# Adversarial Robustness of Activation Steering in Large Language Models

Anonymous ACL submission

## Abstract

Activation steering has become a popular training-free method to control LLM behavior by injecting precomputed direction vectors into the model's residual stream at inference time. Yet its robustness to realistic input variation remains unstudied. We present the first systematic evaluation of activation steering robustness under adversarial text perturbations on the inputs, covering four extraction methods, three attack strategies, six personas from Anthropic Model-Written Evaluation Dataset, and five models ranging from 1.5B to 30B parameters. Attacks succeed broadly across all settings: directional robustness drops by up to 64%, post-attack confidence collapses near or below 0.25 across all methods and models, and steering strength degrades on nearly every steerable input. Layer selection is equally fragile, with the optimal layer identified by an automated method on clean inputs shifting by up to 17 positions under perturbation, a failure that compounds the vector-level breakdown. Extracting vectors from adversarially perturbed inputs partially recovers steerability for PCA and MD on mid-to-large models, but they consistently fail to locate the improved optimal layer, limiting the practical benefit of this mitigation. Together, these findings reveal that the brittleness of activation steering is structural rather than method-specific, and that current layer selection strategies are not robust enough for real-world deployment.

## 1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, yet controlling their behavior at deployment time remains a challenge. The dominant alignment paradigms, Supervised Fine-Tuning (Chung et al., 2024), RLHF (Ouyang et al., 2022), and DPO (Rafailov et al., 2023), update model weights and require substantial compute and carefully curated preference data. They also optimize for a fixed alignment target, making it difficult to adjust behavior on a per-deployment basis without retraining (Turner et al., 2023; Zou et al., 2023).

*Activation steering* has emerged as a highly efficient, training-free alternative (Turner et al., 2023; Zou et al., 2023; Li et al., 2023; Rimsky et al., 2024). Instead of modifying weights, it injects a precomputed direction vector into the model's residual stream during inference, nudging its hidden representations toward a desired behavior. Anthropic has directly employed activation steering to monitor and mitigate undesirable personality traits in Claude (Anthropic, 2025), and its applications now span safety (Arditi et al., 2024), truthfulness (Li et al., 2023), persona control (Rimsky et al., 2024), and mathematical reasoning (Nguyen and Le, 2026). Recent work has progressed along two related directions: *how to steer* or asking how to extract the steering vector, and *where to steer* or asking which layer to intervene at, with methods such as LayerNavigator (Sun et al., 2025) providing principled selection.

Despite this progress, an important question has not yet been explored: *how robust are steering vectors when deployed in real-world conditions?* Vectors are extracted from one curated set of prompts but applied to potentially messier set of inputs at inference time. Users make typos, rephrase questions, and supply prompts whose surface form differs from the clean examples used during extraction. Prior work has noted symptoms of this brittleness in passing (Tan et al., 2024; Arditi et al., 2024), but no study has systematically asked *(1) whether steering holds up under controlled, semantics-preserving perturbations,* and *(2) whether the layer selected as optimal on clean inputs remains reliable once inputs are perturbed.*

To close this gap, this work attempts to answer both questions through a controlled, comprehensive empirical study across four candidate steering methods, three text perturbation attack strategies,

six persona datasets, and two LLM families spanning five model sizes. The results are substantial: attacks succeed broadly across nearly all settings, with directional robustness dropping by up to 64%. Moreover, the layer identified as optimal on clean inputs can shift by up to 17 positions under perturbation. Under adversarial training, we only observe partial recovery in steering robustness, while LayerNavigator consistently fails to locate the optimal layer for adversarially trained vectors. Our results demonstrate that, although activation engineering offers a promising weight-update-free mechanism for controlling LLM behaviors on-the-fly, its robustness under permissible perturbations remains highly fragile and insufficiently understood. These findings raise important concerns about the reliability of current steering methods in realistic deployment settings, where even minor, semantically preserving input variations can substantially alter steering effectiveness and layer sensitivity. We invite the reader to jump directly to the results in Sections 4–6.

In summary, **our main contributions are:**

- We present the first systematic study of activation steering robustness under controlled text perturbations, covering four steering methods and three perturbation strategies across five model sizes and six persona datasets.
- We show that steerability degrades sharply under all perturbations, directional robustness dropping by up to 64% and strength reduction approaching 100%.
- We demonstrate that score-based layer selection method like LayerNavigator is unstable under perturbation: the optimal layer on clean inputs can shift by up to 17 positions, and the steerability score profile can change significantly.
- We show that adversarial training partially recovers robustness for PCA and MD on mid-to-large models, but layer selection consistently fails to locate the improved optimal layer.

## 2 Preliminaries and Related Work

### 2.1 Activation Steering

Activation steering or activation engineering guides a language model towards a target behavior at inference time by intervening its residual stream via a precomputed vector, often called a steering vector, without updating model weights (Turner et al., 2023; Zou et al., 2023; Li et al., 2023; Rimsky et al., 2024). Given a hidden state $h_\ell \in \mathbb{R}^d$ at layer $\ell$, a steering method applies an intervention

$$\tilde{h}_\ell = \mathcal{F}(h_\ell;\, v_\ell), \tag{1}$$

where $\mathcal{F}$ is a method-specific steering function and $v_\ell \in \mathbb{R}^d$ is a precomputed steering vector.

### 2.2 Candidate Activation Steering Methods

We investigate four representative steering paradigms: contrastive activation methods, using Mean Difference (MD) (Rimsky et al., 2024; Belrose et al., 2023), unsupervised decomposition methods, using PCA (Zou et al., 2023), probe-based methods, using ITI (Li et al., 2023), and optimization-based learned interventions, using ODESteer (Zhao et al., 2026).

These methods estimate steering directions in different ways: MD averages differences between positive and negative activations, PCA extracts the dominant direction of activation variation, ITI learns a probe boundary separating positive from negative activations, and ODESteer learns an input-dependent barrier function for adaptive steering.

However, each paradigm may be brittle under distribution shift: MD can depend strongly on the chosen contrastive examples, PCA may capture prompt-distribution structure rather than the target behavior, ITI may overfit to extraction-prompt artifacts, and ODESteer still learns its barrier function from a fixed extraction set. These vulnerabilities motivate our study of whether steering remains stable under adversarially perturbed inference-time inputs. Detailed formulations of each method and additional variants are provided in Appendix A.

### 2.3 Robustness of Activation Steering

**Robustness of Steering Vectors.** Despite rapid progress in activation steering, its robustness under input perturbations remains underexplored. Existing studies show that steering effects can vary substantially across inputs, including sign flips or "anti-steerable" examples (Tan et al., 2024; Braun et al., 2025). Other work finds that commonly used steering evaluations may overstate effectiveness when consistency, magnitude, and specificity are measured more rigorously (Pres et al., 2024), and that refusal directions may transfer imperfectly to paraphrased or jailbroken prompts (Arditi et al., 2024). These findings suggest a key vulnerability: steering vectors estimated from a fixed extraction set may capture brittle or spurious structure rather than a stable behavioral concept.

This concern is especially important in adversarial settings. Prior work in adversarial NLP shows that small, semantics-preserving text perturbations can substantially alter model behavior (Li et al., 2020, 2018; Morris et al., 2020). However, to our knowledge, no prior work systematically studies whether activation steering remains stable when the evaluation prompts are adversarially perturbed.

**Robustness of Steering Layer Selection.** Robustness also depends on *where* steering is applied. Layer choice is known to strongly affect steering performance, often as much as the extraction method itself (Tan et al., 2024; Braun et al., 2025). LayerNavigator (Sun et al., 2025) provides a principled layer-selection criterion by ranking layers using a steerability score:

$$S_\ell = D_\ell + C_\ell, \tag{2}$$

where $D_\ell$ measures how well positive and negative activations are separated, and $C_\ell$ measures how consistently individual contrastive pairs align with the global steering direction. While this provides an efficient alternative to validation sweeps, its robustness under adversarially perturbed inputs has not been tested.

**Overall Robustness.** Together, these gaps motivate our study of adversarial robustness in activation steering along two axes: (1) **How to steer** or the robustness of how a steering method is computed and used to manipulate the residual streams and (2) **Where to steer** or robustness of how the optimal layer is selected for steering. Rather than assuming that a cleanly extracted vector and layer will remain effective at deployment time, we evaluate whether steering behavior is stable under controlled, semantics-preserving perturbations of the inputs at which steering is applied.

## 3 Problem Formulation

### 3.1 Robustness of How to Steer

Building on these observations, we now formalize what it means for a steering intervention to be *robust* to input perturbations.

Following Tan et al. (2024), we measure steering effectiveness on an input $x$ via the *mean Logit Difference* (LD). Since the persona datasets we use pose binary Yes/No questions, we compute LD using the softmax-normalized probabilities of the positive and negative answer tokens:

$$\mathrm{LD}(x) = \mathrm{logit}(y^+ \mid x) - \mathrm{logit}(y^- \mid x), \tag{3}$$

where $y^+ \in \{\text{“Yes”}, \text{“No”}\}$ is the answer token that aligns with the target behavior for question $x$, and $y^-$ is the opposite token. Which token counts as positive depends on the question: for some questions the target behavior is expressed by “Yes”, and for others by “No”. The *steering gain* of a vector $v_\ell$ on input $x$ is the change in LD induced by:

$$\Delta\mathrm{LD}(x; v_\ell) = \mathrm{LD}_{\text{steered}}(x; v_\ell) - \mathrm{LD}_{\text{base}}(x), \tag{4}$$

where $\mathrm{LD}_{\text{base}}(x)$ is the logit difference without intervention. An input $x$ is *steerable* if $\Delta\mathrm{LD}(x; v_\ell) > 0$, and *anti-steerable* otherwise (Tan et al., 2024). We can then define:

**Definition 3.1** (Extraction Robustness)**.** Let $v_\ell$ be a steering vector extracted from $\mathcal{D}_{\text{train}}$, and let $x' = \mathcal{A}(x)$ denote the semantics-preserving perturbation of $x \in \mathcal{D}_{\text{test}}$ produced by attack $\mathcal{A}$. The vector $v_\ell$ is *$\mathcal{A}$-robust* on input $x$ if the perturbed input $x' = \mathcal{A}(x)$ satisfies two conditions Steerability and Strength, simultaneously.

**(i) Steerability**: $\Delta\mathrm{LD}(x'; v_\ell) > 0$, i.e., the steering intervention continues to push behavior in the intended direction after perturbation. We measure this over the *steerable set* $\mathcal{S} = \{x \in \mathcal{D}_{\text{test}} : P(y^+ \mid x; v_\ell) > \delta\}$, where $\delta \in (0, 1)$ is a minimum confidence threshold (conservatively set to 0.3) or when the steered model already assigns sufficient probability to the target token on the clean input, as the *directional robustness rate*:

$$\mathcal{R}_{\text{dir}} = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \mathbf{1}[\Delta\mathrm{LD}(x'; v_\ell) > 0] \tag{5}$$

**(ii) Strength**: $\Delta\mathrm{LD}(x'; v_\ell) \geq \Delta\mathrm{LD}(x; v_\ell)$, i.e., the steering gain under perturbation is at least as large as on the clean input. We measure the converse or how often perturbation causes strength to degrade as the *strength reduction rate*:

$$\mathcal{R}_{\text{str}} = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \mathbf{1}[\Delta\mathrm{LD}(x'; v_\ell) < \Delta\mathrm{LD}(x; v_\ell)] \tag{6}$$

Condition (i) captures *directional robustness*, or whether the steering effect avoids flipping into an anti-steerable failure under attack. Condition (ii) captures *magnitude robustness*, or whether the full strength of the behavioral effect is preserved under perturbation, not just its sign. Any degradation in steering gain counts as a robustness failure, making this a strict criterion that we adopt throughout our experiments. A robust steering method should exhibit high $\mathcal{R}_{\text{dir}}$ and low $\mathcal{R}_{\text{str}}$.

### 3.2 Robustness of WHERE TO STEER

In this work, we adopt LayerNavigator (Sun et al., 2025) as our layer selection strategy across all three extraction methods. Its method-agnostic steerability score provides a principled, consistent basis for comparison, without requiring held-out data or brute-force search. While the analysis above establishes *how* to select layers under clean inputs, it leaves open whether that selection remains stable when inputs are perturbed.

Let $\mathbf{S}(\mathcal{D})=(S_1(\mathcal{D}),\ldots,S_L(\mathcal{D}))$ denote the vector of per-layer steerability scores computed by LayerNavigator over a dataset $\mathcal{D}$, where each $S_\ell(\mathcal{D})$ is the score defined in Eq. (2). Note that $S_\ell$ depends on $\mathcal{D}$ through both the activations and the steering vector $v_\ell$. We extract $v_\ell$ from the same $\mathcal{D}$ at each layer. Let $\ell^*=\arg\max_\ell S_\ell(\mathcal{D})$ be the optimal layer selected from it, we can then define:

**Definition 3.2** (Layer Robustness)**.** Let $\mathcal{A}$ be a text attack and let $\mathcal{D}'_{\text{train}}$ denote the training set obtained by applying $\mathcal{A}$ to each example in $\mathcal{D}_{\text{train}}$. The layer selection is *$\mathcal{A}$-robust* if re-running LayerNavigator on $\mathcal{D}'_{\text{train}}$ satisfies two conditions Profile and Selection Stability, simultaneously.

**(i) Profile stability**: the steerability score profile does not change shape under perturbation. We measure this as the *Dynamic Time Warping* (DTW) distance between the clean and perturbed profiles, averaged across persona datasets:

$$\mathcal{M}_{\text{DTW}} = \text{DTW}\big(\mathbf{S}(\mathcal{D}_{\text{train}}),\, \mathbf{S}(\mathcal{D}'_{\text{train}})\big)\,. \quad (7)$$

**(ii) Selection stability**: the optimal layer does not shift under perturbation. We measure this as the absolute layer shift:

$$\mathcal{M}_{\text{shift}} = |\ell^*(\mathcal{D}_{\text{train}}) - \ell^*(\mathcal{D}'_{\text{train}})|. \quad (8)$$

Condition (i) captures whether the *shape* of the steerability landscape is preserved under perturbation: a profile that shifts substantially signals that the model's internal signal for layer selection is sensitive to small changes in the input. Condition (ii) captures the downstream consequence: whether the actual layer chosen for intervention changes. A robust selection method should exhibit low $\mathcal{M}_{\text{DTW}}$ and low $\mathcal{M}_{\text{shift}}$.

## 4 Robustness of HOW TO STEER

### 4.1 Objective Function

Tan et al. (2024) showed that steering vectors are brittle when context changes — for example, when instructions are prepended or the system prompt is altered. But what if the change is far subtler: a small rewrite of the question itself, one that preserves meaning but uses different words? Would a steering vector, computed once on a fixed set of questions, still work reliably on such perturbed inputs? We now formalize our goal.

**Minimizing steering effectiveness.** Given a question $x_i \in \mathcal{D}_{\text{test}}$ and a steering vector $v_\ell$ extracted at layer $\ell$ from $\mathcal{D}_{\text{train}}$, we seek a perturbed input $x'$ on which steering fails as badly as possible. Since the dataset uses binary Yes/No answers, we have $P(y^+)+P(y^-)=1$, making $P(y^+)$ a monotone transformation of $\text{LD}(x)=\log\frac{P(y^+)}{1-P(y^+)}$. We therefore use $P(y^+)$ as a computationally convenient attack objective, while reporting results in terms of $\Delta$LD throughout:

$$\min_{x'}\ P(y^+ \mid x', v_\ell), \quad (9)$$

where $v_\ell$ and $\ell$ are fixed — extracted once from $\mathcal{D}_{\text{train}}$ and held constant throughout. Only the input text $x_i \in \mathcal{D}_{\text{test}}$ is perturbed.

**Semantic similarity constraint.** Minimizing Eq. (9) without restriction is trivial as we could rewrite the question entirely and produce an unrelated input. We therefore require $x'$ to remain semantically close to $x_i$, meaning $x'$ must lie within $\mathcal{A}(x_i)$, the set of semantics-preserving perturbations defined in Definition 3.1. Concretely, this is enforced by requiring:

$$\text{sim}(x_i, x') \geq \epsilon \quad (10)$$

where $\text{sim}(\cdot)$ is a semantic similarity function and $\epsilon$ is a minimum threshold, both determined by the specific attack method following their original formulations in Jin et al. (2020); Li et al. (2018, 2020).

**Full objective.** Combining the two, we solve:

$$\min_{x'}\ P(y^+ \mid x', v_\ell)\ \text{ s.t. }\ \text{sim}(x, x') \geq \epsilon \quad (11)$$

In practice, the search stops early once $P(y^+ \mid x', v_\ell)<\delta$, where we set $\delta$=0.3 throughout all experiments. A perturbation that drives the steered probability of $y^+$ this low already constitutes a

| | Method | Religion Following | | | Conscientiousness | | | Self-Improvement | | | Impact-Maximization | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ASR | b.Atk | a.Atk | ASR | b.Atk | a.Atk | ASR | b.Atk | a.Atk | ASR | b.Atk | a.Atk |
| Llama3.2-3B | W/o Steer | – | 0.56 | – | – | 0.73 | – | – | 0.71 | – | – | 0.66 | – |
| | PCA | 0.70 | 0.65 | 0.24 (↓0.41) | 0.94 | 0.76 | 0.25 (↓0.51) | 0.80 | 0.72 | 0.28 (↓0.44) | 0.88 | 0.69 | 0.25 (↓0.44) |
| | MD | 0.60 | 0.65 | 0.26 (↓0.39) | 0.86 | 0.77 | 0.26 (↓0.51) | 0.82 | 0.76 | 0.27 (↓0.49) | 0.89 | 0.71 | 0.24 (↓0.47) |
| | ITI | 0.45 | 0.61 | 0.26 (↓0.35) | 0.91 | 0.81 | 0.25 (↓0.56) | 0.90 | 0.81 | 0.25 (↓0.56) | 0.84 | 0.69 | 0.25 (↓0.44) |
| | ODE | 0.78 | 0.78 | 0.25 (↓0.53) | 0.87 | 0.81 | 0.27 (↓0.54) | 0.91 | 0.81 | 0.26 (↓0.55) | 0.89 | 0.75 | 0.25 (↓0.50) |
| Qwen3-14B | W/o Steer | – | 0.81 | – | – | 0.81 | – | – | 0.80 | – | – | 0.71 | – |
| | PCA | 0.72 | 0.81 | 0.25 (↓0.56) | 0.90 | 0.81 | 0.21 (↓0.60) | 0.75 | 0.81 | 0.26 (↓0.55) | 0.79 | 0.71 | 0.24 (↓0.47) |
| | MD | 0.92 | 0.88 | 0.23 (↓0.65) | 0.90 | 0.87 | 0.21 (↓0.66) | 0.85 | 0.84 | 0.23 (↓0.61) | 0.90 | 0.78 | 0.22 (↓0.56) |
| | ITI | 0.95 | 0.86 | 0.22 (↓0.64) | 0.91 | 0.87 | 0.21 (↓0.66) | 0.95 | 0.86 | 0.23 (↓0.63) | 0.91 | 0.77 | 0.22 (↓0.55) |
| | ODE | 0.75 | 0.82 | 0.26 (↓0.56) | 0.91 | 0.82 | 0.21 (↓0.61) | 0.74 | 0.81 | 0.26 (↓0.55) | 0.83 | 0.72 | 0.24 (↓0.48) |
| Q3-30B-A3B | W/o Steer | – | 0.80 | – | – | 0.94 | – | – | 0.87 | – | – | 0.74 | – |
| | PCA | 0.82 | 0.79 | 0.19 (↓0.60) | 0.91 | 0.94 | 0.14 (↓0.80) | 0.80 | 0.87 | 0.19 (↓0.68) | 0.78 | 0.74 | 0.15 (↓0.59) |
| | MD | 0.85 | 0.85 | 0.17 (↓0.68) | 0.90 | 0.93 | 0.14 (↓0.79) | 0.80 | 0.87 | 0.19 (↓0.68) | 0.78 | 0.75 | 0.15 (↓0.60) |
| | ITI | 0.83 | 0.82 | 0.18 (↓0.64) | 0.90 | 0.94 | 0.15 (↓0.79) | 0.81 | 0.87 | 0.18 (↓0.69) | 0.78 | 0.75 | 0.15 (↓0.60) |
| | ODE | 0.87 | 0.86 | 0.17 (↓0.69) | 0.90 | 0.93 | 0.14 (↓0.79) | 0.79 | 0.87 | 0.19 (↓0.68) | 0.79 | 0.76 | 0.14 (↓0.62) |

Table 1: Attack results across four tasks, averaged across three perturbation methods (TextFooler, TextBugger, BERT-Attack). **ASR** is defined in Eq. (12). **W/o Steer** gives the unsteered baseline on clean inputs. **b.Atk** and **a.Atk** denote before attack and after attack, respectively. Values in parentheses show absolute decrease after attack. Refer to Table A1 and Table A2 for remaining tasks and models results.

clear steering failure, and the threshold keeps computational cost reasonable without sacrificing the quality of the found perturbations.

### 4.2 Experiment Set-up

**Text Perturbation and Dataset.** We apply three attack methods: TextFooler (word-level synonym substitution) (Jin et al., 2020), TextBugger (character-level manipulation) (Li et al., 2018), and BERT-Attack (filling masked words with BERT) (Li et al., 2020), with varying text perturbation strategies on $\mathcal{D}_{\text{test}}$, creating $\mathcal{D}'_{\text{test}}$ for six personas on the Anthropic MWE datasets (Perez et al., 2023), notably Religion Following, Conscientiousness, Self-Improvement, Alliance Building, Impact Maximization and Self-Aware.

**Metrics.** Each example in $\mathcal{D}_{\text{test}}$ either belongs to the steerable set $\mathcal{S}$ or does not (Definition 3.1). Examples outside $\mathcal{S}$ are skipped and no attack is attempted on them. For examples in $\mathcal{S}$, the attack either *succeeds* (driving $P(y^+ \mid x', v_\ell)$ below $\delta$) or *fails*. The **Attack Success Rate** (ASR) is the fraction of all $|\mathcal{D}|$ examples for which the attack succeeds:

$$\text{ASR} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{S}} \mathbf{1}[P(y^+ \mid x', v_\ell) \leq \delta]. \quad (12)$$

We additionally report $\mathcal{R}_{\text{dir}}$ and $\mathcal{R}_{\text{str}}$ (Definition 3.1) as our primary robustness metrics, computed over $\mathcal{S}$ across all attack methods and extraction approaches.

**Models.** We test with five models with different sizes: Qwen2.5-1.5B (Qwen et al., 2025), Llama-3.2-3B (Grattafiori et al., 2024), Qwen3-4B, Qwen3-14B and Qwen3-30B-A3B (Yang et al., 2025). Different sizes allow us to validate whether size improves robustness. We also include a Mixture of Experts model Qwen3-30B-A3B to see if sparse activations are helpful to steering robustness.

### 4.3 Findings and Discussion

***Finding #1: Attacks succeed broadly.*** Table 1 shows that ASR is high across almost all models, methods, and tasks. Correspondingly, the post-attack confidence collapse is consistent: *After Attk* probabilities converge near or below 0.25 across all settings, regardless of the pre-attack confidence level. One might expect that Mixture-of-Experts (MoE) models confer some implicit robustness, as perturbations may activate different experts and thus interact differently with the steering vector. Instead, ASR and post-attack confidence collapse on Qwen3-30B-A3B are comparable to dense models of similar active parameter count, suggesting that the brittleness of activation steering is not specific to dense transformer architectures.

***Finding #2: Steering strength degrades in almost every case.*** Table 2 shows that, across all four extraction methods, all three models, and all three tasks, $\mathcal{R}_{\text{str}}$ is high, frequently exceeding 0.9 and approaching 1.0 in many settings. This means that

| | | Rel.Fol. | | Conscient. | | Self-Improv. | |
|---|---|---|---|---|---|---|---|
| | | $\mathcal{R}_{\text{str}}$ | $\mathcal{R}_{\text{dir}}$ | $\mathcal{R}_{\text{str}}$ | $\mathcal{R}_{\text{dir}}$ | $\mathcal{R}_{\text{str}}$ | $\mathcal{R}_{\text{dir}}$ |
| Llama-3.2-3B | PCA | 0.76 | $0.52_{(\downarrow 0.10)}$ | 0.83 | $0.50_{(0.00)}$ | 0.69 | $0.48_{(\downarrow 0.10)}$ |
| | MD | 0.73 | $0.54_{(\downarrow 0.27)}$ | 0.90 | $0.52_{(\downarrow 0.45)}$ | 0.97 | $0.45_{(\downarrow 0.43)}$ |
| | ITI | 0.45 | $0.75_{(\downarrow 0.20)}$ | 0.98 | $0.49_{(\downarrow 0.37)}$ | 0.97 | $0.49_{(\downarrow 0.12)}$ |
| | ODE | 0.88 | $0.51_{(\downarrow 0.13)}$ | 0.90 | $0.46_{(\downarrow 0.44)}$ | 0.96 | $0.49_{(\downarrow 0.12)}$ |
| Qwen3-14B | PCA | 0.75 | $0.16_{(\downarrow 0.52)}$ | 0.63 | $0.21_{(\downarrow 0.28)}$ | 0.85 | $0.10_{(\downarrow 0.64)}$ |
| | MD | 0.87 | $0.48_{(\downarrow 0.05)}$ | 0.89 | $0.41_{(\downarrow 0.49)}$ | 0.94 | $0.33_{(\downarrow 0.48)}$ |
| | ITI | 0.83 | $0.50_{(0.00)}$ | 0.84 | $0.44_{(\downarrow 0.46)}$ | 0.80 | $0.50_{(0.00)}$ |
| | ODE | 0.78 | $0.39_{(\downarrow 0.15)}$ | 0.78 | $0.24_{(\downarrow 0.49)}$ | 0.75 | $0.22_{(\downarrow 0.33)}$ |
| Q3-30B-A3B | PCA | 0.50 | $0.50_{(0.00)}$ | 0.67 | $0.50_{(0.00)}$ | 0.67 | $0.50_{(0.00)}$ |
| | MD | 0.50 | $0.46_{(\downarrow 0.17)}$ | 0.66 | $0.50_{(\uparrow 0.01)}$ | 0.64 | $0.50_{(\downarrow 0.12)}$ |
| | ITI | 0.50 | $0.50_{(\downarrow 0.01)}$ | 0.70 | $0.51_{(\downarrow 0.19)}$ | 0.55 | $0.36_{(\downarrow 0.39)}$ |
| | ODE | 0.57 | $0.46_{(\downarrow 0.21)}$ | 0.70 | $0.50_{(0.00)}$ | 0.66 | $0.47_{(\downarrow 0.03)}$ |

Table 2: Steering robustness evaluation. The lower $\mathcal{R}_{\text{str}}$ the more robust. The higher $\mathcal{R}_{\text{dir}}$ the more robust. Subscripts show the change from the clean-input rate. Table A3 shows full results on all models and tasks.

perturbation reduces the steering gain on nearly every steerable input, even when the steering direction is not fully reversed. This result holds regardless of which extraction method is used, confirming that strength degradation is an intrinsic property of the additive intervention paradigm rather than a limitation of any particular estimation procedure. Even input-conditioned methods such as ODESteer offer no systematic protection against it.

***Finding #3: Directional robustness varies by model, not by method.*** $\mathcal{R}_{\text{dir}}$ tells a more varied story. The largest directional drops occur consistently on mid-to-large dense models, where the gap between clean and perturbed $\mathcal{R}_{\text{dir}}$ can reach 50–64 percentage points (Table 2). By contrast, on some models, $\mathcal{R}_{\text{dir}}$ remains near 0.5 both before and after perturbation. This happens because when the clean $\mathcal{R}_{\text{dir}}$ is already near 0.5, meaning the steered model is at chance on the clean input, there is simply no room for the metric to degrade further. In these cases the high $\mathcal{R}_{\text{str}}$ values confirm that steering is still being weakened, and the steerability floor obscures the failure in $\mathcal{R}_{\text{dir}}$ alone.

***Finding #4: No extraction method is consistently robust.*** Table 2 shows no single extraction method dominates on either metric. A method that achieves relatively low $\mathcal{R}_{\text{str}}$ in one setting typically shows high values elsewhere, and the same holds for $\mathcal{R}_{\text{dir}}$. This suggests that the brittleness we observe is not an artifact of how steering vectors are estimated but a structural phenomena.

# 5 Robustness of WHERE TO STEER

## 5.1 Objective Function

Section 4 that a fixed steering vector might not generalize well in practice when the *test* input is perturbed or written in different ways by the users. The problem may start earlier: at the point where the steering vector is extracted and the intervention layer is selected. For instance, LayerNavigator computes its steerability scores from an extraction or training set. This raises the crucial question: if those inputs are perturbed, does it still identify the same optimal layer, or, is steering layer selection also sensitive to permissible noise?

Given a training set $\mathcal{D}_{\text{train}}$ and a text perturbation $\mathcal{A}$, different from § 4 where we perturb only a single test input, here we apply $\mathcal{A}$ to every example $x_i \in \mathcal{D}_{\text{train}}$, producing a perturbed training set $\mathcal{D}'_{\text{train}} = \{\mathcal{A}(x_i)\}_{i=1}^{N}$. The same attack from § 4 is reused with the same objective function and semantic similarity constraint formalized in Eq. ( 11). We then re-extract steering vectors and re-run LayerNavigator on $\mathcal{D}'_{\text{train}}$, yielding a new score profile $\mathbf{S}(\mathcal{D}'_{\text{train}})$ and a new optimal layer $\ell^*(\mathcal{D}'_{\text{train}})$.

## 5.2 Experiment Set-up

We apply the same three attacks $\mathcal{A}$ to $\mathcal{D}_{\text{train}}$ on the same set of tasks and models as in §4. For each attack and each steering method, we re-extract steering vectors from $\mathcal{D}'_{\text{train}}$ and re-run LayerNavigator on the resulting activations.

**Metrics.** We report $\mathcal{M}_{\text{DTW}}$ and $\mathcal{M}_{\text{shift}}$ (Definition 3.2) as our two robustness metrics, averaged across the three attack methods. A robust layer selection method should exhibit low values on both: low $\mathcal{M}_{\text{DTW}}$ means the steerability score profile is stable under perturbation, and low $\mathcal{M}_{\text{shift}}$ means the chosen intervention layer does not change. We also quantify the $L1$ distance between the true optimal layer and LayerNavigator's selected layer $\ell^*$ via brute-force steering to show that on the perturbed set, LayerNavigator becomes unreliable.

## 5.3 Findings and Discussion

Table 3 reports $\mathcal{M}_{\text{DTW}}$ and $\mathcal{M}_{\text{shift}}$ for all method-model combinations across all tasks and attacks. Figure 1 shows the full steerability score profiles on clean and perturbed training sets. Table 4 shows the $L1$ distance between LayerNavigator $\ell^*$ and the true optimal layer.

| | | Rel.Fol. | | Conscient. | | Self-Improv. | |
|---|---|---|---|---|---|---|---|
| | | DTW | Top1-Shift | DTW | Top1-Shift | DTW | Top1-Shift |
| Llama3.2-3B | PCA | 2.55 | 15.00 | 5.65 | 1.33 | 2.51 | 1.33 |
| | MD | 3.74 | 6.67 | 3.89 | 2.67 | 4.68 | 6.00 |
| | ITI | 1.50 | 0.00 | 3.49 | 1.00 | 3.62 | 4.33 |
| | ODE | 5.14 | 1.33 | 5.16 | 3.33 | 6.95 | 6.33 |
| Qwen3-14B | PCA | 8.21 | 13.33 | 6.33 | 7.67 | 6.89 | 1.67 |
| | MD | 6.80 | 0.67 | 6.41 | 3.00 | 7.15 | 5.00 |
| | ITI | 2.05 | 0.67 | 4.50 | 1.33 | 3.29 | 1.67 |
| | ODE | 5.64 | 1.67 | 6.39 | 3.00 | 6.95 | 3.67 |
| Q3-30B-A3B | PCA | 9.61 | 20.00 | 4.76 | 10.67 | 4.43 | 20.00 |
| | MD | 5.15 | 6.33 | 9.86 | 2.00 | 11.27 | 4.33 |
| | ITI | 1.53 | 0.33 | 3.90 | 7.00 | 3.88 | 3.33 |
| | ODE | 5.26 | 4.33 | 9.09 | 3.33 | 10.63 | 0.00 |

Table 3: DTW scores and Top-1 layer difference in LayerNavigator. Red cells mark high DTW scores ($\geq 5$), while orange cells mark large Top-1 layer shifts ($\geq 10$) or near-high DTW scores. Refer to Table A4 for full results on all models and tasks.

| | | Rel.Fol. | | Conscient. | | Self-Improv. | |
|---|---|---|---|---|---|---|---|
| | | Clean | Perturb. | Clean | Perturb. | Clean | Perturb. |
| Llama3.2-3B | PCA | 1.0 | $14.0_{(\uparrow 13.0)}$ | 1.0 | $13.0_{(\uparrow 12.0)}$ | 1.0 | $2.0_{(\uparrow 1.0)}$ |
| | MD | 1.0 | $0.0_{(\downarrow 1.0)}$ | 1.0 | $1.3_{(\uparrow 0.3)}$ | 1.0 | $3.3_{(\uparrow 2.3)}$ |
| | ITI | 0.0 | $0.0_{(0.0)}$ | 0.0 | $10.0_{(\uparrow 10.0)}$ | 0.0 | $3.7_{(\uparrow 3.7)}$ |
| | ODE | 0.0 | $8.0_{(\uparrow 8.0)}$ | 1.0 | $1.0_{(0.0)}$ | 1.0 | $8.7_{(\uparrow 7.7)}$ |
| Qwen3-14B | PCA | 14.0 | $20.0_{(\uparrow 6.0)}$ | 7.0 | $19.0_{(\uparrow 12.0)}$ | 16.0 | $20.0_{(\uparrow 4.0)}$ |
| | MD | 1.0 | $8.7_{(\uparrow 7.7)}$ | 1.0 | $15.0_{(\uparrow 14.0)}$ | 2.0 | $17.0_{(\uparrow 15.0)}$ |
| | ITI | 1.0 | $5.0_{(\uparrow 4.0)}$ | 0.0 | $4.0_{(\uparrow 4.0)}$ | 1.0 | $21.0_{(\uparrow 20.0)}$ |
| | ODE | 0.0 | $18.0_{(\uparrow 18.0)}$ | 0.0 | $12.3_{(\uparrow 12.3)}$ | 3.0 | $22.0_{(\uparrow 19.0)}$ |
| Q3-30B-A3B | PCA | 35.0 | $10.0_{(\downarrow 25.0)}$ | 1.0 | $3.0_{(\uparrow 2.0)}$ | 3.0 | $10.0_{(\uparrow 7.0)}$ |
| | MD | 1.0 | $4.0_{(\uparrow 3.0)}$ | 23.0 | $5.0_{(\downarrow 18.0)}$ | 1.0 | $18.7_{(\uparrow 17.7)}$ |
| | ITI | 3.0 | $6.3_{(\uparrow 3.3)}$ | 9.0 | $3.0_{(\downarrow 6.0)}$ | 3.0 | $12.3_{(\uparrow 9.3)}$ |
| | ODE | 25.0 | $26.0_{(\uparrow 1.0)}$ | 26.0 | $13.3_{(\downarrow 12.7)}$ | 13.0 | $26.0_{(\uparrow 13.0)}$ |

Table 4: L1 distance between LayerNavigator's selected layer $\ell^*$ and the true optimal layer, on clean and perturbed inputs. The subscript shows the change (Perturbed − Clean). Red cells (↑) indicate the distance increased under perturbation, or LayerNavigator moved further from the optimal layer. Green cells (↓) indicate it decreased. Refer to Table A5 for full results on all models and tasks.

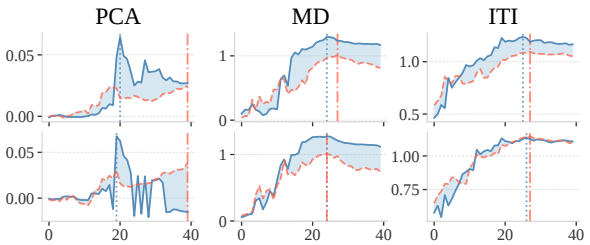


Figure 1: LayerNavigator S-score across two tasks: Religion Following (top) and Conscientiousness (bottom) on Qwen3-14B, averaged over three perturbations. **Blue** and **Red** line denote S-score calculated using steering vectors computed from $\mathcal{D}_{\text{train}}$ and $\mathcal{D}'_{\text{train}}$, respectively. Vertical lines show the largest S-score. The remaining score profiles are show at Figure A1, Figure A2 and Figure A3

***Finding #1: DTW and top-1 layer shift dissociate across steering methods.*** The most striking pattern is that the two robustness metrics do not move together, and the direction of their dissociation depends on which extraction method is used (Table 3). ITI produces the lowest DTW across nearly all models and tasks, meaning perturbation preserves the overall *trend* of its steerability score profile most reliably (Figure 1). Yet ITI still exhibits substantial top-1 layer shifts in several settings. MD and ODESteer show the opposite: their DTW values are the highest across the board, yet their top-1 shifts are often moderate or small. PCA occupies an intermediate and less consistent position: its DTW is low for smaller models but rises considerably for larger ones, while its layer shifts remain among the largest observed.

***Finding #2: Perturbations push LayerNavigator's predicted layer further from the true optimal layer.*** On clean inputs, LayerNavigator selects a layer close to the true optimal in most cases, as shown in Table 4. On perturbed inputs, however, the L1 distance to the true optimal, validated by brute force, grows in the majority of cases. This compounds the failure from Section 3.1: *perturbation both weakens steering at the originally selected layer* and *pushes the layer selection further from where steering would actually work best.*

**Ablation study on top-K shift** We also conduct an ablation study on layer shifts as we increase the number of top layers to 3 and 5. We report Rank Biased Overlap (RBO) score (Webber et al., 2010) to quantify how the ranking positions of top 3 and 5 layers shift. We refer the readers to more details in the Table A6. Overall, most models and steering methods exhibit high instability in rankings, showing that as we increase the number of top layers from LayerNavigator, it is more unreliable on perturbed inputs.

## 6 Discussion

§ 4 and § 5 show that steering vectors break down on perturbed inputs and that LayerNavigator's selected layer shifts when extraction inputs are perturbed. Inspired by adversarial training literature (Goodfellow et al., 2014; Madry et al., 2017), natural follow-up question is: *does extracting steering vectors from adversarially perturbed inputs*

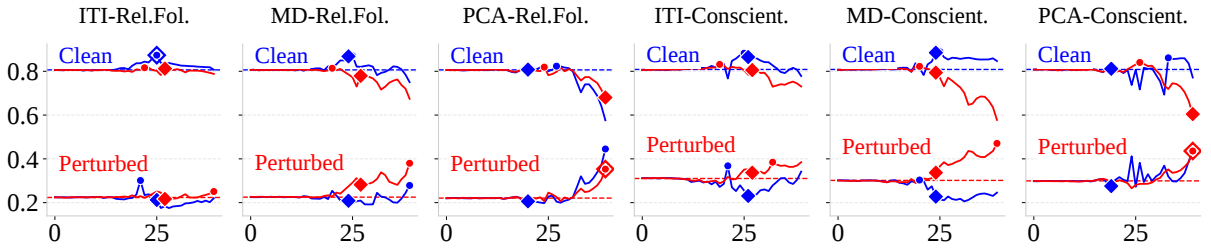


Figure 2: Per-layer averaged performance of Qwen3-14B across three perturbations on Religion Following and Conscientiousness. **Blue** and **Red** denote clean and adversarially trained steering vectors, respectively. The **upper cluster** (solid lines) shows performance on the clean set $\mathcal{D}_{\text{test}}$. The **lower cluster** (dashed lines) shows performance on the perturbed set $\mathcal{D}'_{\text{test}}$. **Diamonds** mark the layer selected by LayerNavigator. White Diamond mark when LayerNavigator selected layer matches the true optimal layer. For remaining models and methods, refer to Figure A4, Figure A5 and Figure A6.

*improve robustness at test time?* To test this, we reuse the perturbed training sets $\mathcal{D}'_{\text{train}}$ from § 5 to extract a new set of *adversarially trained* steering vectors. We evaluate all four combinations of vector type and test set: clean and adversarially trained vectors, each applied to both the clean test set $\mathcal{D}_{\text{test}}$ and the perturbed test set $\mathcal{D}'_{\text{test}}$. LayerNavigator is applied separately to each vector type using its corresponding training set. We report results for Qwen3-14B in Figure 2.

**Adversarial training partially recovers steerability on perturbed inputs.** In the lower cluster of Figure 2, the red dashed line (adversarially trained vectors on perturbed inputs) rises visibly above the blue dashed line (clean vectors on perturbed inputs) for PCA and MD, and sometimes clears the perturbed-set baseline by a meaningful margin at the best layer. This improvement is most consistent on mid-to-large models and weaker or absent on smaller ones. ITI shows smaller but still visible gains in some settings. ODE shows almost no improvement across any model or task: its lower curves remain flat regardless of whether vectors are extracted from clean or perturbed data, suggesting that ODE's learned Barrier function does not transfer well to adversarial inputs even when re-estimated from them.

**Clean-input performance is largely preserved.** In the upper cluster, the solid blue and solid red lines largely overlap across most model-method combinations, meaning adversarial training does not consistently hurt clean-input steerability. Exceptions exist in PCA on larger models where we observe a visible drop in the red solid line at later layers. However, these results are not uniform across tasks. For most methods, the adversarial training set *retains enough behavioral signal to maintain performance on clean inputs while improving robustness on perturbed ones*, though the benefit remains limited and method-dependent.

**LayerNavigator does not track the optimal layer for adversarially trained vectors.** Across all models and methods, the diamond markers for adversarially trained vectors (red diamonds) frequently do not coincide with the corresponding star markers, indicating that LayerNavigator's selected layer is not the best-performing one.

This mismatch is most visible for PCA and MD, where the optimal layer shifts when switching from clean to adversarially trained vectors, but LayerNavigator does not track this shift. As a result, even when adversarially trained vectors can achieve meaningful steerability at the right layer, a practitioner relying on LayerNavigator will often not find that layer. This compounds the finding from § 5: *adversarial perturbations shift both which vector is extracted and where that vector works best, and LayerNavigator's steerability scores fail to reflect either change reliably.*

## 7 Conclusion

This paper presents the first systematic stress test of activation steering under adversarial text perturbations across multiple extraction methods, attacks, models, and datasets. Our findings reveal a surprisingly brittle picture: attacks succeed broadly, directional robustness drops by up to 64 percentage points, and steering strength weakens on nearly all steerable inputs. This fragility extends to layer selection, where the optimal steering layer shifts drastically under perturbation despite seemingly stable score profiles. Furthermore, adversarial training offers only a partial remedy, failing to reliably correct layer selection. Ultimately, these results expose structural weaknesses in activation steering, motivating future work on input-adaptive layer selection and perturbation-aware extraction.

## Limitations

Our work has two main limitations. First, we evaluate a representative set of steering methods rather than exhaustively covering all existing approaches. We believe this still provides meaningful general insight, as the selected methods span distinct families of steering-vector construction. Second, we focus on alignment-oriented steering rather than other applications, such as mathematical reasoning or code generation. Because these tasks differ substantially in their objectives, evaluation criteria, and failure modes, we intentionally restrict the scope to alignment settings in order to provide a more concrete and focused analysis.

## Broader Impact

Although our work solely focuses on analyzing the robustness of activation steering methods to understand when and how model steering can fail during inference, the resulting insights may help practitioners design safer and more reliable deployment strategies for LLMs. We acknowledge that our framework could also be misused for malicious purposes, such as weakening steering mechanisms that improve refusal behavior against harmful or unsafe prompts. Similar to prior work in adversarial robustness and security evaluation, we believe that systematically identifying weaknesses is a necessary step toward building more robust defenses. By exposing failure modes under realistic perturbations, our study aims to encourage the development of steering methods that remain reliable under distribution shifts and adversarial manipulation, rather than relying on assumptions of stability that may not hold in practice.

## Acknowledgments

The authors acknowledge the use of ChatGPT, Claude and Grammarly for editorial assistance and ChatGPT, Gemini for assistance with figure visualization.

## References


Anthropic. 2025. Persona vectors: Monitoring and controlling character traits in language models.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. 2023. Leace: Perfect linear concept erasure in closed form. *Advances in Neural Information Processing Systems*, 36:66044–66063.

Joschka Braun, Carsten Eickhoff, David Krueger, Seyed Ali Bahrainian, and Dmitrii Krasheninnikov. 2025. Understanding (un) reliability of steering vectors in language models. *arXiv preprint arXiv:2505.22637*.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California. Association for Computational Linguistics.

Tuc Nguyen and Thai Le. 2026. Atlas: Adaptive test-time latent steering with external verifiers for enhancing llms reasoning. *arXiv preprint arXiv:2601.03093*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann,

Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, and 44 others. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13387–13434, Toronto, Canada. Association for Computational Linguistics.

Itamar Pres, Laura Ruis, Ekdeep Singh Lubana, and David Krueger. 2024. Towards reliable evaluation of behavior steering interventions in llms. *arXiv preprint arXiv:2410.17245*.

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, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

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.

Hao Sun, Huailiang Peng, Qiong Dai, Xu Bai, and Yanan Cao. 2025. Layernavigator: Finding promising intervention layers for efficient activation steering in large language models. *Advances in Neural Information Processing Systems*, 38:101058–101080.

Daniel Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. 2024. Analysing the generalisation and reliability of steering vectors. *Advances in Neural Information Processing Systems*, 37:139179–139212.

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, and 3 others. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron Wallace, and David Bau. 2024. Function vectors in large language models. In *International conference on learning representations*, volume 2024, pages 17282–17333.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4).

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Hongjue Zhao, Haosen Sun, Jiangtao Kong, Xiaochang Li, Qineng Wang, Liwei Jiang, Qi Zhu, Tarek Abdelzaher, Yejin Choi, Manling Li, and 1 others. 2026. Odesteer: A unified ode-based steering framework for llm alignment. *arXiv preprint arXiv:2602.17560*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

## A Steering Vector Extraction Methods

All methods follow the general steering formulation in Eq. (1), differing in how the steering vector $v_\ell$ is extracted and how $\mathcal{F}$ is instantiated. Let $\{(x_i^+, x_i^-)\}_{i=1}^N$ denote a set of $N$ contrastive pairs of positive and negative prompts, and let $h_\ell^+(x_i)$, $h_\ell^-(x_i) \in \mathbb{R}^d$ denote the hidden states at layer $\ell$ for each.

**Mean Difference (MD).** MD (Rimsky et al., 2024) computes the steering vector as the average difference between positive and negative activations:

$$v_\ell^{\text{MD}} = \frac{1}{N} \sum_{i=1}^{N} \left( h_\ell^+(x_i) - h_\ell^-(x_i) \right). \tag{A1}$$

The steering function is additive: $\mathcal{F}(h_\ell;\, v_\ell) = h_\ell + \alpha v_\ell^{\text{MD}}$. Despite its simplicity, MD is worst-case optimal for linear concept editing (Belrose et al., 2023) and remains a strong baseline. Related variants include In-Context Vectors (Liu et al., 2024), Function Vectors (Todd et al., 2024), and the Refusal Direction (Arditi et al., 2024).

**Principal Component Analysis (PCA).** PCA (Zou et al., 2023) extracts the steering vector as the leading eigenvector of the empirical covariance of contrastive activations. Let $H_\ell \in \mathbb{R}^{2N \times d}$ be the centered matrix stacking all positive and negative activations at layer $\ell$. The PCA steering vector is:

$$v_\ell^{\text{PCA}} = \arg\max_{\|v\|=1} \; v^\top \left( \frac{1}{2N} H_\ell^\top H_\ell \right) v, \tag{A2}$$

with sign aligned to $v_\ell^{\text{MD}}$ so that the vector points toward the target behavior. The steering function is additive: $\mathcal{F}(h_\ell;\, v_\ell) = h_\ell + \alpha v_\ell^{\text{PCA}}$. Related variants include Sparse Autoencoders (Bricken et al., 2023; Templeton et al., 2024).

**Inference-Time Intervention (ITI).** ITI (Li et al., 2023) treats the steering direction as the decision boundary of a logistic regression probe trained to discriminate positive from negative activations at each layer:

$$\min_{w_\ell, b_\ell} \frac{1}{2N} \sum_{i=1}^{N} \sum_{s \in \{+,-\}} \mathcal{L}\Big( \sigma(w_\ell^\top h_\ell^s(x_i) + b_\ell),\, y_i^s \Big), \tag{A3}$$

where $\sigma$ is the sigmoid, $\mathcal{L}$ is binary cross-entropy, and $y_i^+{=}1$, $y_i^-{=}0$. The normalized probe weight defines the steering vector:

$$v_\ell^{\text{ITI}} = \frac{w_\ell}{\|w_\ell\|}. \tag{A4}$$

The steering function is additive: $\mathcal{F}(h_\ell;\, v_\ell) = h_\ell + \alpha v_\ell^{\text{ITI}}$.

**ODESteer.** ODESteer (Zhao et al., 2026) defines the steering vector implicitly through a barrier function that scores activation desirability as the log-density ratio between positive and negative activations:

$$B_\ell(h) = \log \frac{p^+(h)}{p^-(h)} = w^\top \phi(h) + b, \tag{A5}$$

where $\phi : \mathbb{R}^d \to \mathbb{R}^D$ is a nonlinear feature map and $w, b$ are learned from the contrastive pairs. Unlike additive methods, $\mathcal{F}$ is defined as the solution to an ODE:

$$\mathcal{F}(h_\ell;\, v_\ell) = a(T) \tag{A6}$$

$$\frac{da}{dt} = \nabla_a B_\ell(a(t)) \tag{A7}$$

$$a(0) = h_\ell, \tag{A8}$$

where $T$ is a chosen time horizon. Because the vector field $\nabla_a B_\ell(a(t))$ depends on the current activation, ODESteer performs multi-step adaptive steering rather than a fixed additive update. Conventional activation addition is recovered as the first-order Euler approximation of this ODE (Zhao et al., 2026).

## B Text Perturbation Methods

All three attacks instantiate the objective in Eq. (11). Given input $x = (w_1, \ldots, w_n)$ with words $w_i$, each method iteratively substitutes words in order of importance. The importance of word $w_i$ is measured as the drop in steered probability when $w_i$ is removed:

$$I(w_i) = P(y^+ \mid x,\, v_\ell) - P(y^+ \mid x \setminus w_i,\, v_\ell), \tag{A9}$$

where $x \setminus w_i$ denotes $x$ with $w_i$ deleted. Words are substituted in decreasing order of $I(w_i)$. At each step, a set of candidate replacements $\mathcal{C}(w_i)$ is generated (method-specific, described below), and the best candidate is selected as:

$$w_i^* = \arg\min_{w' \in \mathcal{C}(w_i)} P(y^+ \mid x[w_i \leftarrow w'],\, v_\ell), \tag{A10}$$

where $x[w_i \leftarrow w']$ denotes $x$ with $w_i$ replaced by $w'$. All candidates are filtered by the semantic similarity constraint:

$$\mathcal{C}(w_i) \leftarrow \left\{w' \in \mathcal{C}(w_i) : \mathrm{sim}(x,\, x[w_i \leftarrow w']) \geq \epsilon\right\}, \tag{A11}$$

where $\mathrm{sim}(\cdot,\cdot)$ is computed via the Universal Sentence Encoder (Cer et al., 2018) and $\epsilon$ follows each method's original formulation. The attack halts early once $P(y^+ \mid x', v_\ell) < \delta$.

**TextFooler (Jin et al., 2020).** Candidates are retrieved from a counter-fitted word embedding space (Mrkšić et al., 2016) and filtered by two additional constraints:

$$\mathcal{C}^{\mathrm{TF}}(w_i) = \Big\{w' \;:\; \frac{e(w_i)^\top e(w')}{\|e(w_i)\|\|e(w')\|} \geq \epsilon_w, \quad \mathrm{POS}(w') = \mathrm{POS}(w_i)\Big\}, \tag{A12}$$

where $e(\cdot)$ is the word embedding and $\epsilon_w$ is a word-level similarity threshold. Only synonyms that preserve the part-of-speech tag of $w_i$ are considered.

**TextBugger (Li et al., 2018).** Candidates are generated by applying five character-level transformations to $w_i$:

$$\mathcal{C}^{\mathrm{TB}}(w_i) = \mathrm{Insert}(w_i) \cup \mathrm{Delete}(w_i) \cup \mathrm{Swap}(w_i) \cup \mathrm{Substitute}(w_i) \cup \mathrm{Repeat}(w_i), \tag{A13}$$

where Insert adds a random character, Delete removes one character, Swap exchanges two adjacent characters, Substitute replaces a character with a visually similar one, and Repeat duplicates a character. TextBugger also includes word-level substitutions from an embedding space, analogous to TextFooler.

**BERT-Attack (Li et al., 2020).** Candidates are generated by masking $w_i$ and querying a masked language model for contextually fluent replacements:

$$\mathcal{C}^{\mathrm{BA}}(w_i) = \text{top-}K\left(P_{\mathrm{BERT}}(\cdot \mid x[w_i \leftarrow \texttt{[MASK]}])\right), \tag{A14}$$

where $P_{\mathrm{BERT}}(\cdot \mid \cdot)$ is the masked language model distribution and $K$ is the number of candidates considered. Unlike TextFooler and TextBugger, whose candidates are retrieved from a static embedding space, BERT-Attack conditions on the full surrounding context, producing substitutions that are more grammatically and semantically coherent.

# C Implementation Details

## C.1 Models and Hardware

We evaluate on five models of varying size and architecture, spanning both dense transformers and a Mixture-of-Experts model.

- For Qwen2.5-1.5B (Qwen et al., 2025), we use *Qwen/Qwen2.5-1.5B-Instruct*.
- For Llama-3.2-3B!(Grattafiori et al., 2024), we use *meta-llama/Llama-3.2-3B-Instruct*.
- For Qwen3-4B (Yang et al., 2025), we use *Qwen/Qwen3-4B-Instruct-2507*.
- For Qwen3-14B (Yang et al., 2025), we use *Qwen/Qwen3-14B*.
- For Qwen3-30B-A3B (Yang et al., 2025), we use *Qwen/Qwen3-30B-A3B-Instruct-2507*.

All experiments are run on a single NVIDIA H100 80GB GPU.

## C.2 Dataset

We use six persona datasets from the Anthropic Model Written Exams (Perez et al., 2023): *Religion Following*, *Conscientiousness*, *Self-Improvement*, *Alliance Building*, *Impact Maximization*, and *Self-Aware*. Each persona dataset contains 1,000 binary Yes/No questions. We split each dataset into train, validation, and test sets using an 80/10/10 ratio, yielding 800 training, 100 validation, and 100 test examples per persona. Steering vectors are extracted from the training set $\mathcal{D}_{\mathrm{train}}$, and all attack experiments are conducted on the test set $\mathcal{D}_{\mathrm{test}}$.

## C.3 Text Perturbation

All three attacks are implemented using the TextAttack framework (Morris et al., 2020). Unless stated otherwise, we use TextAttack's default settings. The method-specific hyperparameters are as follows. For TextFooler, the word-level similarity threshold is set to $\epsilon_w = 0.5$ for computation convenience. For BERT-Attack, the number of candidate replacements per word is $K = 10$. Across all three attacks, the early stopping threshold is $\delta = 0.3$: the attack halts once the steered probability of the target token drops below this value, following Eq. (11).

### C.4 Steering Configuration

The steering strength is set to $\alpha = 1.0$ for all additive methods (MD, PCA, ITI), following the configuration of LayerNavigator (Sun et al., 2025). For ODESteer, we use a time horizon of $T = 5.0$ with 10 ODE integration steps. Zhao et al. (2026) originally used $T = 1.0$, but we observe this constant show very weak steerability on our dataset, so we modified it. For ITI, we intervene on all layers rather than restricting to a subset of attention heads, as our focus is on layer selection rather than head selection. All steering vectors are extracted using activations at the final prompt token position, following standard practice (Rimsky et al., 2024).

### C.5 LayerNavigator Configuration

We use LayerNavigator (Sun et al., 2025) with top-$K$=1 layer selection throughout all experiments, meaning a single intervention layer is selected per method and dataset. Per-layer steerability scores are computed using Z-score normalized activations, following the default configuration of the original implementation. As described in Section 3.2, steering vectors used to compute the S-score are extracted from the same dataset $\mathcal{D}_{\text{train}}$ as the activations.

## D Supplementary

| | Method | Impact Maximization | | | Alliance Building | | | Self-aware | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | ASR | b.Atk | a.Atk | ASR | b.Atk | a.Atk | ASR | b.Atk | a.Atk |
| Llama3.2-3B | W/o Steer | – | 0.66 | – | – | 0.76 | – | – | 0.60 | – |
| | PCA | 0.88 | 0.69 | 0.25 (↓0.44) | 0.86 | 0.85 | 0.23 (↓0.62) | 0.56 | 0.61 | 0.27 (↓0.34) |
| | MD | 0.89 | 0.71 | 0.24 (↓0.47) | 0.87 | 0.87 | 0.24 (↓0.63) | 0.71 | 0.67 | 0.27 (↓0.40) |
| | ITI | 0.84 | 0.69 | 0.25 (↓0.44) | 0.87 | 0.87 | 0.25 (↓0.62) | 0.64 | 0.63 | 0.27 (↓0.36) |
| | ODE | 0.89 | 0.75 | 0.25 (↓0.50) | 0.91 | 0.91 | 0.22 (↓0.69) | 0.86 | 0.73 | 0.27 (↓0.46) |
| Qwen3-4B | W/o Steer | – | 0.59 | – | – | 0.76 | – | – | 0.68 | – |
| | PCA | 0.62 | 0.60 | 0.10 (↓0.50) | 0.66 | 0.79 | 0.18 (↓0.61) | 0.68 | 0.68 | 0.12 (↓0.56) |
| | MD | 0.62 | 0.59 | 0.12 (↓0.47) | 0.73 | 0.84 | 0.15 (↓0.69) | 0.69 | 0.70 | 0.13 (↓0.57) |
| | ITI | 0.61 | 0.59 | 0.12 (↓0.47) | 0.60 | 0.76 | 0.20 (↓0.56) | 0.68 | 0.69 | 0.12 (↓0.57) |
| | ODE | 0.61 | 0.60 | 0.12 (↓0.48) | 0.65 | 0.79 | 0.18 (↓0.61) | 0.68 | 0.69 | 0.12 (↓0.57) |
| Qwen3-14B | W/o Steer | – | 0.71 | – | – | 0.79 | – | – | 0.79 | – |
| | PCA | 0.79 | 0.71 | 0.24 (↓0.47) | 0.36 | 0.69 | 0.38 (↓0.31) | 0.70 | 0.81 | 0.26 (↓0.55) |
| | MD | 0.90 | 0.78 | 0.22 (↓0.56) | 0.77 | 0.83 | 0.26 (↓0.57) | 0.94 | 0.89 | 0.22 (↓0.67) |
| | ITI | 0.91 | 0.77 | 0.22 (↓0.55) | 0.81 | 0.87 | 0.26 (↓0.61) | 0.95 | 0.86 | 0.23 (↓0.63) |
| | ODE | 0.83 | 0.72 | 0.24 (↓0.48) | 0.69 | 0.80 | 0.28 (↓0.52) | 0.73 | 0.81 | 0.26 (↓0.55) |
| Q3-30B-A3B | W/o Steer | – | 0.74 | – | – | 0.91 | – | – | 0.93 | – |
| | PCA | 0.78 | 0.74 | 0.15 (↓0.59) | 0.80 | 0.94 | 0.21 (↓0.73) | 0.88 | 0.94 | 0.14 (↓0.80) |
| | MD | 0.78 | 0.75 | 0.15 (↓0.60) | 0.78 | 0.93 | 0.22 (↓0.71) | 0.88 | 0.94 | 0.15 (↓0.79) |
| | ITI | 0.78 | 0.75 | 0.15 (↓0.60) | 0.75 | 0.93 | 0.22 (↓0.71) | 0.85 | 0.94 | 0.16 (↓0.78) |
| | ODE | 0.79 | 0.76 | 0.14 (↓0.62) | 0.78 | 0.93 | 0.22 (↓0.71) | 0.87 | 0.94 | 0.15 (↓0.79) |
| Q2.5-1.5B | W/o Steer | – | 0.74 | – | – | 0.75 | – | – | 0.78 | – |
| | PCA | 0.81 | 0.74 | 0.28 (↓0.46) | 0.78 | 0.75 | 0.30 (↓0.45) | 0.78 | 0.78 | 0.29 (↓0.49) |
| | MD | 0.85 | 0.75 | 0.28 (↓0.47) | 0.85 | 0.76 | 0.27 (↓0.49) | 0.82 | 0.78 | 0.28 (↓0.50) |
| | ITI | 0.85 | 0.73 | 0.27 (↓0.46) | 0.86 | 0.76 | 0.28 (↓0.48) | 0.83 | 0.78 | 0.28 (↓0.50) |
| | ODE | 0.86 | 0.75 | 0.27 (↓0.48) | 0.85 | 0.76 | 0.28 (↓0.48) | 0.83 | 0.79 | 0.29 (↓0.50) |

*Note:* b.Atk and a.Atk denote before attack and after attack. Values in parentheses show absolute decrease from b.Atk to a.Atk.

Table A1: Attack results across three tasks, averaged across three perturbation methods (TextFooler, TextBugger, BERT-Attack). **ASR** is defined in Eq. (12). **W/o Steer** gives the unsteered baseline on clean inputs.

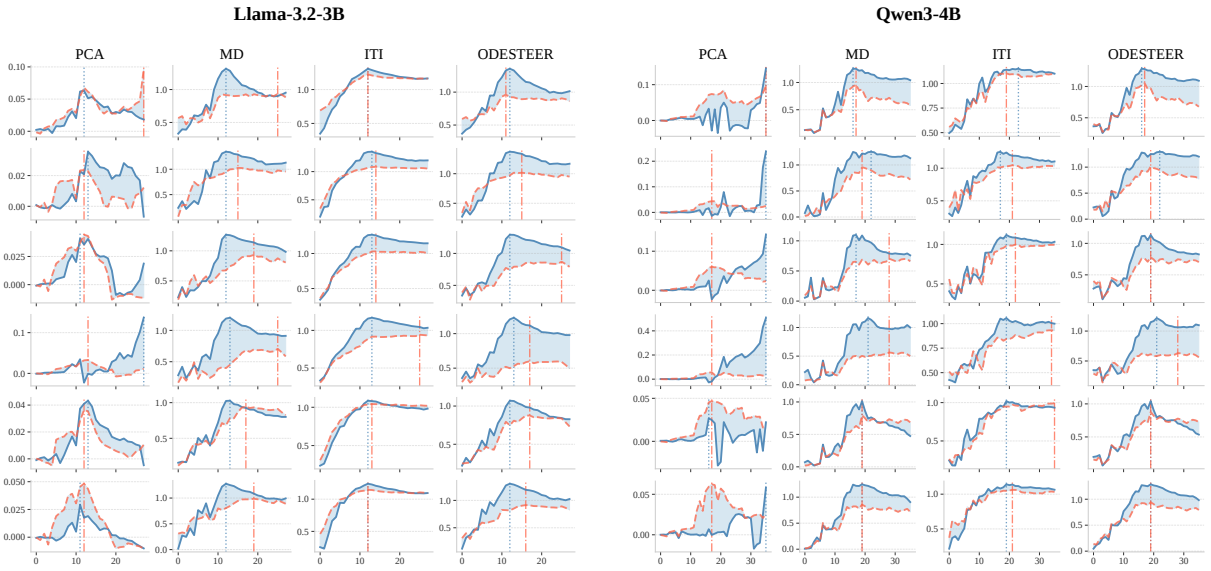


Figure A1: S-score profile for Llama-3.2-3B and Qwen3-4B. Each subplot row is a task in the following order: Religion Following, Conscientiousness, Self-Improvement, Alliance Building, Impact Maximization, Self-Aware.

| | Method | Religion Following | | | Conscientiousness | | | Self-Improvement | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | ASR | b.Atk | a.Atk | ASR | b.Atk | a.Atk | ASR | b.Atk | a.Atk |
| Qwen3-4B | W/o Steer | – | 0.65 | – | – | 0.93 | – | – | 0.72 | – |
| | PCA | 0.61 | 0.66 | 0.16 (↓0.50) | 0.89 | 0.93 | 0.13 (↓0.80) | 0.62 | 0.73 | 0.14 (↓0.59) |
| | MD | 0.66 | 0.70 | 0.16 (↓0.54) | 0.88 | 0.95 | 0.13 (↓0.82) | 0.62 | 0.73 | 0.14 (↓0.59) |
| | ITI | 0.71 | 0.73 | 0.15 (↓0.58) | 0.88 | 0.93 | 0.13 (↓0.80) | 0.63 | 0.74 | 0.14 (↓0.60) |
| | ODE | 0.65 | 0.69 | 0.16 (↓0.53) | 0.90 | 0.93 | 0.12 (↓0.81) | 0.63 | 0.73 | 0.14 (↓0.59) |
| Qwen2.5-1.5B | W/o Steer | – | 0.77 | – | – | 0.70 | – | – | 0.74 | – |
| | PCA | 0.84 | 0.77 | 0.27 (↓0.50) | 0.82 | 0.69 | 0.26 (↓0.43) | 0.78 | 0.74 | 0.29 (↓0.45) |
| | MD | 0.87 | 0.76 | 0.26 (↓0.50) | 0.82 | 0.70 | 0.26 (↓0.44) | 0.83 | 0.75 | 0.28 (↓0.47) |
| | ITI | 0.89 | 0.75 | 0.26 (↓0.49) | 0.83 | 0.69 | 0.26 (↓0.43) | 0.81 | 0.75 | 0.28 (↓0.47) |
| | ODE | 0.90 | 0.76 | 0.26 (↓0.50) | 0.85 | 0.70 | 0.27 (↓0.43) | 0.85 | 0.75 | 0.28 (↓0.47) |

*Note:* b.Atk and a.Atk denote before attack and after attack. Values in parentheses show absolute decrease from b.Atk to a.Atk.

Table A2: Attack results across three tasks for Qwen3-4B and Qwen2.5-1.5B, averaged across three perturbation methods (TextFooler, TextBugger, BERT-Attack). **ASR** is defined in Eq. (12). **W/o Steer** gives the unsteered baseline on clean inputs.

| | | Rel.Fol. | | Conscient. | | Self-Improv. | | Imp.Max. | | All.Bld. | | Self-Awa. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\mathcal{R}_{str}$ | $\mathcal{R}_{dir}$ | $\mathcal{R}_{str}$ | $\mathcal{R}_{dir}$ | $\mathcal{R}_{str}$ | $\mathcal{R}_{dir}$ | $\mathcal{R}_{str}$ | $\mathcal{R}_{dir}$ | $\mathcal{R}_{str}$ | $\mathcal{R}_{dir}$ | $\mathcal{R}_{str}$ | $\mathcal{R}_{dir}$ |
| Llama-3.2-3B | PCA | 0.76 | $0.52_{(\downarrow 0.10)}$ | 0.83 | $0.50_{(0.00)}$ | 0.69 | $0.48_{(\downarrow 0.10)}$ | 0.76 | $0.50_{(0.00)}$ | 0.88 | $0.50_{(0.00)}$ | 0.71 | $0.48_{(\downarrow 0.04)}$ |
| | MD | 0.73 | $0.54_{(\downarrow 0.27)}$ | 0.90 | $0.52_{(\downarrow 0.45)}$ | 0.97 | $0.45_{(\downarrow 0.43)}$ | 0.90 | $0.50_{(\downarrow 0.01)}$ | 0.90 | $0.50_{(\downarrow 0.02)}$ | 0.89 | $0.49_{(\downarrow 0.04)}$ |
| | ITI | 0.45 | $0.75_{(\downarrow 0.20)}$ | 0.98 | $0.49_{(\downarrow 0.37)}$ | 0.97 | $0.49_{(\downarrow 0.12)}$ | 0.89 | $0.47_{(\downarrow 0.42)}$ | 0.91 | $0.50_{(0.00)}$ | 0.82 | $0.49_{(\downarrow 0.01)}$ |
| | ODE | 0.88 | $0.51_{(\downarrow 0.13)}$ | 0.90 | $0.46_{(\downarrow 0.44)}$ | 0.96 | $0.49_{(\downarrow 0.12)}$ | 0.93 | $0.49_{(\downarrow 0.04)}$ | 0.92 | $0.50_{(0.00)}$ | 0.96 | $0.45_{(\downarrow 0.06)}$ |
| Qwen3-4B | PCA | 0.50 | $0.46_{(\uparrow 0.06)}$ | 0.45 | $0.23_{(\downarrow 0.28)}$ | 0.57 | $0.49_{(0.00)}$ | 0.44 | $0.29_{(\downarrow 0.21)}$ | 0.59 | $0.50_{(\uparrow 0.01)}$ | 0.42 | $0.48_{(\uparrow 0.09)}$ |
| | MD | 0.51 | $0.56_{(\downarrow 0.13)}$ | 0.39 | $0.52_{(\uparrow 0.08)}$ | 0.45 | $0.39_{(\downarrow 0.33)}$ | 0.24 | $0.41_{(\downarrow 0.06)}$ | 0.66 | $0.49_{(\downarrow 0.15)}$ | 0.28 | $0.70_{(\downarrow 0.02)}$ |
| | ITI | 0.75 | $0.50_{(0.00)}$ | 0.52 | $0.41_{(\downarrow 0.36)}$ | 0.36 | $0.65_{(\uparrow 0.14)}$ | 0.19 | $0.42_{(\downarrow 0.05)}$ | 0.30 | $0.50_{(\uparrow 0.17)}$ | 0.48 | $0.56_{(\downarrow 0.21)}$ |
| | ODE | 0.61 | $0.47_{(\downarrow 0.21)}$ | 0.66 | $0.37_{(\downarrow 0.31)}$ | 0.52 | $0.46_{(\downarrow 0.26)}$ | 0.22 | $0.41_{(\downarrow 0.06)}$ | 0.65 | $0.45_{(\downarrow 0.33)}$ | 0.44 | $0.55_{(\downarrow 0.34)}$ |
| Qwen3-14B | PCA | 0.75 | $0.16_{(\downarrow 0.52)}$ | 0.63 | $0.21_{(\downarrow 0.28)}$ | 0.85 | $0.10_{(\downarrow 0.64)}$ | 0.71 | $0.08_{(\downarrow 0.50)}$ | 0.38 | $0.50_{(0.00)}$ | 0.86 | $0.08_{(\downarrow 0.59)}$ |
| | MD | 0.87 | $0.48_{(\downarrow 0.05)}$ | 0.89 | $0.41_{(\downarrow 0.49)}$ | 0.94 | $0.33_{(\downarrow 0.48)}$ | 0.87 | $0.46_{(\downarrow 0.15)}$ | 0.89 | $0.28_{(\downarrow 0.65)}$ | 0.93 | $0.47_{(\downarrow 0.18)}$ |
| | ITI | 0.83 | $0.50_{(0.00)}$ | 0.84 | $0.44_{(\downarrow 0.46)}$ | 0.80 | $0.50_{(0.00)}$ | 0.80 | $0.50_{(\downarrow 0.36)}$ | 0.88 | $0.48_{(\downarrow 0.42)}$ | 0.88 | $0.50_{(0.00)}$ |
| | ODE | 0.78 | $0.39_{(\downarrow 0.15)}$ | 0.78 | $0.24_{(\downarrow 0.49)}$ | 0.75 | $0.22_{(\downarrow 0.33)}$ | 0.78 | $0.32_{(\downarrow 0.28)}$ | 0.74 | $0.19_{(\downarrow 0.43)}$ | 0.86 | $0.27_{(\downarrow 0.36)}$ |
| Q3-30B-A3B | PCA | 0.29 | $0.48_{(0.00)}$ | 0.39 | $0.50_{(0.00)}$ | 0.62 | $0.50_{(0.00)}$ | 0.50 | $0.50_{(0.00)}$ | 0.67 | $0.50_{(0.00)}$ | 0.67 | $0.50_{(0.00)}$ |
| | MD | 0.69 | $0.44_{(\downarrow 0.27)}$ | 0.50 | $0.59_{(\uparrow 0.06)}$ | 0.62 | $0.49_{(\downarrow 0.19)}$ | 0.50 | $0.46_{(\downarrow 0.17)}$ | 0.66 | $0.50_{(\uparrow 0.01)}$ | 0.64 | $0.50_{(\downarrow 0.12)}$ |
| | ITI | 0.45 | $0.45_{(\downarrow 0.07)}$ | 0.33 | $0.26_{(\downarrow 0.34)}$ | 0.57 | $0.50_{(\uparrow 0.03)}$ | 0.50 | $0.50_{(\downarrow 0.01)}$ | 0.70 | $0.51_{(\downarrow 0.19)}$ | 0.55 | $0.36_{(\downarrow 0.39)}$ |
| | ODE | 0.74 | $0.47_{(\downarrow 0.31)}$ | 0.51 | $0.55_{(\uparrow 0.03)}$ | 0.58 | $0.50_{(\downarrow 0.13)}$ | 0.57 | $0.46_{(\downarrow 0.21)}$ | 0.70 | $0.50_{(0.00)}$ | 0.66 | $0.47_{(\downarrow 0.03)}$ |
| Qwen2.5-1.5B | PCA | 0.42 | $0.12_{(\downarrow 0.11)}$ | 0.36 | $0.33_{(0.00)}$ | 0.36 | $0.22_{(\uparrow 0.05)}$ | 0.49 | $0.28_{(\uparrow 0.06)}$ | 0.64 | $0.40_{(0.00)}$ | 0.37 | $0.18_{(\downarrow 0.03)}$ |
| | MD | 0.47 | $0.49_{(\downarrow 0.01)}$ | 0.39 | $0.50_{(\downarrow 0.03)}$ | 0.82 | $0.50_{(0.00)}$ | 0.76 | $0.49_{(\downarrow 0.01)}$ | 0.81 | $0.49_{(\downarrow 0.01)}$ | 0.64 | $0.50_{(0.00)}$ |
| | ITI | 0.56 | $0.37_{(\downarrow 0.12)}$ | 0.36 | $0.50_{(0.00)}$ | 0.63 | $0.14_{(\downarrow 0.25)}$ | 0.69 | $0.49_{(\uparrow 0.02)}$ | 0.75 | $0.50_{(0.00)}$ | 0.60 | $0.49_{(\downarrow 0.01)}$ |
| | ODE | 0.47 | $0.49_{(\downarrow 0.01)}$ | 0.44 | $0.52_{(\downarrow 0.05)}$ | 0.84 | $0.50_{(0.00)}$ | 0.79 | $0.50_{(0.00)}$ | 0.83 | $0.50_{(0.00)}$ | 0.64 | $0.50_{(0.00)}$ |

Table A3: Extraction robustness scores with $\mathcal{R}_{\text{str}} \downarrow$ is the fraction of steerable inputs whose steering gain decreases after perturbation. $\mathcal{R}_{\text{dir}} \uparrow$ is the fraction of steerable inputs that remain steerable after perturbation. Subscript shows the change from the clean-input rate.

| | | Rel.Fol. | | Conscient. | | Self-Improv. | | Imp.Max. | | All.Bld. | | Self-Awa. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | DTW | Top1-Shift | DTW | Top1-Shift | DTW | Top1-Shift | DTW | Top1-Shift | DTW | Top1-Shift | DTW | Top1-Shift |
| Llama3.2-3B | PCA | 2.55 | 15.00 | 5.65 | 1.33 | 2.51 | 1.33 | 2.62 | 1.00 | 3.64 | 14.67 | 6.87 | 1.00 |
| | MD | 3.74 | 6.67 | 3.89 | 2.67 | 4.68 | 6.00 | 1.44 | 4.67 | 6.75 | 10.00 | 3.05 | 6.00 |
| | ITI | 1.50 | 0.00 | 3.49 | 1.00 | 3.62 | 4.33 | 1.25 | 3.00 | 3.75 | 12.67 | 1.52 | 0.33 |
| | ODE | 5.14 | 1.33 | 5.16 | 3.33 | 6.95 | 6.33 | 3.37 | 4.67 | 9.39 | 6.33 | 7.86 | 6.00 |
| Qwen3-4B | PCA | 4.22 | 4.67 | 3.03 | 17.33 | 4.06 | 17.00 | 5.92 | 1.00 | 6.09 | 15.67 | 5.39 | 18.00 |
| | MD | 7.54 | 1.00 | 6.93 | 1.67 | 4.35 | 11.33 | 2.83 | 9.33 | 9.47 | 7.67 | 5.42 | 6.00 |
| | ITI | 2.39 | 7.33 | 4.17 | 6.67 | 3.16 | 10.33 | 1.75 | 15.00 | 4.04 | 14.00 | 2.29 | 1.67 |
| | ODE | 7.33 | 1.00 | 6.90 | 3.00 | 4.47 | 10.33 | 2.35 | 0.00 | 9.76 | 8.00 | 5.21 | 6.00 |
| Qwen3-14B | PCA | 8.21 | 13.33 | 6.33 | 7.67 | 6.89 | 1.67 | 7.12 | 1.67 | 8.13 | 17.00 | 3.93 | 13.33 |
| | MD | 6.80 | 0.67 | 6.41 | 3.00 | 7.15 | 5.00 | 4.01 | 3.00 | 9.43 | 4.33 | 8.03 | 2.67 |
| | ITI | 2.05 | 0.67 | 4.50 | 1.33 | 3.29 | 1.67 | 2.79 | 2.00 | 4.69 | 1.67 | 3.93 | 2.00 |
| | ODE | 5.64 | 1.67 | 6.39 | 3.00 | 6.95 | 3.67 | 3.37 | 2.33 | 9.39 | 3.00 | 7.86 | 0.67 |
| Q3-30B-A3B | PCA | 9.61 | 20.00 | 4.76 | 10.67 | 4.43 | 20.00 | 6.60 | 20.00 | 9.30 | 17.00 | 3.16 | 15.00 |
| | MD | 5.15 | 6.33 | 9.86 | 2.00 | 11.27 | 4.33 | 4.56 | 6.00 | 12.86 | 11.00 | 9.48 | 11.33 |
| | ITI | 1.53 | 0.33 | 3.90 | 7.00 | 3.88 | 3.33 | 1.81 | 6.67 | 7.32 | 4.00 | 4.73 | 4.00 |
| | ODE | 5.26 | 4.33 | 9.09 | 3.33 | 10.63 | 0.00 | 3.60 | 6.33 | 13.82 | 6.00 | 10.23 | 15.67 |
| Qwen2.5-1.5B | PCA | 5.68 | 15.00 | 4.47 | 4.33 | 0.13 | 0.00 | 3.39 | 0.00 | 3.54 | 0.00 | 2.02 | 4.33 |
| | MD | 2.64 | 0.00 | 2.83 | 1.33 | 5.28 | 3.33 | 3.26 | 3.67 | 7.92 | 2.33 | 4.57 | 0.00 |
| | ITI | 2.06 | 4.33 | 2.65 | 21.00 | 2.24 | 4.67 | 2.92 | 4.67 | 2.37 | 4.00 | 2.84 | 5.00 |
| | ODE | 2.14 | 3.33 | 2.79 | 1.67 | 4.32 | 4.33 | 2.20 | 4.00 | 6.84 | 2.00 | 4.96 | 0.33 |

Table A4: DTW scores and Top-1 layer difference in LayerNavigator. Red cells mark high DTW scores ($\geq 5$), while orange cells mark large Top-1 layer shifts ($\geq 10$) or near-high DTW scores.

| | | Rel.Fol. | | Conscient. | | Self-Improv. | | All.Bld. | | Imp.Max. | | Self-Awa. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Clean | Perturb. | Clean | Perturb. | Clean | Perturb. | Clean | Perturb. | Clean | Perturb. | Clean | Perturb. |
| Llama3.2-3B | PCA | 1.0 | $14.0_{(\uparrow 13.0)}$ | 1.0 | $13.0_{(\uparrow 12.0)}$ | 1.0 | $2.0_{(\uparrow 1.0)}$ | 1.0 | $14.0_{(\uparrow 13.0)}$ | 1.0 | $13.0_{(\uparrow 12.0)}$ | 2.0 | $15.0_{(\uparrow 13.0)}$ |
| | MD | 1.0 | $0.0_{(\downarrow 1.0)}$ | 1.0 | $1.3_{(\uparrow 0.3)}$ | 1.0 | $3.3_{(\uparrow 2.3)}$ | 1.0 | $11.0_{(\uparrow 10.0)}$ | 0.0 | $5.0_{(\uparrow 5.0)}$ | 1.0 | $5.0_{(\uparrow 4.0)}$ |
| | ITI | 0.0 | $0.0_{(0.0)}$ | 0.0 | $10.0_{(\uparrow 10.0)}$ | 0.0 | $3.7_{(\uparrow 3.7)}$ | 1.0 | $14.0_{(\uparrow 13.0)}$ | 1.0 | $3.0_{(\uparrow 2.0)}$ | 0.0 | $15.0_{(\uparrow 15.0)}$ |
| | ODE | 0.0 | $8.0_{(\uparrow 8.0)}$ | 1.0 | $1.0_{(0.0)}$ | 1.0 | $8.7_{(\uparrow 7.7)}$ | 0.0 | $8.0_{(\uparrow 8.0)}$ | 0.0 | $8.0_{(\uparrow 8.0)}$ | 1.0 | $8.3_{(\uparrow 7.3)}$ |
| Qwen3-4B | PCA | 13.0 | $13.3_{(\uparrow 0.3)}$ | 13.0 | $12.3_{(\downarrow 0.7)}$ | 19.0 | $13.7_{(\downarrow 5.3)}$ | 3.0 | $13.7_{(\uparrow 10.7)}$ | 5.0 | $8.0_{(\uparrow 3.0)}$ | 13.0 | $12.3_{(\downarrow 0.7)}$ |
| | MD | 6.0 | $6.0_{(0.0)}$ | 0.0 | $0.0_{(0.0)}$ | 4.0 | $2.3_{(\downarrow 1.7)}$ | 1.0 | $12.7_{(\uparrow 11.7)}$ | 3.0 | $3.0_{(0.0)}$ | 3.0 | $2.7_{(\downarrow 0.3)}$ |
| | ITI | 0.0 | $3.0_{(\uparrow 3.0)}$ | 5.0 | $5.0_{(0.0)}$ | 2.0 | $1.0_{(\downarrow 1.0)}$ | 2.0 | $1.3_{(\downarrow 0.7)}$ | 2.0 | $1.3_{(\downarrow 0.7)}$ | 2.0 | $2.0_{(0.0)}$ |
| | ODE | 6.0 | $6.0_{(0.0)}$ | 0.0 | $11.7_{(\uparrow 11.7)}$ | 19.0 | $6.3_{(\downarrow 12.7)}$ | 1.0 | $12.0_{(\uparrow 11.0)}$ | 3.0 | $1.0_{(\downarrow 2.0)}$ | 2.0 | $2.3_{(\uparrow 0.3)}$ |
| Qwen3-14B | PCA | 14.0 | $20.0_{(\uparrow 6.0)}$ | 7.0 | $19.0_{(\uparrow 12.0)}$ | 16.0 | $20.0_{(\uparrow 4.0)}$ | 3.0 | $11.0_{(\uparrow 8.0)}$ | 18.0 | $20.0_{(\uparrow 2.0)}$ | 17.0 | $6.0_{(\downarrow 11.0)}$ |
| | MD | 1.0 | $8.7_{(\uparrow 7.7)}$ | 1.0 | $15.0_{(\uparrow 14.0)}$ | 2.0 | $17.0_{(\uparrow 15.0)}$ | 4.0 | $17.0_{(\uparrow 13.0)}$ | 0.0 | $8.0_{(\uparrow 8.0)}$ | 1.0 | $6.0_{(\uparrow 5.0)}$ |
| | ITI | 1.0 | $5.0_{(\uparrow 4.0)}$ | 0.0 | $4.0_{(\uparrow 4.0)}$ | 1.0 | $21.0_{(\uparrow 20.0)}$ | 1.0 | $11.7_{(\uparrow 10.7)}$ | 1.0 | $5.3_{(\uparrow 4.3)}$ | 0.0 | $11.0_{(\uparrow 11.0)}$ |
| | ODE | 0.0 | $18.0_{(\uparrow 18.0)}$ | 0.0 | $12.3_{(\uparrow 12.3)}$ | 3.0 | $22.0_{(\uparrow 19.0)}$ | 4.0 | $5.7_{(\uparrow 1.7)}$ | 2.0 | $17.0_{(\uparrow 15.0)}$ | 2.0 | $12.7_{(\uparrow 10.7)}$ |
| Q3-30B-A3B | PCA | 35.0 | $10.0_{(\downarrow 25.0)}$ | 1.0 | $3.0_{(\uparrow 2.0)}$ | 3.0 | $10.0_{(\uparrow 7.0)}$ | 0.0 | $10.0_{(\uparrow 10.0)}$ | 26.0 | $10.0_{(\downarrow 16.0)}$ | 0.0 | $10.0_{(\uparrow 10.0)}$ |
| | MD | 1.0 | $4.0_{(\uparrow 3.0)}$ | 23.0 | $5.0_{(\downarrow 18.0)}$ | 1.0 | $18.7_{(\uparrow 17.7)}$ | 4.0 | $25.0_{(\uparrow 21.0)}$ | 1.0 | $18.7_{(\uparrow 17.7)}$ | 3.0 | $16.7_{(\uparrow 13.7)}$ |
| | ITI | 3.0 | $6.3_{(\uparrow 3.3)}$ | 9.0 | $3.0_{(\downarrow 6.0)}$ | 3.0 | $12.3_{(\uparrow 9.3)}$ | 4.0 | $12.7_{(\uparrow 8.7)}$ | 4.0 | $6.0_{(\uparrow 2.0)}$ | 1.0 | $8.0_{(\uparrow 7.0)}$ |
| | ODE | 25.0 | $26.0_{(\uparrow 1.0)}$ | 26.0 | $13.3_{(\downarrow 12.7)}$ | 13.0 | $26.0_{(\uparrow 13.0)}$ | 0.0 | $28.7_{(\uparrow 28.7)}$ | 1.0 | $18.3_{(\uparrow 17.3)}$ | 9.0 | $33.0_{(\uparrow 24.0)}$ |
| Qwen2.5-1.5B | PCA | 11.0 | $13.0_{(\uparrow 2.0)}$ | 5.0 | $9.7_{(\uparrow 4.7)}$ | 10.0 | $1.3_{(\downarrow 8.7)}$ | 10.0 | $5.7_{(\downarrow 4.3)}$ | 10.0 | $12.0_{(\uparrow 2.0)}$ | 10.0 | $9.3_{(\downarrow 0.7)}$ |
| | MD | 5.0 | $2.3_{(\downarrow 2.7)}$ | 0.0 | $1.3_{(\uparrow 1.3)}$ | 6.0 | $5.3_{(\downarrow 0.7)}$ | 3.0 | $15.7_{(\uparrow 12.7)}$ | 7.0 | $5.0_{(\downarrow 2.0)}$ | 1.0 | $3.3_{(\uparrow 2.3)}$ |
| | ITI | 3.0 | $3.7_{(\uparrow 0.7)}$ | 2.0 | $2.0_{(0.0)}$ | 3.0 | $14.7_{(\uparrow 11.7)}$ | 2.0 | $12.3_{(\uparrow 10.3)}$ | 7.0 | $4.3_{(\downarrow 2.7)}$ | 1.0 | $11.0_{(\uparrow 10.0)}$ |
| | ODE | 15.0 | $5.0_{(\downarrow 10.0)}$ | 0.0 | $8.7_{(\uparrow 8.7)}$ | 3.0 | $14.7_{(\uparrow 11.7)}$ | 3.0 | $10.7_{(\uparrow 7.7)}$ | 3.0 | $4.3_{(\uparrow 1.3)}$ | 15.0 | $15.3_{(\uparrow 0.3)}$ |

Table A5: L1 distance between LayerNavigator's selected layer $\ell^*$ and the true optimal layer, on clean and perturbed inputs. The subscript shows the change (Perturbed $-$ Clean). Red cells ($\uparrow$) indicate the distance increased under perturbation — LayerNavigator moved further from the optimal layer. Green cells ($\downarrow$) indicate it decreased.

| | | Rel.Fol. | | Conscient. | | Self-Improv. | | Imp.Max. | | All.Build. | | Self-aware | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | @3 | @5 | @3 | @5 | @3 | @5 | @3 | @5 | @3 | @5 | @3 | @5 |
| Llama3.2-3B | PCA | 0.28 | 0.47 | 0.28 | 0.32 | 0.46 | 0.61 | 0.56 | 0.70 | 0.00 | 0.00 | 0.50 | 0.61 |
| | MD | 0.37 | 0.37 | 0.17 | 0.29 | 0.00 | 0.03 | 0.00 | 0.03 | 0.00 | 0.00 | 0.00 | 0.04 |
| | ITI | 0.83 | 0.90 | 0.33 | 0.49 | 0.37 | 0.40 | 0.30 | 0.41 | 0.00 | 0.00 | 0.82 | 0.77 |
| | ODE | 0.32 | 0.44 | 0.22 | 0.35 | 0.00 | 0.04 | 0.00 | 0.06 | 0.00 | 0.06 | 0.00 | 0.09 |
| Qwen3-4B | PCA | 0.70 | 0.67 | 0.09 | 0.10 | 0.00 | 0.00 | 0.50 | 0.54 | 0.00 | 0.00 | 0.00 | 0.00 |
| | MD | 0.60 | 0.62 | 0.28 | 0.34 | 0.17 | 0.23 | 0.33 | 0.30 | 0.00 | 0.01 | 0.37 | 0.37 |
| | ITI | 0.00 | 0.12 | 0.24 | 0.25 | 0.04 | 0.11 | 0.09 | 0.11 | 0.00 | 0.00 | 0.46 | 0.54 |
| | ODE | 0.56 | 0.60 | 0.17 | 0.28 | 0.17 | 0.20 | 0.74 | 0.62 | 0.00 | 0.01 | 0.37 | 0.42 |
| Qwen3-14B | PCA | 0.20 | 0.18 | 0.04 | 0.07 | 0.57 | 0.61 | 0.63 | 0.58 | 0.00 | 0.00 | 0.20 | 0.15 |
| | MD | 0.46 | 0.53 | 0.11 | 0.29 | 0.09 | 0.18 | 0.20 | 0.34 | 0.04 | 0.12 | 0.32 | 0.43 |
| | ITI | 0.46 | 0.53 | 0.39 | 0.41 | 0.22 | 0.33 | 0.04 | 0.12 | 0.26 | 0.30 | 0.24 | 0.26 |
| | ODE | 0.31 | 0.37 | 0.26 | 0.40 | 0.35 | 0.46 | 0.22 | 0.38 | 0.26 | 0.33 | 0.68 | 0.72 |
| Q3-30B-A3B | PCA | 0.00 | 0.03 | 0.09 | 0.12 | 0.00 | 0.00 | 0.00 | 0.00 | 0.17 | 0.22 | 0.30 | 0.27 |
| | MD | 0.28 | 0.27 | 0.39 | 0.43 | 0.41 | 0.45 | 0.37 | 0.36 | 0.00 | 0.00 | 0.00 | 0.08 |
| | ITI | 0.61 | 0.54 | 0.46 | 0.48 | 0.54 | 0.66 | 0.30 | 0.32 | 0.26 | 0.39 | 0.45 | 0.49 |
| | ODE | 0.54 | 0.52 | 0.72 | 0.67 | 0.87 | 0.77 | 0.32 | 0.32 | 0.00 | 0.01 | 0.00 | 0.10 |
| Qwen2.5-1.5B | PCA | 0.09 | 0.12 | 0.44 | 0.43 | 0.65 | 0.61 | 0.61 | 0.52 | 0.76 | 0.67 | 0.70 | 0.65 |
| | MD | 0.61 | 0.52 | 0.17 | 0.23 | 0.04 | 0.14 | 0.13 | 0.25 | 0.17 | 0.28 | 0.65 | 0.72 |
| | ITI | 0.26 | 0.33 | 0.44 | 0.48 | 0.07 | 0.14 | 0.04 | 0.12 | 0.04 | 0.20 | 0.22 | 0.31 |
| | ODE | 0.41 | 0.33 | 0.24 | 0.25 | 0.20 | 0.32 | 0.07 | 0.18 | 0.17 | 0.35 | 0.61 | 0.67 |

Table A6: RBO scores across models and tasks. RBO@K reports rank-biased overlap over the top-$K$ layers. Lower values indicate larger changes in the selected layer ranking. Red cells mark high instability ($< 0.20$), and orange cells mark medium-high instability (0.20–0.40).

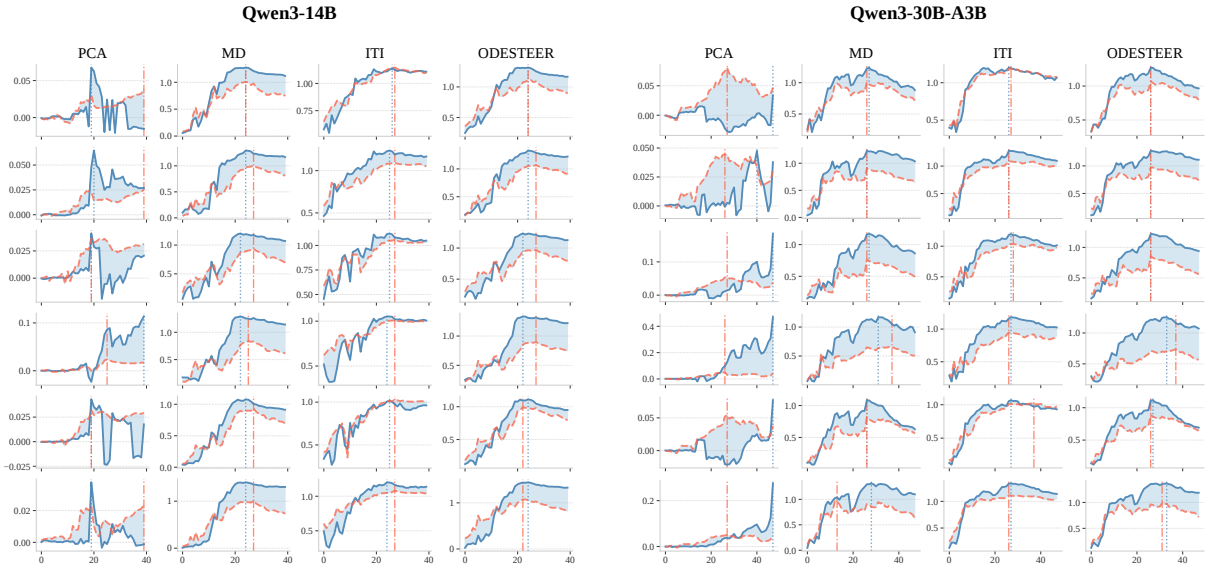


Figure A2: S-score profile for Qwen3-14B and Qwen3-30B-A3B. Each subplot row is a task in the following order: Religion Following, Conscientiousness, Self-Improvement, Alliance Building, Impact Maximization, Self-Aware.

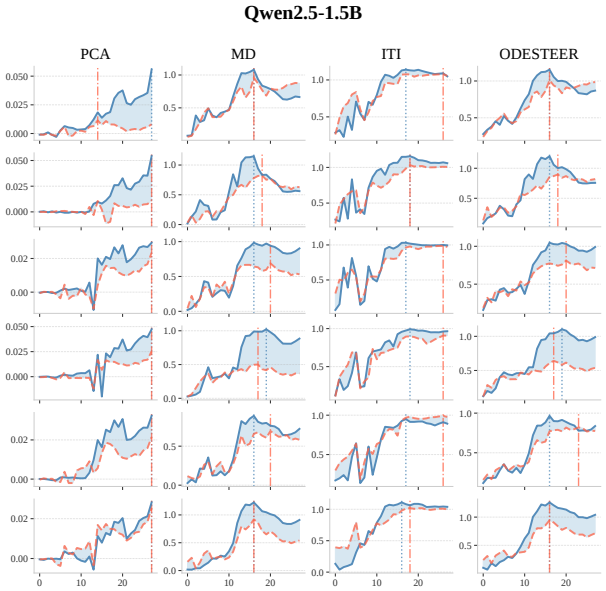


Figure A3: S-score profile for Qwen2.5-1.5B. Each row is a task in the following order: Religion Following, Conscientiousness, Self-Improvement, Alliance Building, Impact Maximization, Self-Aware.

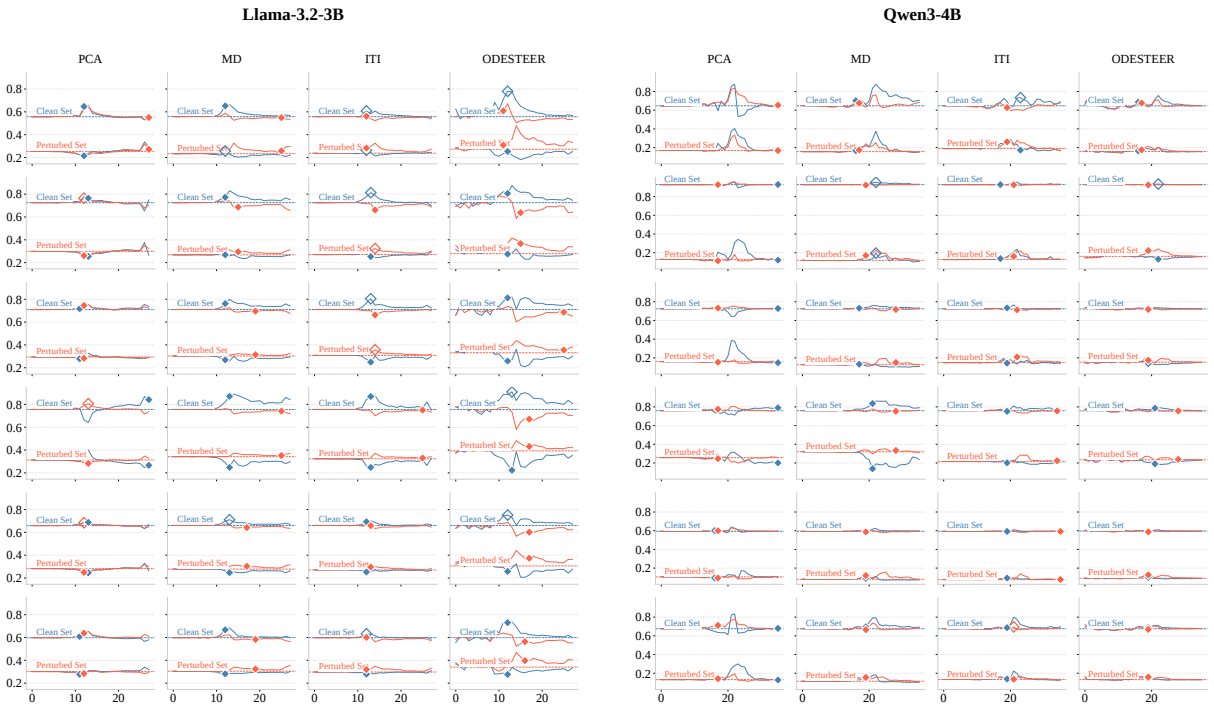


Figure A4: Per-layer performance for Llama-3.2-3B and Qwen3-4B. Each subplot row is a task in the following order: Religion Following, Conscientiousness, Self-Improvement, Alliance Building, Impact Maximization, Self-Aware.

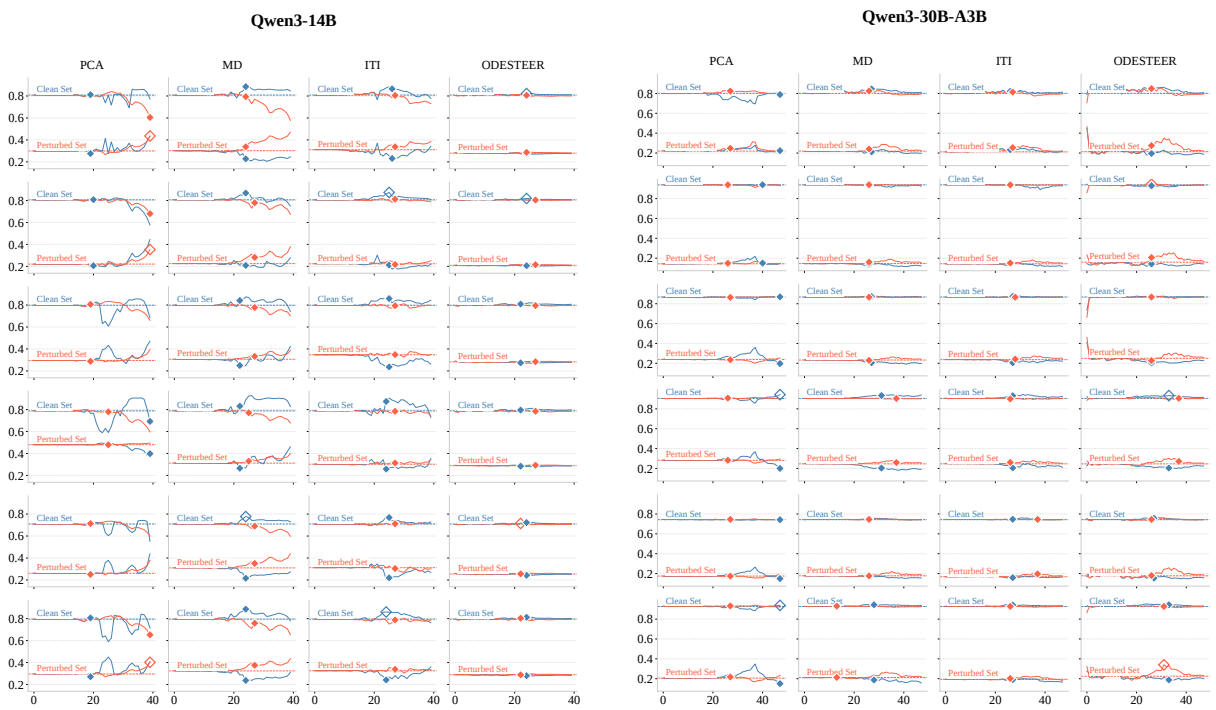


Figure A5: Per-layer performance for Qwen3-14B and Qwen3-30B-A3B. Each subplot row is a task in the following order: Religion Following, Conscientiousness, Self-Improvement, Alliance Building, Impact Maximization, Self-Aware.

Figure A6: Per-layer performance for Qwen2.5-1.5B. Each row is a task in the following order: Religion Following, Conscientiousness, Self-Improvement, Alliance Building, Impact Maximization, Self-Aware.