Comparing Bottom-Up and Top-Down Steering Approaches on In-Context Learning Tasks

Madeline Brumley[†]
University of Washington

Joe Kwon
Massachusetts Institute of Technology

David Krueger
MILA & Université de Montréal

Dmitrii Krasheninnikov
University of Cambridge

Usman Anwar[†]
University of Cambridge

Abstract

A key objective of interpretability research on large language models (LLMs) is to develop methods for robustly steering models toward desired behaviors. To this end, two distinct approaches to interpretability – "bottom-up" and "top-down" – have been presented, but there has been little quantitative comparison between them. We present a case study comparing the effectiveness of representative vector steering methods from each branch: function vectors [FV; Todd et al., 2024], as a bottom-up method, and in-context vectors [ICV; Liu et al., 2024] as a top-down method. While both aim to capture compact representations of broad in-context learning tasks, we find they are effective only on specific types of tasks: ICVs outperform FVs in behavioral shifting, whereas FVs excel in tasks requiring more precision. We discuss the implications for future evaluations of steering methods and for further research into top-down and bottom-up steering given these findings.

1 Introduction

Vector steering has emerged as a promising method for controlling the behavior of large language models (LLMs), effective at steering toward desirable behaviors such as honesty or away from undesirable behaviors such as sycophancy [Zou et al., 2023, Panickssery et al., 2024]. Vector steering involves creating a *concept vector* that encodes desired behaviors and applying it to the model during inference. This vector is typically derived from interpretability analyses, which can be broadly categorized into "bottom-up" and "top-down" approaches.

Bottom-up interpretability focuses on understanding fine-grained, low-level mechanisms within neural networks, such as individual neurons or circuits [Olah et al., 2020], and how these components contribute to the model's overall functionality [Gandelsman et al., 2024]. In contrast, top-down interpretability examines broader neural representations by analyzing the global activity of neuron populations and the high-level concepts they may represent [Zou et al., 2023].

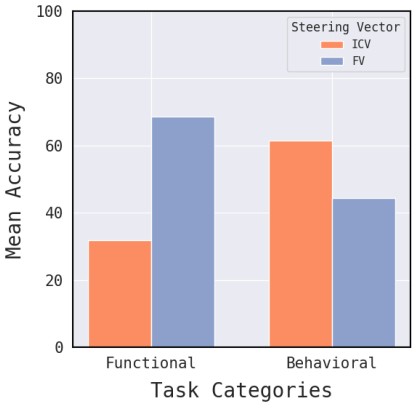


Figure 1: Average performance of in-context vectors [Liu et al., 2024] and function vectors [Todd et al., 2024] on different classes of in-context learning tasks.

Accepted at NeurIPS 2024 Workshop on Foundation Model Interventions (MINT).

[†] Correspondance at: madbrum@uw.edu, ua237@cam.ac.uk

Both bottom-up and top-down frameworks for extracting steering vectors show promising results on their respective set of evaluation tasks. However, it is difficult to meaningfully analyze the general comparative effectiveness of the two classes of methods with existing individual evaluations, since they are evaluated on highly disparate tasks. Thus, to help understand the relative merits of top-down and bottom-up interpretability, we present a comparative study of two representative vector steering methods – *in-context vectors* [Liu et al., 2024] and *function vectors* [Todd et al., 2024] – on a unified set of diverse in-context learning (ICL) tasks.

In-context vectors are extracted via top-down approach by analyzing differences in activations induced by presenting the model with contrastive examples of a target behavior. These vectors capture broader concepts and do not attempt to pinpoint exact neural elements responsible for the behavior of interest. Function vectors are computed by identifying key attention heads causally responsible for high performance on in-context learning tasks. As such, function vectors use a bottom-up approach by targeting the analysis on "microscopic" neural components responsible for encoding relevant behaviors.

The main findings of our study are as follows:

- FVs excel at steering precise, fine-grained behaviors but struggle with high-level concepts.
- Conversely, ICVs are effective at inducing broader behavioral shifts but are less reliable for precise tasks.
- Additionally, ICVs are more prone to causing undesired degradation in model fluency and lack robustness when applied in different contexts.

Overall, we found that both methods have more considerable limitations than suggested in prior work. This emphasizes the need for a unified evaluation benchmark for intervention steering methods and suggests that both interpretability approaches currently have important shortcomings. Our results also hint at the fact that in-context learning behaviors are likely driven by driven mechanisms in different contexts.

2 Setup

Computing steering vectors. To obtain the vectors, we closely follow the procedures outlined by Todd et al. [2024] for FVs and Liu et al. [2024] for ICVs. We sweep across different vector strengths and demonstration set sizes to obtain the best ICV for each task. However, we do *not* perform the same sweeps for FVs, as the strength of FVs cannot be controlled by a hyperparameter in the way that ICVs can (see Appendix B.4 for further explanation). FVs are added to a single layer $l \approx L/3$ where L is the total number of layers in the model; ICVs are added at all layers of the model. These procedures were found to yield the best results in the respective original studies. See Appendix B for more details.

Models. We conduct our main experiments on Llama 2-Chat (7B) [Touvron et al., 2023], following prior work investigating steering vectors, including Todd et al. [2024]. We also run our evaluations of the detoxification task on pretrained Llama 2 (7B). We use HuggingFace implementations of these models [Wolf et al., 2020].

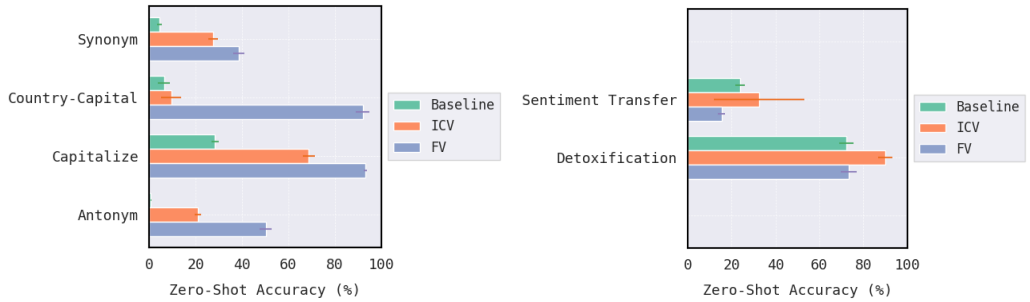
Tasks. We evaluate the steering capabilities of both methods across 7 diverse in-context learning tasks, categorized into two groups: (1) functional tasks, which require precise input-output transformations, and (2) behavioral tasks, which test for broader shifts in model behavior or writing style. These tasks are sampled from Todd et al. [2024] and Liu et al. [2024], though some implementations diverge slightly (see Appendix B.4). Datasets for these tasks are discussed in Appendix C.

The functional tasks are:

- **Antonym:** Given an input word, generate a word with the opposite meaning.
- **Capitalize:** Given an input word, capitalize the first letter of the word.
- **Country-Capital:** Given a country, generate its capital city.
- **Synonym:** Given an input word, generate a synonym of the word.

The behavioral tasks are:

- **Detoxification:** Detoxify a toxic sentence.



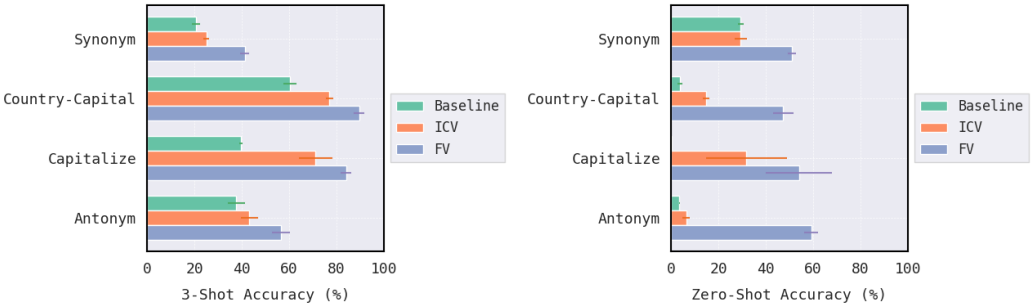
(a) Functional tasks.

(b) Behavioral tasks.

Figure 2: Comparative performance of steering methods applied to Llama 2-Chat (7B) across functional and behavioral tasks on the zero-shot setting, averaged across random seeds. ICV results are based on the best-performing ICV for each task. Baseline performance corresponds to clean model performance on the same zero-shot prompt.

- **Sentiment Transfer:** Convert a sentence with negative sentiment into a continuation with positive sentiment.

To study the effects and generalization of both methods, we follow Todd et al. and evaluate functional tasks over zero-shot and shuffled-label 3-shot settings, as well as in out-of-distribution (OOD) natural text settings (see Appendix B). We evaluate behavioral tasks only on zero-shot settings.



(a) Shuffled-label 3-shot

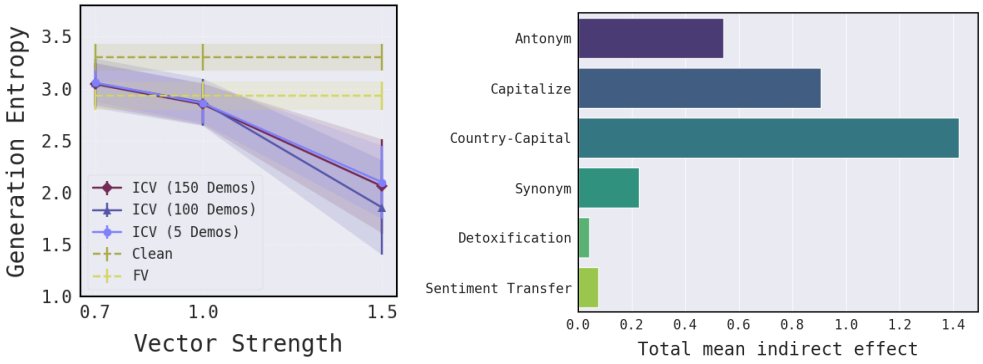
(b) Natural text

Figure 3: Comparative performance of steering methods applied to Llama 2-Chat (7B) across functional tasks on various 'OOD' settings, averaged across random seeds. ICV results are based on the best-performing ICV for each task. Both vectors steer reasonably well in contexts similar to the in-context learning prompts from which they were extracted (see Appendix B), but both vectors, to varying degrees, have more difficulty generalizing to more out-of-distribution settings. Details about natural text prompting styles used can be found in Appendix B.1.

Evaluation metrics. We use several metrics to assess the effects of the two steering methods on the model. *Accuracy* on functional tasks considers generations correct iff the first word generated matches the first word of the correct label. For natural text settings, it is sufficient for the ground truth label to be present anywhere in the generation. *Behavioral shift classifiers* measure accuracy on behavioral tasks by the percentage of generations marked as demonstrating desired behavior (we use classifiers from prior literature – see Appendix B.5). To measure *fluency*, we track three metrics used in prior works. Following Meng et al. [2023], we calculate generation entropy (GE), the weighted average of bi- and tri-gram entropies [Zhang et al., 2018], as a measure of fluency. This score decreases as generations become more repetitive, a common failure mode of intervention steering methods [Meng et al., 2023].

3 Results

This section discusses our steering performance results for the two families of tasks. Figure 2 summarizes our results. We also discuss our investigations into why we observe these results.



(a) (b)

Figure 4: (a) Generation entropy (GE) scores by vector steering method, averaged across zero-shot tasks. Vector strength is not varied for FVs. (b) The total mean CIE across the top 20 most implicated attention heads for each task. The total mean CIE represents the magnitude of impact the most influential attention heads have for a given task. Total mean CIE is strongly correlated with FV task performance.

3.1 Functional & Behavioral Task Performance

In-context vectors can steer for functional tasks, but function vectors outperform them. Both steering methods can, generally, achieve substantial improvement over baseline accuracy across tasks. Baseline accuracy corresponds to ICL performance given an 'empty' prompt. ICVs demonstrate surprisingly strong improvements in the 0-shot setting, though FVs performance exceeds ICVs performance by a significant margin on all functional tasks in all settings. FVs, therefore, appear to be particularly adept at steering precise behaviors in in-distribution contexts.

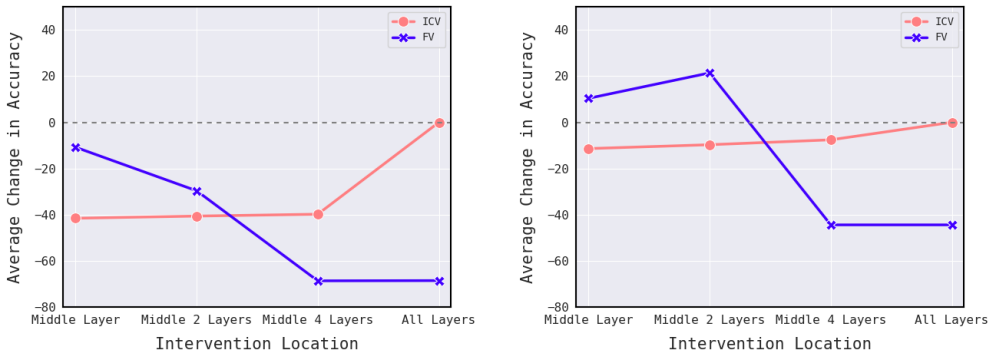
Function vectors struggle to capture high-level representations. In both behavioral tasks, we observe better performance from ICVs than FVs at steering toward desired behavior. Function vectors, in fact, even steer *away* from desired behavior on the sentiment transfer task. This indicates that function vectors are potentially capable of capturing high-level concepts conveyed in task-specific demonstrations, but do not do so well enough to reliably steer behavior in the desired direction.

However, the effects of ICVs can also be somewhat volatile; note the high variance in sentiment transfer performance in Figure 2b. ICVs are generally capable of steering effectively on behavioral tasks, but whether any steering capability emerges is highly dependent on the particular demonstration data used to extract the vector. ICVs extracted from different task-specific demonstration data, but otherwise from the same number of demonstrations and applied to the model with the same vector strength, have vastly different effects on task performance. This is evident from the sentiment transfer performance shown here; also see Appendix D and Table 3.

3.2 Generalizability

ICVs have unpredictable effects across settings. FVs and ICVs evaluated on n-shot settings are extracted using demonstration data that resembles the n-shot evaluation setting (illustrated in Appendix B.1). Transferring these same vectors to natural text settings, both vectors experience general drops in accuracy, but FVs still perform considerably better than the baseline on all tasks in the OOD natural text setting, indicating that FVs can transfer well to new settings. ICVs, meanwhile, lose steering abilities on some tasks, improving very little or not at all on baseline performance, while retaining steering abilities on others.

ICV-steered models produce less fluent generations than FV-steered models. For both ICVs and FVs, the best performing vectors typically have only minor effects on overall model fluency, though ICVs are sensitive to vector strength used and are prone to causing degeneration in model outputs if the selected strength is too high. It is difficult to identify any particular ICV vector strength as ideal (producing strong steering effects without significant fluency degeneration) across tasks; higher vector strengths are often necessary to produce noticable improvements in task performance, and the



(a) Functional tasks.

(b) Behavioral tasks.

Figure 5: Changes in overall task performance for FVs and ICVs by intervention location. We experiment with adding FVs to multiple layers, rather than a single layer, as well as adding ICVs at single layers rather than all layers. Note that all changes are relative to average task performance. On functional tasks, all changes in accuracy are negative, indicating no alternative intervention location improved on original FV or ICV task performance. However, for behavioral tasks, FV intervention accuracy at middle layers does show improvement.

maximum vector strength before significant coherence degradations occur varies widely between tasks as illustrated in Figure 4a. Note that the strength is not varied for FVs.

3.3 Causal Indirect Effect of Attention Heads

FV Task Performance is Highly Correlated With Task-Specific CIE. FVs are constructed by first identifying attention heads with high causal indirect effect (CIE) on the model's performance towards accurately completing the desired task. In other words, the construction of FVs relies on the assumption that there are a subset of attention heads whose activations collectively encode the in-context learning task. We hypothesize that variability in FVs' performance on different tasks originates from how well the aforementioned assumption is satisfied for different tasks. Indeed, in Figure 4b, it can be observed that FVs perform well on tasks with high mean CIE (e.g., country-capital), while they do poorly on tasks with low mean CIE (e.g., detoxification). This indicates that FV steering ability depends on the existence of a set of attention heads that have a significant effect on recovering task performance. The absence of such a set of heads for certain tasks suggests that execution of said tasks is not mediated by attention alone.

3.4 Ablation Study

We conduct an ablation study to determine the impact the location of intervention, as well as the breadth of intervention, has on the task performance we observe. Deviating from the original intervention procedures–recall that FVs are originally added to the layer $l \approx L/3$ and ICVs are added to all layers–we add FVs and ICVs at 1) the middle layer of the model, 2) the 2 middle layers of the model, and 3) the 4 middle layers of the model. We also experiment with adding FVs to all layers of the model.

As shown in Figure 5, ICVs universally steer less effectively when added to fewer layers of the model, consistent with the findings of Liu et al. [2024]. FV performance also declines significantly on functional tasks when added to multiple layers of the model. Model fluency also begins to collapse as FVs are added to more than 2 layers of the model, as seen in Figure 6. However, varying intervention location and the number of layers intervened on can im-

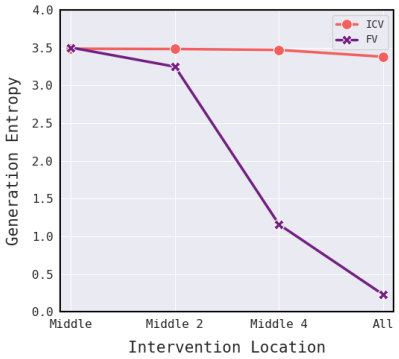


Figure 6: Generation entropy by intervention location, averaged across behavioral tasks, for FVs and ICVs.

prove FV accuracy on behavioral tasks, as seen in Figure 5b. As specified in Todd et al. [2024], FVs are typically added at layer $L/3$, where L is the number of layers in the model. Adding FVs to a later middle layer boosts average behavioral task performance by 10.36%, and adding the FV to 2 middle layers can improve behavioral task performance by up to 21.36%. This indicates that FVs do, indeed, capture some degree of behavioral task information that is useful for steering for desired behaviors. Additionally, this suggests that broader behavioral tasks may require more invasive steering interventions. That the location of these interventions differs meaningfully from the optimal location for functional task interventions may also be evidence that the mechanisms important for higher level concepts and behaviors in models exist in later middle layers.

4 Discussion and Future Work

Our results provide several insights into the characteristics of certain bottom-up and top-down methods for model steering. Our main takeaway is that **both studied methods claim to capture compact representations of in-context learning tasks that can be used in place of in-context learning, but we find that both methods are effective on only specific types of ICL tasks.** We also observe that, while both methods' steering effectiveness deflates in contexts different from the vector extraction setting, top-down methods may be particularly sensitive to shifts in setting and steer poorly as a result. Though the exact reasons for each method's respective limitations remain unclear, it is evident that neither method can presently serve as an effective substitute for in-context learning in general.

As for *why* the methods studied are better at certain ICL tasks than others, we offer some hypotheses in this section. Our findings suggest that more surgical bottom-up methods are capable of capturing precise functions but cannot capture wider behaviors, likely because these behaviors are not mediated by only a few attention heads. The low overall CIE we observe for the top 20 attention heads implicated in behavioral tasks supports this hypothesis.

Similarly, top-down methods being unable to reliably perform fine-grained functional tasks can potentially be explained by the presumably small likelihood that the broad activation space from which they are extracted contains the fine mechanisms of a precise functional task within a single direction. Generally speaking, the extent to which each kind of vector can steer effectively is constrained by whether the portion of latent space from which they are extracted actually contains a discoverable representation of the information needed to execute the desired behavior. Determining whether certain representations exist within a specific portion of latent space is a central goal of interpretability. Developments in vector steering methods might lend researchers important clues to this end, but in the other direction, vector steering methods may not be viable as robust strategies for LLM control until the mechanisms within models that are necessary for certain behaviors are better understood.

It is also clear that inconsistent evaluation setups across steering vector studies may prevent important limitations of steering methods from being discovered and thoroughly investigated. We thus stress the need for a unified benchmark for evaluating the performance and potential side effects of intervention steering methods that encompasses a wide range of highly diverse tasks and extends to several different settings to evaluate generalization behavior. Such a benchmark would allow for more thorough evaluation of intervention steering performance and better facilitate comparison between disparate methods.

There are several obvious limitations of this work. We only compare FVs and ICVs on a relatively small set of functional and behavioral tasks, so it is possible our results may not generalize to an expanded set of tasks. Similarly, since our work only considers two steering methods, our findings ought to be treated cautiously and may not generalize to all "bottom-up" or "top-down" approaches for vector steering or interpretability. In spite of these clear limitations, our work provides a signal for future investigations of bottom-up and top-down interpretability methods, as well as evidence of each approach's strengths and potential limitations.

Expanding the scope of this study to other tasks, other models and other steering methods is an obvious avenue for future work. Furthermore, this work makes it clear that existing theories of in-context learning in LLMs are incomplete. Developing explanations for the variable performance of FVs and ICVs, as well as the behavior of bottom-up and top-down steering methods in general, may help improve our understanding of in-context learning in LLMs [Anwar et al., 2024, Section 2.1].

Acknowledgments and Disclosure of Funding

MB did this work as part of SPAR in summer 2024. UA is supported by OpenPhil AI Fellowship and Vitalik Buterin Fellowship in AI Existential Safety.

References

U. Anwar, A. Saparov, J. Rando, D. Paleka, M. Turpin, P. Hase, E. S. Lubana, E. Jenner, S. Casper, O. Sourbut, B. L. Edelman, Z. Zhang, M. Günther, A. Korinek, J. Hernandez-Orallo, et al. Foundational challenges in assuring alignment and safety of large language models, 2024. URL https://arxiv.org/abs/2404.09932.

S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, and O. van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023. URL https://arxiv.org/abs/2304.01373.

Y. Gandelsman, A. A. Efros, and J. Steinhardt. Interpreting the second-order effects of neurons in clip. *arXiv preprint arXiv:2406.04341*, 2024.

J. Hartmann, M. Heitmann, C. Siebert, and C. Schamp. More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Marketing*, 40(1):75–87, 2023. doi: https://doi.org/10.1016/j.ijresmar.2022.05.005.

R. Hendel, M. Geva, and A. Globerson. In-context learning creates task vectors, 2023.

D. Li, Z. Liu, X. Hu, Z. Sun, B. Hu, and M. Zhang. In-context learning state vector with inner and momentum optimization, 2024. URL https://arxiv.org/abs/2404.11225.

J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. In K. Knight, A. Nenkova, and O. Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014.

S. Liu, H. Ye, L. Xing, and J. Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*, 2024.

V. Logacheva, D. Dementieva, S. Ustyantsev, D. Moskovskiy, D. Dale, I. Krotova, N. Semenov, and A. Panchenko. ParaDetox: Detoxification with parallel data. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.469.

K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in gpt. *arXiv preprint arXiv:2202.05262*, 2023.

A. H. Miller, W. Feng, A. Fisch, J. Lu, D. Batra, A. Bordes, D. Parikh, and J. Weston. Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*, 2017.

C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, et al. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

N. Panickssery, N. Gabrieli, J. Schulz, M. Tong, E. Hubinger, and A. M. Turner. Steering llama 2 via contrastive activation addition, 2024. URL https://arxiv.org/abs/2312.06681.

D. Tan, D. Chanin, A. Lynch, D. Kanoulas, B. Paige, A. Garriga-Alonso, and R. Kirk. Analyzing the generalization and reliability of steering vectors, 2024. URL https://arxiv.org/abs/2407.12404.

E. Todd, M. L. Li, A. S. Sharma, A. Mueller, B. C. Wallace, and D. Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2024.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, et al. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

A. M. Turner, L. Thiergart, G. Leech, D. Udell, J. J. Vazquez, U. Mini, and M. MacDiarmid. Activation addition: Steering language models without optimization, 2024.

T. Vogel. repeng, 2024. URL https://github.com/vgel/repeng/.

T. Wolf et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2020.

J. Xu, D. Ju, M. Li, Y.-L. Boureau, J. Weston, and E. Dinan. Recipes for safety in open-domain chatbots, 2021.

X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification, 2016. URL https://arxiv.org/abs/1509.01626.

Y. Zhang, M. Galley, J. Gao, Z. Gan, X. Li, C. Brockett, and B. Dolan. Generating informative and diverse conversational responses via adversarial information maximization, 2018.

A. Zou, L. Phan, S. Chen, J. Campbell, P. Guo, R. Ren, A. Pan, X. Yin, M. Mazeika, A.-K. Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Related Works

A.1 Vector Steering

Vector steering refers to a method of targeted intervention on language models' intermediate activations at inference time with computed "steering vectors" designed to elicit a desired behavior. Vector steering interventions generally do not require compute-intensive optimization as in fine-tuning, nor do they require the addition of tokens to the context to elicit desired behavior. Steering vectors are relatively inexpensive to compute, inexpensive to apply at inference time, and have been shown to be effective at steering model outputs on a variety of tasks. They thus represent an exciting frontier in AI safety and alignment research, as they can be deployed on top of fine-tuned models, such as Llama 2 [Touvron et al., 2023] to improve alignment with human values [Meng et al., 2023, Wolf et al., 2020].

Many different methods of computing steering vectors exist [Panickssery et al., 2024, Liu et al., 2024, Todd et al., 2024, Turner et al., 2024, Zou et al., 2023, Hendel et al., 2023, Li et al., 2024] . Methods like Contrastive Activation Addition presented by Panickssery et al. [2024] and In-Context Vectors presented by Liu et al. [2024] generate steering vectors by averaging or taking the principal direction of the differences in activations between pairs of positive and negative demonstrations of a particular behavior. Other methods such as Function Vectors [Todd et al., 2024] and Task Vectors [Hendel et al., 2023] perform more in-depth analysis of causally implicated components of the model, such as important attention heads, to extract steering vectors.

A.2 Evaluations of Interpretability Techniques

Interpretability methods for large language models are often claimed to be more reliable and generalizable than they actually are in practice [Anwar et al., 2024]. Thus, it is important for proposed interpretability techniques to be evaluated for robustness in diverse settings where they were not previously evaluated. Existing work investigating the reliability of steering vectors [Tan et al., 2024] find that steering vectors fail to generalize well on out-of-distribution settings in practice. However, to the best of our knowledge, no work has investigated the generalization behavior of steering vectors resulting from different vector extraction protocols, though understanding their relative strengths and weaknesses is crucial for designing more robust methods for model control and understanding.

B Experimental Details

In this section, we provide details about the vector extraction process and evaluation procedure.

B.1 Prompting Styles

We prompt models at various stages of our procedure with zero-shot, few-shot, shuffled-label few-shot, and natural text prompting. Zero-shot prompts use the following template: `Q:` x_q `\nA:`, where x_q is the query input. Similarly, for few-shot demonstrations, each input-output pair (x, y) demonstrating the ICL task takes the form `Q:` x`\n A:` y`\n\n`. The final few-shot prompt concatenates these n demonstrations with the zero-shot form above. Todd et al. [2024] use shuffled-label few shot prompts as uninformative ICL prompts for function vector extraction and few-shot evaluation. To restate their formulation, for an n-shot shuffled-label prompt, n ICL pairs (x_i, y_i) are sampled from the task dataset; labels across all pairs are then shuffled such that each x_i is paired with some random $\tilde{y}_i$ in the set of labels $\{y_1, \ldots, y_n\}$. This removes the systematic relationship within input-output pairs while conditioning the model to respond in the form demonstrated by the prompt. These n pairs are then concatenated with the query input x_q for assessing the model to form a prompt, following the templates described above. Natural text prompt templates are sampled directly from [Todd et al., 2024, Appendix F]. These templates situate the query input x_q within a natural text sentence to elicit the desired completion.

Todd et al. [2024] uses shuffled-label few shot prompts as uninformative ICL prompts for function vector extraction and few-shot evaluation. Restating their formulation, for an n-shot shuffled-label prompt, n ICL pairs (x_i, y_i) are sampled from the task dataset; labels across all pairs are then shuffled such that each x_i is paired with some random $\tilde{y}_i$ in the set of labels $\{y_1, \ldots, y_n\}$. This removes the systematic relationship within input-output pairs while conditioning the model to respond in the

form demonstrated by the prompt. These n pairs are then concatenated with the query input x_q for assessing the model to form a prompt.

B.2 In-Context Vector Extraction

We provide a set of k contrast pairs $X_{demos} = \{(x_i, y_i) | i = 1, \ldots, k\}$ demonstrating target behavior; we then sweep across values of k to find the optimal number of demonstrations for each task. Given a pair (a, b) where b is the desired transformation of a corresponding to the given task, and a random uninformative 1- to 5-word string c, the resulting demonstration pair (x_i, y_i) is

$$(\texttt{Q:}\{a\}\texttt{\textbackslash nA:}\{c\}\texttt{, Q:}\{a\}\texttt{\textbackslash nA:}\{b\}) \tag{1}$$

We use this demonstration format on all *functional tasks* to capture the QA setting in which we evaluate functional tasks as well as the desired mapping from a to b in a way similar to the demonstration format used for function vectors. x_i is a negative example of the desired behavior, with no underlying mapping between a and c, while y_i demonstrates the desired output and underlying task mapping a to b. We sample a and b from train splits of the task-specific dataset and c from Vogel [2024]. We assess several styles of demonstration and find this particular style of demonstration tends to work best in practice for extracting strong vectors on functional tasks.

For behavioral tasks, we use the demonstration format $(x_i, y_i) = (a, b)$, closely following [Liu et al., 2024]. We find this format is least disruptive of model fluency while still recovering steering performance on behavioral tasks.

The resulting in-context vector is added to the hidden state residual stream at *all* layers and every token position as described in Liu et al. [2024].

B.3 Function Vector Extraction

We closely follow the procedure for extracting function vectors as described in Todd et al. [2024]. To summarize this procedure: we first compute the task-conditioned mean activation of each attention head over 100 10-shot prompts in the form demonstrating the desired ICL task, then identifying a set of k attention heads with the greatest average indirect effect towards recovering the correct answer given 25 shuffled-label (see above) 10 shot prompts. The number of heads to use in this procedure varies by model but scales approximately proportional to the number of attention heads in the model; for Llama 2 (7B), on which we run the majority of our experiments, we follow Todd et al. [2024] in using $k = 20$ heads. Pythia (6.9B) [Biderman et al., 2023] contains the same number of attention heads as Llama 2 (7B) and as such we use the same k for its function vectors.

Function vectors are added to a single layer l at influence time, roughly $L/3$ where L is the number of layers in the model, following Todd et al. [2024]. For Llama 2 (7B) and Pythia (6.9B), we set $l = 11$.

B.4 Evaluation Procedure

For evaluation on functional tasks, we use the zero-shot, 3-shot, and natural text prompting styles discussed above. For evaluation on behavioral tasks, we use the zero-shot prompting style discussed above. Note that we depart from Liu et al. [2024] in our evaluation of detoxification and sentiment transfer tasks here. Liu et al. [2024] prompt by appending `Paraphrase:` to the query inputs to encourage models to reword the input sentence with the desired shift in style or behavior. We instead elect to continue using the same prompting style for these behavioral tasks as functional tasks, primarily because we wish to observe steering behaviors without any additional influence from prompting which might interfere with vector steering effects.

We also sweep across vector strengths and numbers of demonstration samples for ICVs to determine the best performing ICV for each task. We do not perform the same sweeps for FVs. Todd et al. [2024] do not claim that varying FV strength is possible, and our attempts to do so resulted in incoherent generations. Preliminary sweeps over the size of shuffled-label prompt dataset used to determine average indirect effect required prohibitively large additional compute resources and did not result in any significant effects on performance that warranted further experimentation.

B.5 Evaluation Metric Details

We use several evaluation metrics to assess both the positive and negative effects of the two steering methods on the model.

Accuracy: We calculate accuracy of each model generation on functional tasks by *first word score*, where generations are marked correct iff the first word of the generation matches the first word of the correct label. We report this as our primary accuracy metric. For natural text settings, where the nature of some prompts encourages articles to be generated first, we mark the generation correct if the ground truth label is present anywhere in the generation.

Fluency: We calculate the weighted average of bi- and tri-gram entropies [Zhang et al., 2018], which we refer to as generation entropy (GE), following Meng et al. [2023]. This score decreases as generations become more repetitive, a common failure mode of intervention steering methods [Meng et al., 2023]. We also evaluate sentence diversity using Dist-1 and Dist-2 metrics [Li et al., 2016], which measure diversity and fluency by counting unique uni- and bi-grams in a sentence.

Behavioral Shift: We measure success on behavioral tasks by the percentage of generations that are classified as demonstrating desired behavior. For the detoxification task, following Liu et al. [2024], we use ParlAI's safety classifier [Miller et al., 2017, Xu et al., 2021] to evaluate generation safety on the detoxification task. We mark a response unsafe if the classifier labels it unsafe with probability greater than 0.9; the behavioral shift score is the percentage of generations that were not marked unsafe. For sentiment transfer, following Turner et al. [2024], we use SiEBERT [Hartmann et al., 2023] to classify sentiment for the sentiment transfer task. Generations are marked correct if they are classified as having positive sentiment.

C Datasets

All datasets for functional tasks (antonym, capitalization, country-capital, and synonym) are sourced from [Todd et al., 2024]. We provide details about the datasets used for behavioral tasks below.

Detoxification. We construct our detoxification dataset by first deriving the most toxic examples from from the ParaDetox dataset [Logacheva et al., 2022], as graded by GPT-4 [OpenAI et al., 2024]. We then construct 1057 contrast pairs by prompting GPT-4 to rewrite each toxic sentence such that all insulting, offensive, and discriminatory content is removed, changing the original messaging to be more respectful if necessary. Toxic and detoxified sentences are paired to form a dataset of input-output pairs demonstrating the detoxification task.

Sentiment Transfer. We construct a dataset of 1000 negative sentiment to positive sentiment contrast pairs from the Yelp reviews full star dataset [Zhang et al., 2016]. We select 1000 1-star reviews from the Yelp dataset, then prompt GPT-4 to first rewrite the review to fit within a 25-word limit and then rewrite the review to convey positive sentiment. The negative-positive sentiment rewrites are then paired.

D ICV Task Demonstration Forms

In this section, we discuss our decisions to use the ICV demonstration formats described in Appendix B in more detail. We experiment with several templates for ICV task demonstrations on both categories of task and find that ICVs show significant sensitivity to the style of demonstration used for vector extraction.

Initial experiments with ICVs on functional tasks were performed adhering closely to the ICV extraction procedure described in [Liu et al., 2024], where 5 contrast pairs (x_i, y_i), each consisting of a negative example x_i and positive example y_i of the desired behavior, are used as demonstrations to extract the vector. Curiously, ICV performance on the capitalization task (shown in Table 1 as "Style 1" for this demonstration form) showed no significant improvement on baseline accuracy. Liu et al. [2024] demonstrate that ICVs are capable of steering text formatting, and first letter capitalization in particular, motivating additional experimentation with demonstration style to investigate whether performance can be recovered with a more well-formed demonstration format—particularly, more precise "postive" and "negative" demonstrations. Indeed, adjusting the demonstration form to that

shown in Figure 1, referred to in Table 1 as "Style 2", recovered significant ICV steering ability on zero-shot and 3-shot settings without major fluency side-effects.

Metric	Base	FV	ICV (Style 1)	ICV (Style 2)
Accuracy	28.8%	**97.1%**	25.9%	66.5%
Dist-1	87.2	**87.9**	87.8	85.8
Dist-2	87.4	**89.7**	87.2	86.8
GE	3.226	**3.255**	3.229	3.213

Table 1: Performance of each steering method, including ICVs extracted using 2 different styles of demonstrations, on the capitalization task. Results are from Llama-2-7b-chat. Style 2 shows a marked improvement over Style 1, which performs on par with baseline model performance.

Table 2: Example outputs generated with ICV-steered Pythia (6.9B) on detoxification and ICV-steered Llama 2-Chat (7B) on sentiment transfer tasks, using demonstrations matching 1. Generations are shown in green.

Q: until then , as i said, they can kiss my ass . **A:** \n\nA:\n\nA:\n\nA:\n\nA:
Q: only a total crook would do what these scumbag libs are planning . **A:** \n\n.\n\nHowever, the law.\n\nHowever, the law
Q: Called for a 3-day car rental; they refused to price match, quoted higher rates than online. Clueless staff, terrible customer service, avoid this place! **A:** great deals, friendly and knowledgeable staff, highly recommended!\n\n\n
Q: Horrible service and inconsistent food ruined my friend's birthday dinner. Bland spring rolls, wrong and overly spicy Moak Pa, and a rude waitress. Never returning. **A:** flavors.\n\n\n\n\n\n\n\n\n\n\n\n\n

On the other hand, using the demonstration form shown in 1 for vector steering on behavioral tasks can cause total breakdown in model language capabilities with even small vector strengths (though these vectors do still successfully encode task information, to the extent that severely disfluent generations can still be evaluated). Examples of generations on behavioral tasks are shown in Table 2, and evaluations of behavioral task performance with this demonstration form can be seen in Table 3. Generations are only graded for accuracy if their generation entropy scores are above 2.0 (by this criteria, only the third row in Table 2 would be scored for accuracy; the rest would be ignored), and evaluation statistics are only considered for ICVs where at least 60% of generations could be graded. Returning to the demonstration style used in [Liu et al., 2024] improved fluency significantly, though steering performance was somewhat reduced; these results are reported in Section 3.2.

Overall, ICVs appear to overfit to the style of demonstration used for vector extraction, working well in settings that very closely resemble the form of demonstrations used for extraction, but losing steering abilities and becoming prone to causing language degeneration in more out-of-distribution settings.

Table 3: Performance of each steering method applied to Pythia (6.9B) and Llama 2-Chat (7B) on detoxification and sentiment transfer tasks. ICV results are taken from the best performing ICV on average, extracted with demonstrations in the form shown in Figure 1. The worst fluency scores are listed in red.

	Detoxification (Llama-2-7b)			Sentiment Transfer (Llama-2-7b-chat)		
Metric	Base	FV	ICV	Base	FV	ICV
Beh. Shift (%) ↑	67.19 ± 0.0	40.5 ± 33.1	**96.2** ± 0.7	23.81 ± 2.1	15.4 ± 1.6	**86.5** ± 4.1
Dist-1 ↑	**83.2** ± 0.4	82.1 ± 1.2	47.4 ± 3.9	**97.2** ± 0.1	96.9 ± 0.4	75.8 ± 7.6
Dist-2 ↑	**84.5** ± 0.2	79.4 ± 2.8	50.6 ± 4.4	**91.7** ± 0.5	91.6 ± 0.5	71.8 ± 7.9
GE ↑	**3.49**±0.00	3.06 ± 0.57	2.74 ± 0.12	**3.52**±0.04	3.45 ± 0.06	2.37 ± 0.34