LLM Generalization with Function Vector Decomposition

Essam Sleiman **Katherine Hunter** **Megan Luu**
Harvard University
{esleiman,katherinehunter,meganluu}@g.harvard.edu

Abstract

Function vectors (FV) are quantitative representations of in-context learning demonstrations and can be added to language models to encourage a particular behavior. Recent work has demonstrated the ability to apply semantic vector arithmetic to function vectors to induce semantic functional behavior. In this paper, we explore if a FV can be decomposed into semantically sound counterparts by exploring if FV vector arithmetic can be utilized to mitigate out-of-distribution bias. We evaluate the effectiveness of FV decomposition through an overarching experiment of subtracting or adding distribution-related components and evaluating the corresponding out-of-distribution (OOD) performance. Our results indicate FV can, to some extent, be decomposed into task and distribution-related components and we show this through our OOD generalization experiment.

1 Introduction

In-context learning (ICL) is a remarkable behavior of large language models (LLMs), where LLMs are able to 'learn' a new task without any fine-tuning, simply by including examples of the task in the prompt (Brown et al., 2020). Pre-training and fine-tuning LLMs for a downstream task can be computationally expensive, particularly as these models continue to grow in size, so ICL makes the use of LLMs far more accessible by decreasing time and computation. Many studies have proposed strategies to improve ICL, often by altering the format or content of demonstrations (Min et al., 2022; Wang et al., 2023). For example, Gao et al. (2023) found that providing demonstrations that are more semantically similar to the query result in higher ICL performance.

While the mechanism of ICL is still poorly understood, there has been some efforts to further understand how ICL works. Some insight into ICL's efficacy can be found in a recent method of generating 'function vectors' (FVs) from the latent states of the LLM after seeing the demonstrations from ICL (Todd et al., 2023; Liu et al., 2023). These FVs are generated by identifying the attention heads that contribute the most information from the context towards the prediction and averaging their activations to produce a vector representation of the task. By shifting the latent states of attention heads by the FV during zero-shot inference, the LLM can perform the task without demonstrations. This use of FVs retains the capabilities of ICL without the need for demonstrations at test time.

While these vectorized task representations appear to do well at shifting the LLM towards the desired task, it remains unclear what exactly is represented in FVs and if the FV can be decomposed into semantically sound components. In this paper, we study the capability of the decomposition of FVs through evaluating an LLM's ability to better generalize to OOD queries. Min et al. (2022) show that models perform poorly on out-of-distribution (OOD) inputs and we seek to evaluate how poorly FVs generalize, and if this can be mitigated with FV arithmetic.

Our paper offers the following two contributions:

(1) We discover FVs are not robust; scaling the FV by a constant before inserting it into an LLM can have a significant impact on task performance. The optimal scaling factor for one task is different than another.
(2) To some extent, FVs can be decomposed into their task-related and distribution-related components. We show that LLM's OOD generalization performance can improve by nearly 50% by adding a corresponding distribution-related FV component.

2 Related Work

2.1 In-Context Learning

In-Context Learning (ICL) was presented in Brown et al. (2020) as an emerging paradigm for Large Language Models (LLMs), where they can learn from a few examples in context without parameter updates. Since then, there has been a rapidly growing body of work seeking to better understand ICL and its biases. In their work, Min et al., (2022) shows the correct ground truth in context matters little, while the distribution from where the context is sampled from does impact performance; context sampled out-of-distribution shows a significant (3-16%) performance hit for both multiple choice and classification tasks. Another work (Gao et al., 2023) found that semantically similar demonstrations to the query improves performance as well as samples that the model tends to misclassify or finds ambiguous. Fei et al., (2023) identify that LLMs are more likely to output a prediction that is frequently found in the pretraining data and has a tendency to favor the last label in the demonstration more than others.

The mechanism behind ICL remains unclear, but Dong et al. (2023) likens it to learning "from analogy", where the LLM applies the same structure found in the demonstrations to its output using its attention mechanism. Min et al. (2022) suggested that LLMs use four aspects of the demonstrations for inference: 1) the input-label mapping, 2) the distribution of the input text, 3) the label space, and 4) the format. Separate experiments altering these aspects confirmed their importance within ICL, as performance dropped when 1) inputs were paired with incorrect labels (although this impact on performance was minimal), 2) out-of-distribution demonstrations were used (3–16% performance gap), 3) labels were replaced with random words (5–16% performance gap), and 4) only inputs or only labels were used as demonstrations.

Yu et al. (2023) showed that information memorized during training and in-context information compete for prediction during inference. They gave the example of a question-answer task querying for the capital of a country where they provide demonstrations with the incorrect capital. They found that the model is more likely to output the memorized information when the model has more parameters or when the factual association appears more often in training data. They also find that certain attention heads contribute more to the probability of outputting the memorized answer while others emphasize the contextual information. As a result, when they down-weight the attention heads that emphasized memorized information, they bias the LLM to make predictions using the in-context information more often (e.g. predicting 'London:Poland' in a capital:country task). While Yu et al. (2023) show that larger language models are more likely to produce memorized information, Wei et al. (2023) show the opposite by claiming that larger language models are better able to override their learned parameters in favor of new, in-context data.

2.2 Function Vectors

An alternative approach to in-context learning is the use of in-context vectors (ICV) or function vectors (FV), as described in the work of Liu et al., 2023. An in-context vector is generated from the latent embeddings of an LLM that has been fed in context demonstrations such that the ICV encodes information about the target task and demonstrations. The ICV can then be used to steer the latent states of an LLM with a new query towards the in-context learning task. This work employed techniques including PCA, normalization, and vector addition/negation to refine the ICVs' influence on the model, and we incorporate some of these techniques in our experiments to evaluate how to best use FVs to mitigate biases induced by demonstration examples within in-context learning. Building off of this, past work has also shown the effectiveness of applying FVs to models in shuffled few-shot and zero-shot settings to steer models to perform target tasks without explicit or correct context, as well as the ability to apply vector arithmetic to FVs to compose/decompose separate ICL tasks. The work also suggested that for certain tasks, the decoded tokens of FVs correspond to the tasks' output space. This prompted us to question the significance of the task's word distribution in shaping the composition of a function vector, and how we might be able to manipulate FVs to improve model performance on out-of-distribution inputs. (Todd et al., 2023)

3 Methodology

3.1 Motivation

Language models struggle to generalize to out-of-distribution (OOD) data and tasks not seen during training (Hosseini et al., 2022). Overcoming this distribution bias can significantly advance the versatility of language models in real-world scenarios.

In-context learning partially addresses this problem where a few demonstrations can be provided to help solve a novel task without parameter updates, but the issue of distribution bias remains (Zhao et al., 2021). Given that a Function Vector (FV) is extracted from the hidden states of a model evaluated with few-shot prompts, we hypothesize that FVs encode information about both the target task and the demonstration's data distribution. Thus, our work aims to investigate if these two components can be decomposed from a FV, resulting in a task vector that can be matched and added to different data distribution vectors during inference for generalization to those datasets. To evaluate the effectiveness of the decomposition, we study zero-shot performance on new distributions.

3.2 Problem Formulation

An autoregressive transformer-based large language model f takes as input prompt p and outputs a probability distribution over its vocab space v. For each task $t \in T$, we have a dataset P_t of in-context learning prompts $p_i^t \in P_t$. Each prompt p_i^t contains a set of input-output exemplar pairs (x, y) and a query x_{iq}. We formulate in-context learning prompts as

$$p_i^t = [(x_{i1}, y_{i1}), ..., (x_{iN}, y_{iN}), x_{iq}] \quad (1)$$

To build a function vector on a dataset P_t, we take the mean of task-conditioned activations attention heads $a_{\ell j}^{-t}$ at the last token position for the j^{th} attention head in layer ℓ.

$$a_{\ell j}^{-t} = \frac{1}{|P_t|} \sum_{p_i^t \in P_t} a_{\ell j}(p_i^t). \quad (2)$$

Causal mediation analysis then identifies a subset of 10 attention heads $\mathcal{A}$ which have the strongest causal effects towards the correct output when conditioned with ICL. The final function vector for task t is calculated as the sum of these mean task-conditioned activations:

$$v_t = \sum_{a_{\ell j} \in \mathcal{A}} a_{\ell j}^{-t} \quad (3)$$

We then evaluate on a new query x_{iq} by adding the function vector v_t to the final token hidden states at layer ℓ (adapted from Todd et al. (2023)).

3.3 Datasets

We evaluate on the tasks and datasets following Todd et al. (2023). Specifically, we construct our main experiments on the color_v_animal_3 and verb_v_adjective_3 datasets, which are designed to capture the tasks of selecting a color out of animals (Table 1) and selecting a verb out of adjectives (Table 2), respectively, out of a list of three words. We chose these datasets, because they can be formulated to have the same task (select the word that doesn't belong out of three) but come from two distinct distributions. The prompts that are used in-context to build the FVs (10-shot) are randomly sampled from the train set, and the query is sampled from the test set. We benchmark how well FVs generalize to new distributions on these datasets and experiment to improve their performance with FV decomposition.

Input	Output
rhinoceros, woodpecker, black.	black
violet, sloth, turtle.	violet

Table 1: Color_V_Animal Dataset Input-Output examples

Input	Output
fearless, climb, light.	climb
idealistic, deliver, intelligent.	deliver

Table 2: Verb_V_Adjective Dataset Input-Output Examples

3.4 Models

We run all our experiments on gpt2-xl (Radford et al., 2019) which is a transformer-based language model containing 1.5 billion parameters and 48 layers.

3.5 Baseline

To baseline FV's performance on OOD data, we build FVs from one distribution and evaluate its performance on another distinct distribution. More specifically, we build a FV from 100 randomly sampled 10-shot prompts from the color_v_animal dataset by averaging the activations from the language model when evaluated on these prompts. Let's call this FV_{c-a}. We then evaluate the performance of inserting FV_{c-a} in the zero-shot queries from the color_v_animal test set and evaluating. Note: we add FV_{c-a} to the 22^{nd} attention layer of the language model. Next, we build a new FV from the verb_v_adjective_3 dataset in the same fashion which we call FV_{v-a}. We then again evaluate

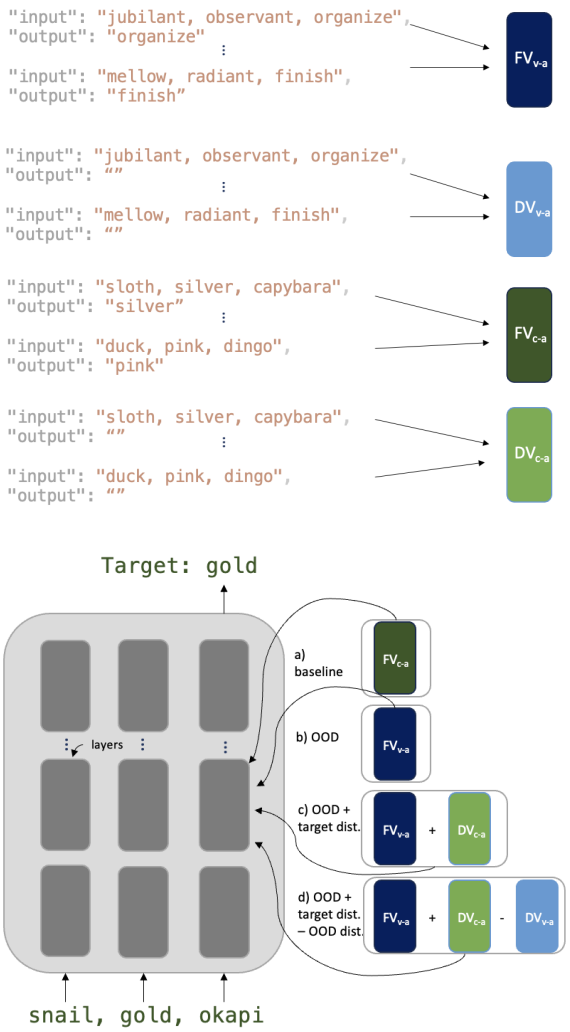


Figure 1: Experimental approach to mitigate out-of-distribution function vector bias. We first generate two function vectors from the datasets color_v_animal_3 (c-a) and verb_v_adjective_3 (v-a), averaging select activations after 100 10-shot ICL prompts were fed into gpt2-xl. Four zero-shot experiments are then run using queries from color_v_animal_3 and applying a) the color_v_animal_3 FV, b) the verb_v_adjective_3 FV, c) the verb_v_adjective_3 FV subtracting out the distribution vector from color_v_animal_3, and d) the previous vector from c) with the addition of the verb_v_adjective_3 distribution vector, all in separate trials.

zero-shot performance on the color_v_animal test set, inserting FV_{v-a} this time. Performance was evaluated using the average probability of predicting the correct output token and top-1, top-5, and top-10 accuracy. By comparing these performance metrics, we attain a baseline understanding of how the FVs improve performance for in-distribution queries and how performance drops when used to evaluate on a new distribution.

3.6 Experiments

Following Todd et al. (2023), we generate a FV for each task. In our baseline, we first test the robustness of FVs when applied to different distributions. We then perform arithmetic operations on extracted FVs from different distributions and/or combinations of distributions to examine their effects on zero-shot tasks. The key idea is that function vectors obtained from mean activations of attention heads contain information about the ICL task, the distribution of the text, and potentially other noise that is not directly related to the task. We hypothesize that a FV can be decomposed into its corresponding Distribution Vector (DV) and Task Vector (TV). The following experiments attempt to test this hypothesis by generating DVs and TVs and evaluating their performance when added and used on different tasks. Along the way, we also uncover issues such as a lack of robustness of FVs due to an inconsistent optimal scaling factor across different tasks and run experiments in an attempt to further understand this phenomenon and to mitigate it.

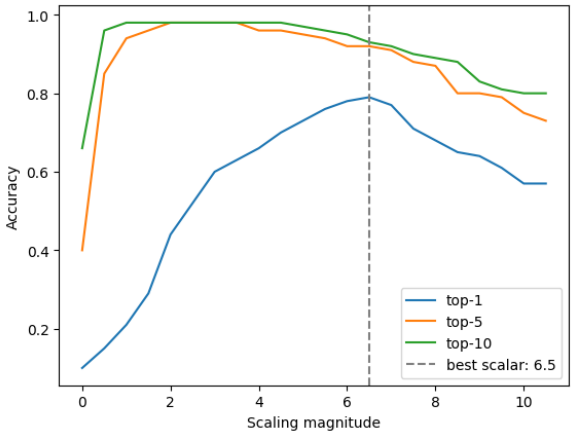
3.6.1 Scaling Function Vectors

We first discover that FVs can be multiplied by a scalar value to alter its influence when applied to shift model activations. On the example of evaluating on the color_v_animal_3 dataset, we build FV_{c-a} from the dataset and evaluate on queries from the same dataset. We test a range of scalar multiples of FV_{c-a} in a zero-shot context when applied to the 22^{nd} layer and find that each dataset has a different optimal scalar. We show the results in Figure 2. The optimal scaling for a FV is later used in arithmetic experiments as a comparison with the unscaled FV.

3.6.2 Function vector arithmetic

We then experiment to address the hypothesis that FVs contain both task and demonstration information that can be decomposed into DV and TV, respectively. We start by generating FVs from each of

(a) Dataset: color_v_animal_3



(b) Dataset: verb_v_adjective_3

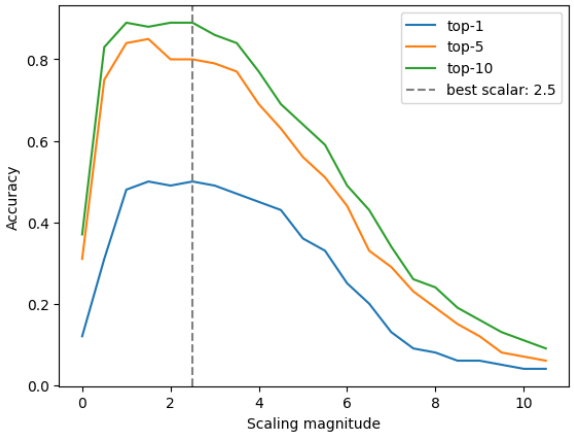


Figure 2: Function vectors derived from separate ICL demonstration distributions have different optimal scaling factors for maximum zero-shot accuracy. For these experiments, the scaled FVs were applied in the 22^{nd} layer of gpt2-xl to the final token activation when testing zero-shot queries from the test set of the same distribution.

the color_v_animal and verb_v_adjective datasets as defined in the problem formulation. These constitute a typical FV. Next, we attempt to build a distribution vector (DV) by modifying the original datasets by removing the question-answer format from them so they no longer represent a specific task, but still maintain the words present in the original task. DVs are constructed from these datasets, which no longer contain task-related information. From our baseline, we build FV_{c-a} from the color_v_animal_3 dataset, and evaluate on the verb_v_adjective_3 dataset to measure drop on performance on OOD tasks. We then build DV_{v-a} on the verb_v_adjective_3 dataset, and add it to FV_{c-a} with vector arithmetic to provide it with the verb_v_adjective_3 distribution information. When evaluating again on the verb_v_adjective_3 dataset, we see a recovery of nearly 50% performance drop in one case (with proper scaling).

3.6.3 Function Vector Normalization

Searching for the optimal scale of a function vector for each task can be time-consuming, computationally costly, and not a robust solution, so we seek to develop a general method to normalize the FVs and mitigate the phenomenon. We hypothesize this can be done by normalizing both the function vector and the last-token activation that it is added to, to place them on the same scale. The FV and activation are divided by their respective L2 norms, before being added and re-scaled by either the activation's original norm or a constant. The results of normalizing in this manner did not do as well as expected as shown in Table 3, but we encourage future research to further explore this direction!

3.6.4 Average Function Vectors

Lastly, we ran a small experiment to evaluate if FVs added from multiple distributions can better generalize to a new task distribution. To do this, we introduce a third dataset, Object_V_Concept, which represents the task of choosing the word corresponding to a physical object out of abstract concepts. Out of the three datasets, we generate FVs from two of the datasets, average the FVs, and apply the averaged FV to a zero-shot prompt from the third distribution. The intuition behind this is that averaging the distribution components of multiple tasks can help in generalizing to a third novel task distribution.

4 Results

4.1 Baseline Results

Our results show that using a FV built from the same task but on a different distribution results in reduced performance in comparison with a FV built and evaluated on the same distribution. Our paper seeks to mitigate this drop in performance such that FVs can better represent the target task and generalize past the distribution it was built upon.

4.2 Optimal scaling factor is unique to function vector

Testing scalar multiples of function vectors on a zero-shot task (applied to the distribution it was

Table 3: Normalizing FV and activation before addition during inference

Dataset	FV	avg_prob	top1	top5	top10
Color/Animal	FV_{c-a}	0.310	0.530	0.980	0.980
	FV_{v-a}	0.162	0.240	0.840	0.930
	FV_{v-a} + DV_{c-a}	0.199	0.370	0.930	0.980
	FV_{v-a} - DV_{v-a} + DV_{c-a}	0.171	0.270	0.850	0.940
	FV_{v-a} -DV_{v-a}	0.089	0.200	0.780	0.840
Verb/Adjective	FV_{v-a}	0.292	0.500	0.840	0.890
	FV_{c-a}	0.086	0.200	0.590	0.710
	FV_{c-a} + DV_{v-a}	0.040	0.090	0.440	0.530
	FV_{c-a}- DV_{c-a} + DV_{v-a}	0.066	0.160	0.540	0.640
	FV_{c-a} - DV_{c-a}	0.072	0.210	0.580	0.700

Table 4: Function vector arithmetic with no scaling

Dataset	Layer	Function Vector	avg_prob	top1	top5	top10
Color/Animal	22	FV_{c-a}	.16	.25	.95	1.0
Verb/Adjective	22	FV_{v-a}	.145	.47	.84	.9
Color/Animal	22	FV_{v-a}	.114	.21	.94	.98
Color/Animal	22	DV_{v-a}	.14	.21	.96	.98
Color/Animal	22	FV_{v-a} -DV_{v-a} + DV_{c-a}	.12	.19	.94	.98

derived from) reveals that function vectors obtained from different tasks require different scaling for optimal zero-shot performance. For example, the function vector constructed from the color_v_animal_3 dataset results in the highest top-1 accuracy for the zero-shot color-v-animal task when scaled by 6.5, achieving a top-1 accuracy of 0.79 compared with 0.21 when there is no scaling (Figure 2). However, the function vector obtained from the verb_v_adjective_3 dataset optimally shifts the LM towards the task when scaled by 2.5. To our knowledge, no prior work has investigated scaling Function Vectors up. Comparisons of the optimal scales for each function vector with their L1 and L2 norms reveal no consistent patterns, so more work remains to systematically identify the optimal scaling of function vectors without requiring an exhaustive search approach.

4.3 Adding/subtracting Distribution Vectors has variable performance

Performing FV arithmetic in which a vector of the target distribution is added (DV) and/or a vector of the source distribution is subtracted results in inconsistent behavior with regard to correcting the OOD bias. For the example of evaluating on the color_v_animal_3 dataset, applying the verb_v_adjective_3 FV (scaling all subsequent FVs by 2.5) results in a top-1 accuracy of 0.27: a decrease of 0.25 from the baseline of 0.52 obtained when using the color_v_animal_3 FV in Table 4. Instead, adding $DV_{c-a}$from the color_v_animal_3 dataset increases the top-1 accuracy to 0.39 as shown in Table 5. Contrary to our expectations, both adding DV_{c-a} and subtracting the DV_{v-a} does not perform as well as the addition only, with a top-1 accuracy of 0.29.

Testing with multiple extractive datasets reveals inconsistent behavior (Appendix A: Table 7). In no case does FV arithmetic restore the baseline accuracy, suggesting that task generalization may be more complex than removing distribution information. Alternatively, it may be that these function vectors do not encode the task as we expect it. For example, it is possible that the task of repeating the word from the list that belongs to a different category cannot be extracted from FV_{c-a}, because the task it encodes may be dependent on the understanding that the task is to repeat a color.

4.3.1 Averaging FVs has little effect on mitigating OOD bias

Lastly, the experiment of averaging FVs from two different distributions and applying it to a zero-shot query from a third distribution revealed little-to-no improvement in model performance. As shown in the results in Table 6, applying the averaged FV on the third distribution generally appears to have

Table 5: Function vector arithmetic scaled by 2.5

Dataset	Layer	Function Vector	avg_prob	top1	top5	top10
Color/Animal	22	FV_{c-a}	.30	.52	.98	.98
Verb/Adjective	22	FV_{v-a}	.31	.52	.8	.9
Color/Animal	22	FV_{v-a}	.1632	.27	.83	.9
Color/Animal	22	FV_{v-a}+DV_{c-a}	.20	.39	.85	.95
Color/Animal	22	FV_{v-a} - DV_{v-a} + DV_{c-a}	.17	.29	.86	.92

similar or poorer performance compared to using a pure FV generated from one of the other two distributions. This could indicate that the averaged FV still suffers from distribution bias, or averaging the FVs may not be an optimal way of distilling out a task representation. However, averaging an in-distribution FV with an OOD FV allows the model to recover some performance, as shown by the higher top-1 accuracies compared to using two averaged OOD FVs. This aligns with what we expect, as having an in-distribution component in the FV allows the model to perform better with that specific dataset. While we expected the two averaged OOD FVs to have better performance compared to using a single OOD FV, we acknowledge, from our previous experiments, that scaling plays an important role in ensuring the consistency of results. Therefore, scaling may be a contributing factor to these observed results, and this could warrant further experimentation.

5 Conclusions & Future Work

Our work can be seen as an introductory investigation into better understanding FVs by analyzing their ability to be decomposed into distinct concepts. More specifically, we hypothesize FVs can be decomposed into task (TV) and data distribution (DV) constituents. Along the way, we discovered that when scaled by a constant, FVs can have varying levels of performance improvement when applied to a task. Additionally, the optimal scaling factor for one task is different than another. This makes it such that FVs aren't necessarily robust, as their effect on performance can be dependent on the particular task or dataset. Thus, directions for future work could involve investigating methods to build more robust FVs by potentially normalizing them such that they are scalar and task invariant. A recent work by Jorgensen et al. (2023) suggests that an approach called mean-centering results in vectors that more directly represent the direction of the desired task in the representation space, and show improvements to the work by Todd et al. (2023). Other options for improving function vector construction and scaling may depend on the finding that some language models store task-specific information in a small subspace within representations, and that a small set of dimensions contain the majority of the variance in representations (Rudman et al., 2023).

Furthermore, future work can focus on how to best decompose a FV into task and distribution components in a way that best allows us to subtract away a DV from the original task and add a new DV. In our experiments, we built a DV from prompts containing the text from the task, with the task (ie: Question and Answer) components of the prompt removed. This may not be the optimal way to build a DV, therefore future work can further investigate this. Potential directions could include using the model's average word embeddings of the demonstrations as the DV, which can then be subtracted from the FV.

Also, as done in Todd et al. (2023), we add the FV to a particular layer, and the optimal layer to insert a FV differs between tasks. Similar to the scaling factor, in the future, we seek to develop a more robust method to insert the FV at an optimal layer, or even at multiple layers. Liu et al. (2023) found that their in-context vectors were most effective when added to all layers of the model, so further work could combine this observation with the method of Todd et al. (2023) for finding the most influential attention heads in ICL and apply function vectors to these selective heads.

Limitations

Due to limited access to computing resources, we could only experiment with models that fit on Google Colab's free GPU resources (gpt2-xl).

6 Impact Statement

The investigation of methods to exert more control over activation-steering with function vectors has

Table 6: Averaging 2 Function Vectors and querying on a third distribution

Dataset	Layer	Function Vector	avg_prob	top1	top5	top10
Color/Animal	22	FV_{c-a}	0.129	0.21	0.95	0.98
Color/Animal	22	FV_{o-c}	0.106	0.12	0.92	0.98
Color/Animal	22	FV_{v-a}	0.117	0.17	0.93	0.98
Color/Animal	22	(FV_{v-a}+FV_{o-c})/2	0.112	0.16	0.95	0.98
Color/Animal	22	(FV_{v-a}+FV_{c-a})/2	0.124	0.19	0.95	0.98
Color/Animal	22	(FV_{o-c} + FV_{c-a})/2	0.117	0.17	0.95	0.98
Object/Concept	22	FV_{o-c}	0.159	0.43	0.78	0.79
Object/Concept	22	FV_{c-a}	0.126	0.36	0.76	0.78
Object/Concept	22	FV_{v-a}	0.134	0.32	0.75	0.78
Object/Concept	22	(FV_{c-a}+FV_{v-a})/2	0.132	0.3	0.76	0.78
Object/Concept	22	(FV_{c-a}+FV_{o-c})/2	0.143	0.4	0.77	0.79
Object/Concept	22	(FV_{v-a}+FV_{o-c})/2	0.147	0.39	0.78	0.79
Verb/Adjective	22	FV_{v-a}	0.148	0.49	0.84	0.89
Verb/Adjective	22	FV_{c-a}	0.082	0.3	0.73	0.8
Verb/Adjective	22	FV_{o-c}	0.105	0.38	0.76	0.82
Verb/Adjective	22	(FV_{o-c}+FV_{c-a})/2	0.093	0.31	0.73	0.8
Verb/Adjective	22	(FV_{o-c}+FV_{v-a})/2	0.124	0.42	0.81	0.85
Verb/Adjective	22	(FV_{v-a}+FV_{c-a})/2	0.111	0.37	0.79	0.83

the potential to remove harmful biases exhibited by language models, such as removing toxic content or producing false information. However, this knowledge could be abused to amplify this harmful content rather than repress it.

Furthermore, it is important to understand the nuances of what constitutes word bias in different cultural contexts. This is important to ensure that our concept of mitigating bias does not unintentionally erase the diversity in linguistic expressions and semantics.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Lingyu Gao, Aditi Chaudhary, Krishna Srinivasan, Kazuma Hashimoto, Karthik Raman, and Michael Bendersky. 2023. Ambiguity-aware in-context learning with large language models.

Arian Hosseini, Ankit Vani, Dzmitry Bahdanau, Alessandro Sordoni, and Aaron Courville. 2022. On the compositional generalization gap of in-context learning.

Ole Jorgensen, Dylan Cope, Nandi Schoots, and Murray Shanahan. 2023. Improving activation steering in language models with mean-centring.

Sheng Liu, Lei Xing, and James Zou. 2023. In-context vectors: Making in context learning more effective and controllable through latent space steering.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work?

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

William Rudman, Catherine Chen, and Carsten Eickhoff. 2023. Outlier dimensions encode task specific knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14596–14605, Singapore. Association for Computational Linguistics.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. 2023. Function vectors in large language models.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023. Large language models are implicitly topic models: Explaining

and finding good demonstrations for in-context learning. In *Workshop on Efficient Systems for Foundation Models @ ICML2023*.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently.

Qinan Yu, Jack Merullo, and Ellie Pavlick. 2023. Characterizing mechanisms for factual recall in language models.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models.

A Appendix: Other figures

Table 7: Arithmetic experiments on other distributions (no scaling). Distributions are abbreviated by the first letter of their two category types, and 'nt' refers to distribution vectors.

Dataset	fv1	avg_prob	top1	top5	top10
color_v_animal_3	FV_{c-a}	0.1302	0.21	0.95	0.98
	FV_{v-a}	0.1169	0.18	0.94	0.98
	FV_{v-a} + DV_{c-a}	0.1417	0.21	0.96	0.98
	FV_{v-a} - DV_{v-a} + DV_{c-a}	0.1201	0.18	0.94	0.98
	FV_{v-a} - DV_{v-a}	0.0854	0.18	0.84	0.95
color_v_animal_3	FV_{o-c}	0.1067	0.12	0.92	0.98
	FV_{o-c} + DV_{c-a}	0.1276	0.16	0.95	0.98
	FV_{o-c} - DV_{o-c} + DV_{c-a}	0.1114	0.11	0.94	0.98
	FV_{o-c} - DV_{o-c}	0.0791	0.15	0.82	0.94
color_v_animal_3	FV_{a-o}	0.1071	0.11	0.94	0.97
	FV_{a-o} + DV_{c-a}	0.1251	0.16	0.96	0.98
	FV_{a-o} - DV_{a-o} + DV_{c-a}	0.1102	0.11	0.95	0.97
	FV_{a-o} - DV_{a-o}	0.0806	0.13	0.82	0.94
verb_v_adjective_3	FV_{v-a}	0.1486	0.5	0.84	0.89
	FV_{c-a}	0.0834	0.3	0.73	0.79
	FV_{c-a} + DV_{v-a}	0.0605	0.19	0.61	0.73
	FV_{c-a} - DV_{c-a} + DV_{v-a}	0.0781	0.27	0.71	0.79
	FV_{c-a} - DV_{c-a}	0.0781	0.27	0.73	0.77
verb_v_adjective_3	FV_{a-o}	0.0939	0.33	0.75	0.8
	FV_{a-o} + DV_{v-a}	0.0697	0.26	0.64	0.76
	FV_{a-o} - DV_{a-o} + DV_{v-a}	0.0888	0.32	0.73	0.79
	FV_{a-o} - DV_{a-o}	0.0853	0.32	0.76	0.79
verb_v_adjective_3	FV_{o-c}	0.1048	0.38	0.76	0.82
	FV_{o-c} + DV_{v-a}	0.0790	0.28	0.71	0.75
	FV_{o-c} - DV_{o-c} + DV_{v-a}	0.1036	0.34	0.76	0.82
	FV_{o-c} - DV_{o-c}	0.0978	0.33	0.76	0.82
object_v_concept_3	FV_{o-c}	0.1593	0.44	0.78	0.79
	FV_{c-a}	0.1284	0.36	0.76	0.78
	FV_{c-a} + DV_{o-c}	0.1149	0.35	0.76	0.79
	FV_{c-a} - DV_{c-a} + DV_{o-c}	0.1236	0.37	0.76	0.78
	FV_{c-a} - DV_{c-a}	0.1103	0.34	0.75	0.78

Table 8: Normalizing FV and activation before addition during inference. Expands on information presented in Table 3.

Dataset	FV	avg_prob	top1	top5	top10
color_v_animal_3	FV_{c-a}	0.310	0.530	0.980	0.980
	FV_{v-a}	0.162	0.240	0.840	0.930
	FV_{v-a} + DV_{c-a}	0.199	0.370	0.930	0.980
	FV_{v-a} - DV_{v-a} + DV_{c-a}	0.171	0.270	0.850	0.940
	FV_{v-a} - DV_{v-a}	0.089	0.200	0.780	0.840
verb_v_adjective_3	FV_{v-a}	0.292	0.500	0.840	0.890
	FV_{c-a}	0.086	0.200	0.590	0.710
	FV_{c-a} + DV_{v-a}	0.040	0.090	0.440	0.530
	FV_{c-a} - DV_{c-a} + DV_{v-a}	0.066	0.160	0.540	0.640
	FV_{c-a} - DV_{c-a}	0.072	0.210	0.580	0.700
color_v_animal_3	FV_{c-a}	0.310	0.53	0.98	0.98
	FV_{o-c}	0.150	0.23	0.94	0.96
	FV_{o-c} + DV_{c-a}	0.184	0.29	0.95	0.98
	FV_{o-c} - DV_{o-c} + DV_{c-a}	0.155	0.24	0.93	0.96
	FV_{o-c} - DV_{o-c}	0.0831	0.11	0.77	0.89
color_v_animal_3	FV_{c-a}	0.310	0.53	0.98	0.98
	FV_{a-o}	0.134	0.14	0.92	0.95
	FV_{a-o} + DV_{c-a}	0.168	0.21	0.95	0.98
	aFV_{a-o} - DV_{a-o} + DV_{c-a}	0.139	0.14	0.91	0.96
	FV_{a-o} - DV_{a-o}	0.0763	0.09	0.76	0.88
verb_v_adjective_3	FV_{v-a}	0.292	0.5	0.84	0.89
	FV_{a-o}	0.0957	0.22	0.66	0.77
	FV_{a-o} + DV_{v-a}	0.0522	0.15	0.45	0.56
	FV_{a-o}- DV_{a-o} + DV_{v-a}	0.0775	0.21	0.63	0.75
	FV_{a-ao} - DV_{a-o}	0.0798	0.26	0.63	0.72
object_v_concept_3	FV_{o-c}	0.270	0.51	0.77	0.78
	FV_{c-a}	0.163	0.28	0.76	0.77
	FV_{c-a} + DV_{o-c}	0.123	0.32	0.68	0.76
	FV_{c-a} - DV_{c-a} + DV_{o-c}	0.159	0.33	0.75	0.77
	FV_{c-a} - DV_{c-a}	0.101	0.29	0.7	0.74
color_v_animal_3	FV_{v-a}	0.292	0.5	0.84	0.89
	FV_{o-c}	0.138	0.33	0.72	0.83
	FV_{o-c} + DV_{v-a}	0.0704	0.17	0.49	0.61
	FV_{o-c} - DV_{o-c} + DV_{v-a}	0.130	0.32	0.7	0.81
	FV_{o-c} - DV_{o-c}	0.115	0.34	0.69	0.82

(a) Scaling expt: color_v_animal_3

Scalar	Avg_prob	top1	top5	top10
0.0	0.020	0.1	0.4	0.66
0.5	0.077	0.15	0.85	0.96
1.0	0.130	0.21	0.94	0.98
1.5	0.183	0.29	0.96	0.98
2.0	0.244	0.44	0.98	0.98
2.5	0.304	0.52	0.98	0.98
3.0	0.357	0.6	0.98	0.98
3.5	0.404	0.63	0.98	0.98
4.0	0.443	0.66	0.96	0.98
4.5	0.476	0.7	0.96	0.98
5.0	0.502	0.73	0.95	0.97
5.5	0.524	0.76	0.94	0.96
6.0	0.536	0.78	0.92	0.95
6.5	0.538	0.79	0.92	0.93
7.0	0.530	0.77	0.91	0.92
7.5	0.513	0.71	0.88	0.9
8.0	0.490	0.68	0.87	0.89
8.5	0.464	0.65	0.8	0.88
9.0	0.435	0.64	0.8	0.83
9.5	0.408	0.61	0.79	0.81
10.0	0.384	0.57	0.75	0.8
10.5	0.362	0.57	0.73	0.8

(b) Scaling expt: verb_v_adjective_3

Scalar	Avg_prob	top1	top5	top10
0.0	0.016	0.12	0.31	0.37
0.5	0.068	0.31	0.75	0.83
1.0	0.146	0.48	0.84	0.89
1.5	0.220	0.5	0.85	0.88
2.0	0.274	0.49	0.8	0.89
2.5	0.306	0.5	0.8	0.89
3.0	0.320	0.49	0.79	0.86
3.5	0.317	0.47	0.77	0.84
4.0	0.301	0.45	0.69	0.77
4.5	0.280	0.43	0.63	0.69
5.0	0.246	0.36	0.56	0.64
5.5	0.202	0.33	0.51	0.59
6.0	0.155	0.25	0.44	0.49
6.5	0.112	0.2	0.33	0.43
7.0	0.078	0.13	0.29	0.34
7.5	0.055	0.09	0.23	0.26
8.0	0.041	0.08	0.19	0.24
8.5	0.031	0.06	0.15	0.19
9.0	0.025	0.06	0.12	0.16
9.5	0.021	0.05	0.08	0.13
10.0	0.018	0.04	0.07	0.11
10.5	0.015	0.04	0.06	0.09

Figure 3: Comparison of scaling experiments