.0	30.8±36.4
2	1.0±1.0	12.0±11.4	3.0±3.6	1.3±1.5	1.2±1.2	1.0±1.2	11.7±14.4
1	1.0±1.0	23.5±18.1	3.8±4.2	1.4±1.5	1.1±1.1	1.2±1.2	2.6±2.7
0	1.0±1.0	31.3±21.6	5.2±5.5	1.2±1.3	1.1±1.1	1.2±1.2	1.4±1.4

(d) 1200 Training Steps.

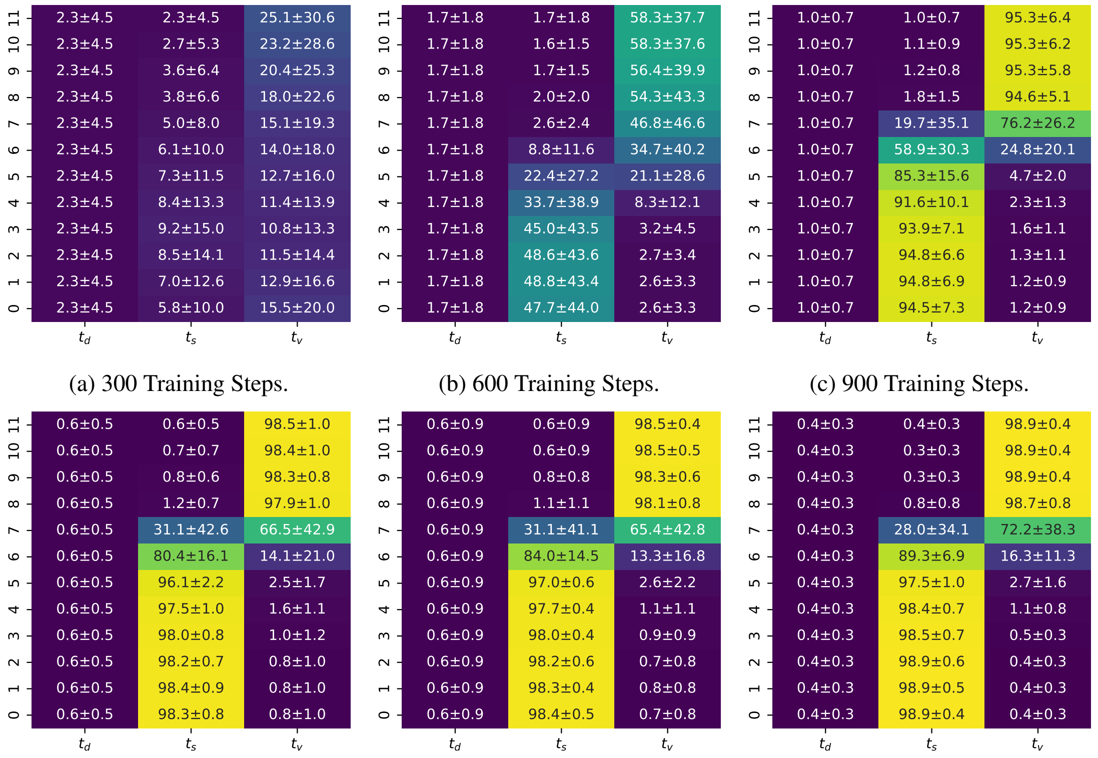
	t_d	t_s	t_v	t_1	t_2	t_3	t_4
11	0.6±0.6	0.6±0.6	0.6±0.6	0.6±0.6	0.6±0.6	0.6±0.6	67.7±25.9
10	0.6±0.6	0.6±0.8	0.6±0.6	0.6±0.6	0.6±0.6	0.6±0.6	67.5±24.8
9	0.6±0.6	0.6±0.8	0.6±0.6	0.6±0.6	0.6±0.6	0.6±0.6	67.4±24.9
8	0.6±0.6	0.9±0.9	0.6±0.6	0.6±0.6	0.6±0.6	0.6±0.6	66.8±25.2
7	0.6±0.6	0.9±0.9	0.6±0.6	0.6±0.6	0.6±0.6	0.6±0.6	64.3±26.2
6	0.6±0.6	1.0±1.1	0.6±0.6	0.6±0.6	0.6±0.6	0.6±0.6	61.2±26.3
5	0.6±0.6	1.4±1.6	0.6±0.6	0.6±0.6	0.6±0.6	0.6±0.6	58.0±27.3
4	0.6±0.6	1.6±1.7	0.7±0.8	0.6±0.7	0.6±0.7	0.6±0.6	55.0±25.1
3	0.6±0.6	7.5±10.7	1.0±1.5	0.6±0.7	0.6±0.6	0.6±0.6	20.3±19.8
2	0.6±0.6	8.9±11.5	2.6±2.7	0.8±0.8	0.6±0.7	0.6±0.7	3.2±2.7
1	0.6±0.6	20.6±17.0	2.3±2.5	0.8±0.8	0.7±0.7	0.6±0.7	1.3±1.3
0	0.6±0.6	30.3±22.0	3.0±3.5	0.7±0.7	0.7±0.7	0.6±0.7	0.7±0.8

(e) 1500 Training Steps.

	t_d	t_s	t_v	t_1	t_2	t_3	t_4
11	1.0±0.5	1.0±0.5	1.0±0.5	1.0±0.5	1.0±0.5	1.0±0.5	84.6±8.5
10	1.0±0.5	1.0±0.6	1.0±0.5	1.0±0.5	1.0±0.5	1.0±0.5	84.4±8.2
9	1.0±0.5	1.1±0.6	1.0±0.5	1.0±0.5	1.0±0.5	1.0±0.5	84.3±7.8
8	1.0±0.5	1.4±0.8	1.0±0.6	1.0±0.5	1.0±0.5	1.0±0.5	83.6±8.5
7	1.0±0.5	1.5±0.9	1.1±0.6	1.0±0.6	1.1±0.7	1.0±0.5	81.7±6.9
6	1.0±0.5	2.0±1.8	1.1±0.7	1.0±0.6	1.1±0.7	1.0±0.5	77.9±9.9
5	1.0±0.5	2.4±2.0	1.2±0.9	1.0±0.6	1.0±0.6	1.0±0.5	73.2±15.9
4	1.0±0.5	2.9±2.1	1.3±0.7	1.0±0.6	1.1±0.8	1.0±0.6	64.8±19.5
3	1.0±0.5	8.8±4.1	2.7±2.7	1.0±0.6	1.1±0.7	1.0±0.5	26.0±20.4
2	1.0±0.5	13.0±5.8	6.7±3.8	1.6±1.1	1.2±0.8	1.0±0.5	5.0±3.2
1	1.0±0.5	33.2±16.2	7.2±2.8	1.2±0.9	1.2±0.8	1.1±0.7	1.9±1.3
0	1.0±0.5	52.3±16.1	6.2±2.4	1.1±0.6	1.1±0.6	1.1±0.6	1.2±0.9

(f) 3000 Training Steps.

Figure 14: Subject–verb agreement interchange intervention accuracies (IIA) for WORDHOP, with confidence intervals across models trained on 5 different random seeds. Vertical axes denote the GPT-2 layer of the intervention, and horizontal axes denote the token position of the intervention. t_d, t_s, and t_v represent the tokens for the determiner, subject, and verb. $t_1 \ldots t_4$ represent the four tokens/words between the verb.



(a) 300 Training Steps.

(b) 600 Training Steps.

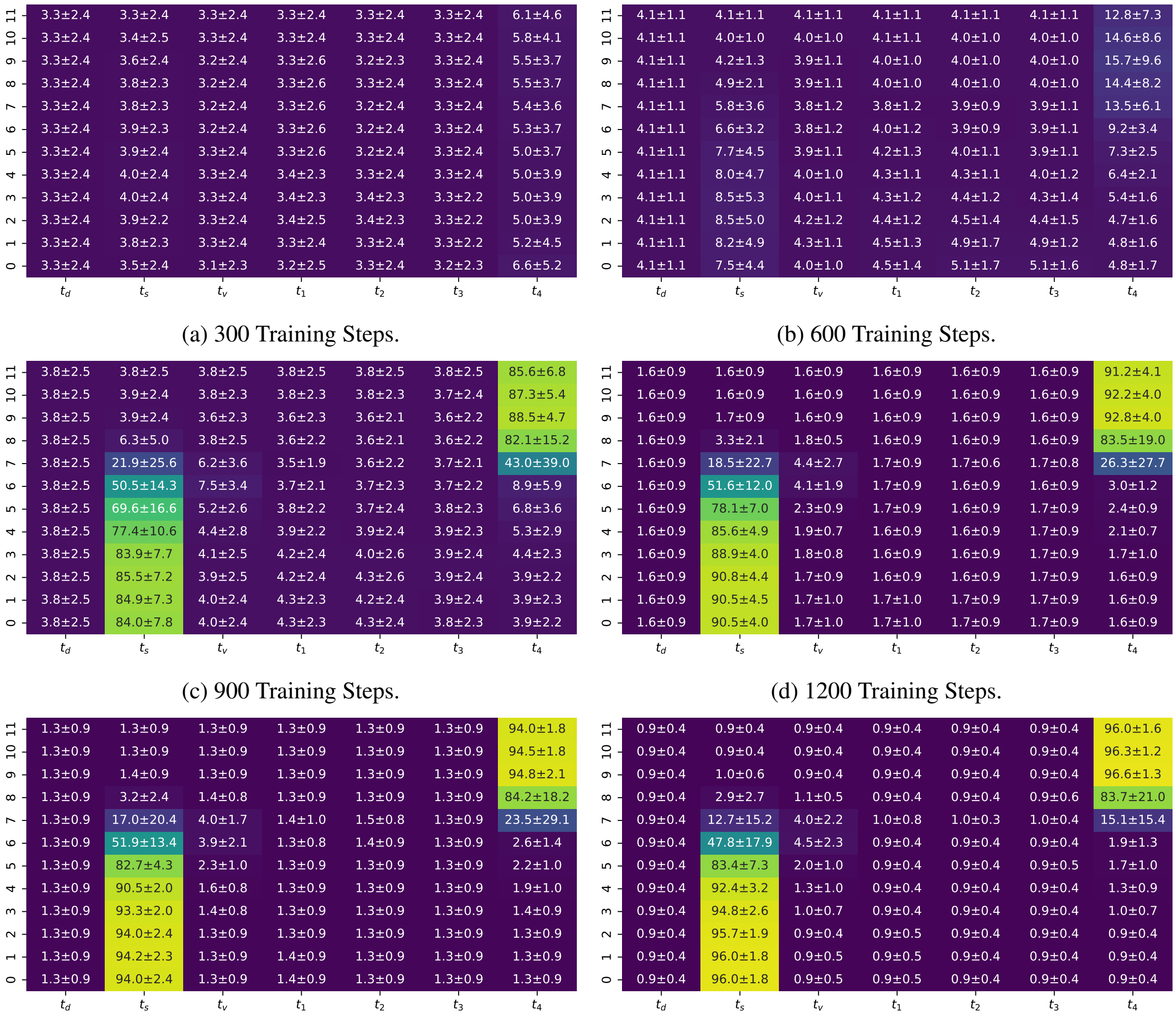
(c) 900 Training Steps.

(d) 1200 Training Steps.

(e) 1500 Training Steps.

(f) 3000 Training Steps.

Figure 15: Subject–verb agreement interchange intervention accuracies (IIA) for the NoHop model trained *without positional encodings*, with confidence intervals across models trained on 5 different random seeds. Vertical axes denote the GPT-2 layer of the intervention, and horizontal axes denote the token position of the intervention. t_d, t_s, and t_v represent the tokens for the determiner, subject, and verb, respectively.



(a) 300 Training Steps.

(b) 600 Training Steps.

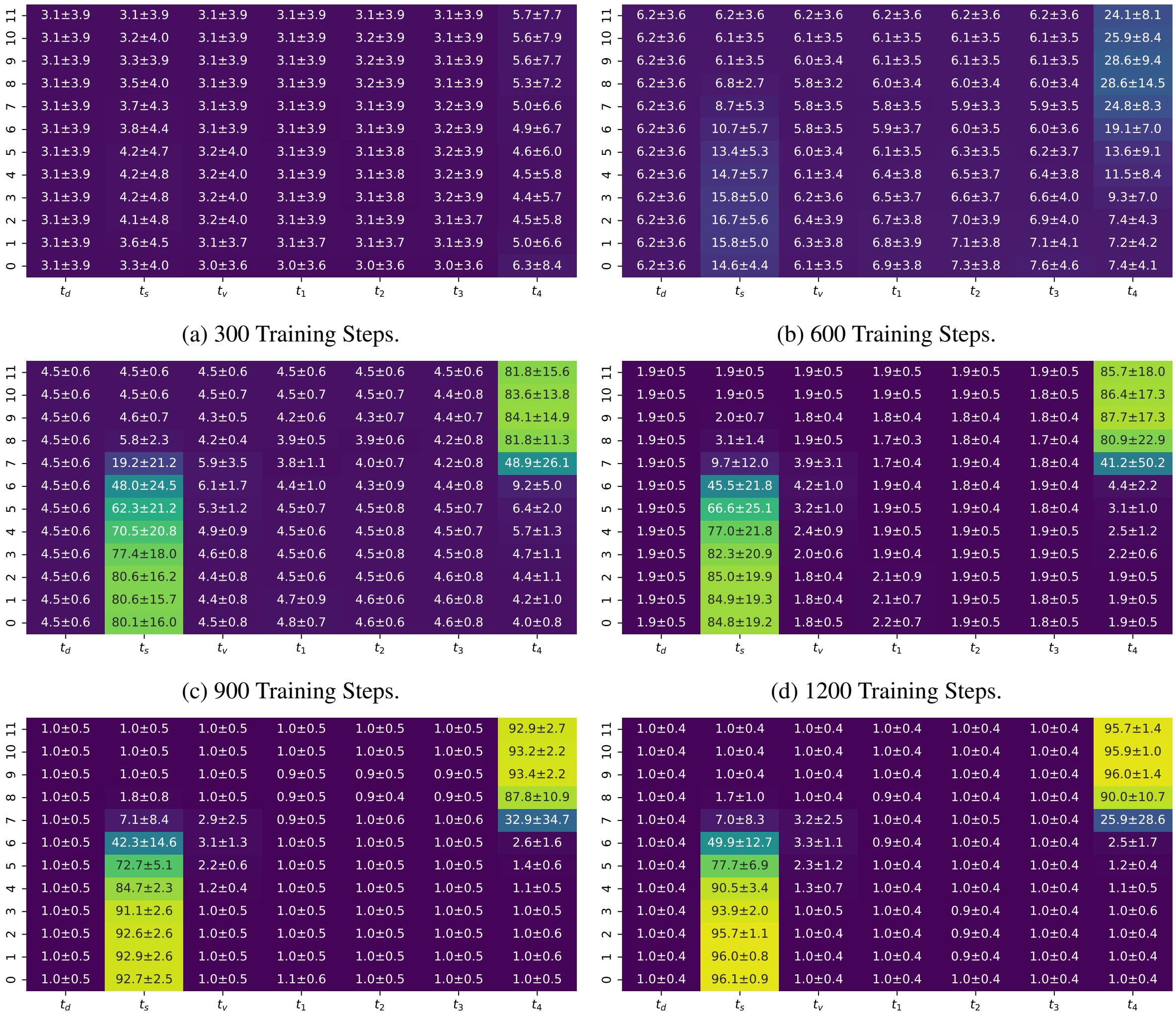
(c) 900 Training Steps.

(d) 1200 Training Steps.

(e) 1500 Training Steps.

(f) 3000 Training Steps.

Figure 16: Subject–verb agreement interchange intervention accuracies (IIA) for the TOKENHOP model trained *without positional encodings*, with confidence intervals across models trained on 5 different random seeds. Vertical axes denote the GPT-2 layer of the intervention, and horizontal axes denote the token position of the intervention. t_d, t_s, and t_v represent the tokens for the determiner, subject, and verb. $t_1 \ldots t_4$ represent the four tokens/words between the verb.



(a) 300 Training Steps.

(b) 600 Training Steps.

(c) 900 Training Steps.

(d) 1200 Training Steps.

(e) 1500 Training Steps.

(f) 3000 Training Steps.

Figure 17: Subject–verb agreement interchange intervention accuracies (IIA) for the WORDHOP model trained *without positional encodings*, with confidence intervals across models trained on 5 different random seeds. Vertical axes denote the GPT-2 layer of the intervention, and horizontal axes denote the token position of the intervention. t_d, t_s, and t_v represent the tokens for the determiner, subject, and verb. $t_1 \ldots t_4$ represent the four tokens/words between the verb.