

Steering off Course: Reliability Challenges in Steering Language Models

Patrick Queiroz Da Silva[♣] Hari Sethuraman[♡]

Dheeraj Rajagopal[♣] Hannaneh Hajishirzi^{♡◇} Sachin Kumar[♣]

[♣]The Ohio State University, Columbus OH [♡]University of Washington, Seattle WA

[♣]Fastino AI, Palo Alto CA [◇]Allen Institute for AI, Seattle WA

dasilva.30@osu.edu

Abstract

Steering methods for language models (LMs) have gained traction as lightweight alternatives to fine-tuning, enabling targeted modifications to model activations. However, prior studies primarily report results on a few models, leaving critical gaps in understanding the robustness of these methods. In this work, we systematically examine three prominent steering methods—DoLa, function vectors, and task vectors. In contrast to the original studies, which evaluated a handful of models, we test up to 36 models belonging to 14 families with sizes ranging from 1.5B to 70B parameters. Our experiments reveal substantial variability in the effectiveness of the steering approaches, with a large number of models showing no improvement and at times degradation in steering performance. Our analysis demonstrate fundamental flaws in the assumptions underlying these methods, challenging their reliability as scalable steering solutions¹.

1 Introduction

Building on a growing array of interpretability tools (Zhao et al., 2023; Ferrando et al., 2024; Rai et al., 2024), steering methods for language models have gained popularity as a way to modify model behavior with specific objectives at inference time (Vig et al., 2020; Meng et al., 2022; Li et al., 2023). They have been applied to steer models toward desirable outputs, such as improved factuality (Chaudhary and Geiger, 2024; Zhao et al., 2024) or away from undesirable traits (O’Brien et al., 2024; Farrell et al., 2024). These techniques are appealing because they require little data compared to fine-tuning and do not require changes to model parameters (Subramani et al., 2022; Turner et al., 2023; Rinsky et al., 2024). However, they are still poorly understood and face significant challenges that hinder their practical applicability.

¹full code, data, and results can be found on our [GitHub](#)

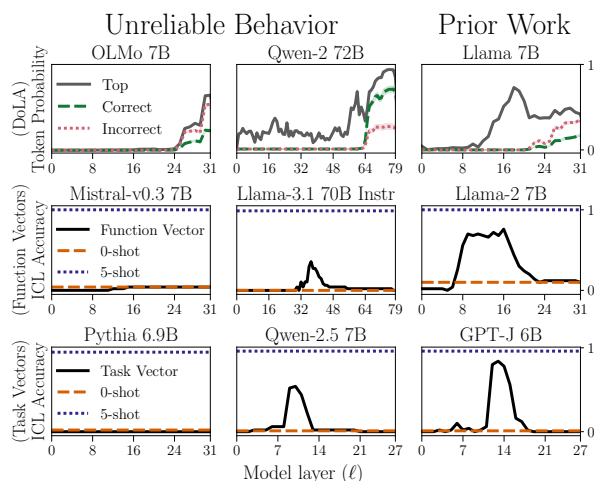


Figure 1: We study the generalization of various LM steering methods to previously unstudied models and find high variance in steering performance. (Top plot) Building on logit lens §3, DoLa contrasts token probabilities across layers to discover factual answers. We show that models do not have similar patterns, limiting the effectiveness of this method. (Middle and bottom plots) Function vectors and task vectors are two methods for steering based on activation patching §4. We show that activation patching results in highly variable performance across many model families and sizes.

The LM steering literature has accumulated a generalization blind spot. Most prior work reports results on only a select number of LMs (Geva et al., 2022; Gurnee et al., 2023; Zou et al., 2023; Chuang et al., 2024; Templeton et al., 2024; Huben et al., 2024), with growing evidence of unintended side effects. Recent work has shown that steering methods can be brittle (Tan et al., 2025) and harm general LM capabilities (Durmus et al., 2024).

Motivated by these studies, we adopt a critical perspective of two popular interpretability tools and the steering methods they inspired. Specifically, logit lens (nostalgebraist, 2020) and activating patching (Vig et al., 2020). We focus on three steering methods based on these tools: DoLa (Chuang et al., 2024) built on logit lens (§3), and

function vectors (Todd et al., 2024) and task vectors (Hendel et al., 2023) built using ideas from activation patching (§4). The original studies evaluated a small number of LMs; our primary goal is to quantify the generalization of these methods across different model families and sizes. Our experiments show a large variance in performance across 36 decoder-only transformer-based LMs from 14 model families with sizes ranging from 1.5B to 70B parameters. Several model families, even after significant hyperparameter tuning, show no improvement or even decline in relevant steering metrics. Our analyses reveal that these steering methods rely on flawed assumptions about the generalization of internal transformer mechanisms. We postulate many more hypotheses for such variance and provide recommendations for the future of evaluating steering methods for LMs in §5.

2 Related Work

Brittleness of Steering Methods Recent work has highlighted the brittleness of different steering methods providing motivation for this work. For example, sparse autoencoders (SAEs) aim to learn sparse, interpretable features from internal activations of LMs (Elhage et al., 2022). The features can be upweighted or downweighted to steer the models (Huben et al., 2024). Durmus et al. (2024) show that targeted steering often induces unpredictable behaviors harming the general capabilities of the model. Related to this work, Wu et al. (2025) show that SAEs are not effective in steering compared to simple baselines like prompting. Additionally, Tan et al. (2025) study steering vectors based on contrastive activation addition (CAA) (Rimsky et al., 2024). They focus on out-of-distribution generalization and show that some concepts in LMs are unsteerable. Most related to our work is Brumley et al. (2024), who report inconsistencies in function and in-context vector steering capabilities across different kinds of tasks. In contrast to our work, they experiment with only two mid-sized models. We test the generalization of steering methods across many model families and sizes with extensive hyperparameter search, and subsequently make a case against the underlying assumptions that steering vectors are built upon.

Brittleness of Model Editing Similar to steering methods, model editing methods do not require training to modify model behavior. Instead of updating model activations at inference time, model

editing aims to locate and update model weights responsible for a specific behavior, such as storing facts (Meng et al., 2022). Similar to our exploration of steering, prior work has shown that the brittleness and side effects of model editing can limit its practical usability. Li et al. (2024) demonstrate two of these unintended consequences, knowledge *conflicts* and *distortions*, in popular locate-then-edit knowledge editing tools (Mitchell et al., 2022; Meng et al., 2022, 2023). Yao et al. (2023); Huang et al. (2024) explore the generalization of knowledge editing and find that edits can struggle to maintain consistency across contexts, often hurting generalization.

3 Logit Lens

Introduced in nostalgebraist (2020), the logit lens provides insights into how LMs refine their prediction across layers (or sublayers). This approach has been applied to interpret activations at various stages (Geva et al., 2020), as well as to discover circuits (Lieberum et al., 2023; Wang et al., 2022). Specifically, the output of any model layer, h_ℓ , can be projected onto the vocabulary space to obtain logits by multiplying it by the unembedding matrix (W_U), $\text{LogitLens}(h_\ell) = h_\ell W_U$, optionally followed by softmax to convert it to a probability distribution over the vocabulary (also known as probits), $q_\ell(\cdot) = \text{softmax}(\text{LogitLens}(h_\ell))$. The tokens with the highest logit values or probabilities can be used to infer what information has been encoded in the hidden layer to the extent that the information is linearly decodable in the space defined by the model’s unembedding matrix. In addition to understanding model dynamics, logit lens has also been used to steer model behavior based on insights from said dynamics (Bhalla et al., 2024). For example, to reduce hallucinations (Gema et al., 2024; Chen et al., 2024; Jiang et al., 2024). In this section, we analyze DoLA (Chuang et al., 2024), one of the most influential works on logit-lens-based steering that focuses on improving factuality.

Central to this work is the hypothesis that neurons that store factual knowledge are distributed among the later layers of the model (Dai et al., 2022). During inference time, while these neurons contribute to an increase in the probability of factually correct outputs, this increase may not be sufficient to ensure that the probability of the correct output is the greatest among all possible outputs. Based on this hypothesis, rather than us-

ing the absolute probability of the following token, [Chuang et al. \(2024\)](#) compute the *relative change* in probability at the final layer compared to an earlier or “premature” layer. The layer ℓ farthest from the final layer L , in terms of Jensen-Shannon Divergence (JSD) between q_L and q_ℓ , is chosen as the premature layer.

More formally, for a model with L layers, the premature layer at inference step t is chosen as $P = \arg \max_{\ell \in \mathcal{B}} \text{JSD}(q_L(\cdot | x_{<t}) \| q_\ell(\cdot | x_{<t}))$. Here, \mathcal{B} refers to the set of all “candidate” layers (except the final). The candidate set, henceforth referred to as buckets, is a hyperparameter. Using P , the output probability is updated as

$$\hat{p}(x_t | x_{<t}) = \text{softmax}(\mathcal{F}(q_L, q_P)) \quad (1)$$

where

$$\mathcal{F}(q_L, q_P)_{x_t} = \begin{cases} \log \frac{q_L(x_t)}{q_P(x_t)} & \text{if } x_t \in \mathcal{V}_{\text{head}} \\ -\infty & \text{otherwise} \end{cases} \quad (2)$$

where $\mathcal{V}_{\text{head}}$ is the set of tokens x_t for which $q_P(x_t) \geq \alpha \max_w q_L(w)$. Here α is another hyperparameter. As α increases, a larger portion of tokens with low probabilities are set to have a probability of zero. We refer to the readers to [Chuang et al. \(2024\)](#) for addit details.

3.1 Experimental Setup

Tasks, datasets, and models Following the original paper, we evaluate DoLa on two multiple choice text completion tasks: TruthfulQA ([Lin et al., 2022](#))² and FACTOR ([Muhlgay et al., 2024](#)). TFQA measures the truthfulness of models in answering questions. It consists of questions spanning various domains to test models’ tendencies to generate false but human-plausible answers. FACTOR measures the factuality of a model. We experiment with the “News” subset. The input consists of a prefix followed by either a correct completion or one of multiple incorrect completions, where the probability of each completion is computed.

We evaluate with a comprehensive set of 10 models that span 7 model families and two scales. We experiment with Llama 1 at 7B ([Touvron et al., 2023a](#)), Llama 2 at 7B and 70B ([Touvron et al., 2023b](#)), Llama 3 at 8B and 70B ([Grattafiori et al., 2024](#)), Pythia at 6.9B ([Biderman et al., 2023](#)), Mistral v0.1 at 7B ([Jiang et al., 2023](#)), OLMo at 7B

²While there exists a generative version of TruthfulQA, it requires GPT3-based evaluation which is no longer offered by OpenAI APIs

([Groeneveld et al., 2024](#)), and Qwen 2 at 7B and 72B ([Yang et al., 2024](#)).

Hyperparameters DoLa uses two primary hyperparameters that may affect its final performance. The first is the choice of \mathcal{B} , the set of candidate layers from which the premature layer is chosen, also referred to as the bucket. The optimal bucket is typically chosen using a validation set. In our experiments, we report results over 4 buckets defined by a range of layers³. For the small models, our buckets are the bottom 50% of the layers (0-50%), middle 50% of the layers (25-75%), top 50% of the layers (50-100%), and all layers (except the final). For the large models, our buckets are the first (0-25%), second (25-50%), third (50-75%), and last (75-100%) quartiles of layers. For each bucket, following the original paper, we only consider even-numbered layers (for efficiency reasons).

The other hyperparameter is α , which determines $\mathcal{V}_{\text{head}}$ (see [Equation 2](#)). α adjusts the threshold in token probability of the mature layer, below which probabilities are set to zero. To avoid a computationally expensive grid search over all buckets and α values for each model, we first run a search over the buckets, with $\alpha = 0.1$, the value used in [Chuang et al. \(2024\)](#). Once we determine the best bucket for each model, we search over $\alpha \in \{0, 0.1, 0.25, 0.5, 0.75, 0.9\}$ values.

Evaluation For both tasks, we compute 6-shot performance for the baseline approach (without any steering) and DoLA. For TruthfulQA, we compute three metrics following prior work: MC1, MC2, and MC3. The task specifies different answer choices to the model depending on the metric to be computed. For MC1, only one of the options is the correct answer. MC1 measures the accuracy of a model’s greedy prediction using the probabilities of different answer choices. For MC2 and MC3, more than one answer may be correct. MC2 measures the normalized score assigned to the set of correct answers. MC3 defines the ranked-choice prediction accuracy—whether the set of correct answers is assigned a higher likelihood than the wrong ones. Regardless of the input, a 6-shot prompt, where each example consists of a question followed by one correct answer, is added as a prefix. For FACTOR, we compute accuracy (same as MC1).

We also note that [Chuang et al. \(2024\)](#) report these metrics by treating \mathcal{F} as log-probabilities

³There are in principle an exponential number of potential buckets, we focus on ranges of layers following prior work.

Model	MC1		MC2		MC3	
	Base	DoLa	Base	DoLa	Base	DoLa
LLama 7B*	0.26	0.32	0.41	0.64	0.19	0.32
Llama 7B	0.26	0.32	0.41	0.52	0.19	0.28
LLama 2 7B	0.29	0.29	0.43	0.44	0.21	0.21
Llama 3 8B	0.32	0.32	0.49	0.49	0.24	0.24
Pythia 6.9B	0.23	0.26	0.37	0.52	0.18	0.23
Mistralv0.1 7B	0.32	0.32	0.48	0.48	0.24	0.24
Qwen 2 7B	0.39	0.37	0.58	0.51	0.30	0.30
OLMo 7B	0.25	0.25	0.40	0.40	0.19	0.19
Llama 2 70B	0.37	0.34	0.54	0.54	0.27	0.27
Llama 3 70B	0.35	0.35	0.56	0.56	0.28	0.28
Qwen 2 72B	0.44	0.39	0.63	0.46	0.33	0.30

Table 1: Performance of DoLA for TruthfulQA with the best hyperparameter combination (chosen based on the highest MC1 value). The results for the full search can be found in Appendix A.1. * refers to the DoLa results reported by [Chuang et al. \(2024\)](#) using \mathcal{F} without applying a softmax.

(which they are not) without computing a softmax and using \hat{p} (using Equation 1). Since all these metrics are computed by aggregating \hat{p} over multiple tokens, directly aggregating logits as output by \mathcal{F} can lead to a length bias where, depending on the signs of logits, a longer output may unfairly be rewarded or punished (more details are provided in Appendix A.1). We report all our results using \hat{p} .

3.2 Results and Analysis

We summarize our main results in Table 1 and Table 2 for TFQA and FACTOR respectively. Each row corresponds to the best result chosen based on MC1 for TFQA and accuracy for FACTOR. We provide complete results for all buckets and α values in Appendix A.1.

[Chuang et al. \(2024\)](#) report results only on Llama 1 family of models. We note that, after correctly computing the metrics (i.e. using \hat{p} instead of \mathcal{F}), the performance improvement offered by DoLa over the baseline approach is *much* lower, although still substantial (more details in Table 7 in Appendix A.1). However, for every other model we evaluate, with the exception of Pythia, the improvements afforded by DoLa in most metrics is not significant, especially compared to the gains of Llama 1. Additionally, in models such as Qwen 2 7B and Llama 2 70B, we see a slight deterioration in some metrics compared to the baseline. The hyperparameters (buckets and α) have little effect on this trend (see Tables 4, 5, 6, 8, 9, and 10 in Appendix A.1). Our results are consistent with several follow-up works that have attempted to apply this approach on top of select fine-tuned models and achieved

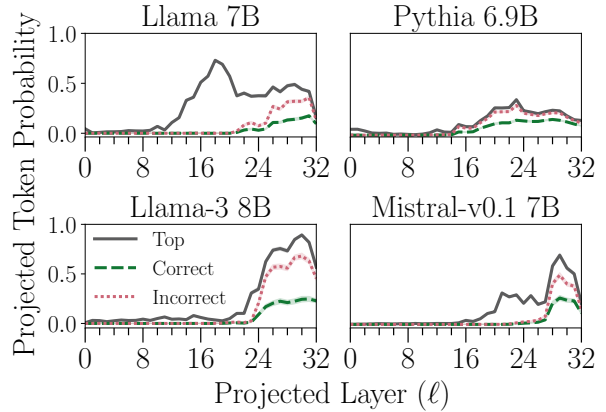


Figure 2: Projected token probabilities from hidden states at each layer of 4 selected LMs on the TruthfulQA dataset (the remaining results can be found in Appendix A.2). The correct and incorrect token probabilities begin spiking at the same layer, which suggests that a contrast with early layers would be relatively uninformative.

little or no improvements ([Tian et al., 2023](#); [Pan et al., 2024](#)).

We hypothesize that the failure of this approach stems from the flawed assumption that “factual knowledge of an LM evolves gradually across layers” ([Meng et al., 2022](#); [Geva et al., 2023](#)), and that contrasting the final with earlier layers offers a meaningful signal. [Wiegrefe et al. \(2025\)](#) show that in a multiple choice setting, there exist model-specific layers that start the “promotion” of the answer tokens. Inspired by their methodology, we measure the probability of 3 token types across each layer using TruthfulQA: the correct and incorrect final answers, and the token with the highest logit value. Figure 2 aligns with the results from [Wiegrefe et al. \(2025\)](#), showing that the correct and incorrect tokens have low probabilities before spiking at the same layer. This suggests that a contrast with early layers is relatively uninformative, especially when comparing probabilities of correct and incorrect tokens. This result contrasts with DoLa’s motivation, which states that the promotion of early tokens should be discounted to encourage more of what the model prefers in later layers.

4 Activation Patching

Activation patching is the technique of replacing or updating internal activations of a neural network with another vector to modify a specific behavior of the model ([Vig et al., 2020](#); [Geiger et al., 2020, 2021](#); [Meng et al., 2022](#); [Chan et al., 2022](#); [Wang et al., 2023](#)). It is typically used to interpret mod-

	LLama 7B*	Llama 7B	Llama 2 7B	Llama 3 8B	Pythia 6.9B	Mistralv0.1	Qwen 2 7B	OLMo	Llama 2 70B	Llama 3 70B	Qwen 2 72B
Base	58	58	72	76	51	76	69	67	83	85	82
DoLa	62	63	73	76	48	76	68	67	82	86	82

Table 2: Results (% accuracy) measuring the performance of DoLA for FACTOR with the best hyperparameter combination (results for the full search can be found in Appendix A.1). * refers to the DoLa results reported by [Chuang et al. \(2024\)](#) using \mathcal{F} without applying a softmax.

els by isolating task-specific circuits and observing relevant changes in model outputs ([Hanna et al., 2023](#); [Heimersheim and Janiak, 2023](#); [Conmy et al., 2023](#)), but it has also been applied to model steering. Previous work has explored several ways to create *steering* vectors ([Nanda et al., 2023](#); [Zou et al., 2023](#); [Tigges et al., 2023](#); [Turner et al., 2024](#)). At inference time, model activations at different layers are added to or replaced by these vectors to change the properties of the generated text.

In this section, we examine two related activation patching techniques which have been used to study in-context learning (ICL) in transformers: **function vectors** (FVs) ([Todd et al., 2024](#)) and **task vectors** (TVs) ([Hendel et al., 2023](#)). For a task t , test example \tilde{x}_i , set of exemplars $S_t^K = \{(x_1, y_1), \dots, (x_K, y_K)\}$ demonstrating the task, and prompt $p_i^{t,K} = [S_t^K, \tilde{x}_i] \in P_t$, both techniques aim to compress the demonstrations into a steering vector $v_t(S)$. When patched into an LM, this vector, in theory, should reproduce the performance of ICL in a zero-shot setup.

To create FVs, [Todd et al. \(2024\)](#) rely on the localization hypothesis ([Olsson et al., 2022](#)), suggesting the that “presence of a handful of attention heads in an LM can mediate many ICL tasks.” The FV is computed in two steps. First, the average activation of a head over all prompts from a task dataset is defined as $\bar{a}_{\ell_j}^t = \frac{1}{|P_t|} \sum_{p_i^t \in P_t} a_{\ell_j}(p_i^{t,10})$ where a_{ℓ_j} is a self-attention head at layer ℓ . It is used to construct the function vector

$$v_t^{FV} = \sum_{a_{\ell_j} \in \mathcal{A}_n} \bar{a}_{\ell_j}^t \quad (3)$$

where \mathcal{A}_n is a subset containing n attention heads ranked by a causal mediation analysis ([Pearl, 2009](#)) and n is a hyperparameter. At a high level, given ICL prompts with random, uninformative pairs, the causal indirect effect of a head is measured as its ability to recover the correct answer when patched with its mean head activation $\bar{a}_{\ell_j}^t$ on the desired task. We report the results for the casual indirect effect of each head in Appendix B.2. For more details on the formulation of FVs, please refer to

[Todd et al. \(2024\)](#).

Task vectors on the other hand do not rely on localization or causal mediation analysis. They directly compress a task into the activation space of a transformer model

$$v_t^{TV} = h_\ell = f_\ell(p_i^{t,5}) \quad (4)$$

where f_ℓ is a transformer model producing the hidden state at layer ℓ . Additional heuristics of TV implementation can be found at [Hendel et al. \(2023\)](#).

Using unseen prompts, the steering vectors $v_t(S)$ can be applied to the hidden states of any layer ℓ as

$$h_\ell \leftarrow \alpha h_\ell + \lambda v_t \quad (5)$$

Here λ is typically set to 1 (we also experiment with other values for FVs). α is set to 1 for FVs and 0 for TVs.

4.1 Experimental Setup

Tasks and datasets We source ICL word-pair samples from [Hendel et al. \(2023\)](#); [Todd et al. \(2024\)](#) and select 11 representative tasks from linguistic, factual knowledge, and translation tasks. This includes generating the antonym of English words, the past tense of a present-tense verb, the capital of a country, and various translations to and from English (eng to [lang], [lang] to eng), using French, Spanish, German, and Italian. Full dataset details can be found in Appendix B.1.

Models We use 36 models ranging from sizes 1.5B to 70B across the GPT-J ([Wang and Komatsuzaki, 2021](#)), Pythia (P) ([Biderman et al., 2023](#)), Llama 1 (L) ([Touvron et al., 2023a](#)), Llama 2 (L2) ([Touvron et al., 2023b](#)), Llama 3.1 (L3.1), Llama 3.2 (L3.2) ([Grattafiori et al., 2024](#)), Mistral v0.1 (M1) and v0.3 (M3) ([Jiang et al., 2023](#)), Gemma 2 (G2) ([Team et al., 2024](#)), Qwen 2 (Q2) ([Yang et al., 2024](#)), Qwen2.5 (Q2.5) ([Qwen et al., 2025](#)), OLMo (O) ([Groeneveld et al., 2024](#)), OLMo 2 (O2) ([OLMo et al., 2025](#)), Amber ([Liu et al., 2023](#)), and Falcon 3 ([Team, 2024](#)) families.⁴ We use only open-

⁴Due to changes to caching and tokenization across HuggingFace versions, we do not replicate Gemma 2 and OLMo

source models, as the experiments require access to model internals.

Hyperparameters FVs and TVs both have what could be considered a hyperparameter; the transformer layer ℓ at which the steering vector is patched. During inference, we apply the steering vector to only a single layer and search across all layers in the model. Besides ℓ , TVs do not have any additional hyperparameters.

Todd et al. (2024) only consider one hyperparameter for FVs— \mathcal{A}_n , the (number of) top attention heads to use when constructing the FV (as shown in Equation 3). While they demonstrate that performance generally saturates in GPT-J after 10 heads, our exploration indicates that other models and tasks may respond differently. We experiment with $n \in \{2, 16, 32, 64, 128, 256, 512, 1024\}$.

We also introduce λ (see Equation 5), a strength multiplier on the function vector. v_t^{FV} is created with a subset of attention heads and can have a low norm compared to h_ℓ ; therefore, we explore $\lambda \in \{0.5, 1, 2, 4, 8, 16, 32, 64\}$.

In addition to the best performance we obtain with hyperparameter search, we also report FV’s performance on Todd et al. (2024)’s default settings. We use $n \in \{2, 16\}$ for small models ($\leq 14\text{B}$) and $n \in \{2, 64\}$ for large models ($> 14\text{B}$). We also set a default $\lambda \in \{1\}$, consistent with both task and function vectors.

Evaluation For each task, we use a test set of 50 word pairs⁵ in FV, and the default number of word pairs using the implementation from Hendel et al. (2023) for TV. For each of our 11 ICL tasks, we create prompts with K exemplars and use the highest probability token as the prediction following prior work. While the FV is created using $K \in \{10\}$, we record baseline performance using $K \in \{0, 5\}$ to be consistent with the defaults from TV implementation. In our tasks, 10-shot performance tends to marginally improve over 5-shot, thus we underestimate brittleness in FVs (see Figure 11 in Appendix B.3). To determine steering vector efficacy, we apply activation patching at layer ℓ with $K \in \{0\}$ and record the corresponding accuracy.

² in task vectors. In function vectors, these models follow similar trends as others. Additional results are provided in Appendix B.4

⁵Considering computational budget and extensive hyperparameter search, we use 50 test samples as our hypotheses are only afflicted by large changes in accuracy.

5-shot Perf.	.50	.75	.90	1.00
FV Default Param	47%	37%	20%	12%
FV Param Search	76%	68%	52%	28%
Task Vector	69%	54%	35%	16%

Table 3: The percent of model-task combinations which surpass a quantile of their respective 5-shot performance using the best combination of hyperparameters: $\{\ell, \lambda, \mathcal{A}_n\}$ for FV and $\{\ell\}$ for TV.

Due to limited space, we report aggregated results across our three hyperparameters in the main paper (with detailed results in Appendix B.4). We accept \mathcal{A}_n and λ to be highly variable across models and thus always take the maximum score across those hyperparameters. Function vectors and task vectors are reported to work over clusters of early/middle layers; therefore, we also report both a peak and average **performance recovery metric** for layers. In general, we report peak (max) and average (mean) performance recovery as their respective aggregated steering vector performance across layers. We normalize these numbers by the respective model’s 5-shot performance. The divisor is intended to remove noise in ICL performance. Exact formulas are defined in Appendix B.3. As a final aggregation, we also report the frequency with which model-task combinations surpass a peak performance recovery at different quantiles of their 5-shot performance.

4.2 Results and Analysis

We begin by optimistically quantifying the brittleness of function vectors and task vectors, summarized in Table 3 and Figure 3. We analyze the interaction between hyperparameters and activation patching, summarized in Figures 4 and 5.

Neither FVs nor TVs are generalizable As shown in Table 3, even allowing for noise, FVs with default parameters recover 5-shot performance in only 20% of model-task combinations, and 52% with parameter search across \mathcal{A}_n and λ ; TVs perform poorly as well at 35%. Even with the best-performing tool, FVs with full hyperparameter search, only 76% of model-task combinations reach 50% of 5-shot performance.

Additionally, aggregate results from Figure 3 demonstrate the low performance and high variability across models and tasks. Function vectors have nearly no pattern in their efficacy, besides eng to [lang] having poor steerability in many models.

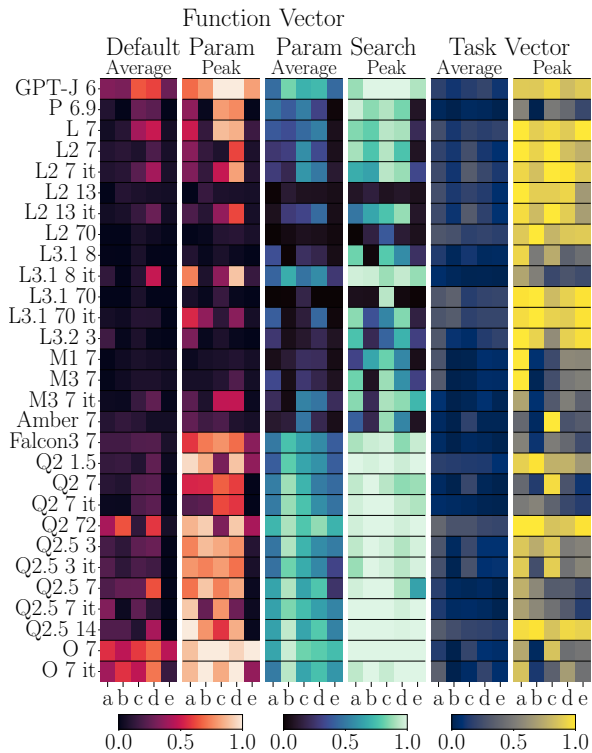


Figure 3: Performance recovery with different activation patching methods across tasks and models. There is large variance in performance across tasks, models, and tools. Tasks: a) Antonym, b) Present-Past, c) Country-Capital, d) [lang] to eng, and e) eng to [lang].

We find TV to be more stable across larger scales. All of the seven larger (≥ 13 B) models we tested worked decently with TV.

We further compare base and post-trained models. Using FV, post-trained models perform better on average, especially when the base model performs poorly as demonstrated in Figures 3 and 5. However, with TVs, we observe a decline in performance with post-trained models (see Figure 16 in Appendix B.4). The uneven impact of post-training on function and task vectors provides further evidence for the fragility of steering based on activation patching.

Impact of function vector strength (λ) and layer (ℓ) We address the assumption that the FV should have a norm strong enough to impact the residual stream with λ . We generally note that models prefer some range of λ (some lower and some higher), as shown in Figure 14 in Appendix B.4, although there is still moderate variability across tasks. Some models work with the default $\lambda \in \{1\}$, but varying this hyperparameter helped to recover performance in many models.

Both TVs and FVs assume that there is at least

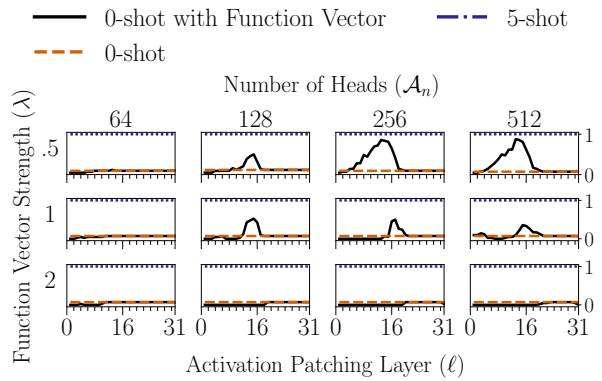


Figure 4: A subspace in the hyperparameter search across $(\ell, \lambda, \mathcal{A}_n)$ for Mistral-v0.3 7B on the Country-Capital task. The FV does not become effective until 128 (10%) heads, and it continues to grow in performance until 512 (50%) heads. This provides some evidence against localization for this model and task, as many heads are required before performance emerges.

one layer ℓ in which the model will respond to the steering vector. However, not only are some models ineffective at all layers, the range of best patching layers varies widely across models and tasks. Figure 15 in Appendix B.4 provides a comprehensive summary of the preferred patching layer for every model and task. Additionally, the high peak but low average recovery for task vectors visible in Figure 3 suggests a strong dependence on the choice of ℓ for task vectors.

Localization hypothesis does not always hold for FVs FVs rely on the assumption that information required for ICL is stored and activated within a small subset of heads, which we test with \mathcal{A}_n . We find that on some tasks, certain models require many heads in their FV before recovering performance, with one example displayed in Figure 4.

The translation to English, linguistic, and factual knowledge tasks show some evidence for localization. They are most effective when n is relatively small, as shown in Figure 5. However, when translating from English to another language, the more localized $n \in \{2, 16, 32\}$ have lower performance, and performance improves with $n \in \{64, 128\}$.

Whether a model is post-trained also has an interaction with \mathcal{A}_n . Figure 5 highlights that post-trained models outperform base models at higher n . This result could suggest that the instruction-tuning process produces models with more distributed access to information.

We summarize the variable levels of steerability with differing number of attention heads n across

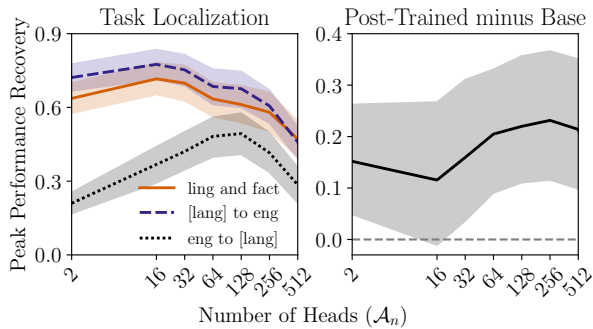


Figure 5: (Left) Some tasks work well with few heads, while translating from English requires more heads to be effective. (Right) Post-trained models outperform their base counterparts on average, especially when more heads are used to construct the function vector. However, there is significant variance when averaging across models and tasks.

all models and tasks in Figure 13 in Appendix B.4.

5 Discussion

Our findings show a clear breakdown in steering tools across models and confusing differences in dynamics between seemingly similar models. In this section, we enumerate several potential hypotheses that may explain these differences. While we conduct some experiments to verify them, our limited resources preclude us from making a conclusive case for any of them.

Model pretraining Recent models have adopted a two-stage pretraining process (Blakeney et al., 2024), also referred to as mid-training (OLMo et al., 2025), involving a second round of training on higher quality raw corpus to increase the context window (Gao et al., 2024) or improve later instruction following abilities for tasks (OLMo et al., 2024). We hypothesize that the second stage, which is performed with a much lower learning rate, may result in late-spiking logit lens dynamics, and hence poor results for techniques such as DoLa, as shown in Figure 2.

To test this hypothesis, we use OLMo, which at the time of running these experiments was the only model with publicly accessible weights at different checkpoints. However, we find no significant differences in dynamics for checkpoints at the end of stage one and two. In fact, we find OLMo starts showing late spiking dynamics very early in the pre-training stage at 100B tokens, as shown in Figure 6 in Appendix A.2. However, we cannot conclusively refute this hypothesis with experiments with only one model.

Model architecture and optimization We investigated several architectural differences among LMs: number of layers and attention heads, number of key-value heads, vocabulary size, hidden dimension sizes, attention types, activation functions, and context length. Model differences are summarized in Table 11 and Table 12. Among these model differences, we find no discernible pattern of brittleness caused by any such changes. One exception is Gemma 2 9B which has been trained via knowledge distillation and shows vastly different dynamics than other models (Team et al., 2024). For example, in logit lens, Gemma2 9B has a top token probability of nearly 1.0 in early layers, demonstrated in Figure 7 in Appendix A.2. We acknowledge that all models we looked at are trained with different training datasets and optimization pipelines. With more resources, future work may conduct controlled experiments to ablate these differences.

Training data Finally, we speculate that the number of tokens, data quality, and style may play a role in model differences. For example, recent models disobey the Chinchilla scaling laws (Hoffmann et al., 2022) and saturate the models with many tokens. It is conceivable that the amount of training tokens supplied to the model plays a role in how interpretable their mechanics are.

6 Conclusion

We study the generalization of three popular steering methods: DoLa, function vectors, and task vectors. We find that all three techniques are extremely brittle across models and tasks. Our analysis reveals that the underlying assumptions upon which these methods are based are flawed. This work adds to the growing evidence of robustness challenges that surgical methods aiming to modify language model behavior face. A large amount of interpretability research from which such steering methods are derived continues to be shared via informal channels such as blog posts with seemingly exciting results but limited formal evaluation. As such, we implore future research in this direction to adopt a rigorous evaluation setup considering a wide array of models and tasks to test the reliability of steering approaches, as models with minor differences may have vastly different manifestations of key behavior.

Limitations

Non-exhaustive hyperparameter search While we performed an extensive hyperparameter search for all steering methods, we did not cover all possible combinations due to our limited computational budget. For example, for DoLa, there are a prohibitively large number of bucket combinations from which the premature layer can be selected, while we experiment with a smaller set. We did, however, find very similar results across all bucket combinations providing confidence that other combinations are unlikely to improve the results.

Non-exhaustive model and task selection We experimented with a wide array of popular open-source models, however, at the time of writing this paper, many newer models have been released that we have not tested which may reveal different results. Additionally, for activation patching, we choose a representative sample of tasks from prior work. It is possible that there are other tasks are more steerable than we report, and some worse. However, the main message of the paper remains the same: there exists high variability in the performance of the steering methods we tested.

Variance explanation We observe that the behaviors which steering methods build upon manifest in many forms and formulate multiple potential hypotheses in §5. Wherever possible, we conduct experiments to test the hypotheses (such as with OLMo checkpoints and Gemma 2 public training details). However, due to our limited resources, we are ultimately unable to run controlled experiments to conclusively verify or refute any of the proposed hypotheses.

Acknowledgments

The authors would like to thank Sarah Wiegrefe, Noah A. Smith, Vidhisha Balachandran, and Yulia Tsvetkov for helpful discussions and feedback on the early draft of this paper.

References

Usha Bhalla, Suraj Srinivas, Asma Ghandeharioun, and Himabindu Lakkaraju. 2024. Towards unifying interpretability and control: Evaluation via intervention. *arXiv preprint arXiv:2411.04430*.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai

Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. *Pythia: A suite for analyzing large language models across training and scaling*. *Preprint*, arXiv:2304.01373.

Cody Blakeney, Mansheej Paul, Brett W Larsen, Sean Owen, and Jonathan Frankle. 2024. Does your data spark joy? performance gains from domain upsampling at the end of training. *arXiv preprint arXiv:2406.03476*.

Madeline Brumley, Joe Kwon, David Krueger, Dmitrii Krasheninnikov, and Usman Anwar. 2024. *Comparing bottom-up and top-down steering approaches on in-context learning tasks*. *Preprint*, arXiv:2411.07213.

Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldwosky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. 2022. *Causal scrubbing, a method for rigorously testing interpretability hypotheses*. *AI Alignment Forum*.

Maheep Chaudhary and Atticus Geiger. 2024. Evaluating open-source sparse autoencoders on disentangling factual knowledge in gpt-2 small. *arXiv preprint arXiv:2409.04478*.

Dingwei Chen, Feiteng Fang, Shiwen Ni, Feng Liang, Ruifeng Xu, Min Yang, and Chengming Li. 2024. *Lower layer matters: Alleviating hallucination via multi-layer fusion contrastive decoding with truthfulness refocused*. *ArXiv*, abs/2408.08769.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. *Dola: Decoding by contrasting layers improves factuality in large language models*. In *The Twelfth International Conference on Learning Representations*.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. *Towards automated circuit discovery for mechanistic interpretability*. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. *Knowledge neurons in pretrained transformers*. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Esin Durmus, Alex Tamkin, Jack Clark, Jerry Wei, Jonathan Marcus, Joshua Batson, Kunal Handa, Liane Lovitt, Meg Tong, Miles McCain, Oliver Rausch, Saffron Huang, Sam Bowman, Stuart Ritchie, Tom Henighan, and Deep Ganguli. 2024. *Evaluating feature steering: A case study in mitigating social biases*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec,

- Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. [Toy models of superposition](#). *Preprint*, arXiv:2209.10652.
- Eoin Farrell, Yeu-Tong Lau, and Arthur Conmy. 2024. Applying sparse autoencoders to unlearn knowledge in language models. *arXiv preprint arXiv:2410.19278*.
- Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. 2024. [A primer on the inner workings of transformer-based language models](#). *Preprint*, arXiv:2405.00208.
- Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2024. How to train long-context language models (effectively). *arXiv preprint arXiv:2410.02660*.
- Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586.
- Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. *arXiv preprint arXiv:2004.14623*.
- Aryo Pradipta Gema, Chen Jin, Ahmed Abdulaal, Tom Diethe, Philip Teare, Beatrice Alex, Pasquale Minervini, and Amrutha Saseendran. 2024. [Decore: Decoding by contrasting retrieval heads to mitigate hallucinations](#). *ArXiv*, abs/2410.18860.
- Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. [Dissecting recall of factual associations in auto-regressive language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.
- Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. [Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. 2024. [Olmo: Accelerating the science of language models](#). *Preprint*, arXiv:2402.00838.
- Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. [Finding neurons in a haystack: Case studies with sparse probing](#). *Transactions on Machine Learning Research*.
- Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. [How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Stefan Heimersheim and Jett Janiak. 2023. [A circuit for python docstrings in a 4-layer attention-only transformer](#). *AI Alignment Forum*.
- Roe Hendel, Mor Geva, and Amir Globerson. 2023. [In-context learning creates task vectors](#). In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. 2022. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA. Curran Associates Inc.
- Baixiang Huang, Canyu Chen, Xiong Xiao Xu, Ali Payani, and Kai Shu. 2024. [Can knowledge editing really correct hallucinations?](#) *Preprint*, arXiv:2410.16251.
- Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. [Sparse autoencoders find highly interpretable features in language models](#). In *The Twelfth International Conference on Learning Representations*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth e Lacroix,

- and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Zhangqi Jiang, Junkai Chen, Beier Zhu, Tingjin Luo, Yankun Shen, and Xu Yang. 2024. Devils in middle layers of large vision-language models: Interpreting, detecting and mitigating object hallucinations via attention lens. *arXiv preprint arXiv:2411.16724*.
- Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. [Inference-time intervention: Eliciting truthful answers from a language model](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. 2024. [Unveiling the pitfalls of knowledge editing for large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. 2023. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [Truthfulqa: Measuring how models mimic human falsehoods](#). *Preprint*, arXiv:2109.07958.
- Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Yonghao Zhuang, Guowei He, Haonan Li, Fajri Koto, Liping Tang, Nikhil Rangan, Zhiqiang Shen, Xuguang Ren, Roberto Iriondo, Cun Mu, Zhiting Hu, Mark Schulze, Preslav Nakov, Tim Baldwin, and Eric P. Xing. 2023. [Llm360: Towards fully transparent open-source llms](#). *Preprint*, arXiv:2312.06550.
- Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. [Locating and editing factual associations in GPT](#). In *Advances in Neural Information Processing Systems*.
- Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. [Mass-editing memory in a transformer](#). In *The Eleventh International Conference on Learning Representations*.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022. [Fast model editing at scale](#). *Preprint*, arXiv:2110.11309.
- Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2024. [Generating benchmarks for factuality evaluation of language models](#). *Preprint*, arXiv:2307.06908.
- Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. [Emergent linear representations in world models of self-supervised sequence models](#). In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore. Association for Computational Linguistics.
- nostalgebraist. 2020. interpreting gpt: the logit lens. *lesswrong*.
- Kyle O’Brien, David Majercak, Xavier Fernandes, Richard Edgar, Jingya Chen, Harsha Nori, Dean Carignan, Eric Horvitz, and Forough Poursabzi-Sangde. 2024. Steering language model refusal with sparse autoencoders. *arXiv preprint arXiv:2411.11296*.
- Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. [2 olmo 2 furious](#). *Preprint*, arXiv:2501.00656.
- Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2024. [2 olmo 2 furious](#). *arXiv preprint arXiv:2501.00656*.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. [In-context learning and induction heads](#). *Transformer Circuits Thread*.
- Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. 2024. Lisa: layer-wise importance sampling for memory-efficient large language model fine-tuning. *Advances in Neural Information Processing Systems*, 37:57018–57049.
- Judea Pearl. 2009. *Causality*, 2 edition. Cambridge University Press.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang,

- Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. 2024. [A practical review of mechanistic interpretability for transformer-based language models](#). *Preprint*, arXiv:2407.02646.
- Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. [Steering llama 2 via contrastive activation addition](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.
- Nishant Subramani, Nivedita Suresh, and Matthew Peters. 2022. [Extracting latent steering vectors from pretrained language models](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581, Dublin, Ireland. Association for Computational Linguistics.
- Daniel Tan, David Chanin, Aengus Lynch, Dimitrios Kanoulas, Brooks Paige, Adria Garriga-Alonso, and Robert Kirk. 2025. [Analyzing the generalization and reliability of steering vectors](#). *Preprint*, arXiv:2407.12404.
- Falcon-LLM Team. 2024. [The falcon 3 family of open models](#).
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. [Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet](#). *Transformer Circuits Thread*.
- Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2023. Fine-tuning language models for factuality. *arXiv preprint arXiv:2311.08401*.
- Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. [Linear representations of sentiment in large language models](#). *Preprint*, arXiv:2310.15154.
- Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2024. [Function vectors in large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. [Llama: Open and efficient foundation language models](#). *Preprint*, arXiv:2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Ulisse Mini, and Monte MacDiarmid. 2024. [Activation addition: Steering language models without optimization](#). Workingpaper, arXiv.org, United Kingdom.
- Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Activation addition: Steering language models without optimization. *arXiv e-prints*, pages arXiv–2308.
- Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural nlp: The case of gender bias. *arXiv preprint arXiv:2004.12265*.
- Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. <https://github.com/kingoflolz/mesh-transformer-jax>.
- Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect ob-

- ject identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*.
- Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. [Interpretability in the wild: a circuit for indirect object identification in GPT-2 small](#). In *The Eleventh International Conference on Learning Representations*.
- Sarah Wiegrefe, Oyvind Tafjord, Yonatan Belinkov, Hannaneh Hajishirzi, and Ashish Sabharwal. 2025. [Answer, assemble, ace: Understanding how LMs answer multiple choice questions](#). In *The Thirteenth International Conference on Learning Representations*.
- Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. 2025. [Axbench: Steering llms? even simple baselines outperform sparse autoencoders](#). *Preprint*, arXiv:2501.17148.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. [Qwen2 technical report](#). *Preprint*, arXiv:2407.10671.
- Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. [Editing large language models: Problems, methods, and opportunities](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240, Singapore. Association for Computational Linguistics.
- Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2023. [Explainability for large language models: A survey](#). *Preprint*, arXiv:2309.01029.
- Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. 2024. [Steering knowledge selection behaviours in llms via sae-based representation engineering](#). *arXiv preprint arXiv:2410.15999*.
- 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. 2023. [Representation engineering: A top-down approach to ai transparency](#). *Preprint*, arXiv:2310.01405.

A Additional Details and Results for Logit Lens

In this section, we provide additional experimental details, results, and analysis to complement §3.

A.1 Results

Length bias in original DoLa results In Table 1 in §3.2, we discuss that [Chuang et al. \(2024\)](#) uses \mathcal{F} (Equation 2) to compute the reported metrics treating them as log-probabilities (when, in fact, they are similar to logits). Using \mathcal{F} gives the illusion that DoLA improves on the baseline. We find that this happens because \mathcal{F} results in logit values that are all > 0 . Hence, an option which is longer in length (in terms of number of tokens) gets a higher score as the logits add up. This length bias leads to several instances in TruthfulQA where the right answer is the longest being predicted correctly.

We show that without determining any premature layers, we can artificially increase or decrease TruthfulQA performance by adding or subtracting a constant value from the logits at the final layer and using the modified logits for the prediction without applying a softmax. To introduce this bias, we first subtract the smallest logit value from all logits to make them all positive. As shown in Table 7, adding a positive constant leads to higher MC1, MC2, and MC3 values and vice versa. We determined that setting this constant to 20 yielded the highest improvement.

Results for all hyperparameters Our preliminary analysis revealed that a lower alpha almost always led to better performance. Hence, with $\alpha = 0.1$ (also the value used by [Chuang et al. \(2024\)](#)), we report results for all buckets we experimented with in Tables 4, 5, 8 and 9. For the best bucket found for each model, we show results over different values of α in Tables 6 and 10. We report the best results (according to MC1) in the main paper (Table 1 and Table 2).

A.2 Analysis

We provide logit lens analysis plots for all models that are not covered in Figure 2 in Figure 7. To further analyze the disparity in the DoLA results

Model	0-50%	25-75%	50-100%	0-100%
MC1				
Llama 7B	0.2375	0.2436	0.2546	0.2436
Llama 2 7B	0.2558	0.2497	0.2534	0.2595
Llama 3 8B	0.2913	0.2876	0.2864	0.2913
Pythia 6.9B	0.2277	0.2179	0.2093	0.2301
Mistralv0.1 7B	0.2987	0.2974	0.2974	0.2987
Qwen 2 7B	0.3550	0.3733	0.3623	0.3623
OLMo 7B	0.2436	0.2399	0.2411	0.2436
MC2				
Llama 7B	0.4061	0.4166	0.4414	0.4234
Llama 2 7B	0.4340	0.4350	0.4408	0.4362
Llama 3 8B	0.5129	0.5128	0.5113	0.5129
Pythia 6.9B	0.4213	0.4213	0.5097	0.4417
Mistralv0.1 7B	0.4841	0.4842	0.4842	0.4841
Qwen 2 7B	0.4935	0.5073	0.4756	0.4756
OLMo 7B	0.4176	0.4205	0.4193	0.4176
MC3				
Llama 7B	0.1739	0.1797	0.1934	0.1824
Llama 2 7B	0.1935	0.1914	0.1929	0.1954
Llama 3 8B	0.2253	0.2236	0.2231	0.2253
Pythia 6.9B	0.1782	0.1733	0.1848	0.1796
Mistralv0.1 7B	0.2249	0.2245	0.2245	0.2249
Qwen 2 7B	0.2840	0.2988	0.2963	0.2963
OLMo 7B	0.1779	0.1781	0.1779	0.1779

Table 4: Bucket search for TFQA using $\alpha = 0.1$ (small models). Bolded are the best scores for MC1 in each model, which is later used for the alpha search. Continue reading in §3.2.

among different models and to better understand how tokens are promoted across layers, in addition to the logit lens analysis, we present an additional analysis strengthening our hypothesis that early layers in many models do not provide a meaningful signal to compute premature layers.

We define a new metric, which we call apathy. The primary motive of apathy is to quantify the extent to which transformer components are making changes to the residual stream. The hidden vector output at a layer, h_ℓ , is a function of the residual stream from the previous layer ($h_{\ell-1}$), and the output from multi-head attention (MHA) and MLP sub-layers. Formally, $h_\ell = h_{\ell-1} + h_{\text{MHA}} + h_{\text{MLP}}$, where, $h_{\text{MHA}} = \text{MHA}(\text{LayerNorm}(h_{\ell-1}))$ and $h_{\text{MLP}} = \text{MLP}(\text{LayerNorm}(h_{\ell-1} + h_{\text{MHA}}))$. We define apathy (A) to quantify the resiliency of the residual stream to the additions of MHA and MLP throughout the layers of a transformer:

$$A(r, h) = \left(1 + \frac{r \cdot h}{\|r\| \|h\|}\right) (\|r\| - \|h\|) \quad (6)$$

$A(h_{\ell-1}, h_{\text{MHA}})$ corresponds to the contribution

of MHA to the residual stream, and $A(h_{\ell-1} + h_{\text{MHA}}, h_{\text{MLP}})$ of MLP. Higher apathy means the residual stream is less affected by the outputs of MHA or MLP.

We plot the apathy metric in Figure 8. Some models persist in low apathy for longer, meaning the residual stream is being updated over more layers before it becomes relatively set in norm and direction. The layer at which apathy does begin to increase seems to match where the predicted token promotion occurs, suggesting some interplay between the two. These findings suggest variability in the that way models promote tokens in vocabulary space, making layer contrasting techniques that rely on logit lens (DoLa) unreliable.

Following the discussion in §5, we show the logit lens analysis for multiple pretraining stages of OLMo in Figure 6. We demonstrate the emergence of token probability spiking early in training (≤ 100 billion tokens). We also show the dynamics of Gemma2 9B in Figure 7, which has a large token probability with logit lens in its early layers.

Model	0-25%	25-50%	50-75%	75-100%
MC1				
Llama 2 70B	0.3317	0.3305	0.3341	0.3403
Llama 3 70B	0.2840	0.2803	0.2712	0.2778
Qwen 2 72B	0.3599	0.3550	0.38807	0.2791
MC2				
Llama 2 70B	0.5503	0.5559	0.5742	0.5368
Llama 3 70B	0.5106	0.5072	0.5064	0.5081
Qwen 2 72B	0.6000	0.5959	0.5968	0.5112
MC3				
Llama 2 70B	0.2319	0.2339	0.2474	0.2711
Llama 3 70B	0.2210	0.2201	0.2143	0.2246
Qwen 2 72B	0.2659	0.2731	0.2895	0.2740

Table 5: Bucket search for TFQA using $\alpha = 0.1$ (large models). Bolded are the best scores for MC1 in each model, which is later used for the alpha search. Continue reading in §3.2.

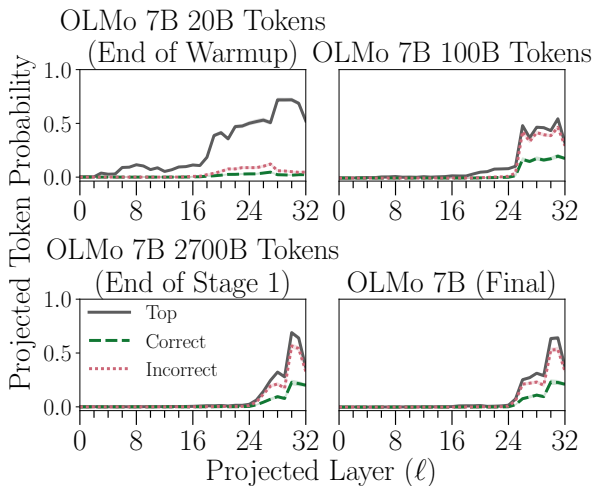


Figure 6: Projected token probabilities from hidden states at each layer of OLMo checkpoints on the TruthfulQA dataset. The correct and incorrect token probabilities begin spiking at the same layer after 100B tokens, suggesting this behavior emerges early in model training, and not with stag 2 of pretraining. Continue reading in §5.

B Additional Details and Results for Activation Patching

In this section, we provide additional experimental details, results, and analysis to complement §4.

B.1 Additional Details on Tasks and Datasets

As described in §4.1, we follow the code implementation which includes full datasets from Hendel et al. (2023); Todd et al. (2024). For FVs, we swap source and target languages to get the "[lang] to eng" task, and the equivalent task is available by default in TV.

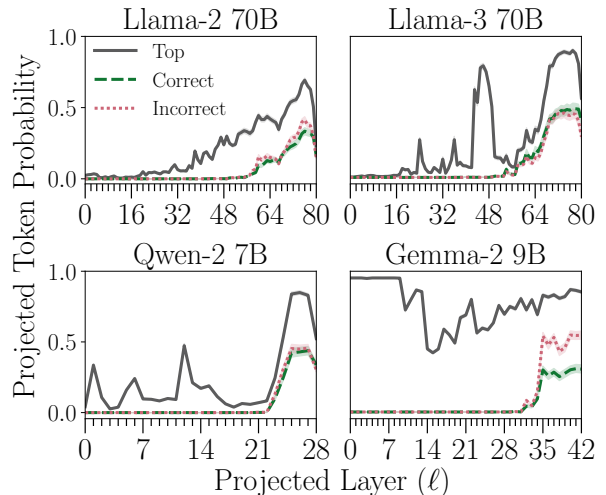


Figure 7: Projected token probabilities from hidden states at each layer of model not reported in the main paper on the TruthfulQA dataset. The correct and incorrect token probabilities begin spiking at the same layer, which suggests that a contrast with early layers would be relatively uninformative. Continue reading in §3.2.

B.2 Additional Details on Function Vector Formulation

As shown in Equation 3, finding heads which modulate a task is a key aspect of the casual mediation analysis in function vectors. We share two representative plots for the contribution of each attention head. GPT-J 6B in Figure 9 shows a few common heads modulate each of its tasks, consistent with Todd et al. (2024). Llama-3.1 8B in Figure 10 shows varied patterns in the heads which modulate each task, reflecting the variability we observe in our experimental results.

To see the causal indirect effect for every

model and task, please use the following link: <https://github.com/patqdasilva/steering-off-course>

B.3 Additional Details on Evaluation

As discussed in §4.1, we provide the full formulation for calculating peak and average performance recovery in FV and TV.

$$\text{peak} = \max_{\mathcal{A}_n, \lambda, \ell} \left(\frac{1}{|P_t|} \sum_{p_i^t \in P_t} \frac{f_{v_t}(p_i^{t,0}) = y_i}{f(p_i^{t,5}) = y_i} \right) \quad (7)$$

$$\text{avg} = \max_{\mathcal{A}_n, \lambda} \left(\frac{1}{L} \sum_{\ell \in L} \frac{1}{|P_t|} \sum_{p_i^t \in P_t} \frac{f_{v_t}(p_i^{t,0}) = y_i}{f(p_i^{t,5}) = y_i} \right) \quad (8)$$

Expanding on K shot evaluation in §4.1, we compare zero-shot, 5-shot and 10-shot performance for all models in Figure 11. We find that on average, 10-shot performance slightly beats out 5-shot. By creating FVs with 10-shot examples, and evaluating the recovery according to 5-shot performance, we slightly overestimate the effectiveness of FVs and underestimate its brittleness. That is, to say our findings hold well for both setups.

B.4 Additional Detail on Activation Patching Results

For a comprehensive display of figures from models not displayed in this paper, please follow <https://github.com/patqdasilva/steering-off-course>

Additional results with OLMo2 and Gemma2 using FVs Corresponding to the results from §4.1, we show additional results for OLMo2 and Gemma2 in Figure 12. OLMo2 shows decent FV performance similar to OLMo, whereas Gemma2 has varying performance which is especially poor in the 27B model.

Task Vectors: Post Training As discussed in §4.2, we find that unlike FVs, instruction tuned models using TVs are not more steerable than their base versions. We display this result across tasks in Figure 16.

Best Hyperparameters As discussed in §4.2, we show the hyperparameters with best performance across all models and tasks in Figure 13 (number of heads \mathcal{A}_n), Figure 14 (strength of the function vector λ), and Figure 15 (layer ℓ).

Function Vectors: Full Hyperparameter Search We show a contrast to the non-localized Mistral-v0.3 7B (Country-Capital task) from Figure 4.

GPT-J 6B shown in Figure 17, which was studied extensively by Todd et al. (2024), does show evidence for localization in the Country Capital task. It has 448 heads while nearly recovering its maximum performance with the FV built with only 2 (0.44%) heads. These two models are just one of several interactions of behavior differences between models that are difficult to parse. We note strong robustness across all hyperparameters. For every model and task tested, please visit <https://github.com/patqdasilva/steering-off-course>

Task Vectors: Full Hyperparameter Search

We provide representative examples of full parameter search as discussed in §4.2. We compare a model replicated from prior work in Figure 18 to a model previously unstudied in Figure 19. For every model and task tested, please visit <https://github.com/patqdasilva/steering-off-course>

C Compute Budget and Hardware

We use V100s with 16 and 32 GB for smaller models, and A100s and H100s with 80GB and 95GB for larger models. We consider small models to be those $< 13B$. The bulk of GPU hours come from the hyperparameter search with function vectors. For our small models, creating the FV takes 40 minutes, and search takes 1.5 hours. For our large models, creating the FV takes 4 hours and search takes 2.5 hours. With 27 smaller models and 9 larger models, across 9 tasks in FVs, this comes out to 524 hours on V100s and 526 hours on A100s and H100s. Total emissions are estimated to be 124.72 kgCO₂eq (Lacoste et al., 2019).

Model	0.0	0.1	0.25	0.5	0.75	0.9	Bucket used
MC1							
Llama 7B	0.3182	0.2546	0.2558	0.2301	0.2387	0.2472	50-100%
Llama 2 7B	0.2889	0.2595	0.2570	0.2595	0.2472	0.2399	0-100%
Llama 3 8B	0.3170	0.2913	0.2717	0.2448	0.2399	0.2387	0-100%
Pythia 6.9B	0.2607	0.2301	0.2007	0.2167	0.2154	0.2142	0-100%
Mistralv0.1 7B	0.3158	0.2987	0.2889	0.2852	0.2766	0.2729	0-100%
Qwen 2 7B	0.3133	0.3733	0.3599	0.3207	0.3170	0.3060	25-75%
OLMo 7B	0.2509	0.2436	0.2472	0.2399	0.2448	0.2411	0-100%
Llama 2 70B	0.2962	0.3403	0.3280	0.3366	0.3378	0.3354	75-100%
Llama 3 70B	0.3537	0.2840	0.2656	0.2338	0.2203	0.2203	0-25%
Qwen 2 72B	0.3329	0.3807	0.3819	0.3917	0.3782	0.3672	50-75%
MC2							
Llama 7B	0.5202	0.4414	0.4968	0.5066	0.5528	0.5562	50-100%
Llama 2 7B	0.4386	0.4362	0.4429	0.4758	0.5067	0.5126	0-100%
Llama 3 8B	0.4884	0.5129	0.5142	0.4998	0.4901	0.4882	0-100%
Pythia 6.9B	0.5253	0.4417	0.4682	0.4981	0.5043	0.5133	0-100%
Mistralv0.1 7B	0.4817	0.4841	0.5021	0.5171	0.5317	0.5329	0-100%
Qwen 2 7B	0.5420	0.5073	0.4744	0.4803	0.4902	0.4826	25-75%
OLMo 7B	0.3952	0.4176	0.4453	0.4764	0.5093	0.5074	0-100%
Llama 2 70B	0.5255	0.5368	0.5293	0.5467	0.5420	0.5426	75-100%
Llama 3 70B	0.5580	0.5106	0.4753	0.4859	0.4855	0.4744	0-25%
Qwen 2 72B	0.5771	0.5968	0.5099	0.4599	0.4486	0.4480	50-75%
MC3							
Llama 7B	0.2803	0.1934	0.1900	0.1745	0.1798	0.1847	50-100%
Llama 2 7B	0.2070	0.1954	0.1852	0.1844	0.1760	0.1693	0-100%
Llama 3 8B	0.2389	0.2253	0.2140	0.1881	0.1815	0.1786	0-100%
Pythia 6.9B	0.2448	0.1796	0.1671	0.1717	0.1763	0.1745	0-100%
Mistralv0.1 7B	0.2394	0.2249	0.2107	0.2100	0.2043	0.2033	0-100%
Qwen 2 7B	0.2911	0.2988	0.2850	0.2588	0.2553	0.2473	25-75%
OLMo 7B	0.1874	0.1779	0.1740	0.1729	0.1757	0.1791	0-100%
Llama 2 70B	0.2996	0.2711	0.2631	0.2528	0.2436	0.2447	75-100%
Llama 3 70B	0.2759	0.2210	0.2080	0.1824	0.1644	0.1571	0-25%
Qwen 2 72B	0.3122	0.2895	0.2870	0.2997	0.2962	0.2862	50-75%

Table 6: α search for TFQA, using the best bucket from Table 4 and Table 5. Bolded are the highest scores MC1 scores across alpha for each model. Continue reading in §3.2.

Model	Base	Best α	Best Bucket	DoLa	DoLa w/o softmax	Base w/ length bias (+20)
MC1						
Llama 7B	0.2570	0.0	50-100%	0.3182	0.3158	0.3182
Llama 2 7B	0.2852	0.0	0-100%	0.2889	0.3207	0.3354
Llama 3 8B	0.3195	0.0	0-100%	0.3170	0.3317	0.3427
Pythia 6.9B	0.2252	0.0	0-100%	0.2607	0.2962	0.3023
Mistralv0.1 7B	0.3158	0.0	0-100%	0.3158	0.3354	0.3623
Qwen 2 7B	0.3550	0.1	25-75%	0.3733	0.3439	0.3831
OLMo 7B	0.2509	0.0	0-100%	0.2509	0.3121	0.3439
Llama 2 70B	0.3500	0.1	75-100%	0.3403	0.2974	0.3341
Llama 3 70B	0.3684	0.0	0-25%	0.3537	0.3305	0.3513
Qwen 2 72B	0.4418	0.5	50-75%	0.3917	0.4015	0.4039
MC2						
Llama 7B	0.4055	0.0	50-100%	0.5202	0.6176	0.6151
Llama 2 7B	0.4340	0.0	0-100%	0.4386	0.6239	0.6182
Llama 3 8B	0.4884	0.0	0-100%	0.4884	0.6335	0.6489
Pythia 6.9B	0.3717	0.0	0-100%	0.5253	0.5968	0.5829
Mistralv0.1 7B	0.4814	0.0	0-100%	0.4817	0.6282	0.6433
Qwen 2 7B	0.4935	0.1	25-75%	0.5073	0.6474	0.6617
OLMo 7B	0.3957	0.0	0-100%	0.3952	0.6117	0.5187
Llama 2 70B	0.5240	0.1	75-100%	0.5368	0.4749	0.6182
Llama 3 70B	0.5817	0.0	0-25%	0.5580	0.6480	0.6476
Qwen 2 72B	0.6256	0.5	50-75%	0.4599	0.5704	0.6804
MC3						
Llama 7B	0.1924	0.0	50-100%	0.2803	0.3010	0.2995
Llama 2 7B	0.2076	0.0	0-100%	0.2070	0.3048	0.3064
Llama 3 8B	0.2392	0.0	0-100%	0.2389	0.3210	0.3276
Pythia 6.9B	0.1787	0.0	0-100%	0.2448	0.3054	0.2787
Mistralv0.1 7B	0.2389	0.0	0-100%	0.2394	0.3177	0.3225
Qwen 2 7B	0.2840	0.1	25-75%	0.2988	0.3297	0.3443
OLMo 7B	0.1878	0.0	0-100%	0.1874	0.3027	0.2588
Llama 2 70B	0.2526	0.1	75-100%	0.2711	0.2202	0.3063
Llama 3 70B	0.2917	0.0	0-25%	0.2759	0.3249	0.3286
Qwen 2 72B	0.3272	0.5	50-75%	0.2997	0.2974	0.3527

Table 7: TFQA best results for DoLa compared with baseline, DoLa w/o softmax, and baseline w/ length bias. Continue reading in §3.2.

Model	0-50%	25-75%	50-100%	0-100%
Llama 7B	0.6253	0.5927	0.5367	0.5784
Llama 2 7B	0.7281	0.7220	0.7169	0.7281
Llama 3 8B	0.7546	0.7454	0.7536	0.7536
Pythia 6.9B	0.4643	0.3996	0.3697	0.4054
Mistralv0.1 7B	0.7607	0.7597	0.7597	0.7607
Qwen 2 7B	0.6782	0.6049	0.5031	0.5061
OLMo 7B	0.6640	0.6650	0.6629	0.6640

Table 8: FACTOR bucket search (small models), using $\alpha = 0.1$. Continue reading in §3.2.

Model	0-25%	25-50%	50-75%	75-100%
Llama 2 70B	0.8214	0.8195	0.7635	0.7124
Llama 3 70B	0.8552	0.8562	0.8571	0.8514
Qwen 2 72B	0.8079	0.8012	0.7056	0.6187

Table 9: FACTOR bucket search (large models), using $\alpha = 0.1$. Continue reading in §3.2.

Model	Baseline	0.0	0.1	0.25	0.5	0.75	0.9	Bucket used
Llama 7B	0.5845	0.5876	0.6253	0.6059	0.5784	0.5815	0.5550	0-50%
Llama 2 7B	0.7220	0.7200	0.7281	0.7138	0.6772	0.6568	0.6568	0-100%
Llama 3 8B	0.7556	0.7587	0.7546	0.7230	0.7037	0.7006	0.6701	0-50%
Pythia 6.9B	0.5125	0.2828	0.4643	0.4797	0.4788	0.4537	0.4614	0-50%
Mistralv0.1 7B	0.7576	0.7617	0.7607	0.7301	0.6792	0.6833	0.6701	0-100%
Qwen 2 7B	0.6925	0.6843	0.6782	0.6680	0.6619	0.6191	0.6171	0-50%
OLMo 7B	0.6660	0.6619	0.6650	0.6415	0.6436	0.6242	0.6375	25-75%
Llama 2 70B	0.8320	0.8176	0.8214	0.7983	0.7905	0.7664	0.7317	0-25%
Llama 3 70B	0.8475	0.8494	0.8571	0.8127	0.7751	0.7847	0.7558	50-75%
Qwen 2 72B	0.8224	0.8243	0.8079	0.7770	0.7375	0.7355	0.7259	0-25%

Table 10: FACTOR α search using buckets from Table 8 and Table 9. Continue reading in §3.2.

Apathy and Token Probabilities on TruthfulQA

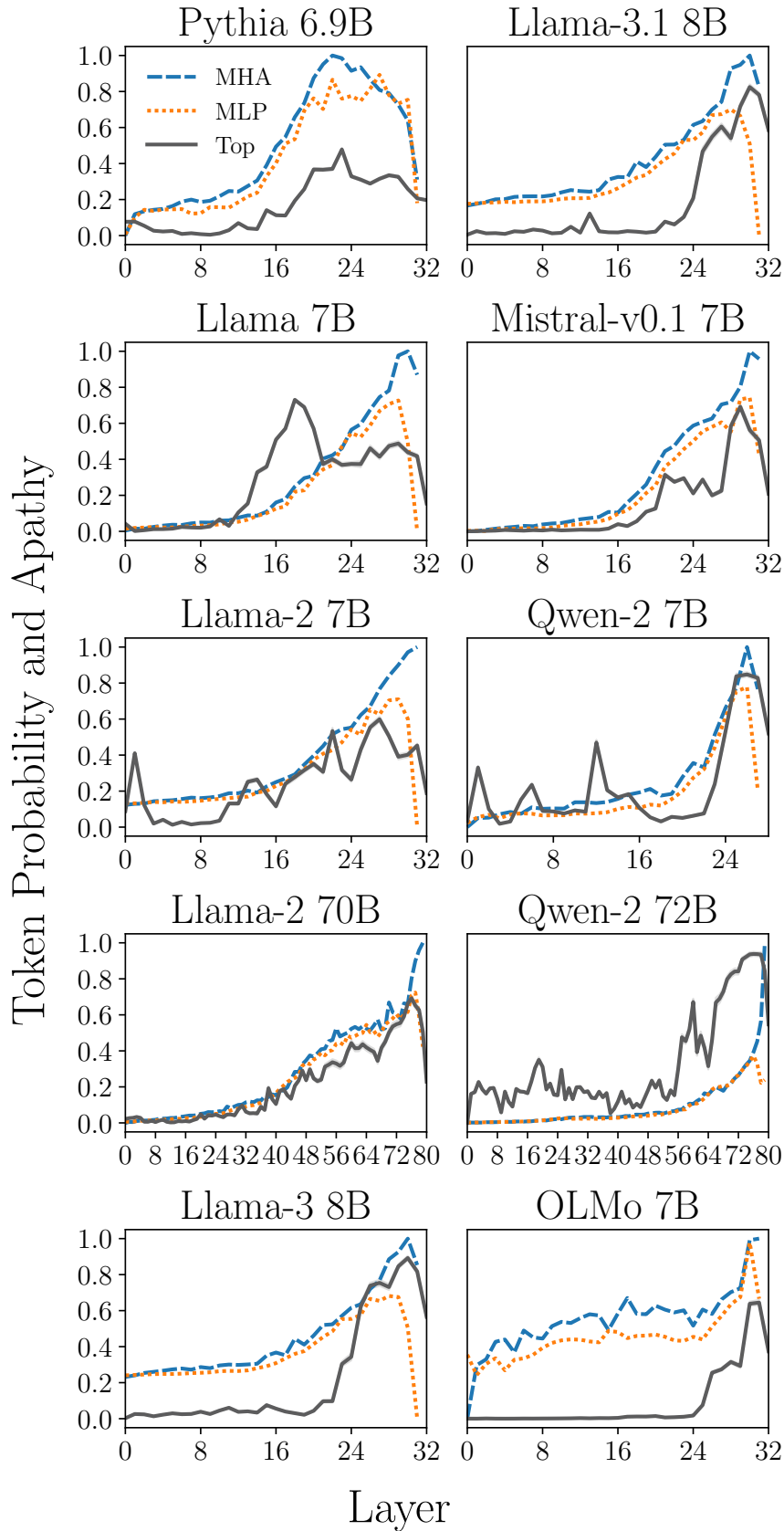


Figure 8: The residual stream starts to ignore MHA and MLP which seems to have some effect on when the top token starts to increase. Each model has its own dynamics, which may help to explain failures using tools which assume that a particular pattern is predicted. Continue reading in Appendix A.2.

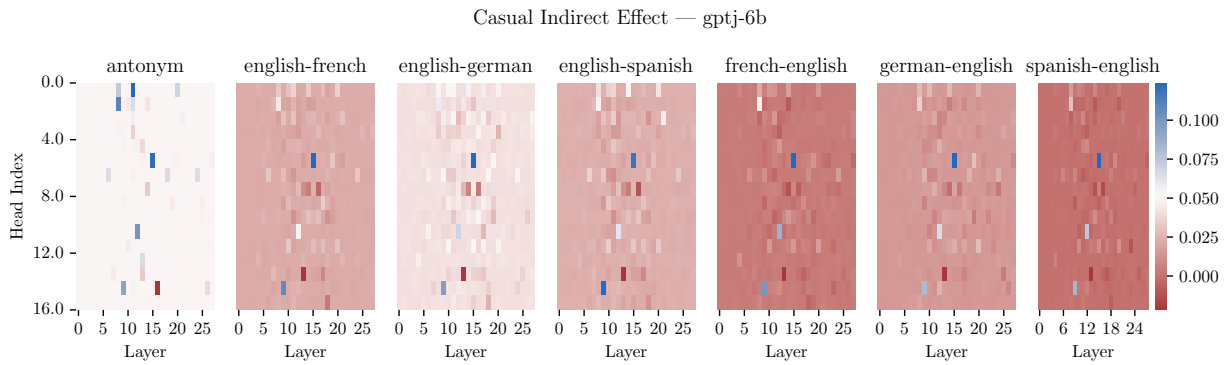


Figure 9: Causal indirect effect of attention heads in GPT-J 6B across all tasks. A few common heads modulate each of its tasks. Continue reading in Appendix B.2.

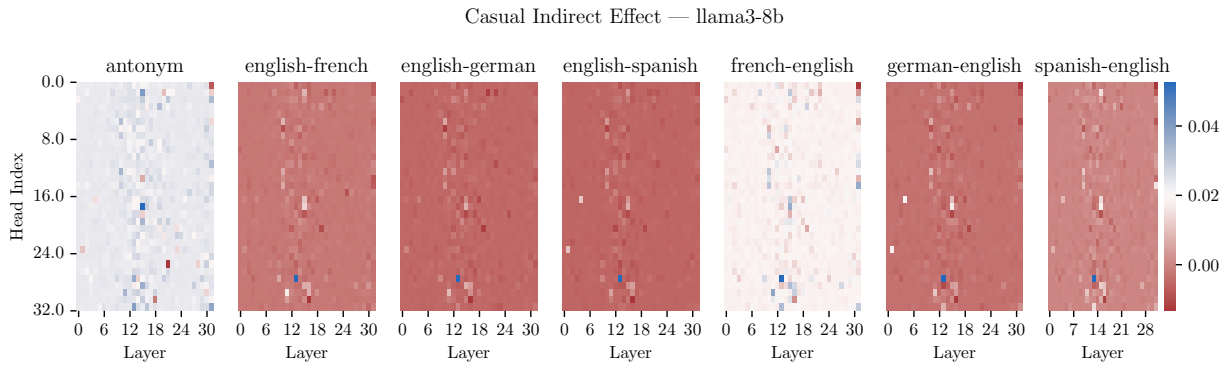


Figure 10: Causal indirect effect of attention heads in Llama-3.1 8B across all tasks. Varied patterns in the heads which modulate each task reflect the variability we observe in our experimental results. Continue reading in Appendix B.2.

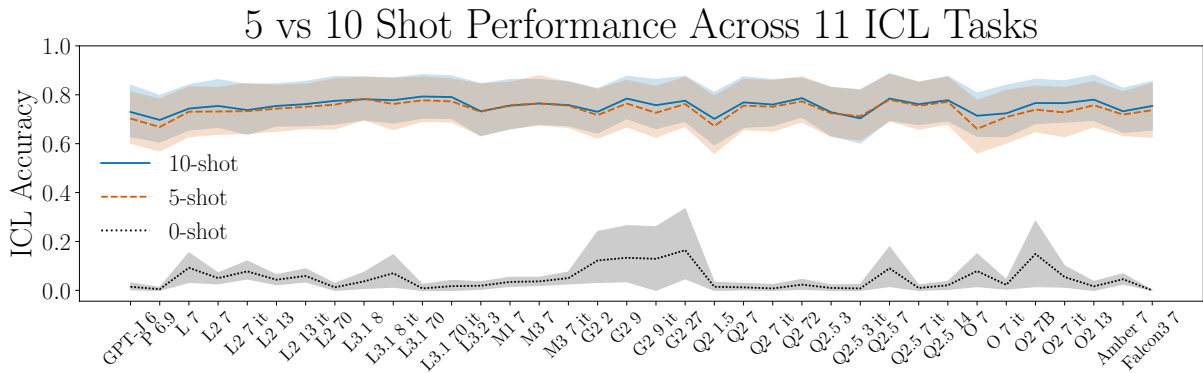


Figure 11: 10-shot and 5-shot performance across all models are relatively consistent, with 5-shot under performing by a small margin in some cases. Continue reading in Appendix B.3

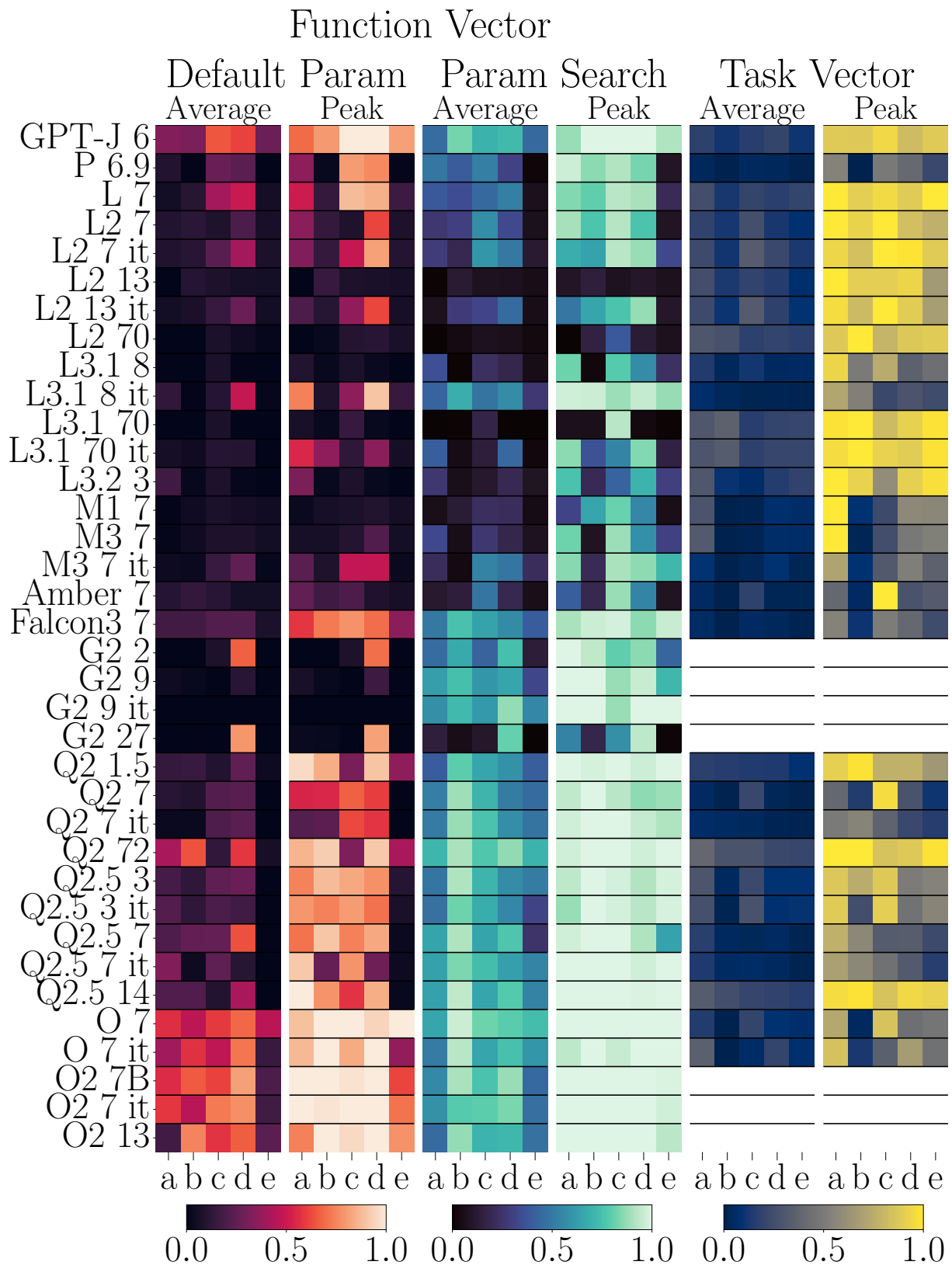


Figure 12: Heatmap showing performance recovery with different activation patching methods across tasks and models. This plot includes OLMo2 and Gemma2, which are omitted from the similar plot in the main paper. There is large variance in performance across tasks, models, and tools. Tasks: a) Antonym, b) Present-Past, c) Country-Capital, d) [lang] to eng, and e) eng to [lang]. Continue reading in Appendix B.4

Architecture													
Model Name	n_layers	n_heads	total_heads	head_size	hidden_dim	mlp_size	layer_norm (eps)	embed (θ)	attention	hidden_act	vocab_size	context_len	
EleutherAI/gpt-j-6b	28	16	448	256	4,096	16,384	LayerNorm (1e-05)	RoPE	MHA	gelu_new	50,400	2,048	
EleutherAI/pythia-6.9b	32	32	1,024	128	4,096	16,384	LayerNorm (1e-05)	RoPE (10k)	MHA	gelu	50,432	2,048	
huggingllama/llama-7b	32	32	1,024	128	4,096	11,008	LayerNorm (1e-06)	RoPE	MHA	SwiGLU	32,000	2,048	
meta-llama/llama-2-7b-hf	32	32	1,024	128	4,096	11,008	LayerNorm (1e-05)	RoPE	MHA	SwiGLU	32,000	4,096	
meta-llama/llama-2-7b-chat-hf	32	32	1,024	128	4,096	11,008	LayerNorm (1e-05)	RoPE	MHA	SwiGLU	32,000	4,096	
meta-llama/llama-2-1.3b-hf	40	40	1,600	128	5,120	13,824	LayerNorm (1e-05)	RoPE	MHA	SwiGLU	32,000	4,096	
meta-llama/llama-2-1.3b-chat-hf	40	40	1,600	128	5,120	13,824	LayerNorm (1e-05)	RoPE	MHA	SwiGLU	32,000	4,096	
meta-llama/llama-2-70b	80	64	5,120	128	8,192	28,672	LayerNorm (1e-05)	RoPE	GQA	SwiGLU	32,000	4,096	
meta-llama/Meta-Llama-3-8B	32	32	1,024	128	4,096	14,336	LayerNorm (1e-05)	RoPE (500k)	GQA	SwiGLU	128,256	8,192	
meta-llama/Meta-Llama-3-70B	80	64	5,120	128	8,192	28,672	LayerNorm (1e-05)	RoPE (500k)	GQA	SwiGLU	128,256	8,192	
meta-llama/llama-3.1-8B	32	32	1,024	128	4,096	14,336	LayerNorm (1e-05)	RoPE (500k)	GQA	SwiGLU	128,256	131,072	
meta-llama/llama-3.1-8B-Instruct	32	32	1,024	128	4,096	14,336	LayerNorm (1e-05)	RoPE (500k)	GQA	SwiGLU	128,256	131,072	
meta-llama/llama-3.1-70B	80	64	5,120	128	8,192	28,672	LayerNorm (1e-05)	RoPE (500k)	GQA	SwiGLU	128,256	131,072	
meta-llama/llama-3.1-70B-Instruct	80	64	5,120	128	8,192	28,672	LayerNorm (1e-05)	RoPE (500k)	GQA	SwiGLU	128,256	131,072	
meta-llama/llama-3.2-3B	28	24	672	128	3,072	8,192	LayerNorm (1e-05)	RoPE (500k)	GQA	SwiGLU	128,256	131,072	
mistralai/Mistral-7B-v0.1	32	32	1,024	128	4,096	14,336	LayerNorm (1e-05)	RoPE	GQA	SwiGLU	32,000	8,192	
mistralai/Mistral-7B-v0.3	32	32	1,024	128	4,096	14,336	LayerNorm (1e-05)	RoPE	GQA	SwiGLU	32,768	8,192	
LLM360/Amber	32	32	1,024	128	4,096	11,008	LayerNorm (1e-06)	RoPE	MHA	SwiGLU	32,000	2,048	
tiuae/Falcon3-7B-Base	28	12	336	256	3,072	23,040	LayerNorm (1e-06)	RoPE (1,000k)	GQA	SwiGLU	131,072	32,768	
Qwen/Qwen2-1.5B	28	12	336	128	1,536	8,960	LayerNorm (1e-06)	RoPE (1,000k)	GQA	SwiGLU	151,646	131,072	
Qwen/Qwen2-7B	28	28	784	128	3,584	18,944	LayerNorm (1e-06)	RoPE (1,000k)	GQA	SwiGLU	151,646	131,072	
Qwen/Qwen2-7B-Instruct	28	28	784	128	3,584	29,568	LayerNorm (1e-06)	RoPE (1,000k)	GQA	SwiGLU	151,646	131,072	
Qwen/Qwen2-72B	80	64	5,120	128	8,192	29,568	LayerNorm (1e-06)	RoPE (1,000k)	GQA	SwiGLU	151,646	131,072	
Qwen/Qwen2.5-3B	36	16	576	128	2,048	11,008	LayerNorm (1e-06)	RoPE (1,000k)	GQA	SwiGLU	151,936	32,768	
Qwen/Qwen2.5-3B-Instruct	36	16	576	128	2,048	11,008	LayerNorm (1e-06)	RoPE (1,000k)	GQA	SwiGLU	151,936	32,768	
Qwen/Qwen2.5-7B	28	28	784	128	3,584	18,944	LayerNorm (1e-06)	RoPE (1,000k)	GQA	SwiGLU	152,064	131,072	
Qwen/Qwen2.5-7B-Instruct	28	28	784	128	3,584	18,944	LayerNorm (1e-06)	RoPE (1,000k)	GQA	SwiGLU	152,064	131,072	
Qwen/Qwen2.5-14B	48	40	1,920	128	5,120	13,824	LayerNorm (1e-05)	RoPE (1,000k)	GQA	SwiGLU	152,064	131,072	
google/gemma-2-2b	26	8	208	256	2,304	18,432	LayerNorm (1e-06)	RoPE (10k)	GQA	GeGLU	256,128	8,192	
google/gemma-2-9b	42	16	672	256	3,584	28,672	LayerNorm (1e-06)	RoPE (10k)	GQA	GeGLU	256,128	8,192	
google/gemma-2-27b	46	32	1,472	256	3,584	28,672	LayerNorm (1e-06)	RoPE (10k)	GQA	GeGLU	256,128	8,192	
google/gemma-2-9b-it	42	16	672	256	3,584	28,672	LayerNorm (1e-06)	RoPE (10k)	GQA	GeGLU	256,128	8,192	
allenai/OLMo-7B-0724-hf	32	32	1,024	128	4,096	11,008	non-parametric	RoPE (10k)	MHA	SwiGLU	50,304	4,096	
allenai/OLMo-7B-0724-Instruct-hf	32	32	1,024	128	4,096	11,008	non-parametric	RoPE (10k)	MHA	SwiGLU	50,304	4,096	
allenai/OLMo-2-1124-7B	32	32	1,024	128	4,096	11,008	LayerNorm (1e-06)	RoPE (500k)	MHA	SwiGLU	100,352	4,096	
allenai/OLMo-2-1124-7B-Instruct	32	32	1,024	128	4,096	11,008	LayerNorm (1e-06)	RoPE (500k)	MHA	SwiGLU	100,352	4,096	
allenai/OLMo-2-1124-13B	40	40	1,600	128	5,120	13,824	LayerNorm (1e-06)	RoPE (500k)	MHA	SwiGLU	100,352	4,096	

Table 11: Architecture differences among the models tested. Continue reading in §5.

Data, Training, and Optimization						
Model Name	n_tokens	pl_learn_rate	staged_pretrain	post_train		
EleutherAI/gpt-j-6b	402B	-	no	no		
EleutherAI/pythia-6.9b	300B	1.2 x 10-4	no	no		
huggyllama/llama-7b	1T	3 x 10-4	no	no		
meta-llama/llama-2-7b-hf	2T	3 x 10-4	no	no		
meta-llama/llama-2-7b-chat-hf	2T	3 x 10-4	no	yes		
meta-llama/llama-2-13b-hf	2T	3 x 10-4	no	no		
meta-llama/llama-2-13b-chat-hf	2T	3 x 10-4	no	yes		
meta-llama/llama-2-70b	2T	1.5 x 10-4	no	no		
meta-llama/Meta-Llama-3-8B	15T+	3 x 10-4	no	no		
meta-llama/Meta-Llama-3-70B	15T+	1.5 x 10-4	no	no		
meta-llama/llama-3.1-8B	15T+	3 x 10-4	yes	no		
meta-llama/llama-3.1-8B-Instruct	15T+	3 x 10-4	yes	yes		
meta-llama/llama-3.1-70B	15T+	1.5 x 10-4	yes	no		
meta-llama/llama-3.1-70B-Instruct	15T+	1.5 x 10-4	yes	yes		
meta-llama/llama-3.2-3B	15T+	-	yes	no		
mistralai/Mistral-7B-v0.1	-	-	no	no		
mistralai/Mistral-7B-v0.3	-	-	no	no		
LLM360/Amber	1.3T	3 x 10-4	no	no		
tiiuae/falcon-3-7B-Base	14T	-	-	no		
Qwen/Qwen2-1.5B	7T+	-	yes	no		
Qwen/Qwen2-7B	7T+	-	yes	no		
Qwen/Qwen2-7B-Instruct	7T+	-	yes	yes		
Qwen/Qwen2-72B	7T+	-	yes	no		
Qwen/Qwen2.5-3B	18T	-	yes	no		
Qwen/Qwen2.5-3B-Instruct	18T	-	yes	yes		
Qwen/Qwen2.5-7B	18T	-	yes	no		
Qwen/Qwen2.5-7B-Instruct	18T	-	yes	yes		
Qwen/Qwen2.5-14B	18T	-	yes	no		
google/gemma-2-2b	2T	-	no	no		
google/gemma-2-9b	8T	-	no	no		
google/gemma-2-9b-it	8	-	no	yes		
google/gemma-2-27b	8T	-	no	no		
allenai/OLMo-7B-0724-hf	2.75T	3 x 10-4	yes	no		
allenai/OLMo-7B-0724-Instruct-hf	2.75T	3 x 10-4	yes	yes		
allenai/OLMo-2-1124-7B	5T	3 x 10-4	yes	no		
allenai/OLMo-2-1124-7B-Instruct	5T	3 x 10-4	yes	yes		
allenai/OLMo-2-1124-13B	5T	9 x 10-4	yes	yes		

Table 12: Data, training, and optimization differences among the models tested. "-" represents information which was not found, or unclear. Continue reading in §5

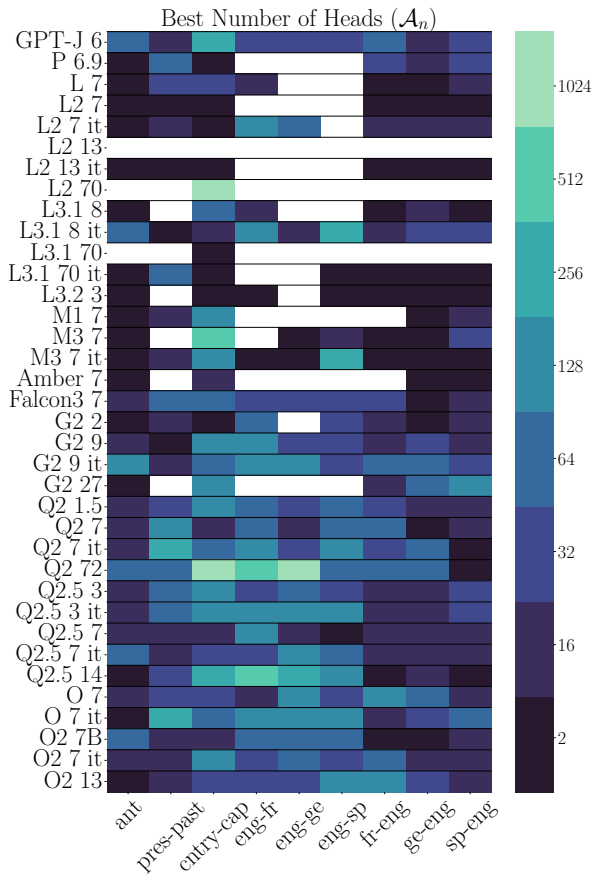


Figure 13: Best number of heads \mathcal{A}_n considering all other hyperparameters for function vectors. Data is masked for models with zero-shot, function vector patched accuracy of at least 10% on the task. Continue reading in §4.2.

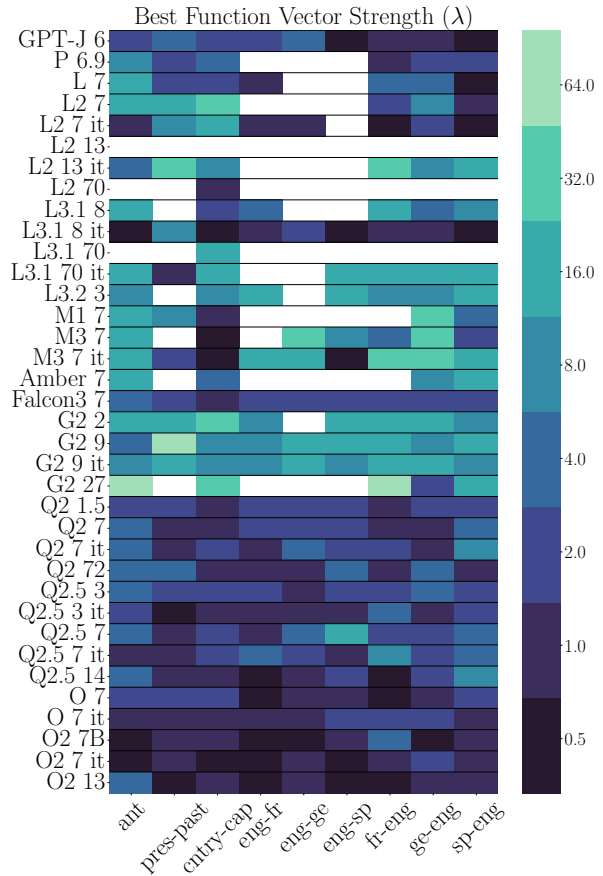


Figure 14: Best function vector strength λ considering all other hyperparameters for function vectors. Data is masked for models with zero-shot, function vector patched accuracy of at least 10% on the task. Continue reading in §4.2.

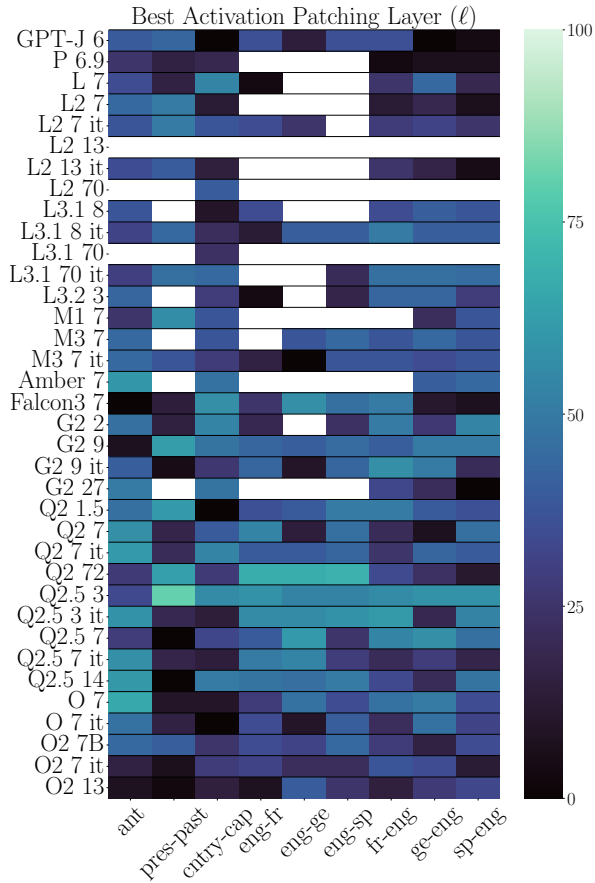


Figure 15: Best layer ℓ considering all other hyperparameters for function vectors expressed as a percent of all layers. Data is masked for models with zero-shot, function vector patched accuracy of at least 10% on the task. Continue reading in §4.2.

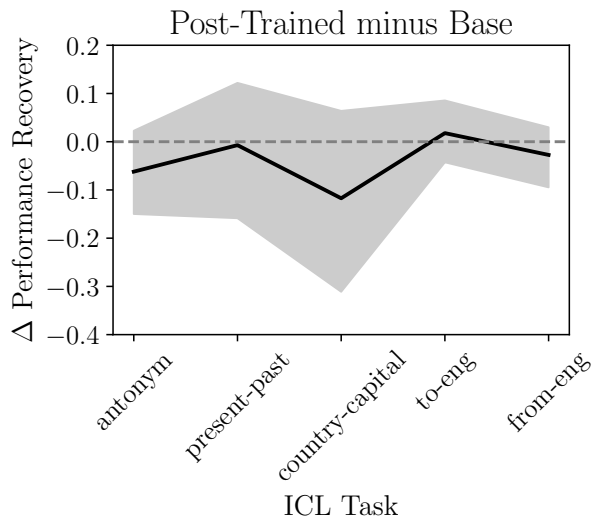


Figure 16: Task vectors performance recovery of post-trained models against their base counterparts. In most cases, post-trained models perform worse or the same when patched with a task vector. Continue reading in §4.2.

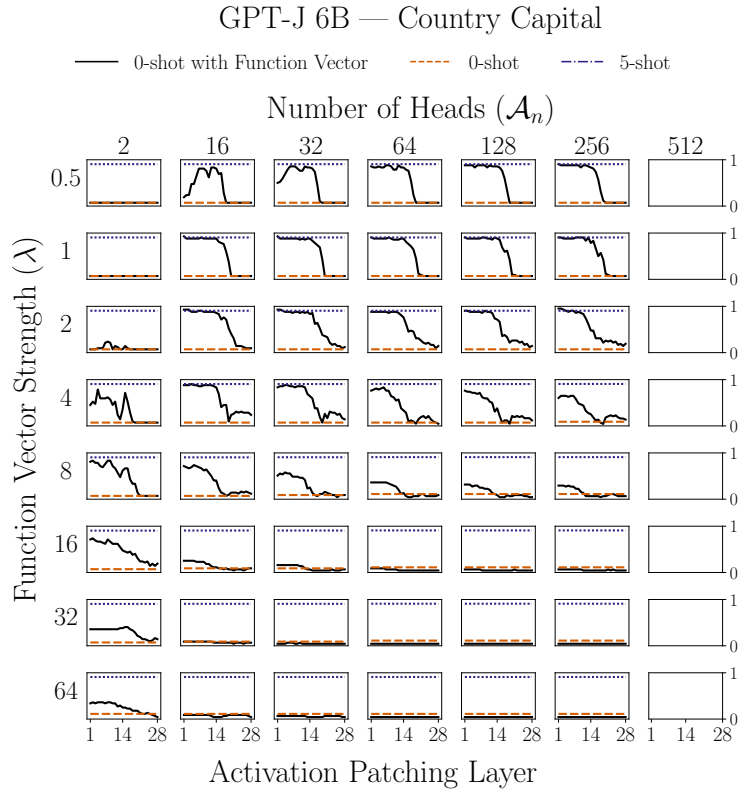


Figure 17: Full parameter search for GPT-J 6B on the country-capital task. Continue reading in §4.2.

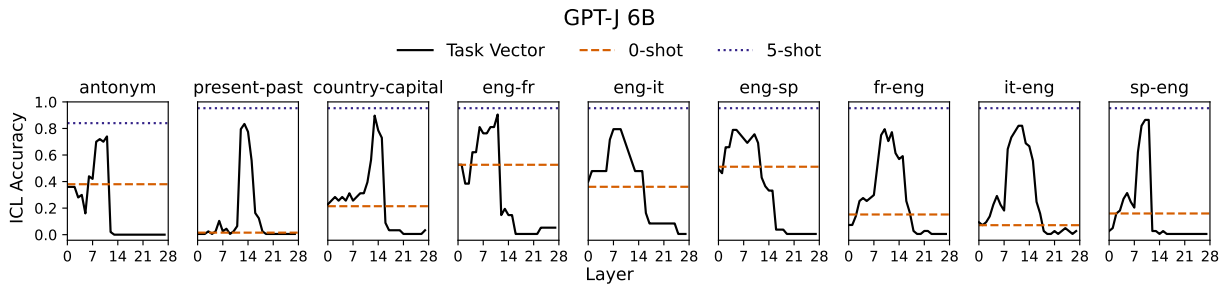


Figure 18: GPT-J 6B accuracy with task vector patched into layer ℓ in a zero-shot prompt. Consistent with [Hendel et al. \(2023\)](#), this model has decent performance recovery across all tasks. Continue reading in Appendix B.4.

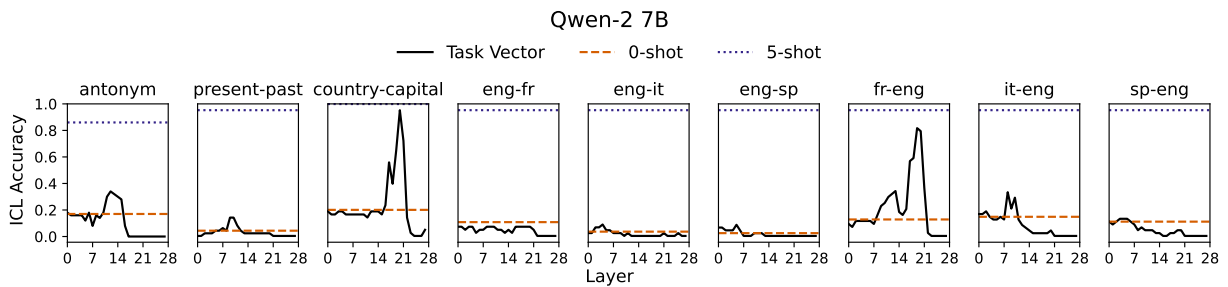


Figure 19: Qwen2 7B ICL accuracy with task vector patched into layer ℓ in a zero-shot prompt. Unlike GPT-J, this model fails to recover performance across many of its tasks. Continue reading in Appendix B.4.