Latent Concept Disentanglement in Transformer-based Language Models

Guan Zhe Hong* HONG288@PURDUE.EDU
Purdue University

Bhavya Vasudeva* BVASUDEV@USC.EDU
University of Southern California

Vatsal Sharan VSHARAN@USC.EDU
University of Southern California

Cyrus Rashtchian CYROID@GOOGLE.COM
Google Research

Prabhakar Raghavan PRAGH@GOOGLE.COM
Google Research

Rina Panigrahy RINAP@GOOGLE.COM
Google Research

Abstract

When large language models (LLMs) use in-context learning (ICL) to solve a new task, they seem to grasp not only the goal of the task but also core, latent concepts in the demonstration examples. This begs the question of whether transformers represent latent structures as part of their computation or whether they take shortcuts to solve the problem. Prior mechanistic work on ICL does not address this question because it does not sufficiently examine the relationship between the learned representation and the latent concept, and the considered problem settings often involve only single-step reasoning. In this work, we examine how transformers disentangle and use latent concepts. We show that in 2-hop reasoning tasks with a latent, discrete concept, the model successfully identifies the latent concept and does step-by-step concept composition. In tasks parameterized by a continuous latent concept, we find low-dimensional subspaces in the representation space where the geometry mimics the underlying parameterization. Together, these results refine our understanding of ICL and the representation of transformers, and they provide evidence for highly localized structures in the model that disentangle latent concepts in ICL tasks.

1. Introduction

Transformer-based Large Language Models (LLMs) demonstrate remarkable in-context learning (ICL) abilities: with only a handful of input-output demonstrations, they can generalize to new inputs without any parameter updates [11]. These successes hint that models might be inferring latent rules or concepts implicit in the prompt. Yet it is still unresolved whether transformers truly form explicit internal representations of such hidden structure, or whether apparent generalization can be explained by more superficial pattern matching. This work therefore asks:

How do LLMs disentangle and manipulate latent concepts during in-context learning?

* Equal contribution.

© G.Z. Hong, B. Vasudeva, V. Sharan, C. Rashtchian, P. Raghavan & R. Panigrahy.

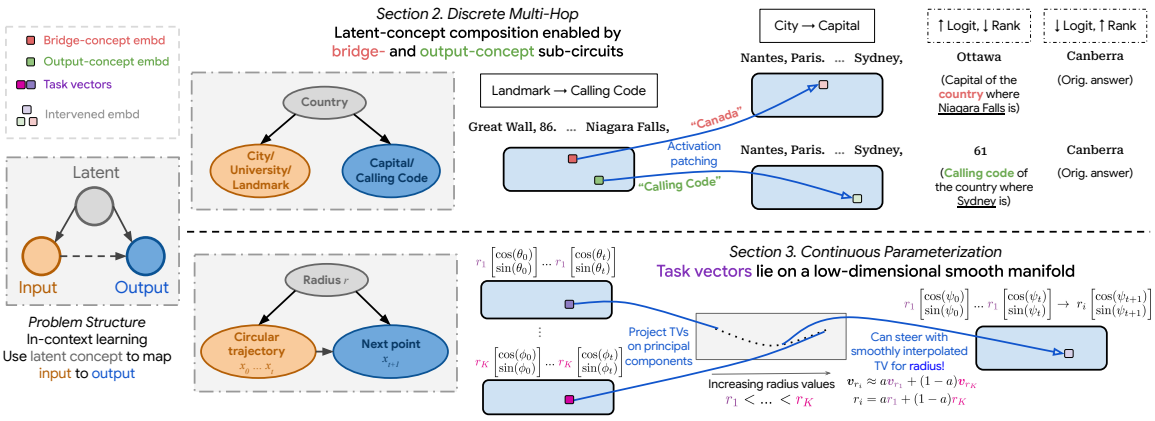


Figure 1: An illustration of problem setup and examples of main findings in this work. We primarily focus on how decoder-only transformer-based language models disentangle and manipulate latent concepts for solving in-context learning (ICL) problems. In the **multi-hop ICL** setting, we discover that transformers *compose* disentangled latent concept representations for predicting the answer. For example, as shown in the upper half of the figure, by intervening on certain "bridge-concept" attention heads, we can push the model's "belief" (reflected in logit and rank) in the *original answer* Canberra (the capital of Australia, which the city Sydney belongs to) to the "type-corrected" *alternative answer* Ottawa (the capital of Canada, which the landmark Niagara Falls belongs to). In the **continuous-parameterization ICL** setting, we discover that transformers' hidden embeddings capture the *geometry* of the latent concepts for our prediction tasks. For instance, for a transformer trained to predict circular trajectories whose radius is randomly chosen from a continuous interval, not only do we obtain causal evidence for *task vectors* which control the trajectory's *radius*, but they also fall on a *smooth* 2D manifold.

Prior mechanistic studies leave this question open: most focus on tasks with simple latent structures which do not require intricate disentanglement nor manipulation of concepts. For instance, recent research has identified linear task or function vectors for problems such as basic arithmetic, single-step reasoning, and linguistic mappings [20, 43], and isolated certain attention heads that drive ICL behavior on these straightforward problems [12, 33, 52]. However, *real ICL usage often involves richer latent structures* — for instance, demonstrations with under-specified intermediate reasoning steps, or that share continuous latent parameters. It remains unclear how transformers *represent* these latent concepts, and how they *disentangle* and *reuse* them.

We address these gaps by probing two challenging ICL settings:

- *Discrete multi-hop reasoning*. A model must map a "source" entity to a "target" entity (e.g., landmark→capital) by first latently inferring the hidden "bridge" entity (e.g., country), allowing us to ask whether the LLM first resolves the "bridge", then refines it to a "target" entity via (causal) concept compositions.

- *Continuous-parameter tasks*. Demonstrations are generated by unknown real-valued parameters (e.g., the radius of a circular trajectory), allowing us to ask whether the model encodes such parameters along a systematic low-dimensional geometry.

Both tasks are *demanding enough* to require intricate latent concept disentanglement and manipulation, yet *sufficiently controlled* to permit causal, feature and circuit-level analysis.

Figure 1 illustrates the main ideas of our paper. Our experiments show that *transformers do learn, and when necessary, compose disentangled latent concept representations in these settings*. In the multi-hop task, we identify a highly sparse circuit of attention heads that first recovers the "bridge" concept, and then specializes it using "target concepts" to produce the answer. In the continuous tasks, we find that the model's task vectors lie on a low-dimensional manifold precisely aligned with the underlying parameter. These results show that transformers internalize latent structure through specialized mechanisms rather than heuristics.

2. Disentanglement for Latent Discrete Task in Multi-hop Reasoning

Prior works have mechanistically analyzed how LLMs solve ICL problems that require a single step of reasoning over world knowledge, such as geography puzzles "Country→Capital", "National Park→Country" [43, 52]. However, whether and how LLMs can solve ICL problems with *under-specified reasoning steps*, or essentially those requiring *latent multi-hop reasoning* remains unclear.

The "Source→Target" problem. We create two-hop ICL puzzles by composing two facts linked by a common "Bridge" entity.[1] That is, we sample fact tuples $\{(S_i, r_1, B_i, r_2, T_i)\}_{i=1}^n$, where the Source entity S_i is related to the Bridge entity B_i via relation r_2, and B_i is related to the Target entity T_i via r_2. We then create the ICL puzzle in the form $[S_1, T_1. S_2, T_2 ... S_n,]$ [Answer: T_n]. Note that the bridge entities B_i's are never specified in the prompt. We primarily work with geography puzzles, with input types {City, University, Landmark}, and output types {Capital, Calling code}. An example problem is

`Sydney, Canberra. Nantes, Paris. Oshawa,`

Here, r_1 is "belongs to the country of", the (unspecified) bridge entities for this example are "Australia", "France", "Canada", and r_2 is "has capital". Therefore, the prompt's answer is `Ottawa`, the capital of Canada, the country `Oshawa` is in. We discuss dataset details in Appendix A.1.

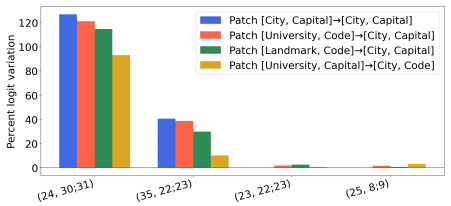
As shown in Figure 7, Gemma-2-27B [17] achieves high accuracy ($> 80\%$) at 20 shots. How is the LLM solving these harder, "source→target" problems that *do not specify the bridge, or even hint at the compositional nature of the problem at all*? In the following, we show a surprising finding that a highly sparse set of attention heads are responsible for "resolving the bridge value".

2.1. Bridge-value resolution for answering the query

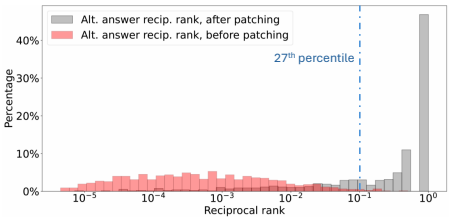
We analyze how the LLM infers the answer (target entity) from the query (source entity), given sufficiently many in-context examples. We will show both causal and correlational evidence that the LLM latently infers the bridge entity as an abstract concept representation, and compose it with output-concept representations to produce the answer. To achieve this, there are two main steps of our experiments, which we discuss below.

We perform activation patching [45, 54] on normal and altered problem pairs with different bridge entities (different countries), across different source and target types, at the last token position. If our hypothesis of certain attention heads resolving the bridge value from S_n for prediction is correct, then this bridge representation must be *transferable across source and target types*: for instance, patching a [University→Calling Code] prompt's activation (i.e. the "altered activation") onto that of

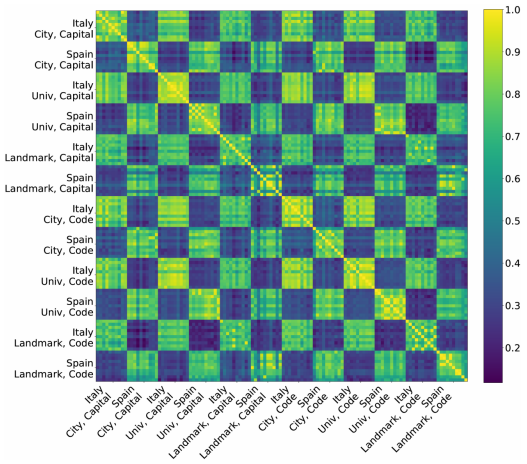
1. We think of this as a systematic ICL version of TWOHOPFACT [51]. Here, the model must figure out relations between the *input-bridge* and *bridge-output* facts from the ICL examples.



(a) Logit difference variations from select activation patching experiments



(b) Reciprocal rank of the altered answer before and after patching head group (24,30;31)



(c) Cosine similarity of "Italy" and "Spain" representations from head group (24,30;31)

Figure 2: Evidence of the bridge-resolving attention heads in Gemma-2-27B. (a) and (b) provide causal evidence and (c) provides correlational evidence, which we elaborate in the main text.

a normal prompt [City→Capital], should cause the model to favor the alternative prompt's answer, but in the form of Capital (instead of Calling Code).

This suggests a slightly unorthodox intervention experiment. Instead of taking the normal-alternative answer pairs to be the ground truths, we convert the alternative answer into the same output type as the normal one, i.e., set $\hat{T}_n^{(alt)} = \text{Type}_{norm}(T_n^{(alt)})$. The logit difference and (reciprocal) rank of the alternative answer would use $\hat{T}_n^{(alt)}$ instead of the raw target $T_n^{(alt)}$. For instance, if the target type of the normal prompt is Capital, and the alternative prompt ends with "Sydney, 61", then $\hat{T}_n^{(alt)} =$ Canberra. We discuss the full detail of our patching experiments in Appendix A.2.

We report the results in Figure 2. In Figure 2(a), we report results on a select set of patching experiments: on *both* tasks with and without overlap in source and target types, we observe that a sparse set of attention heads consistently exhibits very strong causal effects; the head group (24,30;31) is especially dominant.[2] To further understand whether (24,30;31) is really boosting the alternative answer (instead of only decreasing model's confidence on the normal answer, which logit difference might not tell), in Figure 2(b), we show an example patching experiment result of [University, Code]→[City, Capital]. Surprisingly, at least 73% of the time, patching this single head group can boost the model's rank of the alternative prompt's answer into top 10 (and directly become the top-1 answer more than 40% of the time!), when its original rank, obtained on the normal prompt without intervention, is typically in the hundreds to thousands.

Finally, to understand the nature of (24,30;31)'s output embeddings better, in Figure 2(c), we visualize an example cosine similarity matrix of this attention head, with either "Italy" or "Spain" as the bridge values for S_n (the query source entity) in the prompts, across a total of 12 different combinations of bridge, source and target types. Specifically, for each combination of the bridge and source-target type shown in the grid, we sample 10 prompts which obey such requirement[3], giving us a total of 120 prompts. We then obtain head group (24,30;31)'s embedding of these prompts at

2. In our CMA experiments, we account for grouped-query attention by patching heads in groups of 2 on Gemma-2-27B. We noticed that this tends to produce stronger causal effects than with individual heads.
3. We only specify the bridge entity for S_n in the prompt; the bridge for S_i for all $i < n$ are randomly chosen.

the last token position, and compute the pairwise cosine similarities. Observe that the embedding consistently exhibits strong *disentanglement* with respect to the bridge value in the prompt, *regardless of source and target types*. We delay further analysis of the circuit and more general statistics of disentanglement strength of the bridge concepts to the appendix due to space limitations.

3. Disentanglement for Latent Continuous Parameterization

In this section, we consider two problems with numerical or continuous parameterization. For these experiments we study a very small transformer, with a similar architecture to GPT-2 [38]. We use a 2-layer 1-head transformer, with embedding dimension 128, trained with the AdamW optimizer. Additional details about training and hyperparameter choices are in the Appendix.

add-k Problem. Each task is a sequence consisting of pairwise examples $\{(x_i, y_i)\}_{i=1}^{n+1}$, where $y_i = x_i + k$, for a given offset k. Here, we use integer inputs and offsets; all values are in $\{0, \ldots, V-1\}$, each treated as a distinct token. We consider a collection of K tasks parameterized by different offset values in $\{k_i\}_{i=1}^{K}$, where $k_1 = 1$ and we fix $k_{i+1} - k_i = 3$. The model is trained autoregressively to predict the label for each example in the sequence. At test time, the model observes the first n examples and should predict $y_{n+1} = x_{n+1} + k$ for the last example.

Circular-Trajectory Problem. Here a task consists of a sequence $\{\mathbf{x}_i\}_{i=1}^{n+1}$ of points on a circle centered at the origin. Each task is parameterized by the circle's radius r; for K tasks, the set of radii $\{r_i\}_{i=1}^{K}$ is sampled uniformly from $[1, 4]$. A task sequence is generated as follows. We first sample θ_0 uniformly at random in $[0, \frac{\pi}{2}]$, so $\mathbf{x}_1 = r[\cos\theta_0, \sin\theta_0]^T$. Then, we select the *period* p randomly from $\{2, 3, 4\}$, which determines the number of equal consecutive step-sizes. Specifically, we first sample a sequence of $\lfloor \frac{n}{p} \rfloor + 1$ unique step-sizes uniformly between $[0, 1]$, and then get the full sequence of steps $\{a_i\}_{i=1}^{n}$, where $a_j = a_{j+1} = \cdots = a_{j+p-1}$ for $j \in \{0, p, 2p, \ldots\}$. Here, context length $n = 12m + 1$ for integer m. We also sample $c \in \{\pm 1\}$, which denotes if the trajectory is clockwise or anticlockwise. Next, we generate a sequence of angles $\{\theta_i\}_{i=1}^{n}$, where $\theta_i = \theta_0 + c\frac{2\pi}{n}\sum_{j \le i} a_j$. Using the sequence of angles, we generate the sequence, $\mathbf{x}_{i+1} = rR(\theta_i)\mathbf{x}_i$, where $R(\theta)$ is the 2D rotation matrix for θ. Figure 25 shows an example. As in the previous problem, we train the transformer autoregressively on these types of sequences.

Existence of Task Vectors. We first outline the process to identify the task vectors for the add-k problem. We set $V = 100$, $n = 4$, and $K = 2$. Figure 3 shows the cosine similarities between the layer-2 attention embeddings at the last position for 200 input sequences from each of the two tasks. We observe strong clustering between intra-task embeddings. This shows that the model disentangles the concept of different offset values in its representation.

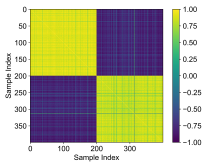


Figure 3: Cosine similarities between the layer-2 attention embeddings for 200 input sequences from two tasks/offsets for the *add-k* problem. Strong clustering between intra-task embeddings shows that the model disentangles the concept of different offsets.

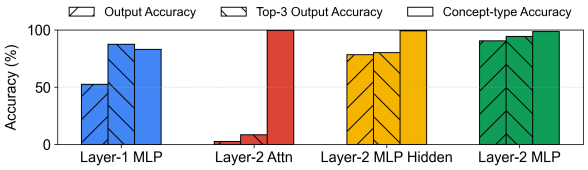


Figure 4: Results for linear probing the embeddings of the trained model at various locations to predict the final output and the task type for the *add-k* problem. The task type becomes disentangled at layer-2 attention, and the output is computed in layer-2 MLP.

To provide causal evidence for disentanglement, and locate where the task vectors emerge in the model, we linear probe the embeddings of the trained model at various locations to predict the final output and the offset/task type. We probe embeddings from the output of the MLP at the first layer, the attention block at the second layer, and the hidden and output layers of the MLP at the second layer. The results are shown in Figure 4. We observe that task type becomes disentangled at layer-2 attention, and the output is computed at layer-2 MLP. For each task, we treat the layer-2 attention embeddings averaged across 200 input sequences from that task as the task vector.

Geometry of Task Vectors. We analyze the geometry of the task vectors for the two problems. We visualize the task vectors by performing PCA and projecting them onto the first two PCs.

Figure 5 presents the 2D PCA projection of the task vectors for the add-k problem, for $K = 4, 8, 16$ tasks/offsets. We observe that in all three settings, the task vectors lie on a 1D linear manifold. In each of the three cases, more than 99.9% of the variance is explained by the first PC. Notably, the model compresses the concept of offsets into a line with the ordering of the offsets (lower to higher) preserved on the manifold (left to right). To corroborate these results, we study the effect of steering using the task vectors, and include the results in the Appendix.

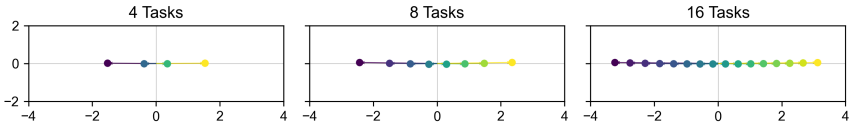


Figure 5: 2D PCA projection of the task vectors for the *add-k* problem. The task vectors lie on a 1D linear manifold. Here the number of tasks refers to the number of values of k.

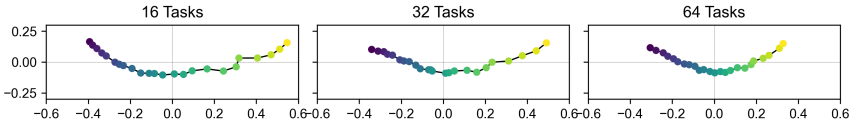


Figure 6: 2D PCA projection of the task vectors for the Circular-Trajectory problem. The task vectors lie on a smooth low-dimensional manifold. Here the number of tasks refers to the number of radius values used for training.

Figure 6 presents the 2D PCA projection of the task vectors for the Circular-Trajectory problem, for $K = 16, 32, 64$ (training) tasks/radii. In this setting, we consider $K = 24$ radii, spaced evenly between $[1, 4]$ to visualize the task vectors, since this task is continuous. We observe that in all three settings, the task vectors lie on a low-dimensional manifold. The variance explained by the first two PCs in the three cases is $97.05\%, 96.44\%, 93.68\%$, respectively. Similar to the previous setting, the order of the radii (lower to higher) is preserved in the compressed representation.

4. Discussion and Conclusion

We showed that transformer-based models latently *disentangle* core concepts in the provided in-context examples, and *manipulate* them well. For 2-hop tasks, we found that models contain sparse sets of attention heads responsible for first inferring the bridge entity and then resolving the output. For tasks with a continuous parameterization, we found that the model uses task vectors which closely capture the underlying parameterization. It would be interesting to understand the extent of this disentanglement across more diverse ICL tasks with various types of latent structures.

Acknowledgments

This work was supported in part by NSF CAREER Award CCF-2239265. The authors acknowledge the use of USC CARC's Discovery cluster.

References

[1] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *Int. Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

[2] Badr AlKhamissi, Greta Tuckute, Antoine Bosselut, and Martin Schrimpf. The LLM language network: A neuroscientific approach for identifying causally task-relevant units. In Luis Chiruzzo, Alan Ritter, and Lu Wang, 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 10887–10911, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.544. URL https://aclanthology.org/2025.naacl-long.544/.

[3] Badr AlKhamissi, Greta Tuckute, Antoine Bosselut, and Martin Schrimpf. The llm language network: A neuroscientific approach for identifying causally task-relevant units, 2025. URL https://arxiv.org/abs/2411.02280.

[4] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction, 2024. URL https://arxiv.org/abs/2406.11717.

[5] David D. Baek and Max Tegmark. Towards understanding distilled reasoning models: A representational approach, 2025. URL https://arxiv.org/abs/2503.03730.

[6] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection, 2023. URL https://arxiv.org/abs/2306.04637.

[7] Aleksandra Bakalova, Yana Veitsman, Xinting Huang, and Michael Hahn. Contextualize-then-aggregate: Circuits for in-context learning in gemma-2 2b. *arXiv preprint arXiv:2504.00132*, 2025.

[8] Daniel Beaglehole, Adityanarayanan Radhakrishnan, Enric Boix-Adserà, and Mikhail Belkin. Aggregate and conquer: detecting and steering llm concepts by combining nonlinear predictors over multiple layers, 2025. URL https://arxiv.org/abs/2502.03708.

[9] Satwik Bhattamishra, Arkil Patel, Phil Blunsom, and Varun Kanade. Understanding in-context learning in transformers and LLMs by learning to learn discrete functions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ekeyCgeRfC.

[10] Jannik Brinkmann, Abhay Sheshadri, Victor Levoso, Paul Swoboda, and Christian Bartelt. A mechanistic analysis of a transformer trained on a symbolic multi-step reasoning task. *arXiv preprint arXiv:2402.11917*, 2024.

[11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[12] Guy Davidson, Todd M. Gureckis, Brenden M. Lake, and Adina Williams. Do different prompting methods yield a common task representation in language models?, 2025. URL https://arxiv.org/abs/2505.12075.

[13] Clément Dumas, Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Separating tongue from thought: Activation patching reveals language-agnostic concept representations in transformers, 2025. URL https://arxiv.org/abs/2411.08745.

[14] Ezra Edelman, Nikolaos Tsilivis, Benjamin L. Edelman, Eran Malach, and Surbhi Goel. The evolution of statistical induction heads: In-context learning markov chains. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=qaRT6QTIqJ.

[15] 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.

[16] Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=flNZJ2eOet.

[17] Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

[18] Tianyu Guo, Hanlin Zhu, Ruiqi Zhang, Jiantao Jiao, Song Mei, Michael I Jordan, and Stuart Russell. How do llms perform two-hop reasoning in context? *arXiv preprint arXiv:2502.13913*, 2025.

[19] Lovis Heindrich, Philip Torr, Fazl Barez, and Veronika Thost. Do sparse autoencoders generalize? a case study of answerability, 2025. URL https://arxiv.org/abs/2502.19964.

[20] Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=QYvFUlF19n.

[21] Guan Zhe Hong, Nishanth Dikkala, Enming Luo, Cyrus Rashtchian, Xin Wang, and Rina Panigrahy. How transformers solve propositional logic problems: A mechanistic analysis. *arXiv preprint arXiv:2411.04105*, 2024.

[22] Xinyan Hu, Kayo Yin, Michael I Jordan, Jacob Steinhardt, and Lijie Chen. Understanding in-context learning of addition via activation subspaces. *arXiv preprint arXiv:2505.05145*, 2025.

[23] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

[24] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic, 2023. URL https://arxiv.org/abs/2212.04089.

[25] Yuxiao Li, Eric J. Michaud, David D. Baek, Joshua Engels, Xiaoqing Sun, and Max Tegmark. The geometry of concepts: Sparse autoencoder feature structure. *Entropy*, 27(4), 2025. ISSN 1099-4300. doi: 10.3390/e27040344. URL https://www.mdpi.com/1099-4300/27/4/344.

[26] Zhuowei Li, Zihao Xu, Ligong Han, Yunhe Gao, Song Wen, Di Liu, Hao Wang, and Dimitris N. Metaxas. Implicit in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=G7u4ue6ncT.

[27] Ziqian Lin and Kangwook Lee. Dual operating modes of in-context learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[28] Sheng Liu, Haotian Ye, Lei Xing, and James Zou. In-context vectors: making in context learning more effective and controllable through latent space steering. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[29] Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets, 2024. URL https://arxiv.org/abs/2310.06824.

[30] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023. URL https://arxiv.org/abs/2202.05262.

[31] Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Language models implement simple Word2Vec-style vector arithmetic. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5030–5047, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.281. URL https://aclanthology.org/2024.naacl-long.281/.

[32] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759/.

[33] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[34] Core Francisco Park, Ekdeep Singh Lubana, and Hidenori Tanaka. Competition dynamics shape algorithmic phases of in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=XgH1wfHSX8.

[35] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*, pages 39643–39666. PMLR, 2024.

[36] Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models, 2025. URL https://arxiv.org/abs/2406.01506.

[37] Judea Pearl. *Direct and Indirect Effects*, page 373–392. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022. ISBN 9781450395861. URL https://doi.org/10.1145/3501714.3501736.

[38] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. In *OpenAI blog*, 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

[39] Nived Rajaraman, Marco Bondaschi, Ashok Vardhan Makkuva, Kannan Ramchandran, and Michael Gastpar. Transformers on markov data: Constant depth suffices. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=5uG9tp3v2q.

[40] Jacob Russin, Ellie Pavlick, and Michael J. Frank. The dynamic interplay between in-context and in-weight learning in humans and neural networks, 2025. URL https://arxiv.org/abs/2402.08674.

[41] Aaditya K. Singh, Ted Moskovitz, Sara Dragutinovic, Felix Hill, Stephanie C. Y. Chan, and Andrew M. Saxe. Strategy coopetition explains the emergence and transience of in-context learning, 2025. URL https://arxiv.org/abs/2503.05631.

[42] Curt Tigges, Oskar John Hollinsworth, Neel Nanda, and Atticus Geiger. Language models linearly represent sentiment, 2024. URL https://openreview.net/forum?id=iGDWZFc7Ya.

[43] Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

[44] Edward Turner, Anna Soligo, Mia Taylor, Senthooran Rajamanoharan, and Neel Nanda. Model organisms for emergent misalignment, 2025. URL https://arxiv.org/abs/2506.11613.

[45] 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. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

[46] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

[47] Dimitri von Rütte, Sotiris Anagnostidis, Gregor Bachmann, and Thomas Hofmann. A language model's guide through latent space, 2024. URL https://arxiv.org/abs/2402.14433.

[48] Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

[49] Zijian Wang and Chang Xu. Functional abstraction of knowledge recall in large language models, 2025. URL https://arxiv.org/abs/2504.14496.

[50] Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=nRfClnMhVX.

[51] Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10210–10229, 2024.

[52] Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning? *arXiv preprint arXiv:2502.14010*, 2025.

[53] Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning?, 2025. URL https://arxiv.org/abs/2502.14010.

[54] Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Hf17y6u9BC.

[55] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models, 2023. URL https://arxiv.org/abs/2205.10625.

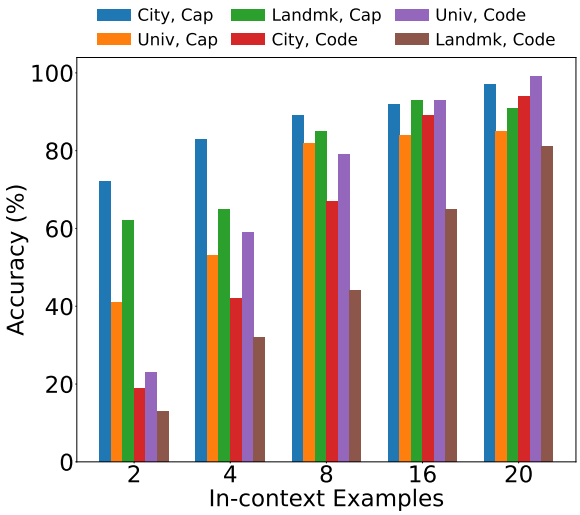


Figure 7: Accuracy of Gemma-2-27B on the two-hop "Source→Target" ICL problems.

Appendix A. Additional Details/Results for Section 2

A.1. Further details on problem setup

We collect the puzzles' data for over 40 countries in the world. For each source type in {City, University, Famous Landmark}, we collect at least 10 entities per country, leading to over 400 possible source entities per source type in the puzzles.

Our 2-hop ICL dataset is constructed by first prompting ChatGPT for the raw data, then clean and add to the data manually. The geography puzzles are cleaned to reduce leakage of source types, for instance, ChatGPT sometimes append city or state/province/region to a landmark, which we remove to ensure that the Landmark source type remains sufficiently distinct from the City source type. University names sometimes cannot avoid such overlap, e.g. University of California, Berkeley indeed has city name in it.

Furthermore, we show the accuracy of Gemma-2-27B on the problems with different number of in-context examples in Figure 7.

A.2. Causal mediation analysis

We primarily rely on causal mediation analysis (CMA), a.k.a. activation patching in the mechanistic interpretability literature, to obtain causal evidence for our claims in the LLM studies.

At a high level, CMA is about the study of indirect effects (IE) and direct effects (DE) in a system with causal relations [37]. Consider the following classical diagram of CMA, in Figure 8.

Suppose we wish to understand whether a certain mediator M plays an important role in the causal path from the input X to the outcome Y. We decompose the "total effect" of X on Y into the sum of *direct* and *indirect* effects (DEs and IEs), as shown in the figure. The indirect effect measures how important a role the mediator M plays in the causal path $X \to Y$. To measure it, we compute Y given X, except that we artificially hold M's output to its "corrupted" version, which is obtained by computing M on a counterfactual ("corrupted") version of the input. A significant change in Y

indicates a strong IE, which implies that M is important in the causal path. On the other hand, a weak IE implies a strong DE, meaning that the mediator does not play a strong causal role in the system (for the distribution of inputs of interest).

There are two common classes of interventions in mechanistic interpretability for localizing model components with strong IE in the causal graph. The first class is simple ablation, such as mean ablation (replace activation of the mediator by its average output on a distribution of interest) [48] or "noising" [30]. While this type of intervention is easy to perform, it typically leads to poor localization, surfacing low-level processing components irrelevant to the study [54].

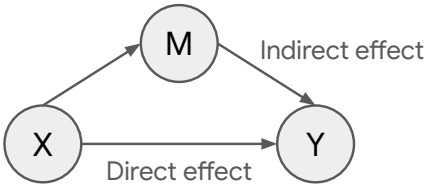


Figure 8: Basic illustration of CMA. X = input (exposure), M = mediator, Y = output (outcome).

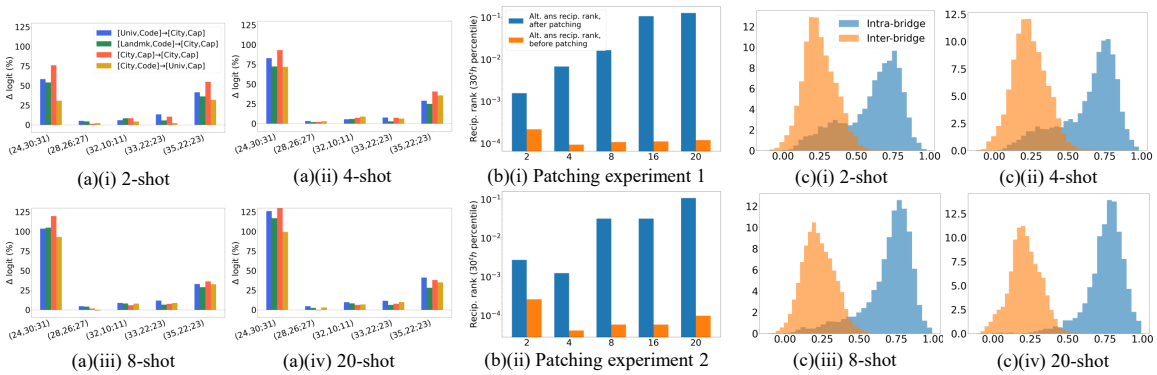
The other class, which we employ, is "interchange" intervention: it requires construction of alterative prompts which differ from the normal prompt in subtle ways, requiring careful consideration of the problem's nature, but allows "causal surgery" which surfaces model components with specific functional roles. Technically speaking, we are measuring the *natural indirect effects* of the mediator. In particular, it works as follows. We first run the system (the LLM) on both normal and alternative (or sometimes called counterfactual) inputs, and cache the output of the mediator M. We then hold M's output to its alternative version, as we run the full system (the LLM) on the normal prompt. Everything downstream in the causal graph from M are also influenced, up to the output Y. This helps us measure how the mediator M causally implicate the answer. Or more intuitively, it measures how "flipping" the output of M causally influences the LLM's "belief" in the alternative answer over the normal answer.

What makes our intervention experiments somewhat novel lies in exactly how we measure the IE. In particular, as we briefly discussed Section 2.1 and 2.2, we do *not* directly use the alternative prompt's ground truth answer to measure how well we are "bending" the model's "belief" through intervention. We discuss our method in greater detail here.

First, to understand whether certain attention heads have functional roles in processing the query source entity S_n which transcend source-target types of the two-hop problems, we work with normal-alternative prompts with distinct source and target types, such as sampling an *alternative* prompt "EPFL, 41. ... University of Tokyo, " ([University, Code] problem), and a *normal* prompt "Okinawa, Tokyo. ... Chicago, " ([City, Capital] problem). We hypothesize that there are certain model components which output the bridge concept, which is then composed with the target/output concept of the problem. For the normal example, this means "Chicago"→"USA" is resolved first, then the model executes $\text{Capital}(\text{USA}) = \text{Washington D.C.}$ as the output. This means that, patching a model component's activation from the alternative prompt onto its activation on a normal prompt, would cause the model to favor the answer of the alternative prompt, but with the same target semantic type as the normal prompt. In our running example, this would be "Tokyo", the capital of "Japan", the country (bridge) of the university "University of Tokyo".

It follows that, to evaluate the "causal effects" of such a bridge-resolving component, we should set $\hat{T}_n^{(\text{alt})} = \text{Type}_{\text{norm}}(T_n^{(\text{alt})})$. We then measure the (expected) intervened logit difference

$$\Delta_{\text{alt}\to\text{norm}} = \mathbb{E}\left[\text{logit}^{\text{alt}\to\text{norm}}(\boldsymbol{p}_{\text{norm}})[T_n^{(\text{norm})}] - \text{logit}^{\text{alt}\to\text{norm}}(\boldsymbol{p}_{\text{norm}})[\hat{T}_n^{(\text{alt})}]\right], \tag{1}$$



(a)(i) 2-shot (a)(ii) 4-shot (b)(i) Patching experiment 1 (c)(i) 2-shot (c)(ii) 4-shot

(a)(iii) 8-shot (a)(iv) 20-shot (b)(ii) Patching experiment 2 (c)(iii) 8-shot (c)(iv) 20-shot

Figure 9: This figure illustrates how the transferability and disentanglement of the bridge representation increases as we increase the number of in-context examples. Figure series (a) and (b) present the transferability result, obtained by performing cross-problem-type patching, and measuring the causal influence of the patched representation. In (a)(i) to (iv), we plot the percentage logit variation of the attention heads found to output "bridge" values, measured on several intervention experiments. For (b)(i) and (ii), we zoom in on head group (24,30;31), and show its causal effects on two patching experiments. The x-axis is the number of in-context examples, and the y-axis is the 30th percentile of the reciprocal rank of the alternative prompt's answer. For (c)(i) to (iv), we plot the disentanglement strength of the representations of head group (24,30;31).

where $\boldsymbol{p}_{\text{norm}} = [S_1^{(\text{norm})}, T_1^{(\text{norm})} ... S_n^{(\text{norm})},]$ is the normal prompt, $\text{logit}^{\text{alt}\to\text{norm}}(\boldsymbol{p}_{\text{norm}})$ indicates the logits of the model obtained after intervention while running the model on the normal prompt, and $\text{logit}(\boldsymbol{p}_{\text{norm}})$ indicates the logits of the model running naturally (un-intervened) on the normal prompt. Moreover, when we measure the rank of the model's answer when intervened, we also use $\hat{T}_n^{(\text{alt})}$ as the target.

Remark. To normalize our logit-difference variations, we compute

$$\bar{\Delta} = \frac{\Delta_{\text{norm}} - \Delta_{\text{alt}\to\text{norm}}}{\Delta_{\text{norm}}}, \tag{2}$$

where

$$\Delta_{\text{norm}} = \mathbb{E}\left[\text{logit}(\boldsymbol{p}_{\text{norm}})[T_n^{(\text{norm})}] - \text{logit}(\boldsymbol{p}_{\text{norm}})[\hat{T}_n^{(\text{alt})}]\right]. \tag{3}$$

A.3. Multi-hop circuit formation and the number of in-context examples

This sub-section focuses on illustrating the relation between the number of in-context examples versus (1) how strong a role the multi-hop mechanism plays in the LLM's inference (via causal interventions), (2) disentanglement strength of key bridge-resolving attention heads. As we will show below, there is a general positive correlation between the number of shots and the two factors.

More demonstrations $\Longrightarrow$ stronger causal score. Figure 9 visualizes the experimental results. From Figure 9(a) and (b) and sub-figures, we observe a correlation between the number of shots (ICL examples) and the bridge-resolving heads' "causal importance" in the model's inference. When the number of shots is low, we find that they tend to exhibit weak causal influence on the model's inference. For instance, as (b)(i) shows, at 2 shots, the 30th percentile of the alternative answer's rank *after* patching at (24,30;31) is on the order of 10^3. This is in stark contrast to how strong this head group's causal influence is at 20 shots as we saw before.

More demonstrations $\Longrightarrow$ stronger disentanglement, with a catch. In Figure 9(c)(i) to (iv), we observe that the intra-bridge cosine similarity tends to cluster better as the number of shots increase, while the inter-bridge cosine similarities decay toward 0.2, with the two distributions overlapping less and less. Interestingly, the bridge-disentanglement strength is still non-trivial with very few shots, mirroring the causal-intervention results: regardless of how disentangled the representations are in the very-few-shot regime, the LLM does not "realize" how it should utilize the multi-hop sub-circuit.

A.4. Further mechanistic analysis and causal evidence

Causal evidence. Recall that in the main text, to provide causal evidence for the bridge-resolving mechanism, we primarily presented causal intervention experiments where we treated [City, Capital] as the problem type we intervene on, using cross-type prompts [University, Calling Code], [Landmark, Calling Code] to show causal evidence for the bridge-resolving heads. Here, we add further evidence by having other source-target types. The results are presented in Figures 10 to 19, indexed as follows:

1. Experiment [City, Capital]→[Landmark, Calling Code]: Figure 10
2. Experiment [University, Capital]→[Landmark, Calling Code]: Figure 11
3. Experiment [City, Capital]→[University, Calling Code]: Figure 12
4. Experiment [Landmark, Capital]→[University, Calling Code]: Figure 13
5. Experiment [University, Capital]→[City, Calling Code]: Figure 14
6. Experiment [Landmark, Capital]→[City, Calling Code]: Figure 15
7. Experiment [City, Calling Code]→[University, Capital]: Figure 16
8. Experiment [Landmark, Calling Code]→[University, Capital]: Figure 17
9. Experiment [City, Calling Code]→[Landmark, Capital]: Figure 18
10. Experiment [University, Calling Code]→[Landmark, Capital]: Figure 19

Every patching experiment is performed on at least 100 prompts. As we can see, the general trend is that there is strong transferability of the bridge representation across the problem types, including when the source and target types have no overlap, giving us causal evidence that head groups (24,30;31), (35,22;23) are "resolving the bridge".

Scaling constant for bridge intervention. We found that for some of the transfer experiments, multiplying the patched representation for the heads (24,30;31), (35,22;23) improves the result, i.e. there is a greater percentage of samples where the alternative answer is boosted into the top-10 (or even top-1) answers of the model after intervention. Therefore, we also report those results. An intriguing property of this scaling constant is that it typically works best around 2.0. At 4.0, we often observe *saturation* or even *decline* in the intervened alternative answer's rank, such as in the [University, Capital]→[City, Calling Code] experiment shown in Figure 14.

The output-concept heads. While the main interest of this work lies in the bridge-resolving mechanism enabled by the sparse set of attention heads discussed above, we also present more

analysis of the output-concept heads, whose embedding tends to cluster with respect to the output concept (Capital versus Calling Code). To localize these heads, we generate normal-alternative prompt pairs where we only change the target/output type of the normal prompt to generate the alternative prompt, but keep the S_i's to be identical across the prompt pairs for all $i \leq n$. This helps us surface components which are independent of the query and bridge value, and sensitive to the output/target type for the ICL problem. The results are shown in Figure 20 and 21, where we run the intervention experiment [Landmark, Calling Code]→[Landmark, Capital] (due to limitations in time and computing resources, we could not sweep all the source-target combinations as of this version of the paper). As we can see, these head groups with the strongest causal scores indeed tend to exhibit sensitivity to output type, and insensitivity to source/input type and query and bridge value. Moreover, they are more concentrated in the deeper layers of the model.

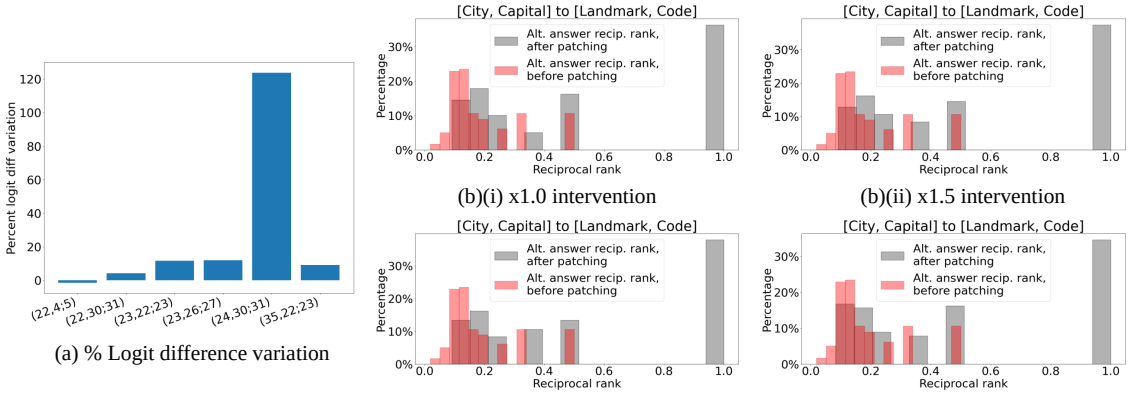
Statistics of alternative-type answers. A natural question to challenge the bridge-resolving mechanism is as follows. Say we are performing intervention by sampling alternative prompts from the problem type [University, Calling Code], and normal prompts from the type [City, Capital]: even though the target types have no overlap in text, perhaps the model still assigns nontrivial confidence to the "Capital" version of the alternative answer (which has target type "Calling Code")? If that is so, then it challenges our hypothesis about the role of the "bridge" representations, since we might just be directly injecting the right version of the alternative answer into the model.

We show evidence to refute this. In particular, in Figure 22, we show that the model places trivial confidence on the altered-type answer, even if they share the same bridge value. Therefore, we add further evidence to the bridge-resolving mechanism.

The role of MLPs. We performed similar intervention experiments on the MLPs at the last token position just like with the attention heads. They are observed to be much less interesting in their functional roles: we find that the MLPs appear to primarily process the output concept type, and do not participate heavily in outputting the bridge concept representation. This is revealed in Figure 23.

Smaller LLM exhibits weaker disentanglement. To contrast against our results for the 27B model, we study a small model in the same family of Gemma 2 models, *Gemma-2-2B*. This smaller model has significantly lower accuracies on the problems, measured at 20 shots. [City, Capital]: 71.67%, [University, Capital]: 25.83%, [Landmark, Capital]: 66.67%, [City, Calling Code]: 47.5%, [University, Calling Code]: 57.5%, [Landmark, Calling Code]: 23.33%.

We perform an intervention experiment [University, Code]→[City, Capital] on *Gemma-2-2B*, similar to the bridge-resolving head localization experiments we did on the 27B model. Intriguingly, we were also able to surface a highly sparse set of attention heads which have nontrivial causal scores (but much lower than that achieved by the 27B model). We find that these heads exhibit noticeable, but *noisy* disentanglement with respect to the bridge representation. We show these results in Figure 24. The weaker causal score of the bridge-resolving heads and their noisier concept disentanglement in the 2B model suggests the conjecture that, the larger the model, the more specialized its concept-processing components are — assuming that the model is well-trained. Such specialization likely benefits the model's generalization accuracy.



(a) % Logit difference variation

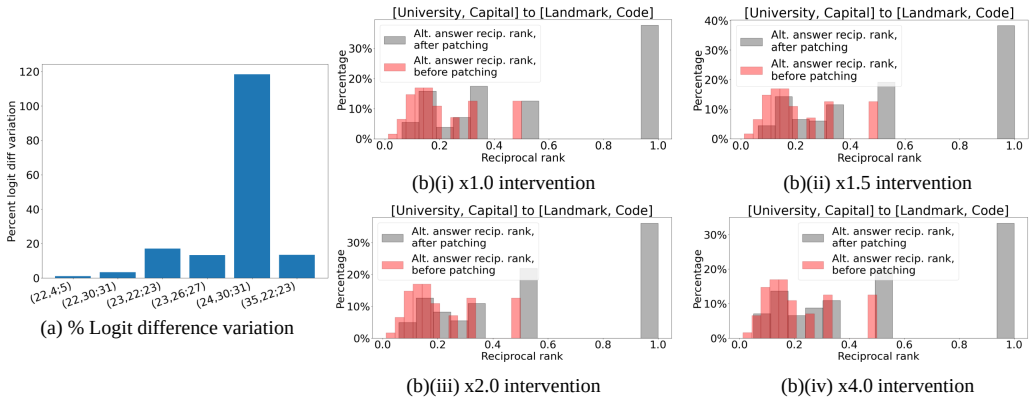
(b)(i) x1.0 intervention

(b)(ii) x1.5 intervention

(b)(iii) x2.0 intervention

(b)(iv) x4.0 intervention

Figure 10: Gemma-2-27B [City, Capital]→[Landmark, Code] transfer experiments, intervening head groups $(24, 30; 31), (35, 22; 23)$. We show the percentage logit-difference variation in (a), and reciprocal rank of the answer answer before and after intervention in the (b) series of figures, with different scaling constants in $\{1.0, 1.5, 2.0, 4.0\}$. We observe strong causal effects of the two attention head groups. Here, the scaling constant does not significantly affect the intervention performance.



(a) % Logit difference variation

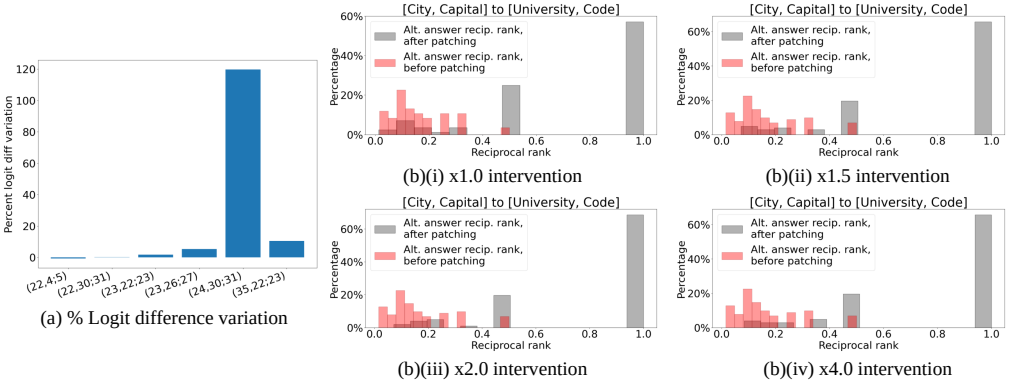
(b)(i) x1.0 intervention

(b)(ii) x1.5 intervention

(b)(iii) x2.0 intervention

(b)(iv) x4.0 intervention

Figure 11: Gemma-2-27B [University, Capital]→[Landmark, Code] transfer experiments, intervening head groups $(24, 30; 31), (35, 22; 23)$. We show the percentage logit-difference variation in (a), and reciprocal rank of the answer answer before and after intervention in the (b) series of figures, with different scaling constants in $\{1.0, 1.5, 2.0, 4.0\}$. We observe strong causal effects of the two head groups. Interesting, past a scaling constant of 1.5, we observe decline in intervention performance.



(a) % Logit difference variation

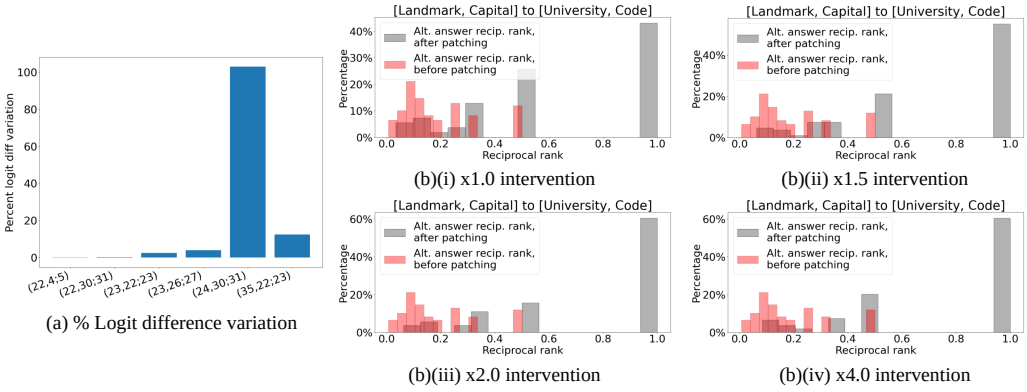
(b)(i) x1.0 intervention

(b)(ii) x1.5 intervention

(b)(iii) x2.0 intervention

(b)(iv) x4.0 intervention

Figure 12: Gemma-2-27B [City, Capital]→[University, Code] transfer experiments, intervening head groups $(24, 30; 31), (35, 22; 23)$. We show the percentage logit-difference variation in (a), and reciprocal rank of the answer answer before and after intervention in the (b) series of figures, with different scaling constants in $\{1.0, 1.5, 2.0, 4.0\}$. We observe strong causal effects of the two attention head groups, boosting the alternative answer into top 1 around 60% of the time! Additionally, the scaling constant does not significantly affect the intervention performance in this experiment.



(a) % Logit difference variation

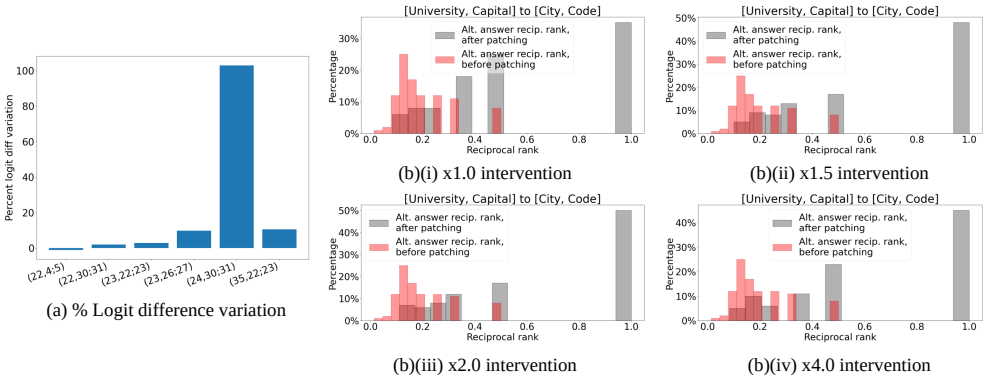
(b)(i) x1.0 intervention

(b)(ii) x1.5 intervention

(b)(iii) x2.0 intervention

(b)(iv) x4.0 intervention

Figure 13: Gemma-2-27B [Landmark, Capital]→[University, Code] transfer experiments, intervening head groups $(24, 30; 31), (35, 22; 23)$. We show the percentage logit-difference variation in (a), and reciprocal rank of the answer answer before and after intervention in the (b) series of figures, with different scaling constants in $\{1.0, 1.5, 2.0, 4.0\}$. We observe strong causal effects of the two attention head groups. Here, the positive effects of the scaling constant saturates around 2.0.



(a) % Logit difference variation

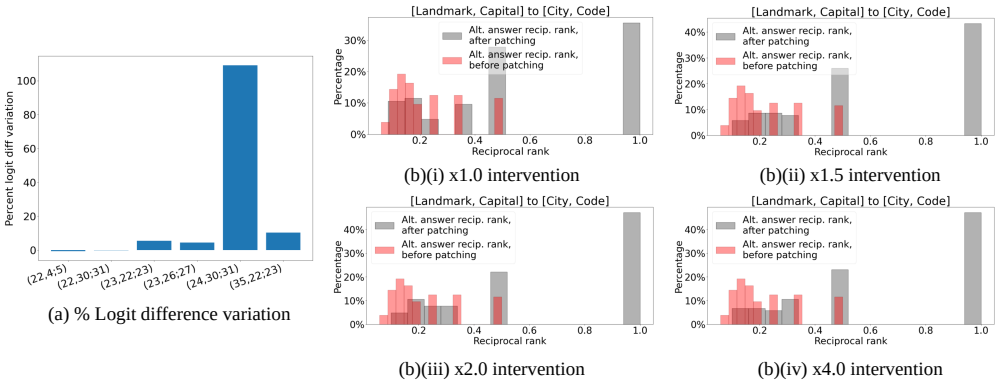
(b)(i) x1.0 intervention

(b)(ii) x1.5 intervention

(b)(iii) x2.0 intervention

(b)(iv) x4.0 intervention

Figure 14: Gemma-2-27B [University, Capital]→[City, Code] transfer experiments, intervening head groups $(24, 30; 31), (35, 22; 23)$. We show the percentage logit-difference variation in (a), and reciprocal rank of the answer answer before and after intervention in the (b) series of figures, with different scaling constants in $\{1.0, 1.5, 2.0, 4.0\}$. Interestingly, we observe decline in the intervention's accuracy as we push the scaling constant from 2.0 to 4.0 (top-1 accuracy decreases from around 50% to slightly above 40%), indicating a subtle regime in which the scaling constant boosts intervention performance.



(a) % Logit difference variation

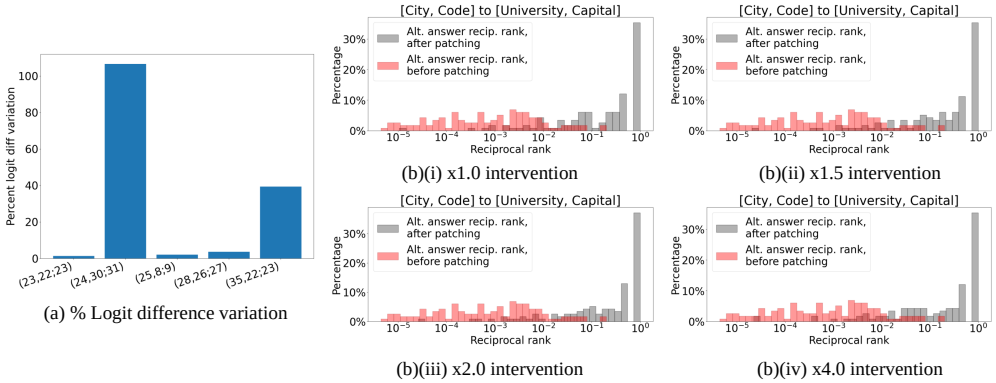
(b)(i) x1.0 intervention

(b)(ii) x1.5 intervention

(b)(iii) x2.0 intervention

(b)(iv) x4.0 intervention

Figure 15: Gemma-2-27B [Landmark, Capital]→[City, Code] transfer experiments, intervening head groups $(24, 30; 31), (35, 22; 23)$. We show the percentage logit-difference variation in (a), and reciprocal rank of the answer answer before and after intervention in the (b) series of figures, with different scaling constants in $\{1.0, 1.5, 2.0, 4.0\}$. We observe strong causal effects of the two attention head groups.



(a) % Logit difference variation

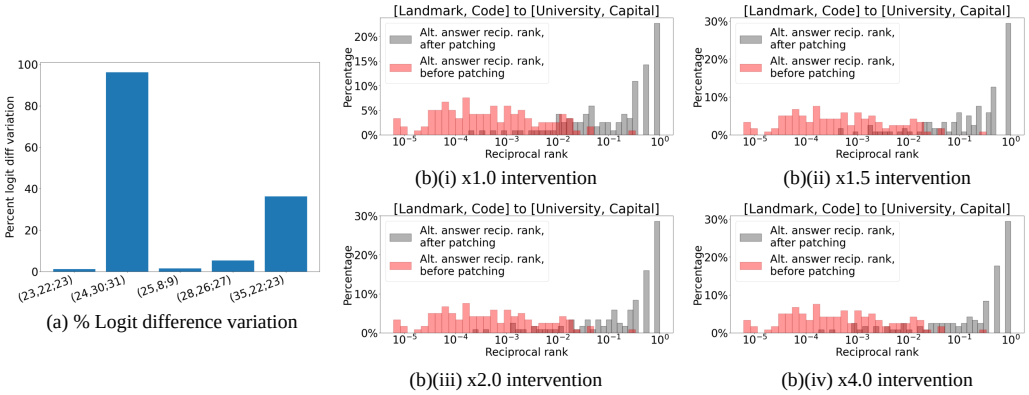
(b)(i) x1.0 intervention

(b)(ii) x1.5 intervention

(b)(iii) x2.0 intervention

(b)(iv) x4.0 intervention

Figure 16: Gemma-2-27B [City, Calling Code]→[University, Capital] transfer experiments, intervening head groups $(24, 30; 31), (35, 22; 23)$. At scaling constant 1.0 (i.e. natural intervention, no additional scaling), the reciprocal rank of 0.1 for the alternative answer after intervention is at the 36th percentile, while before patching, as we can see, the reciprocal rank of the alternative answer is mainly in the range of 10^{-2} to 10^{-5}.



(a) % Logit difference variation

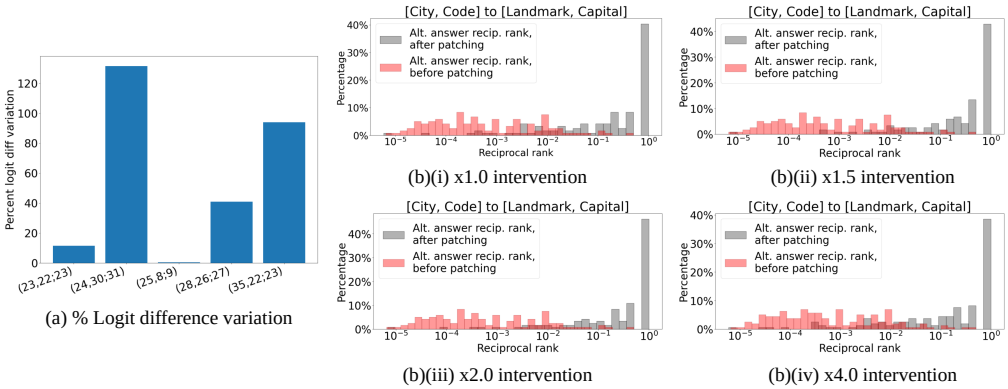
(b)(i) x1.0 intervention

(b)(ii) x1.5 intervention

(b)(iii) x2.0 intervention

(b)(iv) x4.0 intervention

Figure 17: Gemma-2-27B [Landmark, Calling Code]→[University, Capital] transfer experiments, intervening head groups $(24, 30; 31), (35, 22; 23)$. At scaling constant 1.0 (i.e. natural intervention, no additional scaling), the reciprocal rank of 0.1 for the alternative answer after intervention is at the 41th percentile.



(a) % Logit difference variation

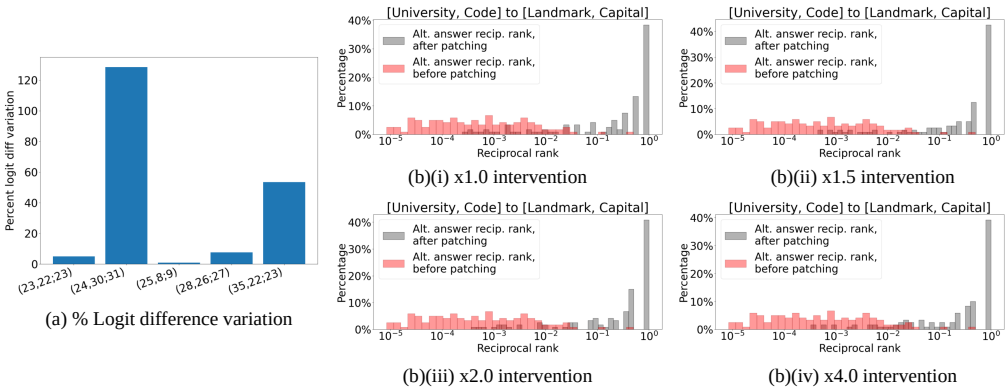
(b)(i) x1.0 intervention

(b)(ii) x1.5 intervention

(b)(iii) x2.0 intervention

(b)(iv) x4.0 intervention

Figure 18: Gemma-2-27B [City, Calling Code]→[Landmark, Capital] transfer experiments, intervening the single head group $(24, 30; 31)$. At scaling constant 1.0 (i.e. natural intervention, no additional scaling), the reciprocal rank of 0.1 for the alternative answer after intervention is at the 34th percentile.



(a) % Logit difference variation

(b)(i) x1.0 intervention

(b)(ii) x1.5 intervention

(b)(iii) x2.0 intervention

(b)(iv) x4.0 intervention

Figure 19: Gemma-2-27B [University, Calling Code]→[Landmark, Capital] transfer experiments, intervening the single head group $(24, 30; 31)$. At scaling constant 1.0 (i.e. natural intervention, no additional scaling), the reciprocal rank of 0.1 for the alternative answer after intervention is at the 31st percentile.

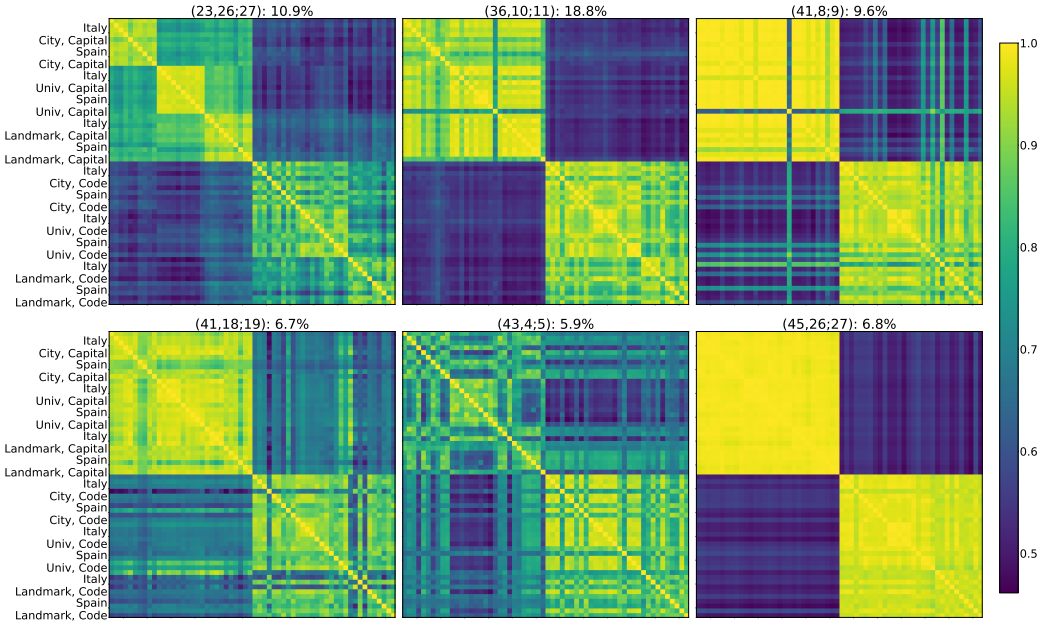


Figure 20: (Best viewed zoomed in) Cosine similarity map of the output-concept head groups (with top causal scores) identified in Gemma-2-27B, along with the percent logit-difference variation of the head groups, serving as the metric for the head groups' causal effects. Observe that they are mostly insensitive to the source type, query value, and bridge value, and primarily sensitive to the output/target type. Note: in this set of visualizations, we are using "Italy" and "Spain" as the bridge values.

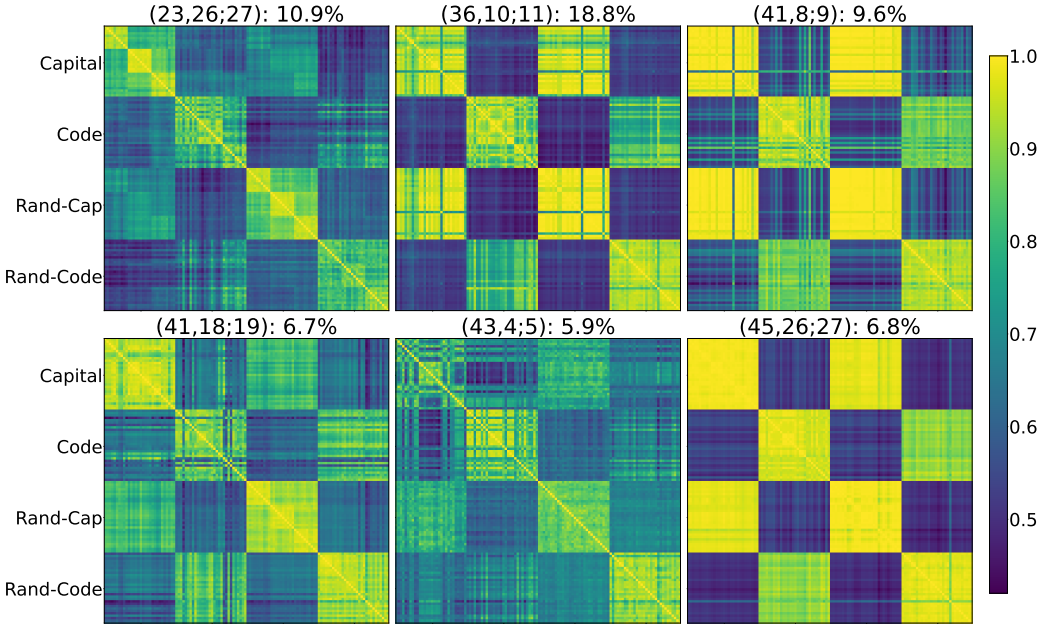


Figure 21: Cosine similarity map of the output-concept head groups identified in Gemma-2-27B. Here, we construct four groups of prompts. The first two groups consist of multi-hop ICL problems with Capital or Calling Code as the target type. The remaining two are created by randomly shuffling the output of the normal multi-hop ICL samples, causing the problem to essentially demand randomly outputting a Capital or a Calling Code; these are the "negative controls" we discussed in the main text. We find the output-concept heads' embeddings on the multi-hop prompts to align strongly with those on the output-concept-only prompts, further confirming their role in the circuit.

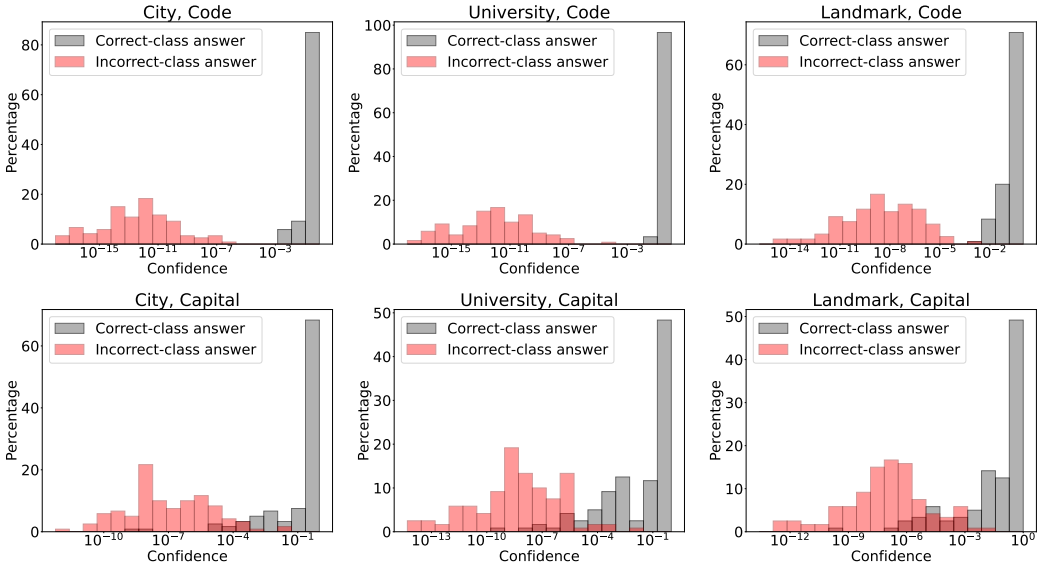


Figure 22: Distribution of Gemma-2-27B's confidence on the correct- and incorrect-type answer on the different problems. When we say "correct-class" answer, we simply mean that the answer's semantic type aligns with that of the problem's target, e.g. the correct-type answer for a prompt "Okinawa, Tokyo. Nantes, Paris. ... Shanghai, " would be "Beijing", while the incorrect-type answer would be "1" (the calling code of China, which the city Shanghai belongs to). We observe a clear separation in the LLM's confidence between the two types of answers.

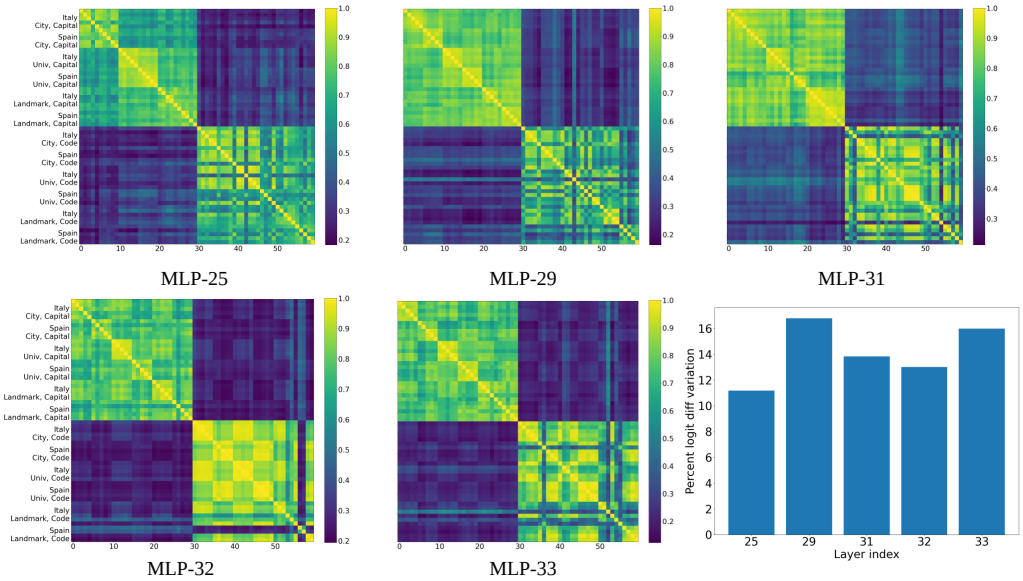


Figure 23: Functional roles of the MLPs in Gemma-2-27B. We perform patching experiments on [University, Code]→[City, Capital] at the last token position, similar to how we localize the bridge-resolving attention heads. We report the percentage logit-difference variation of the top-scoring MLPs, along with their cosine similarity maps computed on prompts sampled with different combinations of bridge values ("Italy" and "Spain") and diverse set of source-target types. Perhaps unsurprisingly, the MLPs at the last token position play a less interesting role: as seen in the cosine similarity maps for the MLPs with the highest causal scores (fairly low compared to the attention heads), they primarily discriminate against the output type. They do not appear to participate much in resolving the bridge concept.

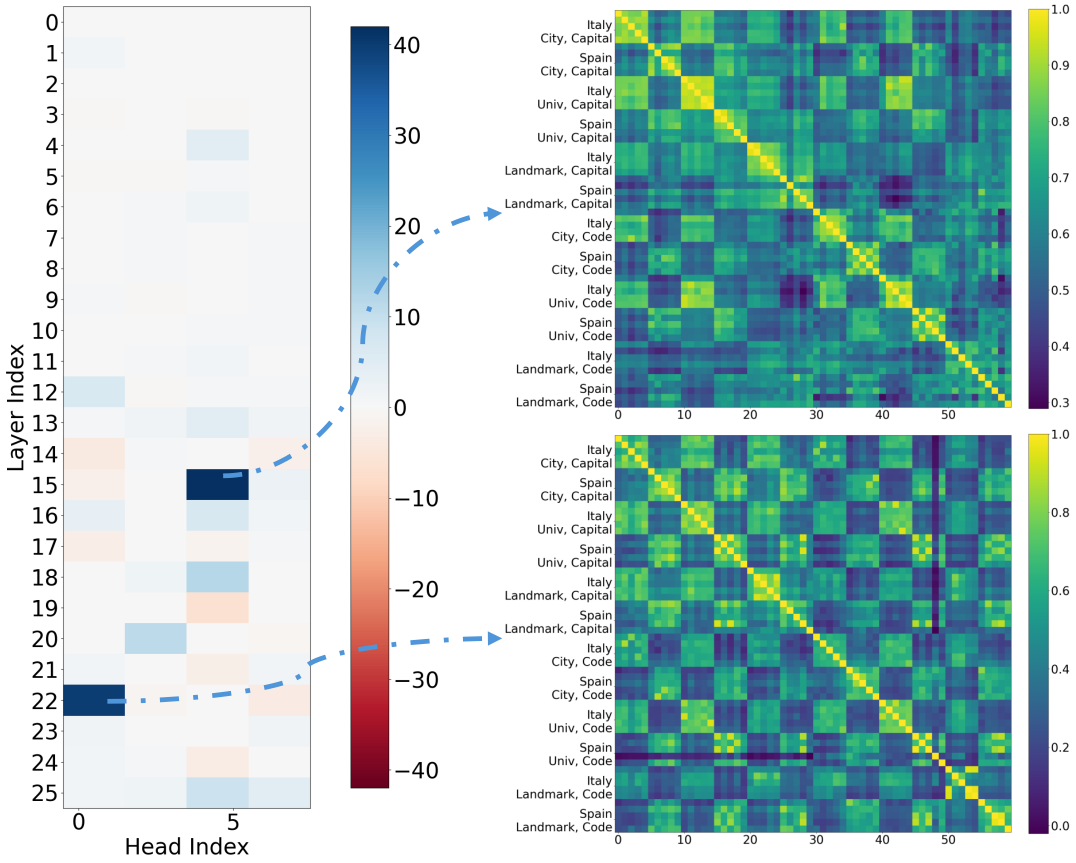


Figure 24: Results of the intervention experiment [University, Code]→[City, Capital], conducted on Gemma-2-2B, with 20-shot ICL. On the left, we show the percentage logit difference variation of the intervention experiment; on the right, we plot the cosine similarity map of the two head groups with the highest causal scores, namely $(15, 4; 5)$ and $(22, 0; 1)$. We find that while the two attention head groups exhibit nontrivial causal effects and disentanglement, they are, in comparison, much weaker than those exhibited by the 27B model. This likely explains the significantly lower accuracy of the 2B model than the 27B model. This also suggests a conjecture: perhaps the larger the model, the more specialized its concept-processing components are?

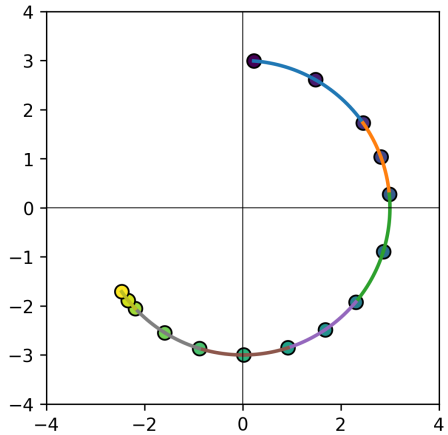


Figure 25: Illustration of an input sequence for the circle trajectory problem. Here, radius $r = 3$, period $p = 2$, sequence length $n = 13$. Every p consecutive steps on the trajectory are equal. We first sample $\lfloor \frac{n}{p} \rfloor + 1$ unique step-sizes in $[0, 1]$, and get the full sequence $\{a_1, a_2, a_3, a_4, \dots\}$, where same colors denote equal step-sizes. Then, we generate the trajectory by rotating point $\mathbf{x}_i$ clockwise by angle $a_i \cdot \frac{2\pi}{n}$ (see text for formal description).

Appendix B. Additional Details/Results for Section 3

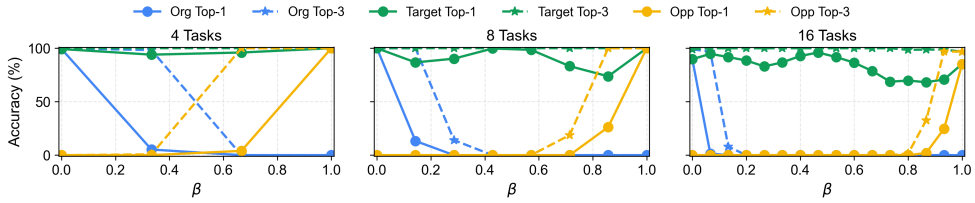


Figure 26: Steering with the task vectors for tasks k_1 and k_K for the *add-k* problem (see text for details). We plot the top-1 and top-3 accuracies for predicting the output based on the original offset k_1 (k_K), the 'opposite' offset k_K (k_1), or the target offset $(1-\beta)k_1 + \beta k_K$ ($(1-\beta)k_K + \beta k_1$), where $\beta \in [0, 1]$, The result shows that the model output can be steered toward the target.

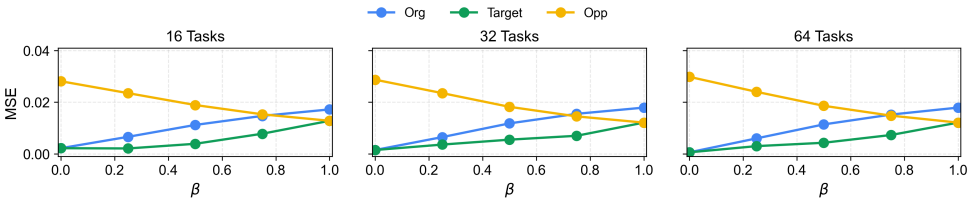


Figure 27: Steering with the task vectors for tasks r_1 and r_K for the Circular-Trajectory problem (see text for details). The MSE between the radius inferred from the model output and the original radius r_1 (r_K), the 'opposite' radius r_K (r_1), or the target radius $(1-\beta)r_1 + \beta r_K$ ($(1-\beta)r_K + \beta r_1$), where $\beta \in [0, 1]$, indicates that the model output can be steered toward the target.

Results for Steering with the Task Vectors. Let $\mathbf{t}_1$ and $\mathbf{t}_K$ denote the task vectors for tasks k_1 and k_K, respectively. Then, for task k_1 (k_K), we consider steering with $(1-\beta)\mathbf{t}_1 + \beta\mathbf{t}_K$ ($(1-\beta)\mathbf{t}_K + \beta\mathbf{t}_1$) and evaluate the accuracy for predicting the output based on the original offset k_1 (k_K), the 'opposite' offset k_K (k_1), or the target offset $(1-\beta)k_1 + \beta k_K$ ($(1-\beta)k_K + \beta k_1$), where $\beta \in [0, 1]$. Figure 26 presents the top-1 and top-3 accuracies for each case. High top-1 accuracies and $\approx 100\%$ top-3 accuracies for the target for all considered values of β indicate that the model output is steered toward the target. This shows that interpolating along the top principal direction is successful at interpolating values of k in the task space, showing that the model is somewhat strikingly successful at capturing the latent concept.

Figure 27 presents the results for steering the model output using the task vectors for radii r_1 and r_K. We follow the same procedure as in the add-k problem, with a different evaluation metric. We compute the norm of the generated output after steering as the model's radius (since the center of the circles is fixed at the origin), and consider the MSE between these radii and the original radius r_1 (r_K), the 'opposite' radius r_K (r_1), or the target radius $(1-\beta)r_1 + \beta r_K$ ($(1-\beta)r_K + \beta r_1$), where $\beta \in [0, 1]$, averaged over 200 sequences from each task. We observe that the MSE with the target radius is the lowest, which indicated the task vector can steer the model's output toward the target.

B.1. More on the Rectangular-Trajectory Problem

In this section we consider a Rectangular-Trajectory problem, parameterized by two parameters, namely the lengths of the two sides of the rectangle, say (a, b). Specifically, the trajectories contain points on axis-aligned rectangles centered at the origin. Let e denote the number of points on each edge of the rectangle spaced uniformly. The starting point of the sequence is randomly sampled from one of the e points on the right vertical edge of the rectangle. The rest of the points are obtained by traversing the rectangle CW or CCW, determined by $c = -1$ or 1. Figure 28 shows an example sequence.

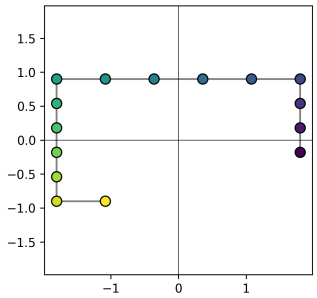


Figure 28: An illustration of a sequence of points for the Rectangular-Trajectory problem with $e = 5$ points per edge, and $n = 15$. See text for details.

Similar to Circular-Trajectory problem, each sequence is obtained by first sampling a and b uniformly between 1 and 4, then sampling the starting point and $c = -1$ or 1, and then following the aforementioned process. For our experiments, we set $e = 5$ and $n = 15$ for this task. The number of tasks K denotes the number of different combinations (a, b).

In Figure 29, we plot the 2D projection of the task vectors obtained for all (a, b) combinations lying on the 2D grid between $a \in [1, 4]$ and $b \in [1, 4]$. Similar to the experiments in Fig. 10, we plot the task vectors for trajectories with $c = -1$ here. We consider $K = 32$ in this experiment. The first two PCs explain 91.97% variance. We observe that all the task vectors lie on a 2D manifold.

This setting goes beyond the Circular-Trajectory problem and shows that transformers represent task vectors corresponding to the problem parameters (radius for circles and edge lengths for rectangles) in low-dimensional (smooth) manifolds in both cases.

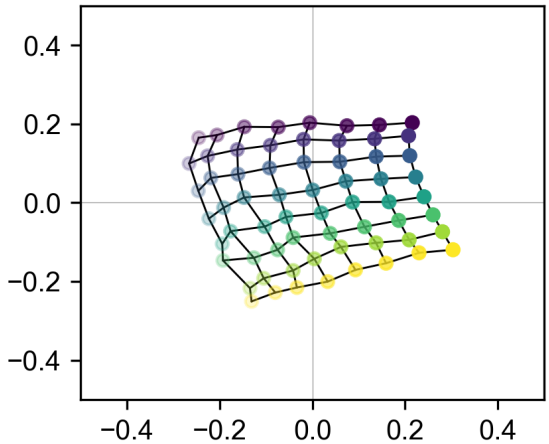


Figure 29: 2D projection of the task vectors obtained for 64 (a, b) combinations. Fixed color or transparency level corresponds to fixed a or b, respectively. The task vectors lie on a (smooth) 2D manifold.

Appendix C. Related Work

Task and Function Vectors. Task vectors **In-context Learning Interpretation.** ICL abilities of transformer-based models were first observed by Brown et al. [11], which sparked work in analyzing this ability. This includes analyzing how pretrained LLMs solve ICL tasks requiring abilities such as copying, single-step reasoning, basic linguistics [20, 32, 33, 43, 52, 55], and smaller models trained on synthetic tasks like regression [1, 6, 16, 18, 46], discrete tasks [9], and mixture of Markov chains [14, 34, 39]. These setups enable discovery of relations between in-context and in-weight learning [27, 40, 41], and internal algorithms that models implement [14, 33, 34, 52]. We contribute to this line of work, by shedding light on how transformers solve ICL problems which have more intricate latent structures.

Linear Representation Hypothesis (LRH). Our results are also connected to the LRH, which essentially speculates that LLMs represent high-level concepts in (almost) linear latent directions [8, 13, 23–25, 31, 35, 36]. Many papers motivated by the LRH then find "concept" vectors that can capture directions of truthfulness [4, 29], sentiment [42], humor [47], toxicity [44], etc. We deepen this study, asking how LLMs' representations capture/disentangle latent input concepts, and compose them during inference. In addition, the LRH is rooted in the field of mechanistic interpretability, which aims to reverse engineer mechanisms in transformer-based LMs [3, 5, 7, 10, 15, 19, 21, 33, 41, 45, 48–50].

Task and Function Vectors. A specific line of work in analyzing ICL mechanisms focus on task or function vectors. They essentially show that there exist certain causal patterns which capture the input-output relationship of the ICL task, on relatively simple problems such as "Country to Capital", "Antonyms", "Capitalize a Word" [12, 20, 43, 53]. Similarly, [2, 26, 28, 31] observed that LLMs tend to compress certain task or context information into sparse sets of vectors. We work with ICL problems with more complex latent structures, and our focus is not solely on (high-level) task vectors, but more on how the model disentangle and manipulate latent concepts useful to answering the query.

In addition, our work also complements the function vector analysis from contemporaneous work [22], providing add-k results for smaller models where we have full control over training and can hence conclude that the geometry of the task vector only arises from the latent task structure. We also compare results from add-k with other ICL tasks, giving additional insights.