

Do Language Models implement compositional solutions for natural language understanding

Stanford CS224N {Custom} Project

Ahmad Jabbar

Department of Linguistics
Stanford University
jabbar@stanford.edu

Abstract

Language models (LMs), both task-specific and pretrained, have improved at compositional generalization. However, all evaluation in the literature has so far been behavioural. In this paper, using the methodology of causal abstractions (Geiger et al., 2021), we seek to assess if LMs also implement compositional solutions for understanding language. We focus on monotonicity inferences. Quantifiers like *all* in sentences like *All birds fly* are taken to take two predicates: *bids* and *fly* such that the sentence *All birds fly* holds true even when the right predicate (*fly*) is replaced with a superset (*move*). Thus, *all* is said to be right upward monotonic. For a model that reliably judges the validity of the monotonicity inference, we seek to localize the crucial semantic representations within its lower-level representations. In this project, we present a new clean dataset for monotonicity inferences. After evaluating a range of state-of-the-art open source models on our dataset, we show that GPT-OSS achieves high accuracy. Using the methodology of interchange interventions, we seek to find the symbolic structure hypothesized by semantics within GPT-OSS low-level representations.

1 Key Information to include

- Mentor: Julie Kallini
- External Collaborators (if you have any): Christopher Potts
- Sharing project: No
- Late Days (if you are using any):

Team member	Late days remaining
Ahmad Jabbar	0
Total pooled	
Team size	1
Late days applied for the report	3

2 Introduction

Compositional benchmarks (Lake and Baroni, 2018; Kim and Linzen, 2020; Wu et al., 2023) seek to evaluate if Language Models (LMs) are compositional. However, such evaluation is often operationalized as semantic parsing of natural language sequences. Semantic parses are model outputs that are evaluated for how closely they match the correct parse. While such evaluation is insightful—with some pitfalls (Jabbar et al., 2025)—it doesn’t help us understand if LMs are compositional in their processes. Nefdt (2020) makes a distinction between output and process compositionality, where evidence of the former cannot be taken to suggest the presence of compositional processes. In principle, we can ask the question about compositionality for LM processes as well. Given a natural

language input, is the LM behavior that displays understanding determined by a causal mechanism? Nefdt (2020, 64) expresses skepticism for making progress towards this answer: “Deep Neural Networks are indeed epistemically opaque in a particular way which prevents an attribution of process and state compositionality to them”. Evaluation metrics that focus on model outputs don’t answer this question about compositional processes. For the present paper, we present a proof of concept of how using the methodology of causal abstraction (Geiger et al., 2021, 2024; Wu et al., 2024), we can find compositional mechanisms within model internals for the task of natural language understanding.

To make the task and its evaluation more concrete and manageable, we exclusively focus on the case study of monotonicity inferences, which are inferences governed by the monotonicity properties of the quantifiers involved in sentences. Consider (1).

- (1) a. All professors are tall women.
- b. All professors are women.

In the above example, the sentence *All professors are tall women* entails the sentence *All professors are women*. However, the entailment doesn’t go the other way; the sentence *All professors are women* does not entail the sentence *All professors are tall women*. The fact that the first inference holds and the second doesn’t is attributed to the following property of the quantifier *all*: if you make *all*’s second predicate less specific, i.e., go from *tall women* (more specific) to *women* (less specific), the resulting sentence with the less specific second predicate is true if the original sentence is true. However, if you make the second predicate more specific, the entailment doesn’t hold. We operationalize the capacity of natural language understanding as judgment on pairs of sentences like (1). In other words, if the LM judges the correct entailment relation between sentences like (1), we take the LM to possess the capacity to at least understand the fragment of natural language understanding comprising sentences with the quantifier *all*.

With natural language understanding operationalized as such, we evaluate a range of state-of-the-art LMs. Our evaluation is carried out with our own original dataset that comprises pairs like 1. Our dataset is generated with a compact Context Free Grammar. GPT-OSS 20B displays the highest accuracy on our task.¹ For our second experiment, we choose GPT-OSS, and using the methodology of interchange interventions (Geiger et al., 2021; Wu et al., 2024), we carry out a causal analysis of GPT-OSS. The causal model that we hypothesize the lower-level neural network to be realizing is a semantic composition tree from semantic theory that crucially encodes the monotonic properties of the quantifier (Potts, 2025). We conclude with little to weak evidence of the LM instantiating the hypothesized causal model. We think that this shouldn’t be taken in a pessimistic stride. First, our investigations provide a proof of concept of how compositional processes can be found within LMs. The possibility of such an investigation is totally compatible with finding out that there aren’t compositional processes behind natural language understanding in LMs. However, for now, this is a hasty conclusion, as we don’t use Distributed Alignment Search (DAS) for the current paper, which is a causal abstraction methodology that is often the most successful one for finding causal mechanisms (Boguraev et al., 2025), which is what we aim to adopt for our future investigations.

3 Related Work

Evaluation of compositionality. There is a range of work that seeks to evaluate if LMs are compositional. Lake and Baroni (2018) propose the SCAN task which requires mapping an input natural language sequence to an action sequence. Kim and Linzen (2020) propose the COGS benchmark that evaluates how well LMs generalize on the task of semantic parsing to new lexical items and structures. Li et al. (2023) is an extension of COGS, focusing solely on structural generalization tasks. While LMs were taken to struggle on these tasks, recent methods have shown that LMs can achieve high, even perfect, accuracy on such tasks: Csordás et al. (2021) show how simple tricks like using relative positional embedding and early stopping can help LMs perform better. More systematically, Lake and Baroni (2023) show that their model trained with a meta-learning regimen can surpass humans on compositional tasks.²

¹All references henceforth are to GPT-OSS 20B and not 120B.

²Lake and Baroni (2023), however, don’t show that their model can solve the structural generalization tasks in COGS; they limit the evaluation of their model only to the lexical tasks.

Causal analysis. Probes are a great way to understand model behavior (Hupkes et al., 2018; Clark et al., 2019). However, Geiger et al. (2021) point out a few problems with interpretability methods that don’t seek to find causal mechanisms within LMs. For probes, they show that probes uncover seemingly important information that doesn’t causally affect the model output. As a result, since Geiger et al. (2021), there has been an effort to causally understand LM behavior. Wu et al. (2024) develop a Python library for efficiently running the causal analysis outlined in Geiger et al. (2021), while Geiger et al. (2025) present the theoretical underpinning of the theory. Moreover, this causal analysis has indeed been shown to work well for understanding the causal processes underpinning model behavior for a range of tasks: entity binding (Gur-Arieh et al., 2025), syntactic transformations such as filler-gap dependencies Arora et al. (2024); Boguraev et al. (2025), etc.

4 Approach

In this section, we explain three things in detail. First, we explain the semantic theory behind monotonicity inferences.³ Second, we explain the methodology of causal abstraction. Third, we show how we can seek to align the semantic theory with the lower-level neural network as per causal abstraction.

4.1 Semantic models for monotonicity inferences

Consider the following pairs of sentences:

- | | |
|---|---|
| <p>(2) Entailment holds:</p> <ul style="list-style-type: none"> a. All farmers are tall people. b. All farmers are people. | <p>(3) Entailment doesn’t hold:</p> <ul style="list-style-type: none"> a. All students are girls. b. All students are hardworking girls. |
|---|---|

First, consider (2). If all farmers are tall people, then all farmers are people as well. However, if all students are girls, that doesn’t mean that all students are hardworking girls. Note another entailment, but this time between

This entailment pattern due to *all* can be captured by the semantics of *all*. In words, *all* can be said to take two predicates *P* and *Q* as in *All P Q* or *All P are Q*. We can think of *P* as the first predicate of *all* and *Q* as the second predicate. If you take a sentence *All P are Q*, and make the second predicate *Q* less specific, call it *Q_{less specific}*, *All P are Q* entails *All P are Q_{less specific}*. However, the entailment doesn’t go in the other direction. Another thing to note is that this property is special to *all*. Contrast how the same entailment patterns don’t hold with *no*.

- | | |
|--|--|
| <p>(4) Entailment doesn’t hold:</p> <ul style="list-style-type: none"> a. No farmers are tall people. b. No farmers are people. | <p>(5) Entailment holds:</p> <ul style="list-style-type: none"> a. No students are girls. b. No students are hardworking girls. |
|--|--|

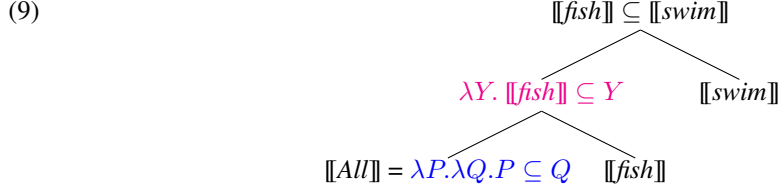
All that we have done to create (4) is just replace *all* in (2) with *no*. This has resulted in a flip of the entailment pattern; *mutatis mutandis* for (3) and (5). Using tools from semantic theory, we can formalize all of the above with set-theoretic relations. For the following part, we use the standard notation in semantic theory, where $\llbracket \cdot \rrbracket$ is the interpretation function.⁴ $\llbracket \cdot \rrbracket$ takes an expression in natural language and maps it to the denotation of that expression. For instance, $\llbracket \text{The Gates Building at Stanford} \rrbracket$ outputs the Gates Building at Stanford that is opposite the Bytes Cafe. Below, we provide the semantics for the three quantifiers: *all*, *no*, and *some*.

- (6) $\llbracket all \rrbracket = \lambda P. \lambda Q. P \subseteq Q$
- (7) $\llbracket no \rrbracket = \lambda P. \lambda Q. P \cap Q = \emptyset$
- (8) $\llbracket some \rrbracket = \lambda P. \lambda Q. P \cap Q \neq \emptyset$

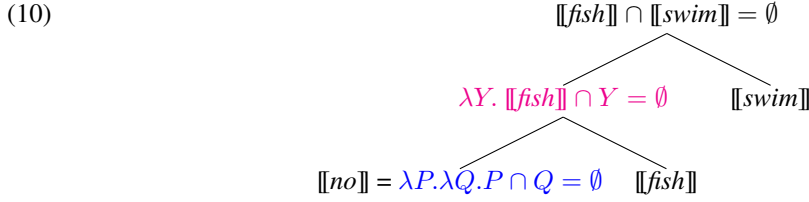
With these semantics, along with the composition trees, we can perform a few calculations and show how we can derive the predicted entailment patterns.

³Thanks to Julie Kallini for the suggestion of explicitly laying out these trees.

⁴This is the standard notation in semantics. See Heim and Kratzer (1998) for its use.



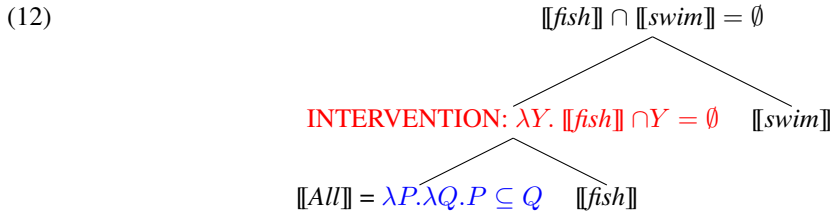
The above tree is to be read from bottom upwards. Each leaf node has its own individual meaning. The meanings of *all* and *fish* compose together to form the meaning at an intermediate stage that is colored in **magenta**. That intermediate meaning is what determines whether the entailment holds or not. To see the importance of this intermediate meaning, we can also build another tree, but this time, with *no*, instead of *all*.



The above composition trees, given the semantics outlined for the quantifiers in (6)-(8), can be thought of as causal models, comprising an intermediate latent variable that we have color-coded in **magenta**.



Our characterization of these composition trees as causal models is also amenable to interchange interventions. We can take the intermediate **magenta** representation from one composition tree and substitute it with the original intermediate representation. That should give us the flip at the root node.



Note that in (12), while the inputs at all the leaf nodes are identical to the inputs at the leaf nodes in (9), due to the intervention at the intermediate variable, the final output at the root node is flipped.

If we hypothesize the lower-level neural network to instantiate the higher-level causal models as proposed by semantic theory, then we should be able to find evidence of the latent intermediate variable via using interchanging interventions as illustrated above. This forms the basis of our dataset and interchange interventions experiments below.

4.2 Interchange interventions in LMs

Above we laid out our hypothesized causal models. Here, we outline how we intend to concretely implement them within LMs.

In (9) and (10), we have an intermediate representation that affects the final output. Using Pyvene (Wu et al., 2024), we run interchange interventions (Geiger et al., 2021; Wu et al., 2024) to evaluate if LMs localize the intermediate representation hypothesized in our causal models within their neurons.

The interchange intervention strategy can be outlined more explicitly here: a model $\mathcal{M}$ conditions on the base input to generate a token; $\mathcal{M}$ is also fed a source input. Call the two states of $\mathcal{M}$ on the base and source inputs $\mathcal{M}_B$ and $\mathcal{M}_S$, respectively. For our experiments, we first seek to find the layer that is most responsible for the entailment judgment. For this, we run the zero intervention, in which the entire block output is zeroed out. We supplement finding out which layer is the most causally efficacious with interchanging representations from sequences that would specifically flip the entailment judgment from *yes* to *no* and *no* to *yes*. This can be done with constructing a bunch of sentences with the quantifier *all*, running $\mathcal{M}_B$ on these sequences, and then constructing another bunch of sentences with the quantifier *no*, and running $\mathcal{M}_S$ on these, and seeing how often the judgment flips.

More specifically, the above strategy is carried out at each token position. For each token position in the input sequence such as (2), for instance, we try to find at which layer, another source representation, if added to the residual stream of the token in lieu of the base representation, reliably flips the model output.

To concretize this further, consider the following pair of sentences:

- (13) a. All farmers are tall people. All farmers are people.
- b. No farmers are tall people. No farmers are people.

Zero interventions. Let the source representations be the representations that are generated with `torch.zeros(model_embedding_size)`. Call this the ‘zero representation’. Instead of adding the block output at a given layer to the residual stream of the token, add the zero representation at that layer. Repeat this for all token positions and all layers. This helps us figure out which layer and which token position is most causally efficacious for the output.

Interchange interventions. Let the source representation be the one that the model builds for a source sequence, rather than a zero representation. Repeat the same process as for Zero interventions. With suitable source representations, we can verify the counterfactual behavior of our hypothesized causal model. For our base sequences, we use 250 pairs of sentences like (1). For our source sequences, we use two sets of sentences: set 1 is created by replacing *all* in our base sequences with *some*; set 2 is created by replacing *all* in our base sequences with *no*. The reason for this is that with Sets 1 and 2, we can check two counterfactual behaviors of our model. With *some*, the model output should not change. With *no*, the model output should change.

Overall. If using zero interventions, we can figure out a particular layer at which zeroing out the block output at a token position changes the output, then we can proceed to interchange interventions. If for the same layer and token position, we see the model output flipping for *no* sequences and staying the same for *some* sequences, then we can make an argument for the localization of the monotonicity information within the model internals.

5 Experiments

This section contains the following.

5.1 Experiment 1

5.1.1 Data

Using a Context Free Grammar, we generated a set of 250 pairs of sentences.

- (14) $S \rightarrow QP V (ADJ) N_2$
- (15) $QP \rightarrow Q N_1$
- (16) $V \rightarrow are$
- (17) $Q \rightarrow all$
- (18) $N_1 \rightarrow farmers, professors, students, workers, soldiers$

Model	Accuracy
GPT-OSS (21B)	0.93
Qwen 3-8B	0.80
OLMO-3 7B	0.77
GPT-2 XL (1.5B)	0.50
Pythia 6.9B	0.51

Table 1: Model performance on monotonicity inference.

- (19) $N_2 \rightarrow \text{men, women, boys, girls, people}$
(20) $ADJ \rightarrow \text{tall, educated, intelligent, hardworking, powerful}$

With this CFG, we were able to construct sentences like the following.

- (21) **Entailment holds:**
a. All farmers are intelligent women.
b. All farmers are women.
- (22) **Entailment doesn’t hold:**
a. All farmers are boys.
b. All farmers are intelligent boys.

5.2 Evaluation method

Using a number of prompt templates, we prompted the seven models, GPT-2 XL (1.5B), Pythia 6.9B, Qwen 3-8B, OLMo 3-7B, and GPT-OSS (21B) to judge entailment between sentences from our dataset. In our prompts, we asked the model to output either yes or no. We took the logits for the tokens yes and no for calculation of the model judgment.

5.3 Experimental details

We loaded all of the five models from HuggingFace using the Transformers library. And all of the experiments just included inference and no training. We ran all of our experiments on A100 GPU.

5.4 Results

Table 1 reports our results. The accuracy of GPT-2 XL and Pythia 6.9B is at about chance. OLMo-3 7B, Qwen 3-8B and GPT-OSS (21B) achieve decent accuracy. However, as GPT-OSS does the best among all, we chose GPT-OSS for our interchange intervention experiments.

5.5 Experiment 2

5.5.1 Data

As outlined in Section 4.2, for running interchange interventions, we crucially need a set of base sequences. We can interchange the model representations at a token position in the base sequence with representations from another source. This source representation can either be an embedding of zeros or an embedding that, for instance, is outputted at a particular layer of a LM at a given token in a source sequence, where the source sequence is not identical to the base sequence. Please see the paragraphs **Zero interventions** and **Interchange interventions** in Section 4.2 for more details. With the above guiding methodology, we kept the base sequences that were generated from the CFG in Experiment 1. However, notably, from this set, we took out all of the sequences that GPT-OSS got wrong in the evaluation of monotonicity inferences. So, we were left with 232 out of 250 sequences. This is still a considerable number, given the high accuracy of GPT-OSS on the task. Call this set ‘Base Set’.

For our source sequences, we created two sets of sentences. In one set, we replaced all occurrences of *All* in the sequences of Base Set and replaced them with *Some*. Given that both *all* and *some* show the same monotonic behavior on the second predicate, this source set provides a good counterfactual check. In the other set, we replaced all occurrences of *All* in the sequences of Base Set and replaced them with *No*. Unlike *some*, *no* does not show the same monotonic behavior on the second predicate as *all*. These two sets provide good counterfactual checks.

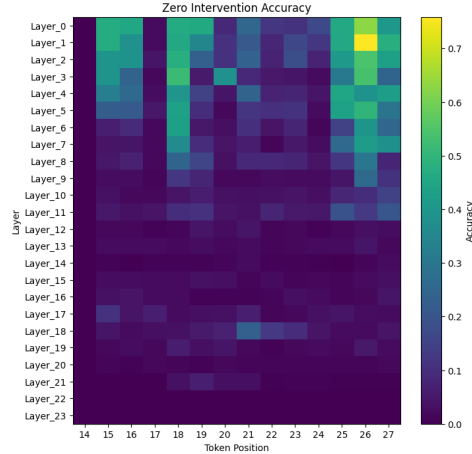


Figure 1: The heatmap shows the interchange intervention accuracy for each token position and layer. The highest accuracy is at Layer 1 at token position 26.

5.5.2 Evaluation method

First, we ran the Zero Intervention on GPT-OSS on all the token positions of the input sequence within the prompt template. Our prompt template was: “Answer ‘Yes’ or ‘No’. Question: Premise: ‘All farmers are tall men’ Hypothesis: ‘All farmers are men’. Does the Premise logically entail the Hypothesis? Answer:”. We looked at the token positions that are colored in teal. Our rationale for this was that, while there may be other factors that are responsible for the final model judgment, the monotonicity information, given the autoregressive nature of the model, should be stored at or after the token *all*. We ran the Zero Intervention on the block outputs for all 24 layers of GPT-OSS. We summed all out of 232 Base Set sequences where intervening in this way flipped the model judgment on the given monotonicity inference.

Next, we ran the Interchange Intervention with the two source sets: the *no* set and the *some* set. Again, we employ the same evaluation method as we did for the Zero intervention.

5.5.3 Experimental Details

We loaded GPT-OSS using the `transformers` library. We used `pyvene` for all of our experiments. With `pyvene`, we were able to wrap the underlying model with an intervenable configuration. For our forward runs, we batched and parallelized the Base Set (and Source Set) sequences for efficient inference. We ran our experiment on A100 GPU.

5.5.4 Results

Our results from the Zero Interventions are reported in Figure 5.5.4. And our results from the interchange interventions using source sequences where *and* is replaced with *no* and *some* are reported in Figure 5.5.4. For token position 26 at Layer 1, zeroing out the block output affects the model judgment on monotonicity inferences. Moreover, for running interchange interventions with source sequences, we see that the accuracies are quite low all throughout. We expand on these results more in the Analysis section next.

6 Analysis

Our results, on their own, especially the one for Zero Intervention is interesting. There is above chance accuracy on token position 26 at Layer 1. However, there are two problems with this result. First, to be able to say that a symbolic causal model is an abstraction of a lower-level model, we want to get as close as possible to 100% accuracy on the interventions (Geiger et al., 2025). Second, even if we sidestep that worry, it is not at all clear what this causal variable is that is being instantiated at Layer 1 for token position 26. Sure, it seems to be contributing to the model judgment at above

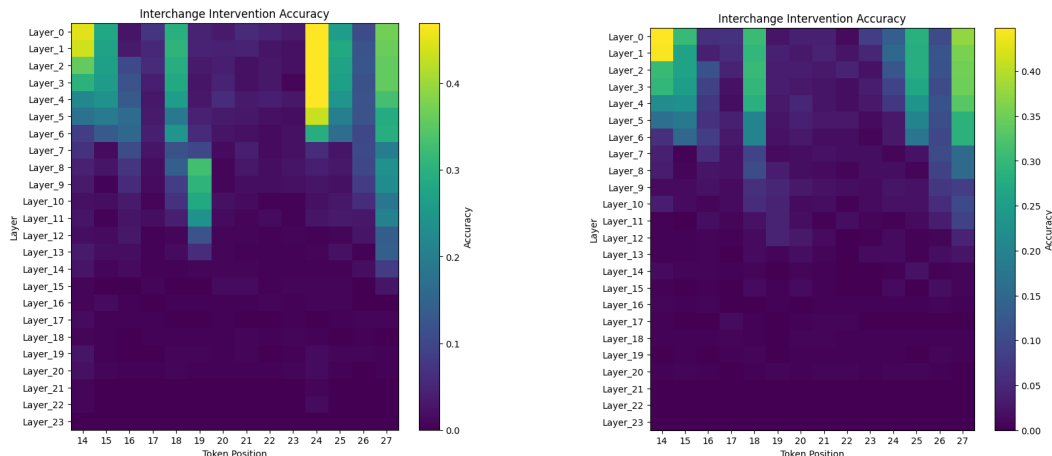


Figure 2: The two heatmaps show the interchange intervention accuracies. The one on the left is for intervening with the representations of *no* and the one on the right is for *some*.

chance rate, but there isn't much else that we can say about it without carrying out more experiments. This is exactly why our other two source datasets with the *no* and *some* manipulation come in handy. If the model is instantiating some monotonicity information at Layer 1 for token 26, then intervening with the *no* representation should also see high accuracy for this position. In addition, intervening with the *some* representation should keep the model outputs the same, resulting in low accuracy. We don't see this result when we compare Figure 5.5.4 with Figure 5.5.4. The map on the left in Figure 5.5.4 is for intervening with the source representations of sequences containing *no* instead of *all*. For these inferences, the output always flips due to the monotonicity properties we outlined in Section 4.1. However, if the localization we found with Zero Interventions is indeed encoding monotonicity, we should see that exact same spot light up for *no*, but as we can see Layer 1 at token 26 in Figure 5.5.4 has quite low accuracy.

7 Conclusion

We started out with highlighting a perspective due to Nefdt (2020), i.e., that deep neural nets are purportedly opaque and it is hard to attribute process compositionality to them. To assuage Nefdt's criticism, we undertook the current investigation to illustrate that it is possible to see whether LMs are process compositional. We operationalized natural language understanding in a reliable way with judgment on monotonicity inferences and created a dataset on which we showed GPT-OSS to be a good judge. Our preliminary causal analysis didn't reveal that GPT-OSS encodes a monotonicity variable that can be intervened on. However, there are a few **limitations** of our present study which strengthen our overall point. First, we didn't run other types of interventions like DAS that are quite customary to run (Boguraev et al., 2025). If we run our current experiments with DAS, we may end up seeing better alignment of our symbolic models with the LMs. Second, what we have shown is that it is, in principle, possible to carry out an analysis of LMs to attribute process compositionality to them. The possibility of such an analysis is quite compatible with eventually finding out that LMs are not process compositional. While of course, currently, it is too early to make any claim, we think that our present work can be taken as a proof of concept contribution towards showing that causal analysis can be undertaken to decide whether LMs are process compositional or not.

Team contributions (Required for multi-person team)

Ahmad did everything for this report himself.

References

- Arora, A., Jurafsky, D., and Potts, C. (2024). Causalgym: Benchmarking causal interpretability methods on linguistic tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14638–14663.
- Boguraev, S., Potts, C., and Mahowald, K. (2025). Causal interventions reveal shared structure across English filler–gap constructions. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V., editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25032–25053, Suzhou, China. Association for Computational Linguistics.
- Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. (2019). What does bert look at? an analysis of bert’s attention. In *Proceedings of the 2019 ACL workshop BlackboxNLP: analyzing and interpreting neural networks for NLP*, pages 276–286.
- Csordás, R., Irie, K., and Schmidhuber, J. (2021). The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634.
- Geiger, A., Ibeling, D., Zur, A., Chaudhary, M., Chauhan, S., Huang, J., Arora, A., Wu, Z., Goodman, N., Potts, C., et al. (2025). Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64.
- Geiger, A., Lu, H., Icard, T., and Potts, C. (2021). Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586.
- Geiger, A., Wu, Z., Potts, C., Icard, T., and Goodman, N. (2024). Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pages 160–187. PMLR.
- Gur-Arieh, Y., Geva, M., and Geiger, A. (2025). Mixing mechanisms: How language models retrieve bound entities in-context. *arXiv preprint arXiv:2510.06182*.
- Heim, I. and Kratzer, A. (1998). *Semantics in generative grammar*. Wiley Blackwell.
- Hupkes, D., Bouwmeester, S., and Fernández, R. (2018). Analysing the potential of seq-to-seq models for incremental interpretation in task-oriented dialogue. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 165–174.
- Jabbar, A., Condoravdi, C., and Potts, C. (2025). Distinguishing fair from unfair compositional generalization tasks. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V., editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20796–20807, Suzhou, China. Association for Computational Linguistics.
- Kim, N. and Linzen, T. (2020). COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105.
- Lake, B. and Baroni, M. (2018). Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.
- Lake, B. M. and Baroni, M. (2023). Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121.
- Li, B., Donatelli, L., Koller, A., Linzen, T., Yao, Y., and Kim, N. (2023). Slog: A structural generalization benchmark for semantic parsing. *arXiv preprint arXiv:2310.15040*.
- Nefdt, R. M. (2020). A puzzle concerning compositionality in machines. *Minds and Machines*, 30(1):47–75.
- Potts, C. (2025). Quantifiers properties. Handout, *Linguist 130a/230a: Introduction to Semantics and Pragmatics*. Winter 2025, February 4.

- Wu, Z., Geiger, A., Arora, A., Huang, J., Wang, Z., Goodman, N., Manning, C. D., and Potts, C. (2024). pyvene: A library for understanding and improving pytorch models via interventions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 158–165.
- Wu, Z., Manning, C. D., and Potts, C. (2023). ReCOGS: How incidental details of a logical form overshadow an evaluation of semantic interpretation. *Transactions of the Association for Computational Linguistics*, 11:1719–1733.