How do language models bind entities in context?

Jiahai Feng FJIAHAI@BERKELEY.EDU
Jacob Steinhardt JSTEINHARDT@BERKELEY.EDU
UC Berkeley

Editors: List of editors' names

Abstract

To correctly use in-context information, language models (LMs) must bind entities to their attributes. For example, given a context describing a "green square" and a "blue circle", LMs must bind the shapes to their respective colors. We analyze LM representations and identify the *binding ID mechanism*: a general mechanism for solving the binding problem, which we observe in every sufficiently large model from the Pythia and LLaMA families. Using causal interventions, we show that LMs' internal activations represent binding information by attaching *binding ID vectors* to corresponding entities and attributes. We further show that binding ID vectors form a continuous subspace, in which distances between binding ID vectors reflect their discernability. Overall, our results uncover interpretable strategies in LMs for representing symbolic knowledge in-context, providing a step towards understanding general in-context reasoning in large-scale LMs.

Keywords: Representations, Interpretability, Language Models

1. Introduction

Modern language models (LMs) excel at many reasoning benchmarks, suggesting that they can perform general purpose reasoning across many domains. However, the mechanisms that underlie LM reasoning remain largely unknown (Räuker et al., 2023). The deployment of LMs in society has led to calls to better understand these mechanisms (Hendrycks et al., 2021), so as to know why they work and when they fail (Mu and Andreas, 2020; Hernandez et al., 2021; Vig et al., 2020b).

In this work, we seek to understand *binding*, a foundational skill that underlies reasoning. How humans solve binding, i.e. recognize features of an object as bound to that object and not to others, is a fundamental problem in psychology (Treisman, 1996). Here, we study binding in LMs.

Binding arises any time the LM has to reason about two or more objects of the same kind. For example, consider the following passage involving two people and two countries:

> Context: Alice lives in the capital city of France. Bob lives in the capital city of Thailand.
>
> Question: Which city does Bob live in? (1)

In this example the LM has to represent the associations *lives(Alice, Paris)* and *lives(Bob, Bangkok)*. We call this the *binding problem*—for the predicate *lives*, *Alice* is bound to *Paris* and *Bob* to *Bangkok*. Since predicates are bound in-context, binding must occur in the activations, rather than in the weights as with factual recall. This raises the question: how do LMs represent binding information in the context such that they can be later recalled?

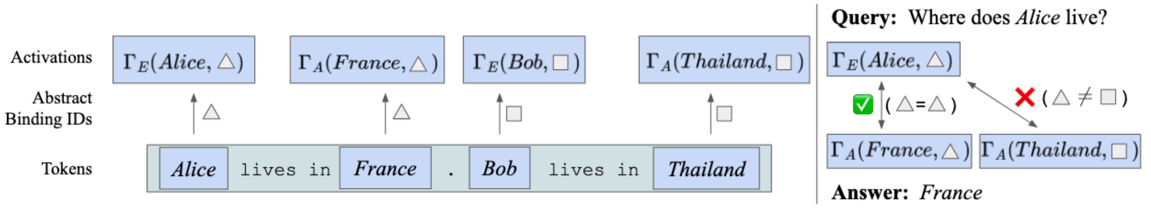


Figure 1: Illustration of the Binding ID mechanism. The LM learns an abstract binding ID (drawn as triangles or squares) which distinguishes between entity-attribute pairs. Binding functions Γ_E and Γ_A bind entities and attributes to their abstract binding ID, and stores the result in the activations. To answer queries, the LM identifies the attribute that shares the same binding ID as the queried entity.

In this work, we identify one frequently-used binding mechanism in LMs and test its robustness and limits. The mechanism relies on *binding IDs*, which are abstract concepts that LMs use internally to mark variables in the same predicate apart from variables in other predicates (Fig. 1). We empirically argue for the existence of binding IDs using causal mediation analysis.

Turning to the structure of binding IDs, we find that binding IDs are represented as vectors and the binding function linearly adds binding ID vectors to representations of variables. Further, we show that binding IDs occupy a metric subspace, in the sense that linear combinations of binding IDs are still valid binding IDs.

Lastly, we find that binding IDs are ubiquitous and transferable. They are used by every sufficiently large model in the LLaMa (Touvron et al., 2023) and Pythia (Biderman et al., 2023) families, and their fidelity increases with scale.

2. Preliminaries

In this section we define the *binding task* and explain causal mediation analysis, our main experimental technique.

Binding task To perform reading comprehension tasks, a necessary skill is to distinguish between entities and bind attributes uniquely to them. We formalize this as the *binding task*. The binding task consists of a set of entities $\mathcal{E}$ and a set of attributes $\mathcal{A}$. An n-entity instance of the binding task consists of a context that is constructed from n entities $e_0, \ldots, e_{n-1} \in \mathcal{E}$ and n attributes $a_0, \ldots, a_{n-1} \in \mathcal{A}$, and we denote the corresponding context as $\mathbf{c} = \text{ctxt}(e_0 \leftrightarrow a_0, \ldots, e_{n-1} \leftrightarrow a_{n-1})$. A template is used to obtain the token representation $\text{Tok}(\mathbf{c})$. For a context $\mathbf{c}$, we use $E_k(\mathbf{c})$ and $A_k(\mathbf{c})$ to denote the k-th entity and the k-th attribute of the context $\mathbf{c}$, for $k \in [0, n-1]$. We will drop the dependence on $\mathbf{c}$ for brevity when the choice of $\mathbf{c}$ is clear from context.

In the CAPITALS task, which is the main task we study for most of the paper, $\mathcal{E}$ is a set of single-token names, and $\mathcal{A}$ is a set of single-token countries. Quote 1 is an example instance of the CAPITALS task with context $\mathbf{c} = \text{ctxt}(\textit{Alice} \leftrightarrow \textit{France}, \textit{Bob} \leftrightarrow \textit{Thailand})$. In this context, E_0 is *Alice*, A_0 is *France*, etc.

Given a context $\mathbf{c}$, we are interested in the model's behavior when queried with each of the n entities present in $\mathbf{c}$. For any $k \in [0, n-1]$, when queried with the entity E_k the model should place high probability on the answer matching A_k. In our running example, the model should predict “Paris” when queried with “Alice”, and “Bangkok” when queried with “Bob”.

Causal structure in LMs Autoregressive language models have inherent causal structure that we utilize. Let an LM have n_{layers} transformer layers and a d_{model}-dimensional activation space. For every token position p, we use $Z_p \in \mathbb{R}^{n_{\text{layers}} \times d_{\text{model}}}$ to denote the stacked set of of internal activations[1] at token p. We refer to the collective internal activations of the context as Z_{context}. In addition, we denote the activations at the token for the k-th entity as Z_{E_k}, and for the k-th attribute as Z_{A_k}. We will sometime write $Z_{A_k}(\mathbf{c}), Z_{\text{context}}(\mathbf{c})$, etc. to make clear the dependence on the context $\mathbf{c}$.

Z_{context} can be viewed as the representation the model constructs for the context. We thus study the structure of Z_{context} using *causal mediation analysis*, a widely used tool for understanding neural networks (Vig et al., 2020a; Geiger et al., 2021; Meng et al., 2022). Causal mediation analysis involves substituting one set of activations in a network for another, and we adopt the /. notation (from Mathematica) to denote this. For example, for activations $Z_* \in \mathbb{R}^{n_{\text{layers}} \times d_{\text{model}}}$, and a token position p in the context, $Z_{\text{context}}/.\{Z_p \to Z_*\} = [Z_0, \ldots, Z_{p-1}, Z_*, Z_{p+1}, \ldots]$. Similarly, for a context $\mathbf{c} = \text{ctxt}(e_0 \leftrightarrow a_0, \ldots, e_{n-1} \leftrightarrow a_{n-1})$, we have $\mathbf{c}/.\{E_k \to e_*\} = \text{ctxt}(e_0 \leftrightarrow a_0, \ldots, e_* \leftrightarrow a_k, \ldots, e_{n-1} \leftrightarrow a_{n-1})$.

3. Binding ID mechanism

We claim that to bind attributes to entities, the LM learns abstract binding IDs that it assigns to entities and attributes, so that entities and attributes bound together have the same binding ID (Fig. 1). In more detail, our informal description of the binding ID mechanism is that:

1. For entity E_k, encode both the entity E_k and the binding ID k in the activations Z_{E_k}.
2. For attribute A_k, encode both the attribute A_k and the binding ID k in the activations Z_{A_k}.
3. To answer a query for entity E_k, retrieve from Z_{context} the attribute that shares the same binding ID as E_k.

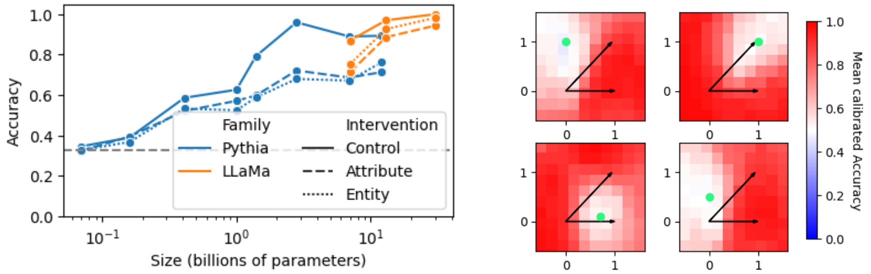
Further, for activations Z_{E_k} and Z_{A_k}, the binding ID and the entity/attribute are the only information they contain that affects the query behavior. See Appendix A for a formal statement.

The binding ID mechanism predicts two testable properties about Z_{context}:

- **Factorizability:** if we replace Z_{A_k} with $Z_{A'_k}$, then the model will bind E_k to A'_k instead of A_k, i.e. it will believe $\mathbf{c}./\{A_k \to A'_k\}$. This is because Z'_{A_k} encodes $\Gamma_A(A'_k, k)$ and Z_{A_k} encodes $\Gamma_A(A_k, k)$. Substituting $Z_{A_k} \to Z_{A'_k}$ will overwrite $\Gamma_A(A_k, k)$ with $\Gamma_A(A'_k, k)$, causing the model to bind E_k to A'_k. We also expect Z_{E_k} to be similarly factorizable.
- **Position independence:** if we e.g. swap Z_{A_0} and Z_{A_1}, the model still binds $A_0 \leftrightarrow E_0$ and $A_1 \leftrightarrow E_1$, because it looks up attributes based on binding ID and not position in the context. We also expect Z_{E_k} to have similar position independence.

[1] These are the the pre-transformer layer activations, sometimes referred to as the *residual stream*.

These properties can be tested by experimentally intervening on Z_{context} with the relevant substitutions, and measuring the causal effects on behavior. Appendix C and D contain these experiments which verify the binding ID mechanism. Appendix B argues that binding ID is the only mechanism consistent with these two properties.



(*a*) Pythia and LLaMa models on CAPITALS task

(*b*) Metric structure in binding subspace

Figure 2

4. Properties of Binding ID

Additivity of binding functions A simple hypothesis is that both entity/attribute representations and abstract binding IDs are vectors in activation space, and that the binding function simply adds the vectors for entity/attribute and binding ID. We let the binding ID k be represented by the pair of vectors $[b_E(k), b_A(k)]$, and the representations of entity e and attribute a be $f_E(e)$ and $f_A(a)$ respectively. Then, we hypothesize that the binding functions can be linearly decomposed as:

$$\Gamma_A(a,k) = f_A(a) + b_A(k), \quad \Gamma_E(e,k) = f_E(e) + b_E(k). \tag{1}$$

Binding ID vectors seem intuitive and plausibly implementable by transformer circuits. To experimentally test this, we seek to extract $b_A(k)$ and $b_E(k)$ in order to perform vector arithmetic on them. We use (1) to extract the *differences* $\Delta_E(k) := b_E(k) - b_E(0)$, $\Delta_A(k) := b_A(k) - b_A(0)$. Rearranging (1), we obtain

$$\Delta_A(k) = \Gamma_A(\alpha,k) - \Gamma_A(\alpha,0), \quad \Delta_E(k) = \Gamma_E(a,k) - \Gamma_E(a,0). \tag{2}$$

We estimate the mean differences $\Delta_A(k)$ by sampling $\mathbb{E}_{\mathbf{c},\mathbf{c}'}[Z_{A_k}(\mathbf{c}) - Z_{A_0}(\mathbf{c}')]$, and likewise for $\Delta_E(k)$. In Appendix F we show that adding or subtracting the estimated mean differences to the binding vectors successfully changes the binding information stored in the context activations, thus validating the additivity of binding functions.

Geometry of binding vectors Using mean interventions, we find that linear interpolations or extrapolations of binding vectors are also valid binding vectors. This suggests that binding vectors occupy a continuous *binding subspace*. We find evidence of a *metric structure* in this space, such that nearby binding vectors are hard for the model to distinguish, but far-away vectors can be reliably distinguished and thus used for the binding task (Fig. 2b). Details in Appendix F.2.

Generality of binding ID We find that sufficiently large models in the Pythia and LLaMa families exhibit the binding ID mechanism by measuring the effectiveness of the *mean interventions* (Fig. 2a). Further, the effectiveness of the mean interventions **increases with scale**, suggesting that large models converge to the same robust representational strategy of using binding IDs.

Additional properties We additionally find that binding IDs are used across synthetic binding tasks with different surface forms, and binding vectors from one task transfer to other tasks. However, despite their ubiquity, binding IDs are not universal: using causal mediations we identify an alternate binding mechanism, *direct binding*, that is used for a question-answering task. Details in Appendix G.

References

Yonatan Belinkov and James Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, 2019. doi: 10.1162/tacl_a_00254. URL https://aclanthology.org/Q19-1004.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Knowledge neurons in pretrained transformers. *ArXiv*, abs/2104.08696, 2021. URL https://api.semanticscholar.org/CorpusID:233296761.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128, 2019.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

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

Mor Geva, R. Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *ArXiv*, abs/2012.14913, 2020. URL https://api.semanticscholar.org/CorpusID:229923720.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767*, 2023.

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ml safety. *arXiv preprint arXiv:2109.13916*, 2021.

Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2021.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. Measuring and manipulating knowledge representations in language models. *arXiv preprint arXiv:2304.00740*, 2023a.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. *arXiv preprint arXiv:2308.09124*, 2023b.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*, 2022.

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

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 464–483. IEEE, 2023.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8: 842–866, 2021.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. In *Association for Computational Linguistics*, 2019a. URL https://arxiv.org/abs/1905.05950.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=SJzSgnRcKX.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Anne Treisman. The binding problem. *Current opinion in neurobiology*, 6(2):171–178, 1996.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020a.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in Neural Information Processing Systems*, 33:12388–12401, 2020b.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Zhengxuan Wu, Atticus Geiger, Christopher Potts, and Noah D Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. *arXiv preprint arXiv:2305.08809*, 2023.

Appendix A. Binding ID mechanism

Formally, the binding ID mechanism states that there are *binding functions* $\Gamma_E(e, k)$ and $\Gamma_A(a, k)$ that fully specify how Z_E and Z_A bind entities/attributes with binding IDs. Specifically, if $E_k = e \in \mathcal{E}$, then we can replace Z_{E_k} with $\Gamma_E(e, k)$ without changing the query behavior, and likewise for Z_A.

Appendix B. Necessity of Binding ID mechanism

From the factorizability of activations Z_{context} we can construct candidate binding functions Γ_E as follows: For any $e \in \mathcal{E}$ and any binding ID $k \in [0, n-1]$, pick any context $\mathbf{c}$ such that $E_k(\mathbf{c}) = e$. Then, let $\Gamma_E(e, k) = Z_{E_k}(\mathbf{c})$. Γ_A can be constructed similarly.

From position independence we argued that the apparent positions of Z_{E_k} and Z_{A_k} are mostly causally irrelevant for the belief of the LM.

Putting these two properties together, we have the finding that, for all entities $e_0, e_1 \in \mathcal{E}$, and all attributes a_0, a_1 the language model have opposite beliefs in these two contexts:

$$Z^0 := Z_{\text{context}}/.\{Z_{E_0} \to \Gamma_E(e_0, 0), Z_{A_0} \to \Gamma_A(a_0, 0), Z_{E_1} \to \Gamma_E(e_1, 1), Z_{A_1} \to \Gamma_A(a_1, 1)\}$$

$$Z^1 := Z_{\text{context}}/.\{Z_{E_0} \to \Gamma_E(e_0, 1), Z_{A_0} \to \Gamma_A(a_0, 0), Z_{E_1} \to \Gamma_E(e_1, 0), Z_{A_1} \to \Gamma_A(a_1, 1)\}$$

For the LM to be able to have different behavior on Z^0 and Z^1, there must be *something* in $\Gamma_E(e_0, 0)$ that marks it out as corresponding to a different attribute than $\Gamma_E(e_0, 1)$, so that the LM knows to bind e_0 to a_0 in Z^0 and e_0 to a_1 in Z^1.

Now consider

$$Z^2 := Z_{\text{context}}/.\{Z_{E_0} \to \Gamma_E(e_0, 0), Z_{A_0} \to \Gamma_A(a_0, 1), Z_{E_1} \to \Gamma_E(e_1, 1), Z_{A_1} \to \Gamma_A(a_1, 0)\}.$$

Similarly, for the LM to behave differently in Z^0 and Z^2, there must be *something* in $\Gamma_A(a_0, 0)$ that marks it as corresponding to a different entity than $\Gamma_A(a_0, 1)$.

Thus, it appears that the binding information must be contained in the activations Z_{E_k} and Z_{A_k} themselves. This leads the formulation of the binding ID mechanism:

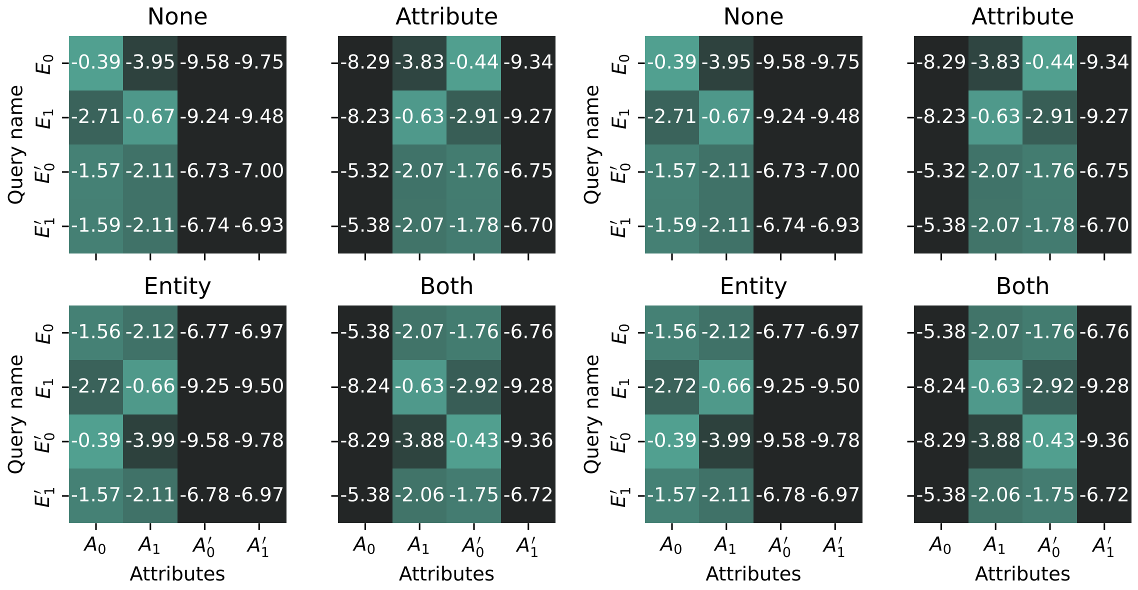
- For all attributes a, for all entities e, and for all $k \in [0, n-1]$, $\Gamma_A(a, k)$ binds a to an abstract binding ID that is referred to by the index k, and $\Gamma_E(e, k)$ binds e to an abstract binding ID that is referred to by the index k
- These binding functions are highly localized to the representations of the tokens carrying information for a and e respectively
- There is a query mechanism that decides if $\Gamma_A(a, k)$ is bound to $\Gamma_E(e, l)$ by checking if k is equal to l

Appendix C. Factorizability

The first property of Z_{context} we test is *factorizability.* We first explain in more detail why our claimed mechanism implies factorizability, then provide experimental verification.

In the binding ID mechanism, information is highly localized—it claims that information about A_k is located at Z_{A_k}. Therefore, we expect LMs that implement the binding ID mechanism to have factorizable activations in that for any contexts $\mathbf{c}, \mathbf{c}'$, substituting $Z_{E_k}(\mathbf{c}) \to Z_{E_k}(\mathbf{c}')$ into Z_{context} will cause the model to believe $\mathbf{c}/.\{E_k \to E'_k\}$, and substituting $Z_{A_k}(\mathbf{c}) \to Z_{A_k}(\mathbf{c}')$ cause the model to believe $\mathbf{c}/.\{A_k \to A'_k\}$.

In practice, we find that the entity encoding is diffused across two token activations, and thus for all experiments in the paper, we expand the definition of Z_{E_k} to include the token activations immediately after E_k. We use LLaMa 30-b unless otherwise stated.



(a) (E_0, A_0)

(b) (E_1, A_1)

Figure 3: Factorizability results for each pair of attributes. Each row corresponds to querying for a particular entity. Plotted are the mean log prob for all four attributes.

Experiments. To test the factorizability hypothesis, we conduct causal mediation experiments on the CAPITALS task with $n = 2$, i.e. with two entity-attribute pairs. Consider two sets of contexts, the target context $\mathbf{c} = \text{ctxt}(e_0 \leftrightarrow a_0, e_1 \leftrightarrow a_1)$ and the source context $\mathbf{c}' = \text{ctxt}(e_2 \leftrightarrow a_2, e_3 \leftrightarrow a_3)$. We choose either (E_0, A_0) (Fig. 3a) or (E_1, A_1) (Fig. 3b) to intervene on. We will intervene on either just the entity ($Z_{E_K} \to Z'_{E_k}$), just the attribute, neither, or both. The mean log probs for each of these settings are shown in Fig. 3.

The results support the factorizability hypothesis. As an example, consider Fig. 3a. In the None setting, we see high log probs for A_0 when queried for E_0, and for A_1 when queried for E_1. This indicates that the LM is able to solve this task. Next, consider the Attribute intervention setting: querying for E_0 now gives high log probs for A_0', and querying for E_1 gives A_1 as usual. Finally, in the Both setting, A_0' ends up bound to E_0' as can also be seen in the log probs.

Appendix D. Position Independence

We next turn to position independence, which is the other property we expect LMs implementing the binding ID mechanism to have. This says that permuting the order of the Z_{E_k} and Z_{A_k} should have no effect on the output, because the LM looks only at the binding IDs and not the positions of entities or attributes activations.

To test this experimentally, recall that transformers use positional embeddings to encode the (relative) position of each token in the input. We can intervene on these embeddings to "move" one of the Z_k's to another location k'. In Appendix E we describe how to do

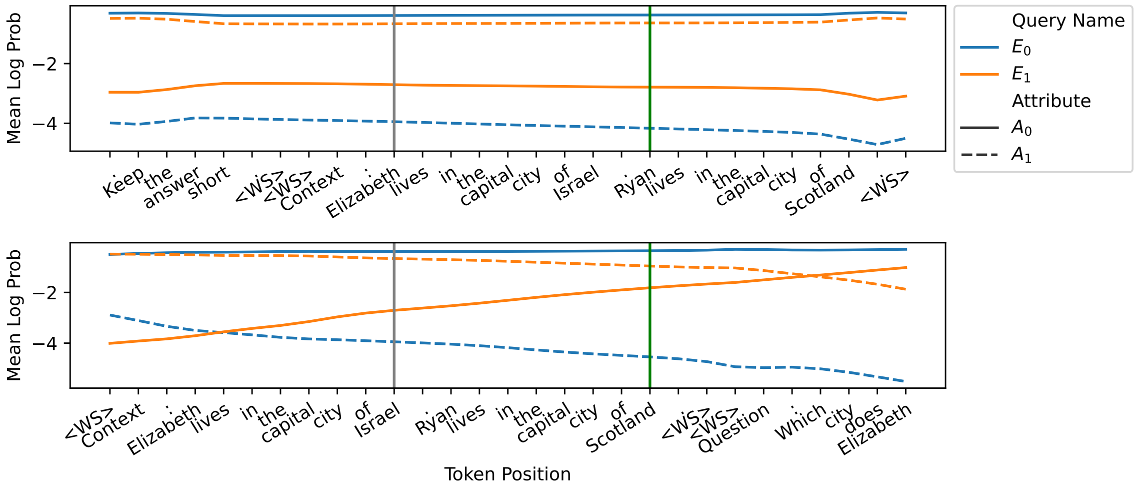


Figure 4: Top: Mean log probs for entity interventions. Bottom: Mean log probs for attributes. For brevity, let Z_k refer to Z_{E_k} or Z_{A_k}. The grey and green vertical line indicate the starting position for Z_0 and Z_1 respectively. The x-axis marks Z_0's position. For every step Z_0 takes to the right, Z_1 takes one step to the left. Thus, the grey line is the *control condition* with no interventions, and the green line is the *swapped condition* where Z_0 and Z_1 have swapped positions.

this for rotary position embeddings (RoPE), which underlie all the models studied in this paper. For now, we will assume this intervention as a primitive and discuss experimental results.

Experiments. We first intervene on the positions of the entity activations in a $n = 2$ context, Z_{E_0} and Z_{E_1}, by simultaneously shifting them by an equal and opposite amount. We then query the model with E_0 and E_1 and measure the mean log prob over the two attributes A_0 and A_1. We repeat the same experiment with attribute activations.

Fig. 4 shows that as predicted, position interventions result in little change in model behavior. Consider the *swapped condition* at the green line. Had the binding information been entirely encoded in position, we expect a complete switch in beliefs compared to the *control condition.* In reality, we observe almost no change in mean log probs for entities and a small change in mean log probs for attributes that is due to a *position dependent bias.* We discuss the position dependent bias and other experimental details in Appendix E.

Appendix E. Details for Position Independence

We can equivalently think of Z_{context} as a set of pairs: $Z_{\text{context}} = \{(p, Z_p) \mid p \text{ is an index for a context token}\}$. LMs that use Rotary Position Embedding (RoPE) (Su et al., 2021), such as those in the LLaMa and Pythia families, have architectures that allow arbitrarily intervention on the apparent position of an activation $(p, Z_p) \to (p', Z_p)$, even if this results in overall context activations that cannot be written down as a list of activations. This is because position information is applied at every layer, and not injected into the residual stream like in absolute position embeddings. Specifically, equation 16 in Su et al. (2021) provides the definition of

RoPE (recreated verbatim as follows):

$$q_m^\intercal k_n = (\mathbf{R}^d_{\Theta,m}\mathbf{W}_q x_m)^\intercal(\mathbf{R}^d_{\Theta,n}\mathbf{W}_k x_n) \tag{3}$$

Then, making the intervention $\mathbf{R}^d_{\Theta,n} \to \mathbf{R}^d_{\Theta,n^*}$ changes the apparent position of the activations at position n to the position at n^*.

Appendix F. Additivity

F.1. Mean interventions

To experimentally test additivity, we would like to extract $b_A(k)$ and $b_E(k)$ in order to perform vector arithmetic on them. We use (1) to extract the *differences* $\Delta_E(k) := b_E(k) - b_E(0)$, $\Delta_A(k) := b_A(k) - b_A(0)$. Rearranging (1), we obtain

$$\Delta_A(k) = \Gamma_A(\alpha, k) - \Gamma_A(\alpha, 0), \quad \Delta_E(k) = \Gamma_E(a, k) - \Gamma_E(a, 0). \tag{4}$$

We estimate $\Delta_A(k)$ by sampling $\mathbb{E}_{\mathbf{c},\mathbf{c}'}[Z_{A_k}(\mathbf{c}) - Z_{A_0}(\mathbf{c}')]$, and likewise for $\Delta_E(k)$.

In our experiments, we fix $n = 2$ and use 500 samples to estimate $\Delta_E(1)$ and $\Delta_A(1)$. We then perform four tests. The first test is a control test where no interventions are done. In the second test, we switch the binding ID vectors in Z_{A_0} and Z_{A_1} by intervening $Z_{A_0} \to Z_{A_0} + \Delta_A(1), Z_{A_1} \to Z_{A_1} - \Delta_A(1)$. In the third test, we switch the binding ID vectors in Z_{E_0} and Z_{E_1} by intervening $Z_{E_0} \to Z_{E_0} + \Delta_E(1), Z_{E_1} \to Z_{E_1} - \Delta_E(1)$. We term these the *mean intervention* on attribute and entity binding IDs respectively, and expect them to result in a complete switch in model beliefs so that the accuracy is near 0. Results are displayed in Table 1, and show agreement with this prediction: all accuracies are below 3%.

As a further check, we intervene on both the attribute and entity IDs simultaneously, which should cancel out and thus restore accuracy. Indeed, Table 1 shows that accuracy in this setting is above 97%. Finally, to show that the *specific* directions obtained by binding IDs matter, we apply a random rotation to the difference vectors, and perform the same mean intervention with the rotated vectors. These random vectors have no effect on the model behavior.

F.2. The Geometry of Binding ID Vectors

Appendix F shows that we can think of binding IDs as pairs of ID vectors, and that randomly chosen vectors do not function as binding IDs. We next investigate the geometric structure of valid binding vectors and find that linear interpolations or extrapolations of binding

Test condition	Control	Attribute	Entity	Both	Attribute	Entity	Both
Querying E_0	0.99	0.00	0.00	0.97	0.98	0.98	0.97
Querying E_1	1.00	0.03	0.01	0.99	1.00	1.00	1.00

Table 1: Left: Mean calibrated accuracies for mean interventions on four test conditions. Columns are the test conditions, and rows are queries. Right: Mean interventions with random vectors.

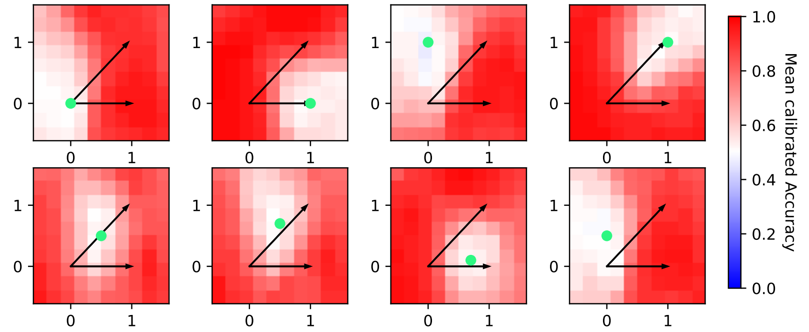


Figure 5: The plots show the mean median-calibrated accuracy when one pair of binding ID, v_0, is fixed at the green circle, and the other, v_1, is varied. Each cell on the grid represents a value of v_1. The binding IDs $b(0)$, $b(1)$, and $b(2)$ are shown as the origin of the arrows, the end of the horizontal arrow and the end of the diagonal arrow respectively. We use LLaMa-13b for computational reasons.

vectors are often also valid binding vectors. This suggests that binding vectors occupy a continuous *binding subspace*. We find evidence of a metric structure in this space, such that nearby binding vectors are hard for the model to distinguish, but far-away vectors can be reliably distinguished and thus used for the binding task.

To perform our investigation, we start with an $n = 2$ context, and thus obtaining representations $Z_0 = (Z_{E_0}, Z_{A_0})$ and $Z_1 = (Z_{E_1}, Z_{A_1})$. We first erase the binding information by subtracting $(\Delta_E(1), \Delta_A(1))$ from Z_1, which reduces accuracy to chance. Next, we will add vectors $v_0 = (v_{E_0}, v_{A_0})$ and $v_1 = (v_{E_1}, v_{A_1})$ to the representations Z; if doing so restores accuracy, then we view (v_{E_0}, v_{A_0}) and (v_{E_1}, v_{A_1}) as valid binding pairs.

To generate different choices of v, we take linear combinations across a two-dimensional space. The basis vectors for this space are $(\Delta_E(1), \Delta_A(1))$ and $(\Delta_E(2), \Delta_A(2))$ obtained by averaging across an $n = 3$ context. Fig. 5 shows the result for several different combinations, where the coordinates of v_0 are fixed and shown in green while the coordinates of v_1 vary. When v_1 is close to v_0, the LM gets close to 50% accuracy, which indicates confusion. Far away from v_1, the network consistently achieves high accuracy, demonstrating that linear combinations of binding IDs (even with negative coefficients) are themselves valid binding IDs.

The geometry of the binding subspace hints at circuits (Elhage et al., 2021) in LMs that process binding vectors. For example, we speculate that certain attention heads might be responsible for comparing binding ID vectors, since the attention mechanism computes attention scores using a quadratic form which could provide the metric over the binding subspace.

Appendix G. Generality and Limitations of Binding ID

The earlier sections investigate binding IDs for one particular task: the capitals task. In this section, we evaluate their generality. We first show that binding vectors are used for a variety of tasks and models. We then show evidence that the binding vectors are task-

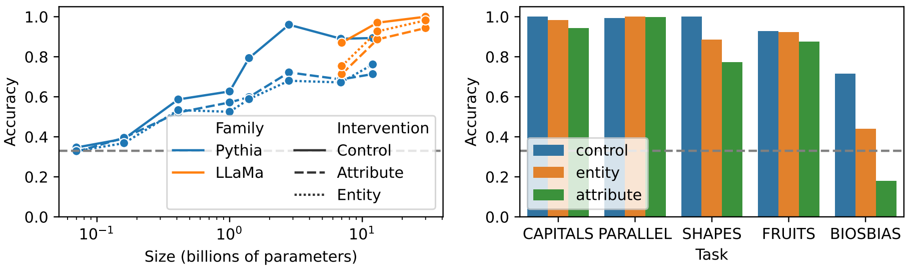


Figure 6: Left: models in Pythia and LLaMa on CAPITALS. LLaMa-65b not present for computational reasons. Right: LLaMa-30b on binding tasks. Unlike others, the BIOS task has attributes that are several tokens long.

Task	CAPITALS	PARALLEL	SHAPES	FRUITS	BIOS	Zeros	Random
Mean accuracy	0.88	0.87	0.71	0.80	0.47	0.30	0.31
Mean log prob	-1.01	-1.07	-1.18	-1.21	-1.64	-1.86	-2.15

Table 2: The mean median calibrated accuracy and mean log prob for mean interventions on $n = 3$ CAPITALS using binding ID estimates from other tasks. Random chance has 0.33 mean accuracy.

agnostic: vectors from one task transfer across many different tasks. Finally, we show that our mechanism is not fully universal, by exhibiting a question-answering task that uses an alternative binding mechanism.

Generality of binding ID vectors. We evaluate the generality of binding vectors across models and tasks. For a (model, task) pair, we evaluate the model's mean median-calibrated accuracy on the $n = 3$ context under three conditions: (1) the control condition in which no interventions are performed, and the (2) entity and (3) attribute conditions in which entity or attribute mean interventions are performed. The mean interventions modify the binding pairs by a cyclic shift, and we measure accuracy according to this cyclic shift. As shown in Figure 6, these mean interventions induce the expected behavior on most tasks; moreover, their effectiveness increases with model scale, suggesting that perhaps larger models generalize better because they have more robust structured representations.

Transfer across tasks. We next show that binding vectors often transfer across tasks. Without access to the binding vectors $[b_E(k), b_A(k)]$, we instead test if the difference vectors $[\Delta_E(k), \Delta_A(k)]$ from a source task, when added to binding vectors from a target task, result in valid binding IDs. To do so, we follow a similar procedure to Appendix F.2: First, we erase binding information by subtracting $[\Delta_E(k), \Delta_A(k)]$ for the target task from each target-task representation $[Z_{E_k}, Z_{A_k}]$, which results in near-chance accuracy. Then, we add back in $[\Delta_E(k), \Delta_A(k)]$ computed from the *source* task with the hope of restoring performance.

Table 2 shows results for a variety of source tasks when using CAPITALS as the target task. Accuracy is consistently high, even when the target task has limited surface similarity to the target task. For example, the SHAPES task contains descriptions about geometrical shapes and their colors, and PARALLEL has a parallel sentence structure. In addition, we include

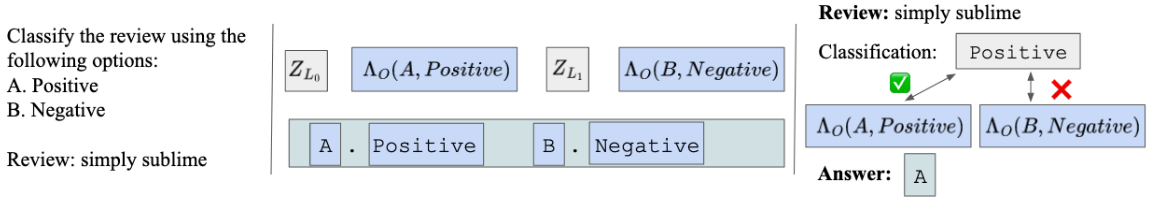


Figure 7: Direct binding in MCQ task. O_k and L_k denote options and labels respectively. Z_{L_0} and Z_{L_1} are causally irrelevant. Z_{O_0} and Z_{O_1} are represented by a binding function Λ_0 that directly binds option and label together.

two baselines: "Zeros" corresponds to not performing the second step of re-introducing Δ's, and "Random" corresponds to reintroducing a randomly rotated vector (as in Table 1). Both lead to chance accuracy. See Appendix H for more details on the tasks.

The fact that binding vectors transfer across tasks, together with the binding subspace results from Appendix F.2, suggests that there could be a task-agnostic subspace in the model's activations reserved for binding vectors.

Appendix H. Binding Task Details

H.1. capitals

Construct a list of one-token names and a list of country-capital pairs that are also each one-token wide. Then, apply the following template:

```
Answer the question based on the context below. Keep the answer short.

Context: {E_0} lives in the capital city of {A_0}.
{E_1} lives in the capital city of {A_1}.

Question: Which city does {qn_subject} live in?

Answer: {qn_subject} lives in the city of
```

The LM is expected to answer with the capital of the country that is bound to the queried entity. Note that the LM is expected to simultaneously solve the factual recall task of looking up the capital city of a country.

H.2. parallel

The PARALLEL task uses the same country capital setup, but with the prompt template:

```
Answer the question based on the context below. Keep the answer short.

Context: {E_0} and {E_1} live in the capital cities of {A_0} and {A_1}
    respectively.

Question: Which city does {qn_subject} live in?

Answer: {qn_subject} lives in the city of
```

This prompt format breaks the confounder in the CAPITALS task that entity always appear in the same sentence as attributes, suggesting binding ID is not merely a syntactic property.

H.3. fruits

The FRUITS task uses the same set of names, but for attributes it uses a set of common fruits and food that are one-token wide. The prompt format is:

```
Answer the question based on the context below. Keep the answer short.

Context: {E_0} likes eating the {A_0}. {E_1} likes eating the {A_1} respectively.

Question: What food does {qn_subject} like?

Answer: {qn_subject} likes the
```

H.4. shapes

The SHAPES tasks have entities which are one-token wide *colors*, and attributes which are one-token wide *shapes*. The prompt looks like:

```
Answer the question based on the context below. Keep the answer short.

Context: The {A_0} is {E_0}. The {A_1} is {E_1}.

Question: Which shape is colored {qn_subject}?

Answer: The {qn_subject} shape is
```

This task inverts the assumption that entities have to be nouns, and attributes are adjectives.

H.5. Bios

This task is adapted from the bias in bios dataset De-Arteaga et al. (2019), with a prompt format following Hernandez et al. (2023a). The entities are the set of one-token names, and the attributes are a set of biography descriptions obtained using the procedure from Hernandez et al. (2023a). The LM is expected to infer the occupation from this description. This time, the attributes are typically one sentence long, and are no longer one-token wide. We thus do not expect the mean interventions for attributes to work, although we may still expect entity interventions to work. Just inferring the correct occupation is also a much more challenging task than the other synthetic tasks.

The prompt format is:

```
Answer the question based on the context below. Keep the answer short.

Context:
About {E_0}: {A_0}
About {E_1}: {A_1}
```

```
Question: What occupation does {qn_subject} have?
Answer: {qn_subject} has the occupation of
```

Appendix I. Related work

Symbolic representations in connectionist systems Many have studied how neural networks embed concepts in activation space (Mikolov et al., 2013; Tenney et al., 2019a,b; Rogers et al., 2021). These tend to rely on correlational rather than causal relationships, leading to a propensity to overestimate the role of these activations (Belinkov and Glass, 2019). Our approach is centered around evaluating the representations' causal effect on model behavior.

Recent works (Nanda et al., 2023; Li et al., 2022) have studied representations in small transformer based language models trained on toy algorithmic tasks. This work extends the study of representations to large language models trained on natural language data.

Hernandez et al. (2023a) studied the representations of in-context statements and found directions corresponding to attributes in activation space, that when injected to the activations for a subject, seem to bind the attribute to the subject. Our work extends this in two ways. First, we test binding in a rigorous setting that requires discrimination between choices. Second, we investigate the binding mechanism inherent in LMs instead of an ad hoc, hand-written binding mechanism.

Knowledge recall. A line of work studies recalling factual associations that LMs learn from pretraining (Geva et al., 2020; Dai et al., 2021; Meng et al., 2022; Geva et al., 2023). (Hernandez et al., 2023b), in particular, concurrently studied the representation of factual relations learned from *pretraining* and how they are recalled from model *weights*. In contrast, we study representations of relations learned from *context*, and how they are recalled from model *activations*.

Mechanistic Interpretability. Mechanistic interpretability aims to uncover circuits (Elhage et al., 2021; Wang et al., 2022; Wu et al., 2023), often composed of attention heads, that are embedded in language models. In our work, we study language model internals on a more coarse-grained level. We identified structures in representations that have causal influences on model behavior, but how circuits construct these representations or utilize them is left as future work.