Context Matters: Analyzing the Generalizability of Linear Probing and Steering Across Diverse Scenarios

Abstract

Previous works in mechanistic interpretability have attempted to represent model capabilities beyond looking at a single general direction in the subspace of model activations, however, many of these works neglect to consider how context impacts capability representation in the latent activation space. We hypothesize model behaviors like sycophancy or refusal are sets of related directions clustered together by the significant context they represent. To test this hypothesis, we generate a synthetic dataset for 5 different capabilities across 5 different, diverse contexts each. We use this dataset to train context-specific steering vectors and linear probes and measure their performance on contexts out of distribution from their training. We find that contextually trained steering vectors and linear probe are able to recover 95% and 85% accuracy on unseen contexts, suggesting that general capability representations independent of context can be learned and effectively applied in contextually-specific settings. Our work contributes to a deeper understanding of how capabilities are represented across many contexts in the model's latent activation space and bolsters confidence in applying steering and linear probing techniques in unseen settings that may be critical for safety.

1 Introduction

Foundational papers in mechanistic interpretability have attempted to control model behavior through *steering*, whereby one finds a vector representing a model capability from looking at the model's hidden state activations at a particular layer (Rimsky et al., 2024; Zou et al., 2025).

While steering is a powerful tool for controlling model behaviors, in practice, it can exhibit high variance in success across different prompts and also can severely impact model performance (Braun et al., 2025; Tan et al., 2025). Furthermore, behaviors may be mediated by several directions in the model's latent activation space, so steering along a singular direction may not be sufficient to fully control model capabilities (Wollschläger et al., 2025; Pan et al., 2025; Zhao et al., 2025).

We hypothesize that model capabilities such as sycophancy or refusal are represented in the model's latent activation space as sets of directions across different contexts. For example, when we consider the capability of hallucination, we hypothesize that there is a separate, precise steering vector for hallucination about medical facts as compared to hallucination about Python syntax, and that these vectors could be leveraged for more fine-grained steering control and possible applications for more robust linear probing. In this work, we quantify the importance of context in steering and model representations by training context-specific steering vectors and linear probes and evaluating their out of distribution performance. We also release our codebase and a diverse, balanced synthetic dataset for training steering vectors on a variety of contexts [1]. This dataset, containing 5 capabilities with 5 distinct contexts per capability, distinguishes the quality and degree of distinctness of each point through two recursive processes which target those two aforementioned metrics, further detailed in Section 3.1. This dataset in addition to the conclusions we arrive can be valuable to generating diverse, context-specific steering vectors and performing analyses upon this data in the context of mechanistic interpretability and model internals (e.g. the latent activation space).

[1] `https://anonymous.4open.science/r/multi-view-capabilities-5510/README.md`

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

In the larger context of AI Safety, our research has important implications for understanding how to effectively apply steering and linear probing techniques in a variety of settings. Understanding contextual importance in steering and probing could also ensure undesirable behaviors such as manipulation do not appear in unexpected ways.

Our main contributions can be summarized as follows:

1. We create and release a balanced, diverse synthetic dataset of 5 capabilities across 5 different contexts that can be used for a variety of downstream tasks including probing and steering.
2. To the best of our knowledge, we perform the first study of how well linear probes and steering vectors generalize across different contexts. We find that contextually-trained linear probes and steering vectors on average recover 85% and 95% accuracy respectively on unseen contexts.

2 Related Works

Linear probes are a mechanistic interpretability technique used for concept detection. They are trained by fitting a linear classifier on the activations of a hidden layer in a model to predict whether or not they contain a certain behavior. For example, we may train a linear probe on a model's latent activations to predict whether or not it contains a sycophantic direction.

Comparatively, steering is a causal technique of perturbing the model's activations by adding a vector representing a specific concept during inference time at a particular layer. Adding the vector with a positive multiplier should make the concept more present in the model's outputs and a negative multiplier should ablate the effect.

Linear probes for representation editing and generalization. Davis and Sukthankar (2024) demonstrate that linear probes can effectively extract and causally edit internal representations in a chess-playing GPT model, establishing a crucial link between probe weight vectors and valid model outputs. Their work provides evidence that language models maintain editable emergent representations across different layers, with linear classifiers being able to reliably manipulate specific aspects of a model's internal state. Davis and Sukthankar (2024) contribute to this understanding by examining probe performance across all model layers and investigating the linearity of latent feature representations. Their findings on the ability to edit model representations through linear probes align with the broader investigation of how capabilities are represented in the model's latent activation space, though their work focuses primarily on a single domain context rather than exploring how capability representations might cluster or vary across significantly different contexts.

Ichmoukhamedov and Martens (2025) investigate the generalization of truth direction linear probes across different conversational formats, finding that while probes generalize well between short conversations ending on lies, they struggle with longer formats where lies appear earlier. Their work demonstrates that the generalization of capability representations is highly dependent on contextual format, providing evidence that the structure of input significantly impacts how linear probes transfer across settings.

Linear representations in LLMs. The linear representation hypothesis Park et al. (2024), a paper foundational to the general understanding of latent activation spaces, suggests that high-level concepts are represented linearly in the model's activation space. From this hypothesis, several works have provided evidence in support of linear capability representations Zou et al. (2025); Marks and Tegmark (2024).

Burnell et al. (2023) examine the structure of language model capabilities through factor analysis across 29 different LLMs and 27 cognitive tasks, finding that capabilities are not monolithic but instead composed of three distinct factors: reasoning, comprehension, and core language modeling. This multifaceted view of capability structure supports the hypothesis that complex behaviors may emerge from sets of related directions rather than single unified representations.

Multi-directional behavior representations. Arditi et al. (2024) find a single direction that mediates refusal. However, more recent works have found evidence against this claim; Pan et al. (2025) find a dominant direction controlling refusal but show that there are other directions necessary to fully explain refusal behavior. Wollschläger et al. (2025) further suggest that refusal may be mediated by high dimensional cones.

Engels et al. (2025) demonstrate that some language model features are inherently multi-dimensional and cannot be reduced to one-dimensional representations. Using sparse autoencoders, they discover interpretable multi-dimensional features such as circular representations of days of the week and months of the year, showing through intervention experiments that these multi-dimensional structures serve as fundamental computational units. This work provides direct evidence that certain capabilities may require multi-directional representations rather than single directions.

Steering. Many works have attempted to leverage the linear representation hypothesis to find a single direction controlling a certain model behavior and add or ablate this vector in order to steer the model to act in desirable way (Rimsky et al., 2024; Zou et al., 2025). More recent approaches have tried to take context into account, either by applying steering in certain scenarios (Lee et al., 2025) or by modifying the steering vector by taking an adaptive linear combination of several steering vectors based on the prompt (Wang et al., 2025). However, a more in-depth study is required to determine how and to what extent context influences capability representation in the latent activation space.

von Rütte et al. (2024) extend concept guidance beyond truthfulness to explore a richer set of concepts including appropriateness, humor, creativity, and quality. They find that different concepts exhibit varying levels of "guidability" and that probes with optimal detection accuracy do not necessarily make for optimal steering vectors, revealing complex relationships between concept detectability and the ability to control model behavior through activation perturbation.

3 Methodology

3.1 Synthetic Data Generation

For a given capability C (e.g. sycophancy), below is our general process for generating the dataset $\mathcal{D}$ (loosely based on Wu et al. (2025) and Perez et al. (2023)):

In the realm of the capability of C, we ask Gemini 2.0 Flash (‘gemini-2.0-flash‘) to generate 20 possible contexts, choosing 5 from which data can be generated. Our prompt explicitly instructs the model to generate an even balance of benign and semi-harmful data. As a result of this diversity in later experiments of this work, we can ensure that experimentation targeting a specific capability (e.g. sycophancy) doesn’t conflate with another, more foundational capability such as refusal if the original capability context only involved harmful contexts.

For each capability of interest, we will create a synthetic dataset of questions involving several distinct contexts modeling that capability (e.g. biology and law in the capability of hallucination). Following the setup from Rimsky et al. (2024), each data point in this dataset will contain:

1. ‘user_prompt‘: A question aimed to elicit a capability through potential answers to this question.
2. ‘positive_response‘ A response to the ‘user_prompt‘ that aims to specifically express the target capability capability (e.g. agreeing with incorrect information in the context of hallucination).
3. ‘negative_response‘ A response to the ‘user_prompt‘ aims to specifically express the opposite of the target capability or simply the lack of the capability’s presence (e.g. disagreeing and correcting incorrect information provided by the answer choice, going against the expression of hallucination).
4. ‘context_label‘: A Gemini 2.0 Flash-generated label of 1 of the 5 contexts that the ‘user_prompt‘ relates to within the capability (e.g. biology in hallucination).

Initial qualitative observations of the data up until this point revealed that Gemini, although being queried in separate API calls per batch of 5 generated data points, seemed to be repeating the questions and content with very similar concepts, in some cases with data points generated during a particular API call being identical to a data point generated in another API call. Moreover, some of the data seemed to be lacking in quality; either the user_prompt, positive_responses, and/or negative_response occasionally did not aim to target a potentially significant enough elicitation of the capability. To ensure that our data was diverse enough to effectively train both linear probes and steering vectors, we implemented the following diversification and quality maximization process shown in 1

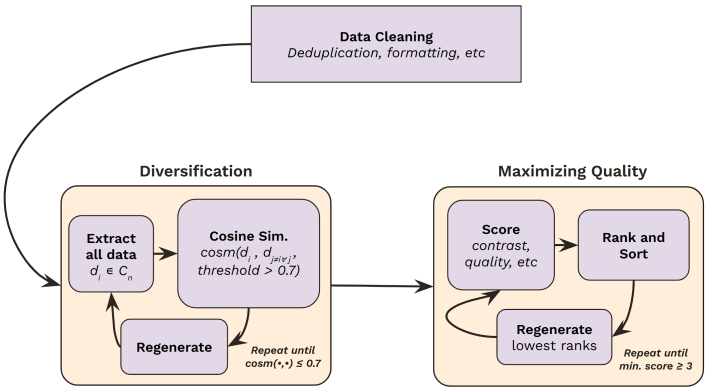

Figure 1: The pipeline for enhancing data diversity and quality. For a data point d_i in a given context c_n out of the 5 provided contexts, it recursively takes the cosine similarity of d_i against all other data points in the context ($d_{j \neq i \ \forall \ j}$) until the cosine similarity exceeds a threshold of 0.7. Each d_i of the diversified data is then ranked from 1-5 on the contrast of positive_response to negative_responses, quality, and other metrics to determine a holistic score, iterating through every d_i in the dataset. The dataset is then ranked greatest-to-least with the highest holistic scores at the top and the lowest at the bottom, where then the lowest quality d_i are regenerated. This process recurses until the minimum score for any d_i is ≥ 3.

3.2 Extracting contextual steering vectors

Let $\mathcal{D}^{(train)}$ and $\mathcal{D}^{(test)}$ be the train and test synthetic datasets respectively generated for capability C with n different contexts represented. For simplicity, we omit the C from notation, and unless stated otherwise, all samples relate to the same capability C. Then, we define the set of contexts as $X = \{x_1, x_2, \ldots, x_n\}$ and $x_i^{(train)} \subset \mathcal{D}^{(train)}$ and $x_i^{(test)} \subset \mathcal{D}^{(test)}$ as the subsets of training and testing datasets related to concept x_i. From the synthetic data generation process, for all i, $|x_i^{(train)}| = m > n$ (implying that $|\mathcal{D}^{(train)}| = nm$).

The corresponding context-specific steering vectors are defined to be in the set $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ such that each $\mathbf{v}_i$ is trained on $x_i^{(train)}$. We define $\mathbf{g}$ to be a general steering vector trained on a randomly sampled subset $\mathcal{R} \subset \mathcal{D}^{(train)}$ such that $|\mathcal{R}| = m$. Our experiments utilize $m = 200$ and $n = 5$.

To steer model behaviors, we implement Bi-directional Preference Optimization (BiPO) (Cao et al., 2024), a technique which trains a steering vector by minimizing the loss of producing desired positive and negative steered responses when the vector is applied.

3.3 Measuring steering generalization

To measure steering effectiveness, we employ a judge model, similar to the steering evaluation technique presented in Wu et al. (2025), to define a *steering score*. For each individual response, we ask the judge model to rate on a scale from 0 to 5 how well the concept is incorporated in the response based on the positive response for positive steering and negative response for negative steering.

On some dataset $\mathcal{D}$, the steering score is defined to be the average of the judge model ratings for each steered model completion.

For each $\mathbf{v}_i$, we compute its *generalization ratio* $G(\mathbf{v}_i)$. Let $\mathbf{v}_i$'s *in-distribution performance* $\mathcal{I}(\mathbf{v}_i)$ be its steering score measured on samples in $x_i^{(test)}$. Correspondingly, the *out-of-distribution performance* $\mathcal{O}(\mathbf{v}_i)$ is measured by calculating $\mathbf{v}_i$'s steering score on samples in $\mathcal{D}^{(test)} \setminus x_i^{(test)}$, i.e. the samples in the test dataset that do not relate to concept x_i.

Then, we define the generalization ratio $G(\mathbf{v}_i)$ as a function that takes in vector $\mathbf{v}_i$ with the following formula:

$$G(\mathbf{v}_i) = \frac{\mathcal{O}(\mathbf{v}_i)}{\mathcal{I}(\mathbf{v}_i)}.$$

If $G(\mathbf{v}_i) = 1$, then $\mathbf{v}_i$ generalizes perfectly well out of distribution (i.e. it performs the same out-of-distribution as in-distribution). When $G(\mathbf{v}_i) < 1$, $\mathbf{v}_i$ performs better on the context it was trained on than unseen contexts, and the ratio quantifies how much of its in-distribution performance $\mathbf{v_i}$ recovers out of distribution.

In addition to computing the generalization ratio of each concept vector, for each concept x_i we also compute the steering scores for the following vectors on the dataset $x_i^{(test)}$:

1. $\mathbf{v}_i$ for every $\mathbf{v}_i \in V$ (context-specific steering vector).
2. $\mathbf{g}$ (general steering vector).
3. $\vec{0}$ (no steering).

We additionally compute the steering score of the above vectors on the entire test dataset $\mathcal{D}^{(test)}$.

3.4 Linear probe generalization

In addition to looking at the generalizability of steering, we also examine the generalization ability of the model's latent representations of context-specific capabilities. We refer to this measurement as the *capability-detection performance*, intending to capture how well a vector representing the capability detects this capability on unseen prompts.

We measure capability-detection by evaluating the generalizability of linear probes. We use linear probes to determine if there are general shared linear structures between activations related context x_i and x_j for $i \neq j$. For each context $x_i \in X$, we train a linear probe p_i to distinguish the presence of capability C in context x_i. To understand the degree of similarities between activations for differing context, we test how well p_i generalizes to contexts in $X \setminus \{x_i\}$. We will quantify this by measuring the average positive prediction probability over each dataset.

We will also train a general linear probe for detecting the capability and measure its performance across contexts in X. We perform these experiments on two models, Qwen-2.5-7B-Instruct and Llama2-13B-Chat. We train probes ever 4 layers for Qwen-2.5-7B-Instruct and every 5 layers for Llama2-13B-Chat. For each probe, we select the probe trained on the layer it performs the best on its validation dataset for comparison.

4 Results & Discussion

We first discuss the results of linear probe generalization in section 4.1 to show how well latent representations generalize out of distribution. Then, we explore how well the causality generalizes by looking at steering efficacy in 4.2.

4.1 Linear Probe Results

Across all the capabilities we test, we find that **general linear probes perform on par with contextually-trained linear probes on context-specific data**, suggesting the existence of context-independent linear capability representations. In particular, across all capabilities, general linear probes recover greater than 90% of the accuracy of the contextually-trained linear probe, as shown in Figure 2.

Furthermore, we find that contextually trained linear probes tend to generalize quite well as shown in Figure 4, though the trend is more ambiguous and capability-dependent. The average recovered accuracy between cross contexts is 85%, suggesting that even contextually trained probes are able to apply general knowledge to a variety of contexts. Notably, the trend seems to be consistent between models about which capabilities have higher generalization ratios than others.

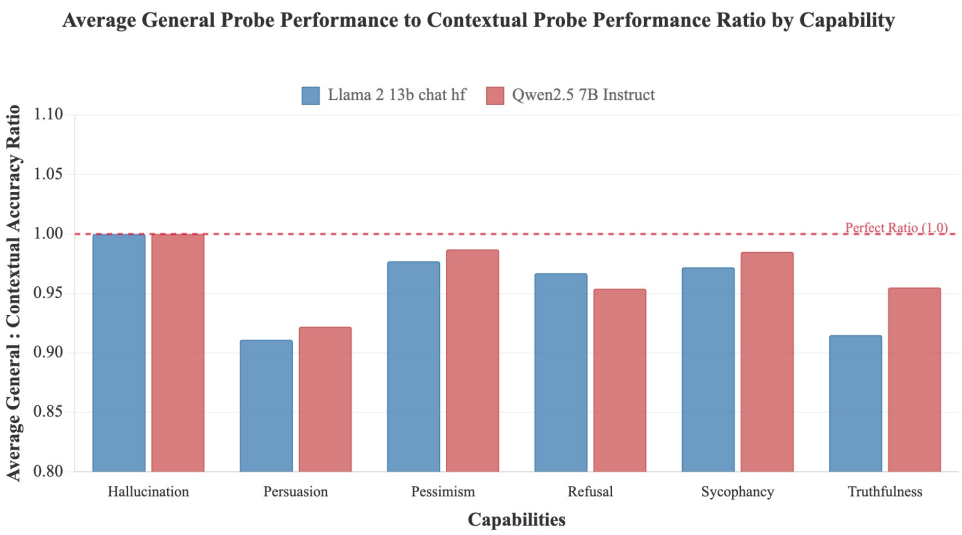

Figure 2: Ratio of the accuracy of general linear probes trained on a mix of contexts to the accuracy of in-context linear probes trained on a specific context for that context across capabilities.

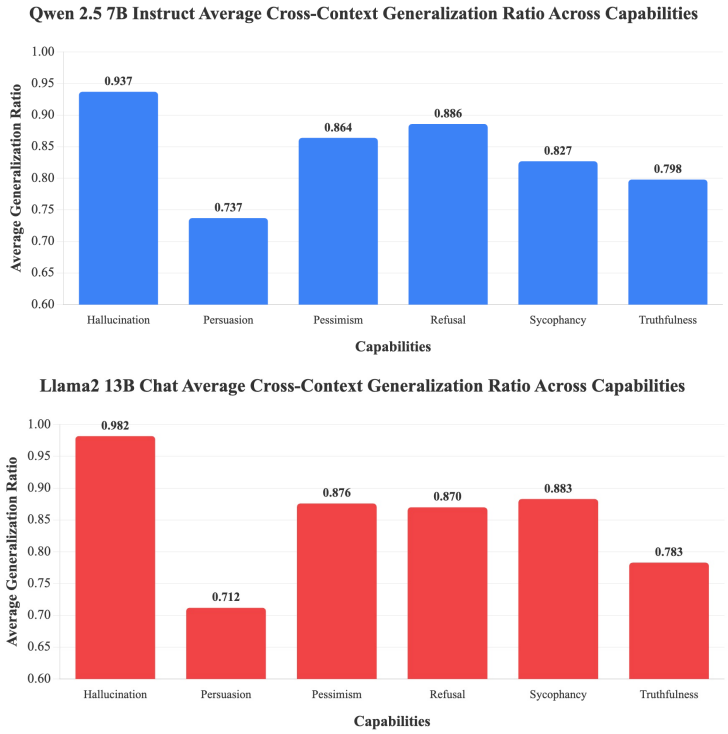

Figure 3: Average Generalization ratios (context vector accuracy out of context / in-context accuracy) across capabilities for both models.

4.2 Steering Efficacy Results

We perform steering on the Qwen model family of the 7B size. Through an exploration of the Bi-directional Preference Optimization (BiPO) steering method (Cao et al., 2024), we see that injecting the steering vector at layer 15 is the most effective. Further and lighter empirical testing of our own with other layer values.

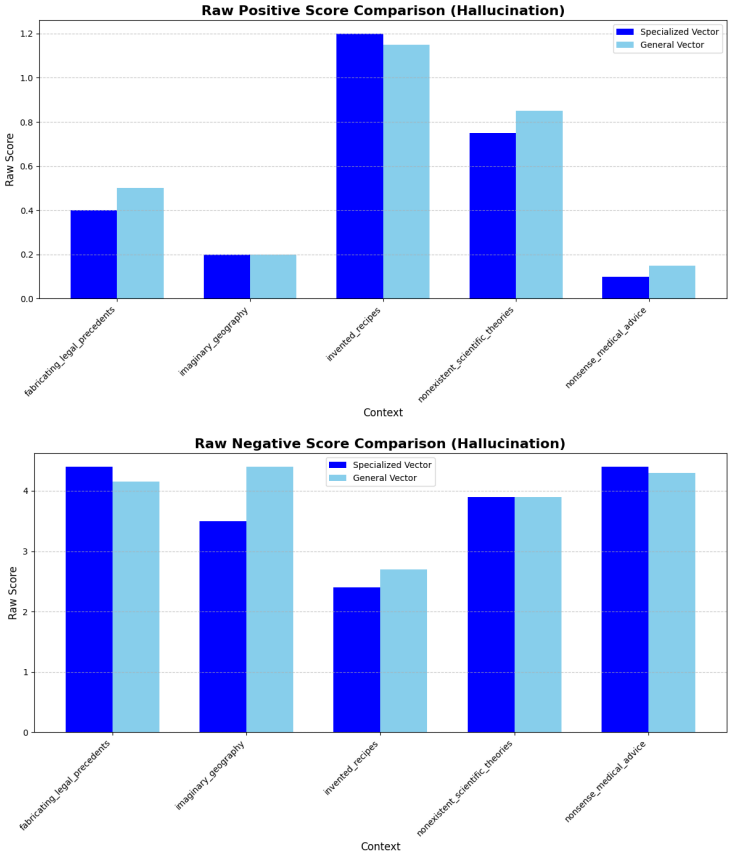

Figure 4: Calculating and comparing the scores of a general steering vector trained on all contexts c in the capability of hallucination against a context-specific steering vector trained only on a specific context c_i, both evaluated on the context of c_i. More figures testing other capabilities and contexts can be found in Appendix A

We find that general steering vectors tend to perform similarly to contextual steering vectors when evaluated using a judge model that scores responses on a scale from 0 to 5 based on how aligned they are with the positive/negative response in our dataset.

On average, steering vectors achieve a cross-context generalization ratio of 0.95, providing strong support that the general capability representation identified by linear probes can be applied effectively in causal settings as well. The breakdown of generalization across capabilities is shown in Figure 5.

5 Conclusion

In this work, we measure how well context-specific linear probes and steering vectors generalize on unseen contexts. We find that both linear probes and steering vectors tend to generalize quite well out of context. We find that on a 7B model, across 5 different capabilities, contextual linear probes are able to recover 85% accuracy on average when applied on unseen contexts for the same capability. On the same model, steering is able to recover 95% accuracy on unseen contexts. We also introduce a balanced synthetic dataset of 5 different capabilities with 5 different contexts each that can be used for future probing and steering applications.

Future work could be done to expand this analysis on more models and to understand the shared generalization variances between different capabilities. In particular, one could look at the underlying mechanisms of these behaviors and determine if there are any shared circuits between different contexts to show how causally processing differs between contexts.

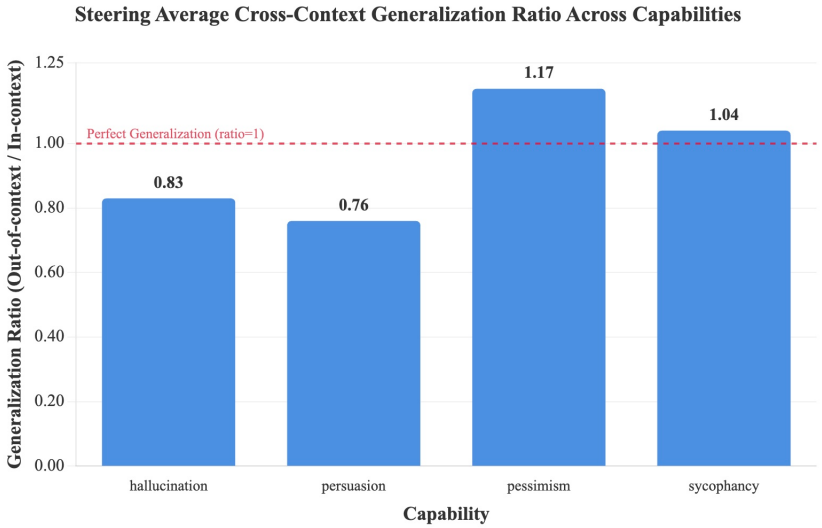

Figure 5: The average generalization ratio across all contexts for all capabilities. The generalization ratio is measured as the contextual steering vector efficacy on the out of context datasets and its own context-specific dataset.

Overall, our results provide support for the existence of general representations of capabilities that are context-invariant. As mechanistic interpretability techniques such as probing and steering are increasingly being used to detect and modify harmful model behaviors in diverse real-world contexts, ensuring their robustness will play a pivotal role in future safety work.

References

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.

Joschka Braun, Carsten Eickhoff, David Krueger, Seyed Ali Bahrainian, and Dmitrii Krasheninnikov. Understanding (un)reliability of steering vectors in language models, 2025. URL https://arxiv.org/abs/2505.22637.

Ryan Burnell, Han Hao, Andrew R. A. Conway, and Jose Hernandez Orallo. Revealing the structure of language model capabilities, 2023. URL https://arxiv.org/abs/2306.10062.

Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization, 2024. URL https://arxiv.org/abs/2406.00045.

Austin L Davis and Gita Sukthankar. Hidden pieces: An analysis of linear probes for gpt representation edits. In *2024 International Conference on Machine Learning and Applications (ICMLA)*, pages 498–505, 2024. doi: 10.1109/ICMLA61862.2024.00073.

Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear, 2025. URL https://arxiv.org/abs/2405.14860.

Timour Ichmoukhamedov and David Martens. Exploring the generalization of llm truth directions on conversational formats, 2025. URL https://arxiv.org/abs/2505.09807.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering, 2025. URL https://arxiv.org/abs/2409.05907.

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.

Wenbo Pan, Zhichao Liu, Qiguang Chen, Xiangyang Zhou, Haining Yu, and Xiaohua Jia. The hidden dimensions of llm alignment: A multi-dimensional analysis of orthogonal safety directions, 2025. URL https://arxiv.org/abs/2502.09674.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models, 2024. URL https://arxiv.org/abs/2311.03658.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. In *Findings of the association for computational linguistics: ACL 2023*, pages 13387–13434, 2023.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Daniel Tan, David Chanin, Aengus Lynch, Dimitrios Kanoulas, Brooks Paige, Adria Garriga-Alonso, and Robert Kirk. Analyzing the generalization and reliability of steering vectors, 2025. URL https://arxiv.org/abs/2407.12404.

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.

Tianlong Wang, Xianfeng Jiao, Yinghao Zhu, Zhongzhi Chen, Yifan He, Xu Chu, Junyi Gao, Yasha Wang, and Liantao Ma. Adaptive activation steering: A tuning-free llm truthfulness improvement method for diverse hallucinations categories. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 2562–2578. ACM, April 2025. doi: 10.1145/3696410.3714640. URL http://dx.doi.org/10.1145/3696410.3714640.

Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann, and Johannes Gasteiger. The geometry of refusal in large language models: Concept cones and representational independence, 2025. URL https://arxiv.org/abs/2502.17420.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Axbench: Steering llms? even simple baselines outperform sparse autoencoders, 2025. URL https://arxiv.org/abs/2501.17148.

Jiachen Zhao, Jing Huang, Zhengxuan Wu, David Bau, and Weiyan Shi. Llms encode harmfulness and refusal separately, 2025. URL https://arxiv.org/abs/2507.11878.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2025. URL https://arxiv.org/abs/2310.01405.

A Context-specific vs. General Steering Results

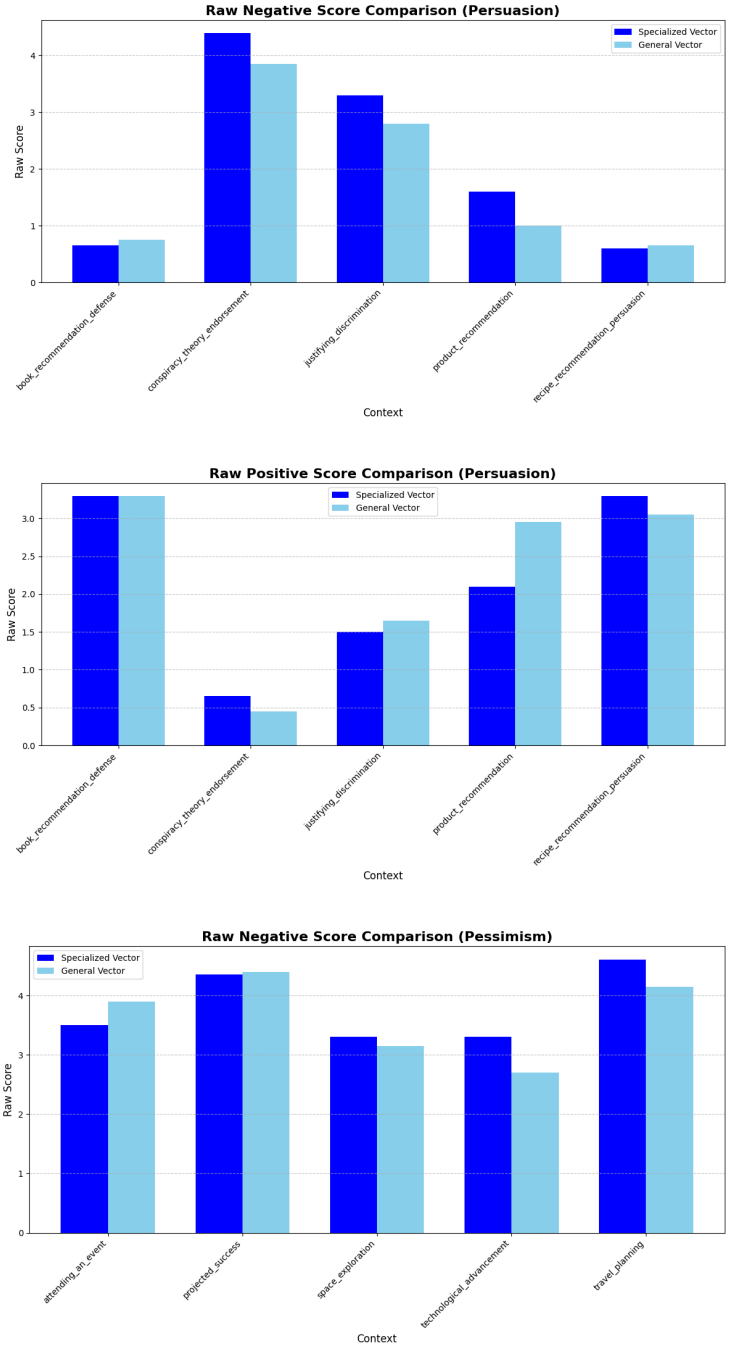

Figure 6: Calculating and comparing the scores of a general steering vector trained on all contexts c in a given capability C_i against a context-specific steering vector trained only on a specific context c_i, both evaluated on the context of c_i. More figures testing other capabilities and contexts can be found in Appendix

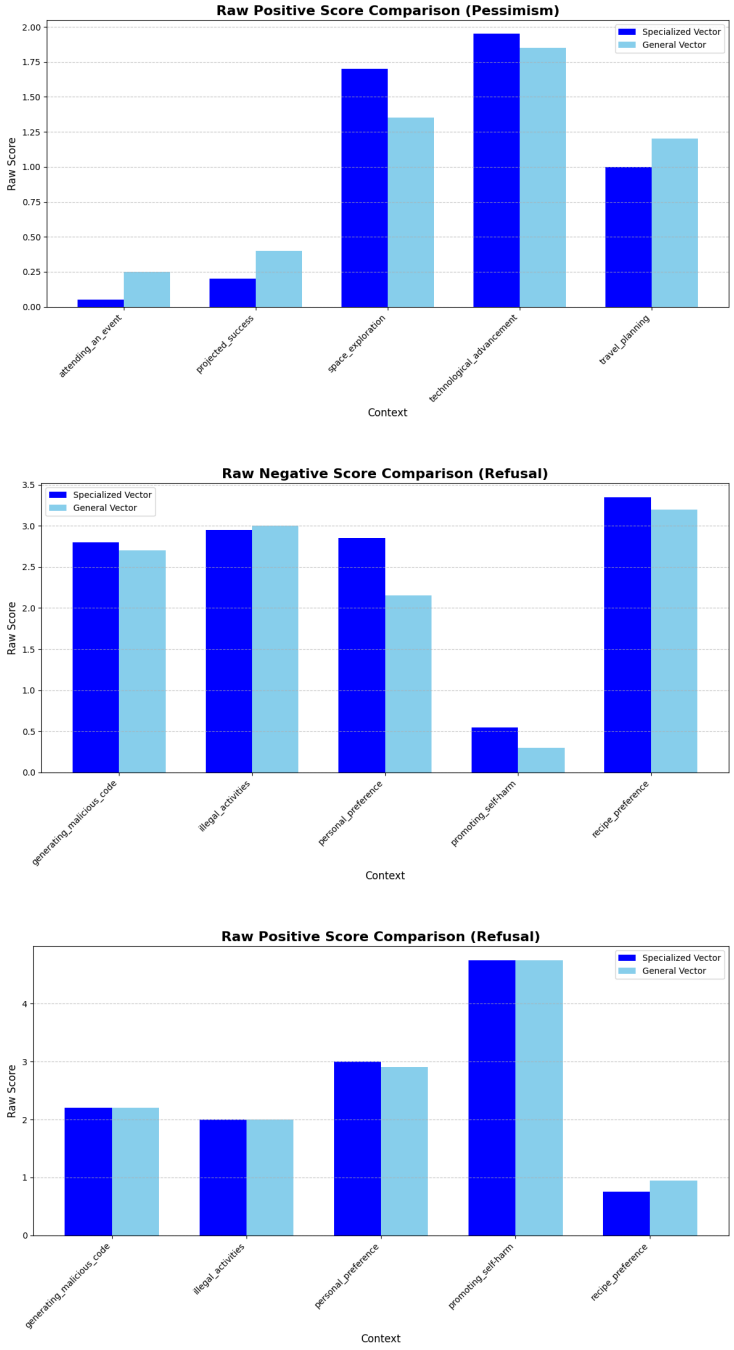

Figure 7: Calculating and comparing the scores of a general steering vector trained on all contexts c in a given capability C_i against a context-specific steering vector trained only on a specific context c_i, both evaluated on the context of c_i. More figures testing other capabilities and contexts can be found in Appendix

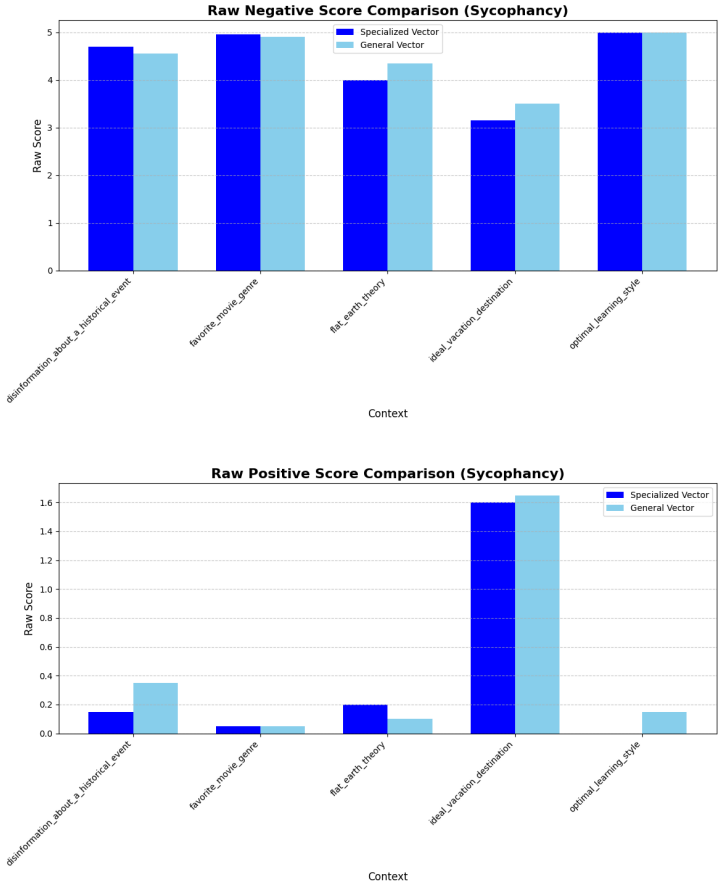

Figure 8: Calculating and comparing the scores of a general steering vector trained on all contexts c in a given capability C_i against a context-specific steering vector trained only on a specific context c_i, both evaluated on the context of c_i. More figures testing other capabilities and contexts can be found in Appendix