Causal Interventions on Continuous Features in LLMs: A Case Study in Verb Bias

Zhenghao Herbert Zhou **R. Thomas McCoy** **Robert Frank**
Department of Linguistics
Yale University
New Haven, CT 06511, USA
{herbert.zhou, tom.mccoy, robert.frank}@yale.edu

Abstract

We investigate how large language models (LLMs) encode and causally use continuous, context-dependent properties in syntactic processing, focusing on verb bias and its influence on structural priming. Building on prior work that localized binary morphosyntactic features in non-basis-aligned subspaces, we introduce a simple and efficient method combining principal component analysis with beta regression to identify verb-bias subspaces in function vectors. Function vectors are compact task representations derived from in-context learning sequences. We show that function vectors can be employed to simulate structural priming in LLMs. Our method supports counterfactual continuous manipulation of the verb-bias subspace, and doing so yields the predicted shifts in priming magnitudes, confirming that the subspace is causally involved in syntactic choice. Our method thus extends causal interpretability methods to continuous linguistic variables, and our application of this method supports the proposal that the same mechanism is responsible for in-context learning and structural priming in LLMs.

1 Introduction

Causal interpretability methods have emerged as a promising method for identifying internal representations and processing mechanisms in neural networks [8]. Such work proceeds by localizing linguistic features in the representations of large language models (LLMs) and verifying their role in processing through causal intervention. Previous studies have focused on *discrete, local* properties — morphosyntactic features (with a small number of possible values) that are fully determined by the local sentence. Examples of features studied in this way include relative clause boundary information [12], the subject number feature [4], Hindi case agreement [9], filler-gap dependencies [1], and others. In this study, we extend the methods of causal intervention to *continuous, context-dependent* properties. Since many aspects of human language processing and conceptual knowledge are inherently graded and contextually variable, expanding in this direction is important for enabling causal interventions to handle a broader range of features that might influence LLMs.

We present a case study based on **verb bias**, a term which refers to the preference for a verb to occur with one semantically equivalent structure as compared to another. We focus here on two structures associated with ditransitive verbs: the double object (DO) '*give* [$_{\text{NP}}$ *the child*] [$_{\text{NP}}$ *the ball*]' and the preposition dative (PD) '*give* [$_{\text{NP}}$ *the ball*] [$_{\text{PP}}$ *to the child*]'. The strength of the preference varies continuously and is modulated by a number of factors including verb-specific preferences [5]. As a result, the representation of verb bias must encode a continuous range of values. We hypothesize that this continuous information is encoded in a subspace of the LLM's hidden states. To test this hypothesis, we present a simple, efficient, supervised search method involving

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: First Workshop on CogInterp: Interpreting Cognition in Deep Learning Models.

principal component analysis (PCA) and beta regression that is designed for identifying and causally manipulating continuous variables, thereby complementing previous work.

To operationalize the causal effect of verb bias, we leverage the psycholinguistic phenomenon of structural priming (see 2 for a review), the tendency for speakers to reproduce recently encountered syntactic structures (primes). In the case of ditransitives, encountering a DO sentence increases the likelihood that a speaker will produce another DO sentence following a subsequent verb as compared to the semantically equivalent PD structure, and vice versa. Structural priming is robustly found both in humans (e.g., 10) and in neural language models [15, 11, 13, 7, 16]. Because the lexical properties of the priming verb impact the size of the priming effect, we can use changes in the amount of priming as an indicator of successful causal manipulation of verb bias.

To simulate priming in LLMs, we use function vectors, a compact representations of in-context learning (ICL) task representations [6, 14]. Zhou et al. [16] propose that structural priming should be conceptualized as a form of ICL, where the "task" is implicitly defined as "repeat the structure from the context" [3]. It is thus natural to apply the function vector method on structural priming to obtain an abstract representation of structures from context. There are two motivations for this choice of method: (1) Function vectors have been demonstrated to capture abstract representations from context, which fits our need to study the context-dependent property of verb bias. (2) The current study tests Zhou et al.'s proposal that *ICL and structural priming leverage the same types of mechanisms*, as function vectors were originally proposed as a tool for studying ICL.

2 Experiment 1: Using Function Vectors to Induce Structural Priming

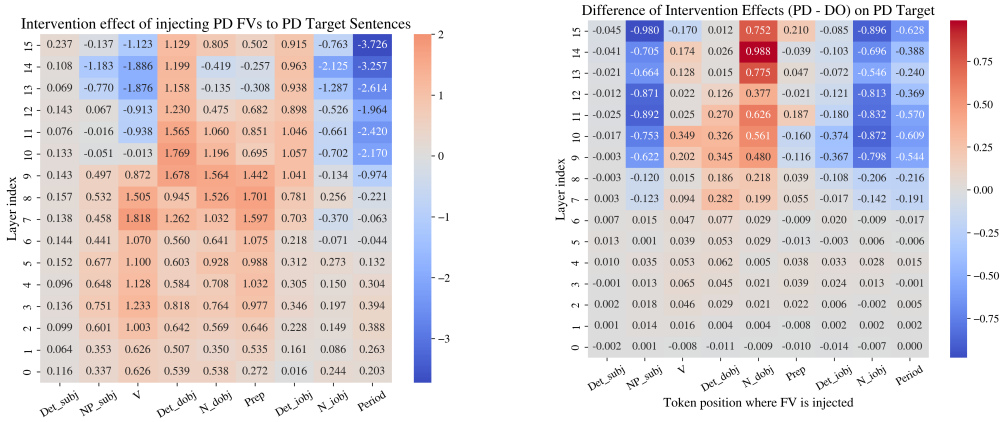
We first verify the feasibility of extracting function vectors to simulate structural priming.

Extracting Function Vectors We use the dataset from Zhou et al. [16], which is adapted from the Core Dative PRIME-LM Corpus from Sinclair et al. [13]. The dataset contains 22 ditransitive verbs and 23,100 synthetically generated DO and PD sentences. Following [6], we define an ICL sequence as a concatenation of demonstrations (primes) followed by a query (target). For a verb-structure combination $\langle V, S \rangle$, we sample 11 sentences with verb V and structure S without lexical repetition for content words (except for the verb V) and with counterbalanced function words. The first 10 sentences are concatenated and serve as the prime sentence context, and they are followed by the single remaining sentence that serves as the target sentence. For every layer l and token position i of the target sentence, we extract the hidden state h_i^l. We average the h_i^l obtained from 10 ICL sequences of the same $\langle V, S \rangle$ to obtain a set of function vectors $\text{FV}_{i,\langle V,S \rangle}^l$. We hypothesize that a function vector obtained in this way encodes abstract information about the "implicitly defined task", namely, producing the next sentence with structure S influenced by verb V.

Simulating Priming with Full-Layer Intervention To simulate structural priming, given a target sentence s, we inject $\text{FV}_{i,\langle V,S \rangle}^l$ to layer l and token position i one at a time by adding $\text{FV}_{i,\langle V,S \rangle}^l$ to the corresponding original hidden state of s. We define the difference between the negative log probability of s before and after FV intervention to be the priming effect: $\text{PE}_{\text{FV}} = \sum_i -\log P(w_i|\overline{w}_{<i})_{\text{FV}} - \sum_i -\log P(w_i|\overline{w}_{<i})_{\text{raw}}$ for each token w_i in s.

Predictions Since a function vector is hypothesized to encode the contextual influence of $\langle V, S \rangle$, two predictions should hold if the function vector succeeds in inducing structural priming: (1) $\text{PE}_{\text{FV}} > 0$, so that the FV intervention increases the probability of the target sentence in all cases we study (because they all involve primes and targets that are structurally related or structurally identical); (2) a larger PE when function vectors encode the same structure as the target sentence: $\text{PE}_{\text{FV}_{\text{PD}}} > \text{PE}_{\text{FV}_{\text{DO}}}$ with PD target sentences, and $\text{PE}_{\text{FV}_{\text{DO}}} > \text{PE}_{\text{FV}_{\text{PD}}}$ with DO target sentences.

Results We scale up from previous studies and examine the Llama family of models: Llama-2-13B, 3-1B, 3-8B. In our figures, we display PE results for Llama3-1B for the verb *show*; the same general patterns hold across verbs and models. Figure 1a shows the average PE of 50 $\langle show, \text{PD} \rangle$ function vectors on 100 PD target sentences. We observed large, positive PE at the positions of the verb through preposition tokens—the region that particularly differentiates DO vs. PD sentences—suggesting that injecting function vectors of the same structure indeed increases the target sentence probability, thereby producing the priming effect. The fact that function vectors are most effective at middle layers



(a) PE of PD function vectors on PD targets.

(b) PE difference between PD and DO function vectors on PD targets.

Figure 1: Priming Effect of injecting function vectors at every layer and token position (Llama3-1B).

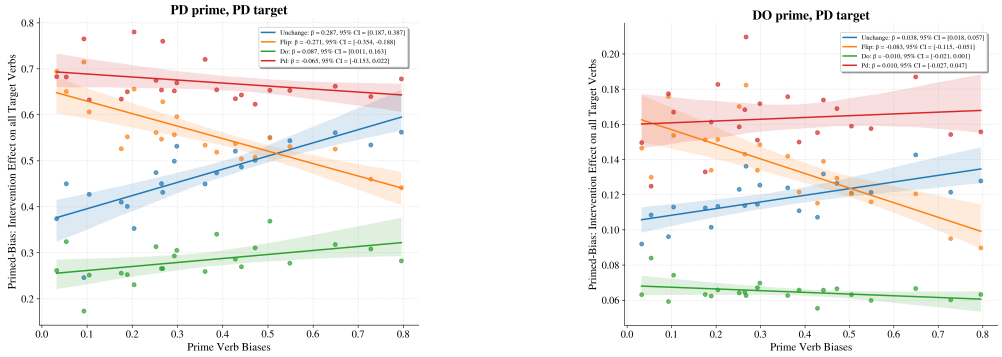
($l = 7$ through 9) also suggests that the abstract structural representations are formed in middle layers. To test prediction (2), Figure 1b shows the difference between average $\text{PE}_{\text{FV}_{\text{PD}}}$ and $\text{PE}_{\text{FV}_{\text{DO}}}$. Large positive differences are observed at intermediate to later layers ($l = 7$ through 15) at the position of the end of the first object argument, suggesting that function vectors are most effective in steering the structural choice when deciding on preposition choice and indirect object argument animacy. Earlier layers do not display a substantial difference between the two prime structures, suggesting again that the abstract structural representations are formed at intermediate layers. In sum, we confirm both predictions and conclude that function vectors are effective in inducing structural priming behaviors. Because function vectors are generally viewed as a signature of ICL, the fact that function vectors also work for structural priming supports the proposal that structural priming is a form of ICL.

3 Experiment 2: Causal Intervention on Verb Bias

The results from Experiment 1 indicate that function vectors encode abstract, structural, context-sensitive information that impacts structural choices. In Experiment 2, we aim to: (1) localize how verb bias is encoded in function vectors; and (2) investigate whether verb bias information has a causal effect in subsequent production. Given our primary interest in studying the effect of verb bias, we select the function vector at the *Verb* position from the *best* layer (i.e., the layer where the function vector induces the largest PE) to represent the batch of ICL sequences it is extracted from.

Measuring Verb Bias and Priming Magnitude of Individual Prime Verbs Following Zhou et al. [16], we quantify individual verbs' verb biases by computing the probability ratio between a PD sentence and its DO counterpart. Given a set of sentences S_V with verb V that contains pairs of semantically equivalent sentences $(t_{\text{PD}}, t_{\text{DO}})$, the PD-bias of verb V is the mean normalized probability of sentences in structure PD: $bias(V, \text{PD}) = \frac{1}{|\mathcal{S}_V|} \sum_{t_{\text{PD}} \in \mathcal{S}_V} \frac{\mathcal{P}(t_{\text{PD}})}{\mathcal{P}(t_{\text{PD}}) + \mathcal{P}(t_{\text{DO}})}$, where $\mathcal{P}(t_{\text{PD}})$ is the probability of sentence t_{PD} assigned by the LLMs. Correspondingly, we define the priming effect of an individual prime verb V as $\text{PrimeEffect}(\text{PD}|\text{DO}, V) = \frac{1}{|T_{\text{PD}}| \cdot |P^V_{\text{DO}}|} \sum_{t_{\text{PD}} \in T_{\text{PD}}} \sum_{p^V_{\text{DO}} \in P^V_{\text{DO}}} \frac{\mathcal{P}(t_{\text{PD}}|p^V_{\text{DO}})}{\mathcal{P}(t_{\text{DO}}|p^V_{\text{DO}}) + \mathcal{P}(t_{\text{PD}}|p^V_{\text{DO}})}$ (we refer readers to Zhou et al. [16] for detailed explanation).

Finding the Verb Bias Subspace Unlike previous efforts to identify non-basis-aligned subspaces for discrete linguistic properties [12, 4, 9, 1], we take a hybrid approach of unsupervised and supervised probing methods that fits the goal of localizing continuous properties. For each verb and structure in our dataset, we extract 50 function vectors using the method in Section 2. We then reduce the dimensionality of function vectors into 50 principal components (PCs) using principal component analysis. We then perform beta regression using the 50 PC values as features to predict the verb biases of the corresponding verbs. We retain PCs whose coefficients have $p < 0.001$ and rank them in terms of $\sqrt{\lambda_i}|\beta_i|$, where λ_i is the sample variance PC_i explains and β_i is the β-coefficient for



(a) PD function vectors applied to PD targets.

(b) DO function vectors applied to PD targets.

Figure 2: Priming magnitudes of the 22 verbs across the 4 causal manipulation conditions.

PC_i. Top-ranked PCs are predicted to have the largest impact on perturbing verb bias. We take the top-ranked PC, PC^*, as a 1D non-basis-aligned subspace that encodes verb bias information.

Causal Manipulation of Verb Biases To investigate whether the identified verb bias subspace is causally implicated in LLM processing, we build on results from Experiment 1 and leverage the behavioral correlation between verb bias and priming magnitude. Specifically, we create counterfactual function vectors by scaling the verb bias values within the identified subspace of the original function vector. Given the continuous nature of verb bias, we consider the following conditions: (1) `Unchange`: the control condition keeping the original verb bias information; (2) `Flip`: given a verb with bias α, modify it to $1-\alpha$; (3) `Fully Biased`: for all verbs, modify their biases to 1 (fully PD-biased) or 0 (fully DO-biased). We compute a scaling factor for the verb bias subspace based on its beta-coefficient β^* and the difference between the current verb bias L and the target verb bias L': $\Delta^* = c\frac{L-L'}{\beta^*}$, where c is a constant. We predict that the priming magnitudes of `Flip` and `Unchange` will be inversely correlated, while there should be no difference across verbs with respect to priming magnitudes in the two `Fully Biased` conditions as the difference in verb bias has been eliminated.

Results For each of the 4 conditions, we fit a line of PrimingEffect(V) against $bias(V, \text{PD})$ to show the correlation between verb bias and priming magnitude. The positive slope for the `Unchange` condition and negative slope for the `Flip` condition suggest that the correlation between verb bias and priming magnitude is effectively inverted through our manipulation of the counterfactual function vectors encoding verb bias. The two essentially flat slopes that are observed for the two `Fully Biased` (DO and PD) conditions suggest that when the relevant subspace of the counterfactual function vector's subspace is saturated, existing difference in verb biases are eliminated. Figures 2a and 2b share the same slope pattern, while the intercept of 2a is significantly higher than that of 2b, suggesting the standard priming effect (PD has a larger priming magnitude than DO on PD target). In sum, continuous causal manipulation in the identified verb bias subspace shows the predicted downstream effect on priming magnitude. We conclude that the verb bias subspace identified with our method is causally involved in the production of target sentences.

4 Discussion and Conclusions

The present study extends causal interpretability in LLMs from discrete morphosyntactic features to continuous, context-dependent variables. Using function vectors, we have shown that abstract contextual information associated with verb bias is compactly represented and manipulable, and that gradient manipulations of the associated subspaces play a predictable causal role in downstream production choices. At the methodological level, the PCA+beta regression framework offers a scalable and interpretable approach for probing continuous properties in neural representations. On the empirical side, our findings support the proposal that in-context learning mechanisms and structural priming share a representational basis. Future work could test whether similar subspace manipulations can uncover the causal role of other gradient linguistic features. Together, this work

highlights the potential of combining causal interventions with psycholinguistic paradigms to yield deeper insights into the interpretability of the underlying mechanisms in LLMs.

References

[1] Boguraev, S., Potts, C., and Mahowald, K. (2025). Causal interventions reveal shared structure across english filler-gap constructions. *arXiv preprint arXiv:2505.16002*.

[2] Branigan, H. P. and Pickering, M. J. (2017). An experimental approach to linguistic representation. *Behavioral and Brain Sciences*, 40:e282.

[3] Chen, Y., Zhao, C., Yu, Z., McKeown, K., and He, H. (2024). Parallel structures in pre-training data yield in-context learning. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8582–8592, Bangkok, Thailand. Association for Computational Linguistics.

[4] Hao, S. and Linzen, T. (2023). Verb conjugation in transformers is determined by linear encodings of subject number. arxiv.

[5] Hawkins, R., Yamakoshi, T., Griffiths, T., and Goldberg, A. (2020). Investigating representations of verb bias in neural language models. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4653–4663, Online. Association for Computational Linguistics.

[6] Hendel, R., Geva, M., and Globerson, A. (2023). In-context learning creates task vectors. *arXiv preprint arXiv:2310.15916*.

[7] Jumelet, J., Zuidema, W., and Sinclair, A. (2024). Do language models exhibit human-like structural priming effects? *arXiv preprint arXiv:2406.04847*.

[8] Mueller, A., Brinkmann, J., Li, M., Marks, S., Pal, K., Prakash, N., Rager, C., Sankaranarayanan, A., Sharma, A. S., Sun, J., et al. (2024). The quest for the right mediator: A history, survey, and theoretical grounding of causal interpretability. *arXiv preprint arXiv:2408.01416*.

[9] Ozaki, S., Bhatt, R., and Dillon, B. (2025). A lstm language model learns hindi-urdu case-agreement interactions, and has a linear encoding of case. *Society for Computation in Linguistics*, 8(1).

[10] Pickering, M. J. and Branigan, H. P. (1998). The representation of verbs: Evidence from syntactic priming in language production. *Journal of Memory and Language*, 39(4):633–651.

[11] Prasad, G., van Schijndel, M., and Linzen, T. (2019). Using priming to uncover the organization of syntactic representations in neural language models. In Bansal, M. and Villavicencio, A., editors, *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

[12] Ravfogel, S., Prasad, G., Linzen, T., and Goldberg, Y. (2021). Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction. *arXiv preprint arXiv:2105.06965*.

[13] Sinclair, A., Jumelet, J., Zuidema, W., and Fernández, R. (2022). Structural persistence in language models: Priming as a window into abstract language representations. *Transactions of the Association for Computational Linguistics*, 10:1031–1050.

[14] Todd, E., Li, M. L., Sharma, A. S., Mueller, A., Wallace, B. C., and Bau, D. (2023). Function vectors in large language models. *arXiv preprint arXiv:2310.15213*.

[15] van Schijndel, M. and Linzen, T. (2018). A neural model of adaptation in reading. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4704–4710, Brussels, Belgium. Association for Computational Linguistics.

[16] Zhou, Z., Frank, R., and McCoy, R. T. (2025). Is in-context learning a type of error-driven learning? evidence from the inverse frequency effect in structural priming. In Chiruzzo, L., Ritter, A., and Wang, L., editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11712–11725, Albuquerque, New Mexico. Association for Computational Linguistics.