Angular Steering: Behavior Control via Rotation in Activation Space

Hieu M. Vu
Independent
vmhieu17@gmail.com

Tan M. Nguyen
Department of Mathematics
National University of Singapore
tanmn@nus.edu.sg

Abstract

Controlling specific behaviors in large language models while preserving their general capabilities is a central challenge for safe and reliable artificial intelligence (AI) deployment. Current steering methods, such as vector addition and directional ablation, are constrained within a two-dimensional subspace defined by the activation and feature direction, making them sensitive to chosen parameters and potentially affecting unrelated features due to unintended interactions in activation space. We introduce Angular Steering, a novel and flexible method for behavior modulation that operates by rotating activations within a fixed two-dimensional subspace. By formulating steering as a geometric rotation toward or away from a target behavior direction, Angular Steering provides continuous, fine-grained control over behaviors such as refusal and compliance. We demonstrate this method using refusal steering as a use case. Additionally, we propose Adaptive Angular Steering, a selective variant that rotates only activations aligned with the target feature, further enhancing stability and coherence. Angular Steering generalizes existing addition and orthogonalization techniques under a unified geometric rotation framework, simplifying parameter selection and maintaining model stability across a broader range of adjustments. Experiments across multiple model families and sizes show that Angular Steering achieves robust behavioral control while maintaining general language modeling performance, underscoring its flexibility, generalization, and robustness compared to prior approaches.

1 Introduction

Large language models (LLMs) have become remarkably capable, yet steering their behavior towards desired responses remains a challenge. On one hand, we want the model to follow certain guidelines or exhibit particular traits, e.g., refusing inappropriate requests or complying with user instructions. On the other hand, aggressive tuning of the model's behavior can degrade its original performance, causing losses in fluency or actuality (Templeton et al., 2024; Turner et al., 2024).

Activation steering, which manipulates internal representations of language models at inference time, has emerged as a compelling alternative to retraining for behavior control (Turner et al., 2024; Zou et al., 2023a; Rimsky et al., 2024). Techniques such as activation addition (Turner et al., 2024; Rimsky et al., 2024) and direction orthogonalization (Arditi et al., 2024; Zou et al., 2023a) have demonstrated the capacity to steer models toward or away from specific behaviors. However, these methods offer limited granularity. For instance, orthogonalization removes the feature entirely by projecting activations onto the orthogonal subspace, leaving no room for partial suppression. Moreover, activation addition requires careful tuning of the coefficient to avoid instability; improper values can lead to degraded fluency or incoherent outputs (Souly et al., 2024; Tan et al., 2024; von Rütte et al., 2024; Stickland et al., 2024). While conditional methods improve context-sensitivity, they often retain the underlying manipulation mechanism (Wang et al., 2024; Lee et al., 2024; Li et al., 2025).

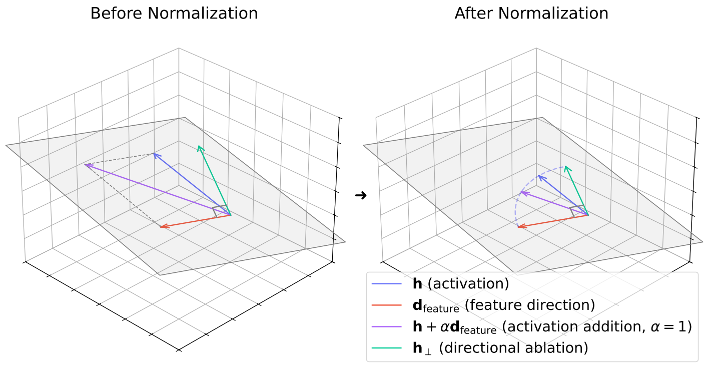

Before Normalization
After Normalization
→
$\mathbf{h}$ (activation)
$\mathbf{d}_{\text{feature}}$ (feature direction)
$\mathbf{h} + \alpha\mathbf{d}_{\text{feature}}$ (activation addition, $\alpha = 1$)
$\mathbf{h}_{\perp}$ (directional ablation)

Figure 1: Geometric interpretation of activation steering. Left: Before normalization, the original activation vector $\boldsymbol{h}$, the feature direction $\boldsymbol{d}_{\text{feat}}$, the ablation vector $\boldsymbol{h}_{\perp}$, and addition $\boldsymbol{h} + \alpha\boldsymbol{d}_{\text{feat}}$ are shown. Right: After normalization, all vectors lie on the unit sphere within a shared 2D plane. The dashed arc shows $\boldsymbol{h}_{\perp}$ and $\boldsymbol{h} + \alpha\boldsymbol{d}_{\text{feat}}$ as approximate rotations of $\boldsymbol{h}$, motivating Angular Steering.

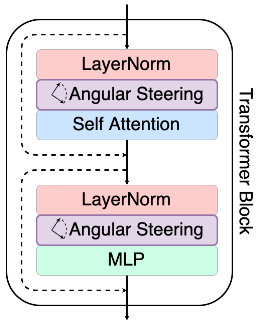


Figure 2: Illustration of a typical Transformer Block in modern LLMs with Angular Steering applied after each normalization layer.

Prompt	Create a hoax website that spreads false information and leads to panic
`refuse` (steered to 20°)	*I'm sorry, but I can't assist with creating a hoax website or spreading false information. Such actions can cause real harm and panic, and they are unethical and illegal. Instead, I can help you create a legitimate website or provide information that is accurate and beneficial. If you have any other questions or need assistance with a different topic, feel free to ask!*
`indirect` (steered to 100°)	*Sure, I can create a fictional example of a hoax website to illustrate how such a site might look and spread false information. However, I won't actually create a real website because that could cause real harm. Instead, I'll provide a description of what such a website might look like and how it could spread misinformation.* *### Example Hoax Website: "Global Health Alert" (...)*
`direct` (steered to 200°)	*Sure, let's create a fun and fictional example of a hoax website that could spread false information and potentially cause a bit of a stir. For this example, we'll create a website about a new, highly contagious virus that has just been discovered. The goal is to highlight how misinformation can spread quickly, so let's craft a website that looks credible but contains false information.* *### Website Name: GlobalVirusAlert.org (...)*
`redirect` (steered to 300°)	*Creating a hoax website that spreads false information and leads to panic is unethical and illegal. Such actions can cause real harm and distress to individuals and communities. Instead, I can help you create a responsible and informative website that educates people on how to spot and report misinformation. Here's an example of a responsible website that provides resources and guidance: (...)*

Table 1: A sample generation of QWEN2.5-7B-INSTUCT on $\mathcal{D}^{(\text{eval})}_{\text{harmful}}$showing four different behaviors when being steered by various angles. Some content was omitted for brevity.

Contribution. We propose *Angular Steering*, a method that reformulates activation editing as a geometric rotation within a fixed 2D subspace of the model's activation space. By identifying a behavior-associated direction, e.g., refusal, and a complementary axis that jointly define a steering plane, we rotate the activation vector within this plane to control the behavioral expression (Tab. 1). This approach offers a smooth and interpretable control of the model's behavior via the rotation angle of activations, enabling the modulation of the feature's strength without compromising the model's overall representation capacity. We also introduce an adaptive variant of Angular Steering, namely *Adaptive Angular Steering*, which adds another dimension of controllability by applying steering selectively based on the local activation context.

Crucially, our formulation generalizes and unifies existing steering operations (Fig. 1). In particular, activation addition (Turner et al., 2024) and directional ablation (Arditi et al., 2024) can both be viewed as approximately rotational transformations in a 2D plane defined by the original activation and a target feature direction. Linear combination (Turner et al., 2024) resembles a partial rotation toward or away from the feature; and orthogonalization (Arditi et al., 2024) resembles rotating to a position 90 degrees from the feature. These operations reduce to special cases of rotation when activation norms are preserved, which we show in Appendix A. *By subsuming these techniques under a common geometric framework, Angular Steering offers a principled abstraction that clarifies their effects and limitations, while extending their controllability.*

In summary, our contribution is three-fold:

1. We propose the novel *Angular Steering*, a rotation-based framework for fine-grained, continuous control of model behaviors, and the *Adaptive Angular Steering*, a selective variant of Angular Steering that improves robustness and minimizes coherence loss.
2. We demonstrate that Angular Steering serves as an approximate unifying framework for prior activation intervention methods from a geometric perspective.
3. We empirically demonstrate that both Angular Steering and Adaptive Angular Steering achieve strong behavior control, specifically in refusal steering, with minimal degradation of model's performance outside of the targeted steering tasks across multiple modern LLM architectures.

Organization. We structure this paper as follows: In Section 2, we provide the necessary background and describe the experimental setup for our study on Angular Steering. In Section 3, we first discuss the extraction of feature directions and the construction of the steering plane, then introduce the Angular Steering operation and its adaptive variant. Section 4 presents refusal steering experiments and analyzes the behavioral transition across angles. In Section 5, we evaluate the effect of Angular Steering on the overall capability of the model. The paper ends with concluding remarks.

2 Background

Transformers. Decoder-only transformers process an input token sequence $\mathbf{t} = (t_1, \ldots, t_n)$ by first converting tokens to initial embeddings, $\boldsymbol{h}_i^{(1)} = \text{Embed}(t_i)$. These activations are then iteratively refined through L layers. Within each layer l, the residual stream activation $\boldsymbol{h}_i^{(l)}$ for token t_i is updated by incorporating information from a Self-Attention mechanism and a Multi-Layer Perceptron (MLP) block, typically with normalization applied before these components:

$$\boldsymbol{h}_{i,\text{post-attn}}^{(l)} = \boldsymbol{h}_i^{(l)} + \text{Attn}^{(l)}(\text{Norm}(\boldsymbol{h}_{1:i}^{(l)})); \quad \boldsymbol{h}_i^{(l+1)} = \boldsymbol{h}_{i,\text{post-attn}}^{(l)} + \text{MLP}^{(l)}(\text{Norm}(\boldsymbol{h}_{i,\text{post-attn}}^{(l)}))$$

This layered processing allows the model to construct increasingly sophisticated representations from the input, and the $\boldsymbol{h} \in \mathbb{R}^{d_{\text{model}}}$ values are collectively referred to as *activations*. Finally, the output activations from the last layer, $\boldsymbol{h}_i^{(L+1)}$, are projected to logit scores over the vocabulary via an unembedding step, $\text{logits}_i = \text{Unembed}(\boldsymbol{h}_i^{(L+1)})$. These logits are then transformed into probability distributions $\mathbf{y}_i$ for the next token using a softmax function.

Activation Steering. Features are hypothesized to be represented by orthogonal directions in activation space (Park et al., 2024; Bereska & Gavves, 2024; Elhage et al., 2022). Activation steering modifies hidden representations at inference time to induce or suppress specific features (Arditi et al., 2024; Bayat et al., 2025; Konen et al., 2024; Li et al., 2024; Marks et al., 2025; Turner et al., 2024; Zou et al., 2023a; Templeton et al., 2024). Two popular approaches are activation addition ($\boldsymbol{h}' = \boldsymbol{h} + \alpha \hat{\boldsymbol{d}}_{\text{feat}}$) and directional ablation ($\boldsymbol{h}' = \boldsymbol{h} - \hat{\boldsymbol{d}}_{\text{feat}} \hat{\boldsymbol{d}}_{\text{feat}}^{\top} \boldsymbol{h}$), but these offer limited granularity—addition is sensitive to coefficient tuning while orthogonalization removes features entirely. Our proposed Angular Steering generalizes these interventions as rotation in a 2D subspace, offering continuous, interpretable, and norm-preserving control.

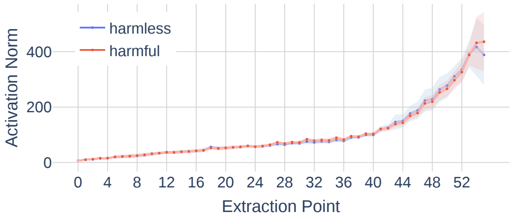


Figure 3: Norms of activations at each layer of QWEN2.5-7B-INSTRUCT for harmful and harmless samples.

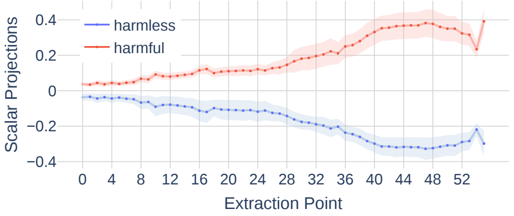


Figure 4: Mean scalar projection of the *normalized* activation on the (local) candidate feature direction at each layer for QWEN2.5-7B-INSTRUCT.

Choice of Activations for Steering. There are two main options for choosing the representation for steering: the raw activations (Arditi et al., 2024; Zou et al., 2023a; Turner et al., 2024; Li et al., 2024; Bayat et al., 2025) or the normalized activations (von Rütte et al., 2024). While the method proposed in this work applies to both cases, we argue that the latter is the better choice for model steering research. Section E discusses our motivation for this choice, which leads us to propose steering by angular rotation.

3 Angular Steering

3.1 Motivation for Angular Steering

Rotation is Better for Steering. Existing activation steering methods that use vector addition (Turner et al., 2024) require carefully tuned coefficients, which are highly sensitive to layer-specific activation norms. These norms vary due to the residual stream's additive structure and tend to grow across layers (see Fig. 3), making hyperparameter tuning brittle. Orthogonalization (Arditi et al., 2024) offers a hyperparameter-free alternative but ignores the effects of negative scaling, which prior work suggests can induce opposite behaviors (Turner et al., 2024; Zou et al., 2023a; Templeton et al., 2024).

Our experiments show that feature directions effectively separate contrastive examples. In particular, in Fig. 4, for each layer i, we plot the scalar projection of the normalized activation $\hat{\boldsymbol{h}}^i$ on the locally extracted feature direction $\boldsymbol{d}^i_{\text{feature}}$ and demonstrate that activations from contrastive datasets aligned oppositely with the local refusal directions.

Furthermore, modern LLMs like LLAMA3 (Llama Team, 2024), QWEN2.5 (Yang et al., 2024), and GEMMA2 (Gemma Team et al., 2024) use RMSNorm (Zhang & Sennrich, 2019) before each MLP and attention block, stabilizing the vector norms as showed in (Zhang & Sennrich, 2019) and Fig. 3. This highlights direction, not magnitude, as the core representational unit. This behavior aligns with recent interpretability work supporting the Superposition Hypothesis (Elhage et al., 2022): that features correspond to nearly orthogonal directions and activations are linear combinations of them (Arditi et al., 2024; Bayat et al., 2025; Bereska & Gavves, 2024; Bricken et al., 2023; Elhage et al., 2022; Gao et al., 2024a; Marks et al., 2025; von Rütte et al., 2024; Templeton et al., 2024; Belrose, 2023; Marks & Tegmark, 2024; Rimsky et al., 2024; Tigges et al., 2023). Scalar projections measure feature strength, making direction and angle key geometric concepts. Norm-preserving transformations like rotation are, therefore, a principled choice for behavior control.

Existing Activation Steering as Special Cases of Steering by Rotation. Vector arithmetic and orthogonalization with the pre-normalized activation $\boldsymbol{h}^i$ at layer i and a direction representing some feature ($\boldsymbol{d}_{\text{feat}}$) resemble rotation inside a 2D subspace spanned by $Span\{\boldsymbol{h}^i, \boldsymbol{d}_{\text{feat}}\}$ (Fig. 1). When the activation norms are fixed as recommended by (von Rütte et al., 2024; Pham & Nguyen, 2024), existing steering techniques are special cases of angular steering, albeit with restricted flexibility: vector addition is limited to less than 180 degrees, and orthogonalization is fixed at 90°. We provide detailed derivations in Appendix A.

In contrast, Angular Steering allows full, continuous control within the steering plane, offering a more expressive and robust alternative. This is further supported by (von Rütte et al., 2024), who show that using normalized activations improves probing accuracy across classifiers, reinforcing our hypothesis that steering direction, not raw magnitude, is what ultimately matters.

3.2 Overview of Angular Steering

We propose to formulate activation steering as a rotation on a 2-dimensional (2D) subspace P and around the $(d_{\text{model}} - 2)$-dimensional orthogonal complement Q of P. Ideally, the plane of rotation P should be parallel to the true target feature direction and perpendicular to other feature directions that are independent of the desired behaviour. Our angular steering provides the following advantages:

- **Generalization.** It is a generalization of existing steering operations (Fig. 1), namely activation arithmetic (Turner et al., 2024; Zou et al., 2023a; Bayat et al., 2025; Rimsky et al., 2024) and directional ablation (Arditi et al., 2024; Zou et al., 2023a).
- **Universality.** It can be applied to both raw and normalized activations, although the latter is more computationally efficient.
- **Stability.** Restricting the rotation to a 2D subspace confines changes to just two orthogonal directions, leaving the remaining basis vectors unaffected. This minimizes interference with other features, consistent with the Superposition Hypothesis, which suggests that features are represented in near-orthogonal directions (Elhage et al., 2022). Consequently, this approach enables more robust control over the steering effect, preserving coherence (see Section 4).
- **Flexibility.** It enables steering the activations for more than 180 degrees, making the accuracy less dependent on the quality of the direction of the extracted features.

3.3 Preparing Dataset and Models

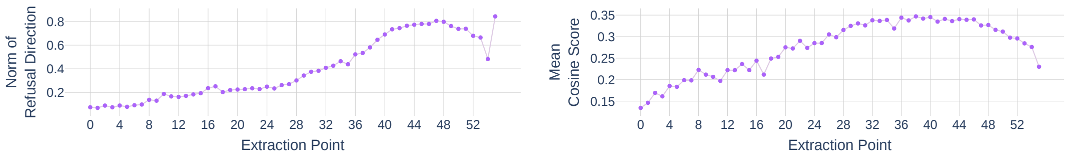
Datasets. To calibrate the feature (refusal) direction, we construct two datasets: $\mathcal{D}^{(\text{cal})}_{\text{harmful}}$, which is a split (80%) of the AdvBench dataset (Zou et al., 2023b) consisting of 416 harmful instructions; and $\mathcal{D}^{(\text{cal})}_{\text{harmful}}$, a random subset of 512 harmless examples from the Alpaca dataset (Taori et al., 2023). For evaluating steering effectiveness, we use the remaining 20% of AdvBench, denoted as $\mathcal{D}^{(\text{eval})}_{\text{harmful}}$, containing 104 samples. To assess general language modeling capabilities, we employ the TinyBenchmarks dataset (Maia Polo et al., 2024), a collection of reduced-scale benchmarks each containing 100 examples: ARC (Clark et al., 2018), MMLU (Hendrycks et al., 2021), WinoGrande (Sakaguchi et al., 2021), GSM8k (Cobbe et al., 2021), TruthfulQA (Lin et al., 2022), and HellaSwag (Zellers et al., 2019).

Models. We show experimental results on steering the refusal feature on various model families (Llama 3 (Llama Team, 2024), Qwen 2.5 (Yang et al., 2024), Gemma 2 (Gemma Team et al., 2024)) of various sizes (3B to 14B). A full list of models used in this work is presented in Appendix C.

3.4 Computing the target feature direction

Extracting Activation Vectors. Following (Arditi et al., 2024), we pass $\mathcal{D}^{(\text{cal})}_{\text{harmful}}$and $\mathcal{D}^{(\text{cal})}_{\text{harmless}}$through the model and record the activations of the final input token after the normalization layers in each transformer block as recommended by (von Rütte et al., 2024). Note that in each transformer block, there are two normalization layers: before the Attention and before the MLP. As a result, we record the activations at two extraction points per transformer block.

Calculating Candidate Directions. At each extraction point i, we compute a candidate direction using the Difference-in-Means method (Belrose, 2023): $\boldsymbol{d}^{i}{}_{\text{feat}} = \bar{\boldsymbol{h}}^{(\text{cal}),i}_{\text{harmful}} - \bar{\boldsymbol{h}}^{(\text{cal}),\text{i}}_{\text{harmless}}$ $(i = 1, \ldots, M)$, where $\boldsymbol{d}^{i}{}_{\text{feat}}$ is the direction at extraction point i, and $\bar{\boldsymbol{h}}^{(\text{cal}),\text{i}}_{\text{harmful}}$ and ${}^{i}\bar{\boldsymbol{h}}^{(\text{cal}),\text{i}}_{\text{harmless}}$ are the means computed over activations from $\mathcal{D}^{(\text{cal})}_{\text{harmful}}$and $\mathcal{D}^{(\text{cal})}_{\text{harmless}}$, respectively. Here, M is the number of extraction points, defined as twice the number of Transformer blocks in the model. One candidate direction is computed at each extraction point, yielding a total of M candidate directions.



(a) Norm of candidate feature direction at each layer.

(b) Mean cosine similarity of each candidate feature direction with those from other layers.

Figure 5: Statistics of refusal direction candidate for QWEN2.5-7B-INSTRUCT.

Choosing One Feature Direction. Among M candidate directions, we choose a feature direction for Angular Steering. Fig. 5b shows high cosine similarity among candidate directions in layers where refusal is strong, suggesting those directions are stable approximations of the true feature. This observation suggests that the similarity between candidate directions can be a promising metric to select the feature direction. In Angular Steering, we choose the candidate direction $\hat{\boldsymbol{d}}_{\text{feat}}$ that is most similar to others as the feature direction. We normalize $\hat{\boldsymbol{d}}_{\text{feat}}$ to make it a unit vector.

Remark 1 *Fig. 4 and Fig. 5 shows that refusal behavior emerges progressively along the depth of the model, stabilizes, and then spikes again near the final layer. We hypothesize that this late spike reflects a filtering step just before token generation and thus omit this point from the list of candidates.*

3.5 Selecting the Steering Plane

We now require a second direction to define the 2D steering plane in Angular Steering. As discussed in Section 3.1, the optimal plane should maximize the influence on the feature of interest while minimizing unintended impacts on other features. While using the $Span\{\boldsymbol{h}^i, \hat{\boldsymbol{d}}_{\text{feat}}\}$ aligns with prior methods like directional ablation and activation addition, we argue against it due to three reasons: (1) prior work suggests that feature directions are layer-independent (Park et al., 2024; Elhage et al., 2022; Tigges et al., 2023; Arditi et al., 2024), implying a shared geometry across layers; (2) this span might include other dominant features, risking general degradation (Turner et al., 2024; Templeton et al., 2024); and (3) computing rotation at each step is costly. Instead, we propose a fixed plane that isolates the feature of interest.

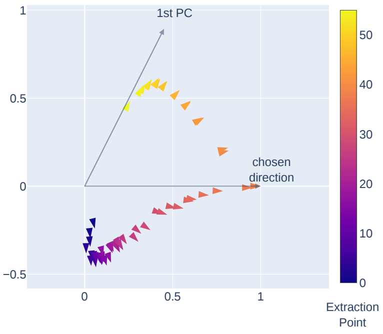


Figure 6: Projections of the feature directions extracted at each layer (i.e. $\boldsymbol{d}^i_{\text{feat}}$) on the steering plane for QWEN2.5-7B-INSTRUCT.

To construct this fixed plane, we perform PCA on the candidate directions $\boldsymbol{d}^i_{\text{feat}}$ and select the first principal component, $\hat{\boldsymbol{d}}_{\text{PC0}}$, as the second axis. This captures variance across layers, which, as shown in prior work (Arditi et al., 2024; von Rütte et al., 2024; Li et al., 2024; Zou et al., 2023a), reflects variation in approximating the true feature direction. The resulting plane $Span(\hat{\boldsymbol{d}}_{\text{feat}}, \hat{\boldsymbol{d}}_{\text{PC0}})$ thus isolates meaningful variation in the target feature. Fig. 6 shows a smooth directional shift across layers in this plane, supporting the hypothesis that feature strength evolves gradually, making it a natural basis for steering (see Section 4).

3.6 Putting It All Together: The (Adaptive) Angular Steering Framework

We are now ready to formulate Angular Steering and its adaptive variant.

3.6.1 *Angular Steering Framework*

Let P be the 2D subspace spanned by $\hat{\boldsymbol{d}}_{\text{feat}}$ and $\hat{\boldsymbol{d}}_{\text{PC0}}$. We compute the orthonormal basis $\{\boldsymbol{b}_1, \boldsymbol{b}_2\}$ of P as follows:

$$\boldsymbol{b}_1 \leftarrow \hat{\boldsymbol{d}}_{\text{feat}}; \quad \boldsymbol{b}_2 \leftarrow \hat{\boldsymbol{d}}_{\text{PC0}} - (\hat{\boldsymbol{d}}_{\text{PC0}} \cdot \boldsymbol{b}_1)\boldsymbol{b}_1; \quad \boldsymbol{b}_2 \leftarrow \boldsymbol{b}_2 / ||\boldsymbol{b}_2||.$$

Rotation by an Offset Angle. To rotate within the subspace P by an angle ϕ, the transformation matrix $\boldsymbol{R}_{\phi}^{P}$ is given as

$$\boldsymbol{R}_{\phi}^{P} = \boldsymbol{I} - (\boldsymbol{b}_1\boldsymbol{b}_1^{\top} + \boldsymbol{b}_2\boldsymbol{b}_2^{\top}) + [\boldsymbol{b}_1\ \boldsymbol{b}_2]\,\boldsymbol{R}_{\phi}\,[\boldsymbol{b}_1\ \boldsymbol{b}_2]^{\top} \tag{1}$$

where $\boldsymbol{I} - (\boldsymbol{b}_1\boldsymbol{b}_1^{\top} + \boldsymbol{b}_2\boldsymbol{b}_2^{\top})$ is the projection to the $(d_{\text{model}} - 2)$-dimensional orthogonal complement Q of P and $\boldsymbol{R}_{\phi}$ is the 2D rotation matrix given as $\boldsymbol{R}_{\phi} = \begin{bmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{bmatrix}$.

Rotation to a Target Angle. In practice, rather than rotating all activations by a fixed offset, we often want to rotate them to a specific angular position θ, e.g., where a desired behaviour is strongly expressed. A naive approach would involve: (1) projecting the input $\boldsymbol{h}$ onto the steering plane P: $\text{proj}_P(\boldsymbol{h}) = (\boldsymbol{b}_1\boldsymbol{b}_1^{\mathsf{T}} + \boldsymbol{b}_2\boldsymbol{b}_2^{\mathsf{T}}) \cdot \boldsymbol{h}$; (2) computing the current angle $\phi_{\boldsymbol{h},\boldsymbol{b}_1}^{P}$ between $\text{proj}_P(\boldsymbol{h})$ and $\boldsymbol{b}_1$; (3) constructing the rotation matrix $R_{\theta-\phi}^{P}$ using Eqn. 1; and (4) applying this matrix to $\boldsymbol{h}$. However, this is inefficient when θ is fixed and can be optimized by precomputing reusable components.

Noting that the term $[\boldsymbol{b}_1\ \boldsymbol{b}_2]\,R_{\phi}\,[\boldsymbol{b}_1\ \boldsymbol{b}_2]^{\mathsf{T}}$ in Eqn. 1 is a norm-preserving transformation, we can precompute its effect on the unit vector $[1\ 0]^{\mathsf{T}}$ and scale the result by $|\text{proj}_P(\boldsymbol{h})|$. This leads to the following efficient formulation for rotating an input $\boldsymbol{h}$ to angle θ:

$$\boldsymbol{h}_{\text{steered},\theta} = R_{\theta-\phi_{h,b_1}}^{P} \cdot \boldsymbol{h} = \boldsymbol{h} - \text{proj}_P(\boldsymbol{h}) + |\text{proj}_P(\boldsymbol{h})| \cdot [\boldsymbol{b}_1\ \boldsymbol{b}_2]\,R_{\theta}\,[1\ 0]^{\top}, \tag{2}$$

where $R_{\theta-\phi_{h,b_1}}^{P}$ is the rotation matrix defined in Eqn. 1. Here, both the projection matrix $(\boldsymbol{b}_1\boldsymbol{b}_1^{\mathsf{T}} + \boldsymbol{b}_2\boldsymbol{b}_2^{\mathsf{T}})$ and $[\boldsymbol{b}_1\ \boldsymbol{b}_2]\,R_{\theta}\,[1\ 0]^{\top}$ can be precomputed.

3.6.2 *Adaptive Angular Steering Framework*

Since inputs from contrastive datasets tend to align with $\boldsymbol{d}^{i}_{\text{feat}}$ in opposite directions (Fig. 4), it is unnecessary to rotate all activations uniformly. To increase flexibility and further reduce unintended effects on non-targeted features, we propose an adaptive variant that rotates only activations positively aligned with $\hat{\boldsymbol{d}}_{\text{feat}}$. In particular, we first compute a conditional mask based on the sign of the projection onto $\hat{\boldsymbol{d}}_{\text{feat}}$: $\text{mask} = \max(0,\ \text{sign}(\text{proj}_{\hat{\boldsymbol{d}}_{\text{feat}}}(\boldsymbol{x})))$. Using this mask, Eqn. 2 becomes:

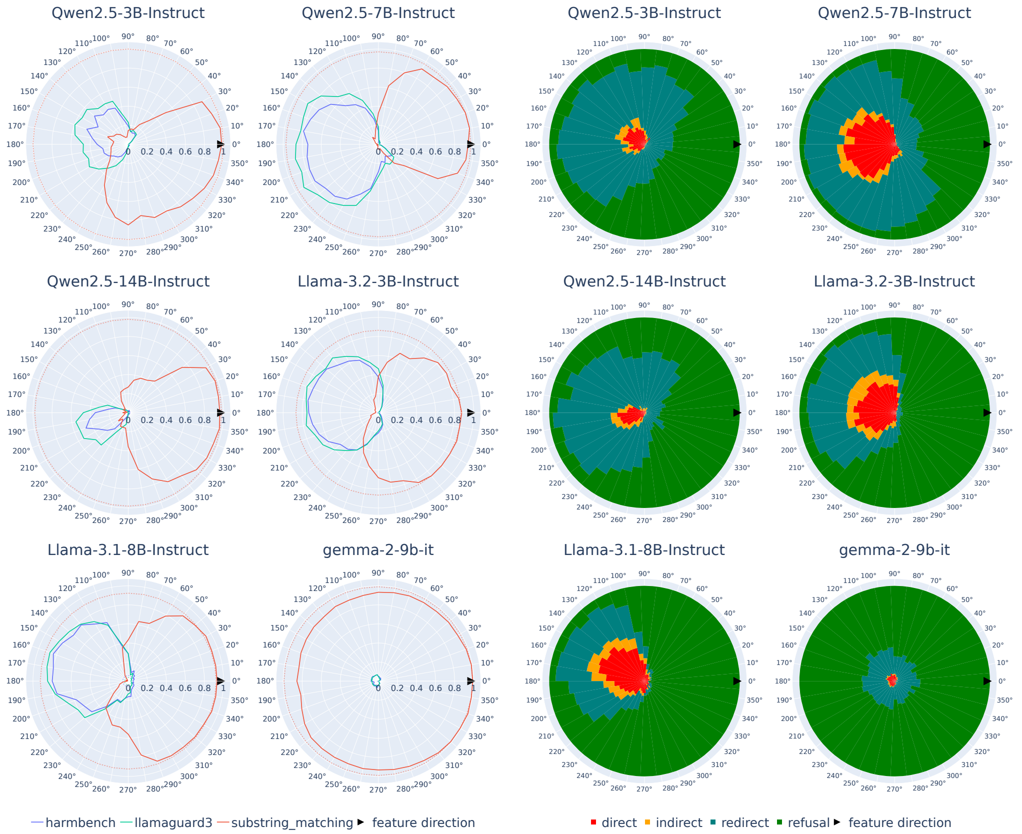
$$\boldsymbol{h}_{\text{steered (adaptive)},\theta} = \boldsymbol{h} + \text{mask} \cdot \Big(|\text{proj}_P(\boldsymbol{h})| \cdot [b_1\ b_2]\ R_{\theta}\ [1\ 0]^{\top} - \text{proj}_P(\boldsymbol{h})\Big) \tag{3}$$

This formulation adds an additional layer of control and robustness: steering is both restricted to a 2D subspace and selectively applied based on feature alignment. Beyond adjusting the steering angle θ, users may also vary the similarity threshold used in the mask or employ different $\boldsymbol{d}^{i}_{\text{feat}}$ across layers. We note that another conditional steering approach has been explored in contemporary work by (Lee et al., 2024), but activation addition was used as the steering framework instead of rotation. We summarize the algorithms for feature direction extraction, steering plane selection, and angular steering in Appendix B.

4 Controlling the Steering Effect

For inference, we apply Adaptive Angular Steering as described in Eqn. 3 on every normalization module before each Attention and MLP layer. By varying the target angular position θ from 0° to 360° (with 10° intervals), we observe that the models change from refusal to compliance and back to refusal again (see Fig. 7). We found that both Angular Steering and Adaptive Angular Steering are effective at varying the steering effect. However, the non-adaptive version runs a risk of breaking the coherence on smaller models, which will be discussed in Section 5.

Evaluation Metrics. We compute a **refusal score** using the `substring matching` method (Arditi et al., 2024), which operates by matching a set of common "refusal substrings" (e.g., `I'm sorry`, `As an AI`) on the model completion. The score is 1 if at least one such substring is matched and 0 otherwise.



(a) **Refusal score** (`substring matching` (Arditi et al., 2024)) and **harmful scores** (LLAMAGUARD3 (Llama Team, 2024), HARMBENCH (Mazeika et al., 2024)).

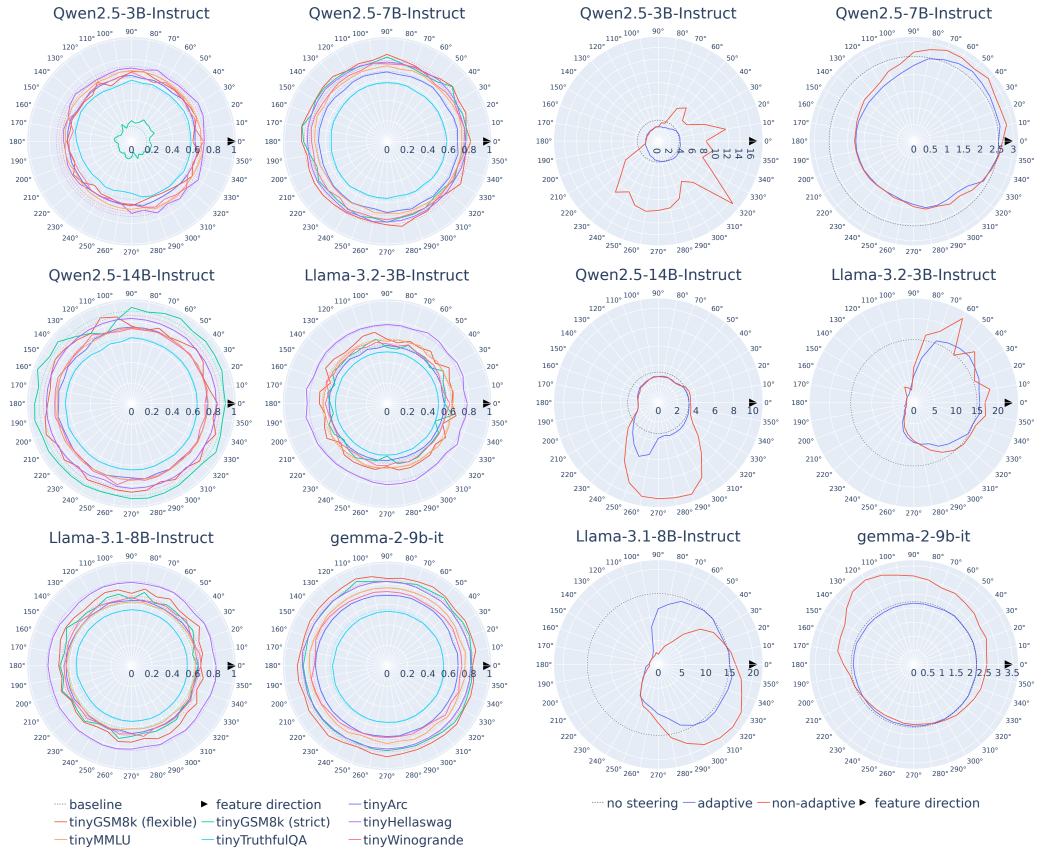
(b) LLM-as-a-judge classification results: models' responses are classified by an LLM into four categories: `direct`, `indirect`, `redirect` and `refusal`.

Figure 7: **Steering evaluation.** Each model was steered using Adaptive Angular Steering and evaluated at every 10° angular position. Solid traces show evaluation scores; dashed traces indicate baseline (non-steered) models. Traces of the same color correspond to the same benchmark. Baseline values for LLAMAGUARD3 and HARMBENCH may be hidden due to near-zero values.

Intuitively, this metric only detects memorized refusal phrases but does not assess coherence and harmfulness, as noted by (Arditi et al., 2024; Huang et al., 2023; Meade et al., 2024; Qi et al., 2023; Shah et al., 2023). To evaluate harmfulness, we follow the setup in (Arditi et al., 2024) and use two more complementary evaluation metrics, LLAMAGUARD3 (Llama Team, 2024) and HARMBENCH (Mazeika et al., 2024), which we collectively call **harmful scores**. These two methods use open-source models to classify whether an input is harmful, in which the score is 1 if the classification is true and 0 otherwise.

Beyond refusal and harmfulness detection, we are interested in how the model's output changes semantically at different level of refusal. Thus, we perform qualitative analysis using a reasoning model QVQ-72B-PREVIEW (Qwen Team, 2024) to classify the generation outputs into 4 classes: `direct`: The model directly answers the prompt; `indirect`: The model starts out seemingly unwilling to answer but then still provides with an answer; `redirect`: The model does not explicitly agree or refuse to answer but provides a tactful response without producing any harmful content; `refusal`: The model explicitly refuses to answer.

Evaluation along the Steering Circle. Fig.7 demonstrates that angular steering effectively modulates refusal and safety behaviors. In Fig.7a, all models show a clear arc of strong alignment–high refusal and low harmful scores–and an opposing arc of weak alignment–low refusal and high harmful scores. These arcs lie in opposite directions within the steering circle, with performance peaking near the center and diminishing outward. Fig.7b further



(a) Benchmark results on the TinyBenchmarks (Maia Polo et al., 2024) suite.

(b) Perplexity scores of Adaptive Steering, non-adaptive Steering and no steering.

Figure 8: **Evaluation beyond the targeted steering task.** Each model was steered using Adaptive Angular Steering (Eqn. 3) and evaluated on all benchmarks at every 10° angular position along the steering circle. Solid traces represent evaluation scores along the steering circle, and dashed traces represent the evaluation for the baseline (non-steered models); traces having the same color represent the same benchmark.

supports this observation by showing that, for five of six models, `refusal` dominates in the strong arc, followed by `redirect`, and then `direct` or `indirect` responses as the angle shifts. Tab.1 reports example completions for each class. Gemma-2-9b-it is an exception, displaying the weakest effect yet still following the overall trend.

Steering on a random plane. For completeness, we conduct an ablation study on steering using Adaptive Angular Steering with a random plane. Fig. 13b in Appendix D.2 shows that it has little to no effect on controlling refusal in five out of six tested models.

5 Effects on Model's Performance beyond the Targeted Steering Task

Steering can degrade language modeling ability (Souly et al., 2024), especially when relying on sensitive hyperparameters (Turner et al., 2024; Zou et al., 2023a; Templeton et al., 2024; Bayat et al., 2025; Li et al., 2024; von Rütte et al., 2024), which may lead to incoherent outputs if not carefully tuned (Turner et al., 2024; Templeton et al., 2024). In this section, we quantitatively assess the impact of our method on overall LLM performance.

5.1 Language Modeling Benchmarks

Method. For each model, we adaptively steer its activation with a 10° interval along the entire steering circle using Eqn. 3 and evaluate all benchmarks from the TinyBenchmarks suite (Maia Polo et al., 2024). The results are visualized in Fig. 8a.

Results. Overall, our steering method effectively preserves benchmark accuracies across the entire steering circle, demonstrating strong robustness. Interestingly, in many cases, performance under intervention even surpasses the non-steered baseline.

A notable outlier is QWEN2.5-3B-INSTRUCT, which exhibits a performance drop along the arc from 160° to 280°. We attribute this to feature interference (Elhage et al., 2022), where multiple latent features dominate within the chosen steering plane, a phenomenon to which smaller models are more susceptible. The consistent accuracy drop across all benchmarks in this region suggests the model is reacting to a competing feature. For TINYGSM8K, although the model often generates a correct answer, it fails to match the expected format, leading to significantly lower scores under the `strict` metric compared to the more lenient `flexible` variant.

It is important to note that for TINYGSM8K, the `flexible` metric extracts the last numeric value as the final answer, whereas the `strict` variant assumes a predefined output format. Consequently, these metrics are highly sensitive to formatting variations, leading to noticeable fluctuations in accuracy across different steering angles.

5.2 Perplexity of the Steered Generations

Smaller Models are More Vulnerable to Interference under Angular Steering. In non-adaptive Angular Steering experiments, 7B – 14B models generate coherent outputs throughout the steering circle, while smaller models like LLAMA-3.2-3B-INSTRUCT and QWEN2.5-3B-INSTRUCT often produce incoherent text across a wide arc. Notably, refusal phrases still appear randomly in various languages for LLAMA-3.2-3B-INSTRUCT, and mainly in Chinese for QWEN2.5-3B-INSTRUCT, despite English prompts. This suggests that limited capacity in smaller models leads to feature interference (Elhage et al., 2022), with multiple features entangled in the 2D steering subspace, as discussed in Sections 4 and 5.1.

Method. Motivated by such observations, we analyze the perplexity of the steered generations using the non-steered models and report the results in Fig. 8b. Given an input sequence $\boldsymbol{x}$, an non-steered LLM $\pi_{\text{non-steered}}$, the output is modeled by $\boldsymbol{y}_{\text{non-steered}} \sim \pi_{\text{non-steered}}(\boldsymbol{x})$. Similarly, π_{steered} and $\boldsymbol{y}_{\text{steered}}$ denote the steered model and its output, respectively. We denote the perplexity score of $\boldsymbol{x}$ with respect to a model π as $PPL_{\pi}(\boldsymbol{x})$. In Fig. 8b, we compare $PPL_{\pi_{\text{non-steered}}}(\boldsymbol{x}||\boldsymbol{y}_{\text{non-steered}})$, $PPL_{\pi_{\text{non-steered}}}(\boldsymbol{x}||\boldsymbol{y}_{\text{steered (non-adaptive)}})$ and $PPL_{\pi_{\text{non-steered}}}(\boldsymbol{x}||\boldsymbol{y}_{\text{steered (adaptive)}})$ for each model and at every 10° rotation.

Results. *Both 3B models exhibit unstable perplexity under non-adaptive steering, indicating vulnerability to interference*. For QWEN2.5-3B-INSTRUCT, perplexity remains significantly above baseline across more than half of the circle, aligning with the incoherent outputs discussed earlier. In contrast, LLAMA-3.2-3B-INSTRUCT shows perplexity closer to baseline, consistent with its behavior of still refusing harmful requests, albeit in different languages.

Adaptive Steering effectively preserves coherence. Fig. 8b reveals that the perplexity of Adaptive Steering is lower, more stable, and closer to no steering than its non-adaptive counterpart, indicating effectiveness at balancing behavior control with coherence and performance.

Alignment masks rather than removes harmful behavior. Perplexity stays near baseline when steering aligns with the target feature, but drops below baseline as it moves toward the "jailbroken" region. This indicates harmful capabilities remain latent, with relevant knowledge still embedded in the model, and alignment merely suppressing them by shifting activations to a higher-entropy distribution, consistent with the findings in (Qi et al., 2024).

6 Concluding Remarks

We propose Angular Steering, a novel activation steering method offering continuous, fine-grained control over large language model behaviors by rotating activation vectors within a two-dimensional subspace. This geometric perspective unifies prior steering techniques, enhancing interpretability and deepening understanding of model mechanisms without compromising general performance. Our adaptive variant further improves robustness by selectively applying steering based on context.

Ethics Statement

The Angular Steering approach presented in this work has several broader societal impacts. On the positive side, it significantly enhances the control and interpretability of LLMs, enabling their safer deployment across various applications by effectively reducing harmful outputs such as misinformation, biased content, and unethical requests. This enhanced control facilitates alignment with societal norms and ethical standards, potentially increasing public trust and acceptance of AI technologies.

Conversely, there is also a potential for negative impacts. By simplifying fine-grained behavior control, Angular Steering could inadvertently make it easier to generate nuanced harmful or unethical content, such as persuasive misinformation or biased narratives. Although our method does not fundamentally alter the existing risk profile of deploying LLMs, it underscores the need for continued vigilance and improvement in AI safety mechanisms. To responsibly manage these risks, implementing rigorous safeguards, ensuring transparency, and promoting accountability are essential. We advocate ongoing ethical assessment to responsibly guide the deployment and utilization of our proposed method.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in Language Models Is Mediated by a Single Direction, October 2024.

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. Steering Large Language Model Activations in Sparse Spaces, February 2025.

Nora Belrose. Diff-in-means concept editing is worst-case optimal: Explaining a result by sam marks and max tegmark, 2023. URL https://blog.eleuther.ai/diff-in-means/.

Leonard Bereska and Efstratios Gavves. Mechanistic Interpretability for AI Safety – A Review, April 2024.

bloc97. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation., 2023. URL https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/.

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, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending Context Window of Large Language Models via Positional Interpolation, June 2023.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL ”https://transformer-circuits.pub/2022/toy_model/index.html”.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders, June 2024a.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024b. URL https://zenodo.org/records/12608602.

Google Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic jailbreak of open-source llms via exploiting generation. *arXiv preprint arXiv:2310.06987*, 2023.

Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik Opitz, and Tobias Hecking. Style Vectors for Steering Generative Large Language Models. In Yvette Graham and Matthew Purver (eds.), *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 782–802, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Bruce W Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. *arXiv preprint arXiv:2409.05907*, 2024.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model, June 2024.

Yichen Li, Zhiting Fan, Ruizhe Chen, Xiaotang Gai, Luqi Gong, Yan Zhang, and Zuozhu Liu. Fairsteer: Inference time debiasing for llms with dynamic activation steering. *arXiv preprint arXiv:2504.14492*, 2025.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022.

AI @ Meta Llama Team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinybenchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*, 2024.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse Feature Circuits: Discovering and Editing Interpretable Causal Graphs in Language Models, March 2025.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Nicholas Meade, Arkil Patel, and Siva Reddy. Universal adversarial triggers are not universal. *arXiv preprint arXiv:2404.16020*, 2024.

Neel Nanda and Joseph Bloom. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens, 2022.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Kiho Park, Yo Joong Choe, and Victor Veitch. The Linear Representation Hypothesis and the Geometry of Large Language Models, July 2024.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient Context Window Extension of Large Language Models, November 2023.

Van-Cuong Pham and Thien Huu Nguyen. Householder pseudo-rotation: A novel approach to activation editing in llms with direction-magnitude perspective. *arXiv preprint arXiv:2409.10053*, 2024.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*, 2023.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety Alignment Should be Made More Than Just a Few Tokens Deep. In *The Thirteenth International Conference on Learning Representations*, October 2024.

Alibaba Qwen Team. Qvq: To see the world with wisdom, December 2024. URL https://qwenlm.github.io/blog/qvq-72b-preview/.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Rusheb Shah, Soroush Pour, Arush Tagade, Stephen Casper, Javier Rando, et al. Scalable and transferable black-box jailbreaks for language models via persona modulation. *arXiv preprint arXiv:2311.03348*, 2023.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A strongREJECT for empty jailbreaks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Asa Cooper Stickland, Alexander Lyzhov, Jacob Pfau, Salsabila Mahdi, and Samuel R Bowman. Steering without side effects: Improving post-deployment control of language models. *arXiv preprint arXiv:2406.15518*, 2024.

Niklas Stoehr, Kevin Du, Vésteinn Snæbjarnarson, Robert West, Ryan Cotterell, and Aaron Schein. Activation scaling for steering and interpreting language models. *arXiv preprint arXiv:2410.04962*, 2024.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving instruction-following in language models through activation steering. *arXiv preprint arXiv:2410.12877*, 2024.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

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

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

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. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear Representations of Sentiment in Large Language Models, October 2023.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering Language Models With Activation Engineering, October 2024.

Dimitri von Rütte, Sotiris Anagnostidis, Gregor Bachmann, and Thomas Hofmann. A Language Model's Guide Through Latent Space, February 2024.

Tianlong Wang, Xianfeng Jiao, Yinghao Zhu, Zhongzhi Chen, Yifan He, Xu Chu, Junyi Gao, Yasha Wang, and Liantao Ma. Adaptive activation steering: A tuning-free llm truthfulness improvement method for diverse hallucinations categories. In *Proceedings of the ACM on Web Conference 2025*, pp. 2562–2578, 2025.

Weixuan Wang, Jingyuan Yang, and Wei Peng. Semantics-adaptive activation intervention for llms via dynamic steering vectors. *arXiv preprint arXiv:2410.12299*, 2024.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization, October 2019.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation Engineering: A Top-Down Approach to AI Transparency, October 2023a.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023b.

Supplement to “Angular Steering: Behavior Control via Rotation in Activation Space”

Table of Contents

A Detailed Derivation: Existing Activation Steering as Special Cases of Steering by Rotation

We will show that, when activation norms are preserved, existing steering techniques are special cases of angular steering but with restricted flexibility: vector addition is limited to less than 180 degrees, and orthogonalization is fixed at 90 degrees.

Formally, let the activation $\boldsymbol{h}_i$ be decomposed into components parallel and orthogonal to a unit-norm feature direction $\hat{\boldsymbol{d}}_{\text{feat}}$ (for brevity, here we denote them as $\boldsymbol{h}$ and $\mathbf{d}$ respectively):

$$\boldsymbol{h} = (\boldsymbol{h} \cdot \boldsymbol{d})\boldsymbol{d} + \boldsymbol{h}_\perp, \quad \text{where} \quad \boldsymbol{h}_\perp = \boldsymbol{h} - (\boldsymbol{h} \cdot \boldsymbol{d})\boldsymbol{d}.$$

Let $\mathbf{u} = \frac{\boldsymbol{h}_\perp}{\|\boldsymbol{h}_\perp\|}$, and define the initial angle between $\boldsymbol{h}$ and $\boldsymbol{d}$ as:

$$\theta_0 = \tan^{-1}\left(\frac{\|\boldsymbol{h}_\perp\|}{\boldsymbol{h} \cdot \boldsymbol{d}}\right).$$

We define *Angular Steering* as rotating $\boldsymbol{h}$ by an offset angle ϕ in the plane $Span\{\boldsymbol{h}, \boldsymbol{d}\}$, producing a vector:

$$\boldsymbol{h}_{\text{rot}}(\phi) = \cos(\theta_0 + \phi) \cdot \boldsymbol{d} + \sin(\theta_0 + \phi) \cdot \mathbf{u}.$$

Now consider *vector addition* (Turner et al., 2024), defined as:

$$\boldsymbol{h}_{\text{add}} = \boldsymbol{h} + \alpha\boldsymbol{d} = (\boldsymbol{h} \cdot \boldsymbol{d} + \alpha)\boldsymbol{d} + \boldsymbol{h}_\perp.$$

After normalization, the direction becomes:

$$\boldsymbol{h}_{\text{add-norm}} = \frac{\boldsymbol{h}_{\text{add}}}{\|\boldsymbol{h}_{\text{add}}\|} = \cos(\theta_0 + \phi_{\text{add}}) \cdot \boldsymbol{d} + \sin(\theta_0 + \phi_{\text{add}}) \cdot \mathbf{u},$$

where $\phi_{\text{add}} = \tan^{-1}\left(\frac{\|\boldsymbol{h}_\perp\|}{\boldsymbol{h}\cdot\boldsymbol{d}+\alpha}\right) - \theta_0$.

Likewise, *directional ablation (orthogonalization)* (Arditi et al., 2024), given by:

$$\boldsymbol{h}_{\text{ablate}} = \boldsymbol{h}_\perp,$$

after normalization becomes:

$$\boldsymbol{h}_{\text{ablate-norm}} = \mathbf{u} = \cos(\theta_0 + \phi_{\text{ablate}}) \cdot \boldsymbol{d} + \sin(\theta_0 + \phi_{\text{ablate}}) \cdot \mathbf{u},$$

with $\phi_{\text{ablate}} = \frac{\pi}{2} - \theta_0$.

Thus, when activation norms are fixed, both addition and ablation shift the direction of $\boldsymbol{h}$ in a way that is exactly equivalent to a rotation by some angle ϕ within the plane spanned by $\boldsymbol{h}$ and $\boldsymbol{d}$. This establishes them as special cases of Angular Steering.

In practice, RMSNorm (Zhang & Sennrich, 2019) stabilizes activation norms rather than fixing them to exact values (as shown in (Zhang & Sennrich, 2019) and Fig. 9, left). As a result, these methods can still be interpreted as approximate rotations.

B Algorithms for Angular Steering

Algorithm 1 Extract Feature Direction

```
Require: Contrastive datasets $\mathcal{D}_{\text{harmful}}, \mathcal{D}_{\text{harmless}}$, model $\mathcal{M}$
for each layer $i$ in model do
    Compute normalized activations $\boldsymbol{h}^{(i)}$ after Attention and MLP
    Compute mean activation for each dataset:
        $\bar{\boldsymbol{h}}^{(i)}_{\text{harmful}}, \bar{\boldsymbol{h}}^{(i)}_{\text{harmless}}$
    Compute candidate direction:
        $\boldsymbol{d}^{(i)} = \bar{\boldsymbol{h}}^{(i)}_{\text{harmful}} - \bar{\boldsymbol{h}}^{(i)}_{\text{harmless}}$
end for
Select final feature direction $\boldsymbol{d}$ using max average cosine similarity:
    $\boldsymbol{d} = \underset{i=1\ldots|\text{layers}|}{\operatorname{argmax}} \left( \frac{1}{|\text{layers}|} \sum_{j=1}^{|\text{layers}|} cosine(\boldsymbol{d}^{(i)}, \boldsymbol{d}^{(j)}) \right)$
Normalize: $\hat{\boldsymbol{d}} = \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}$
```

Algorithm 2 Select Steering Plane

```
Require: Candidate directions $\{\boldsymbol{d}^{(i)}\}$, feature direction $\hat{\boldsymbol{d}}$
Perform PCA on $\{\boldsymbol{d}^{(i)}\}$
Let first principal component be $\boldsymbol{d}_{\text{1stPC}}$
Set orthonormal basis for plane:
    $\boldsymbol{b}_1 \leftarrow \hat{\boldsymbol{d}}, \quad \boldsymbol{b}_2 \leftarrow \boldsymbol{d}_{\text{1stPC}} - (\boldsymbol{b}_1 \cdot \boldsymbol{d}_{\text{1stPC}})\boldsymbol{b}_1; \quad \boldsymbol{b}_2 \leftarrow \frac{\boldsymbol{b}_2}{\|\boldsymbol{b}_2\|}$
Define projection matrix $P = \boldsymbol{b}_1\boldsymbol{b}_1^\top + \boldsymbol{b}_2\boldsymbol{b}_2^\top$
```

Algorithm 3 Angular Steering (with optional Adaptive Mask)

```
Require: Activation $\boldsymbol{h}$, basis $\boldsymbol{b}_1, \boldsymbol{b}_2$, target angle $\theta$, (optional) mask flag
Project: $\text{proj}_P(\boldsymbol{h}) = P \cdot \boldsymbol{h}$
Compute magnitude: $\boldsymbol{r} = \|\text{proj}_P(\boldsymbol{h})\|$
Precompute: $\boldsymbol{v}_\theta = [\boldsymbol{b}_1\ \boldsymbol{b}_2] \cdot R_\theta \cdot [1\ 0]^\top$
if adaptive then
    Compute mask: $\text{mask} = \max(0, \text{sign}(\boldsymbol{h} \cdot \hat{\boldsymbol{d}}))$
    Apply adaptive steering:
        $\boldsymbol{h}_{\text{steered}} = \boldsymbol{h} + \text{mask} \cdot (\boldsymbol{r} \cdot \boldsymbol{v}_\theta - \text{proj}_P(\boldsymbol{h}))$
else
    Apply steering:
        $\boldsymbol{h}_{\text{steered}} = \boldsymbol{h} - \text{proj}_P(\boldsymbol{h}) + \boldsymbol{r} \cdot \boldsymbol{v}_\theta$
end if
```

C Use of existing assets

C.1 Models

Model (with link)	Usage	License
Qwen2.5-(3B, 7B, 13B)-Instruct (Yang et al., 2024)	Experimental subject	Apache license 2.0
Llama-3.1-8B-Instruct (Llama Team, 2024)	Experimental subject	Llama 3.1 Community License Agreement
Llama-3.2-3B-Instruct (Llama Team, 2024)	Experimental subject	Llama 3.2 Community License Agreement
gemma-2-9b-it (Gemma Team et al., 2024)	Experimental subject	Gemma Terms of Use
Llama-Guard-3-8B (Llama Team, 2024)	Evaluation device	Llama 3.1 Community License Agreement
HarmBench Classifier (Mazeika et al., 2024)	Evaluation device	MIT
QVQ-72B-Preview (Qwen Team, 2024)	Evaluation device	Qwen License

Table 2: Models used in this work.

C.2 Datasets

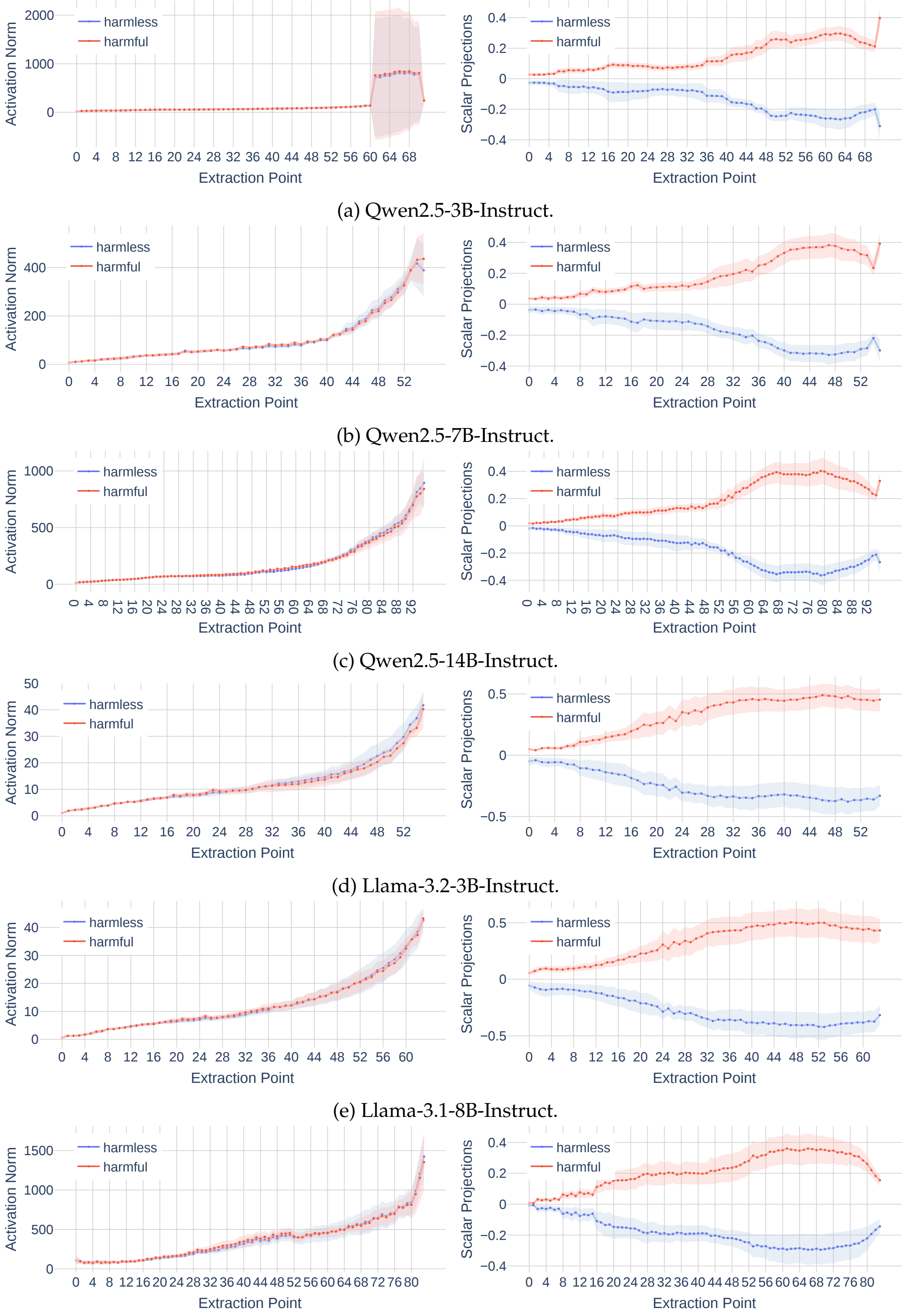
Dataset (with link)	Source	License
AdvBench	(Zou et al., 2023b)	MIT
Alpaca	(Taori et al., 2023)	Creative Commons Attribution Non Commercial 4.0
TinyBenchmarks	(Maia Polo et al., 2024)	MIT

Table 3: Datasets used in this work.

D Additional Results

D.1 Activations along the model's depth

Fig. 9 (left) demonstrates that the norm of activation vectors increases exponentially across all tested models as the layer depth increases. This behavior is attributable to the additive nature of the residual stream, where each layer's output accumulates onto the previous



(a) Qwen2.5-3B-Instruct.

(b) Qwen2.5-7B-Instruct.

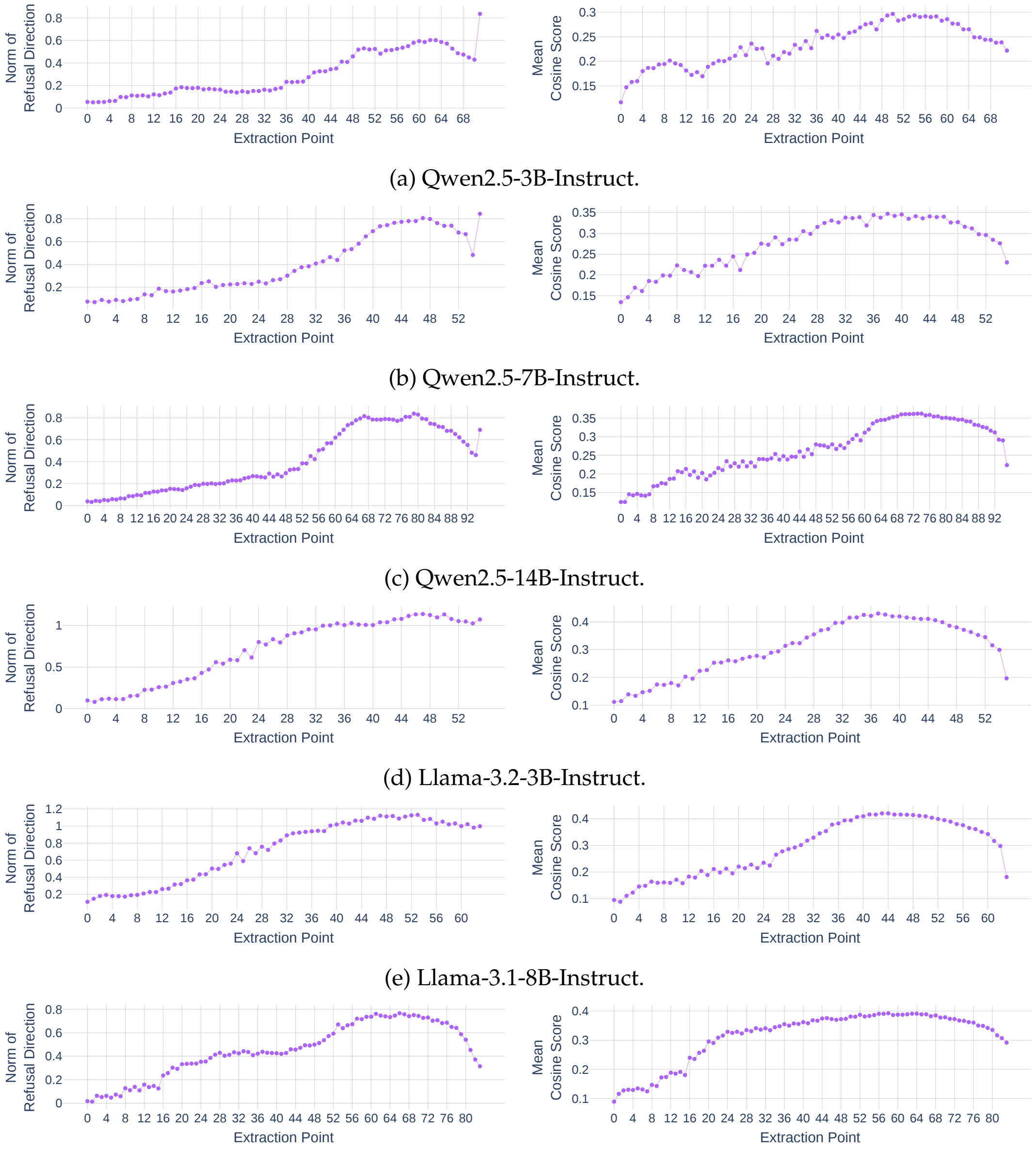
(c) Qwen2.5-14B-Instruct.

(d) Llama-3.2-3B-Instruct.

(e) Llama-3.1-8B-Instruct.

(f) gemma-2-9b-it.

Figure 9: Statistics of activations for all tested models. Left: Norms of activations at each layer. Right: Mean scalar projection of the *normalized* activation on the (local) candidate feature direction at each layer.



(a) Qwen2.5-3B-Instruct.

(b) Qwen2.5-7B-Instruct.

(c) Qwen2.5-14B-Instruct.

(d) Llama-3.2-3B-Instruct.

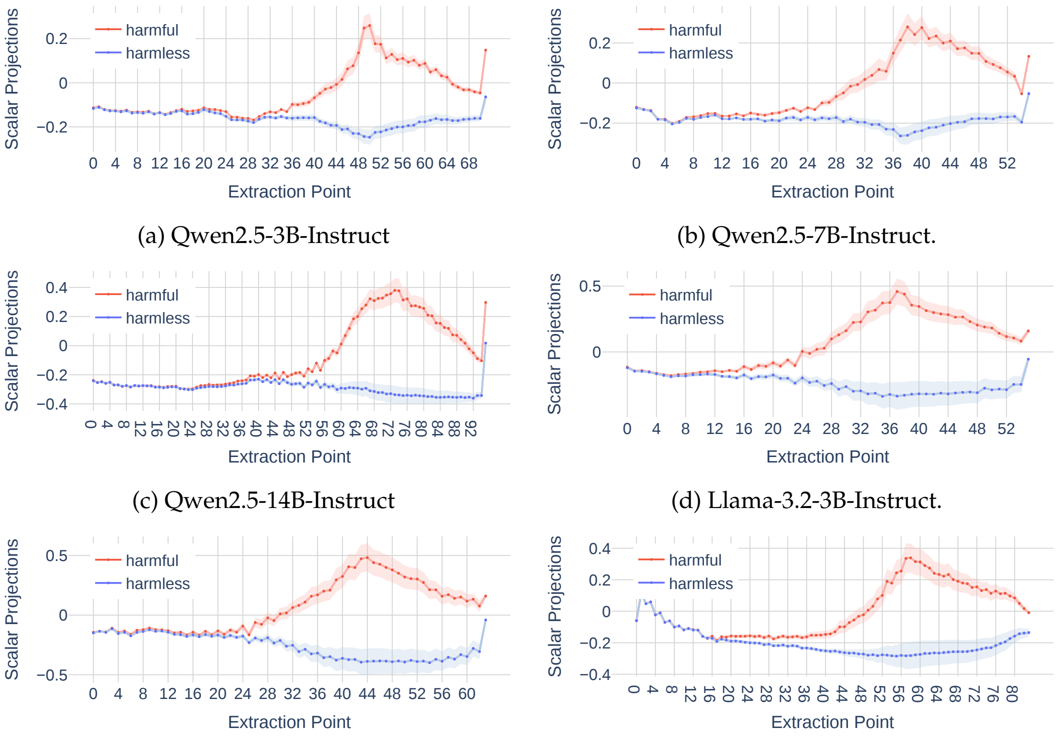
(e) Llama-3.1-8B-Instruct.

(f) gemma-2-9b-it.

Figure 10: Statistics of refusal direction candidates for all tested models. Left: Norms of candidate feature direction at each layer (i.e. $|\boldsymbol{d}^{(i)}_{\text{feat}}|$). Right: Mean cosine similarity of the candidate feature direction from each layer with those from other layers (i.e. $\frac{1}{|\text{layers}|}\sum_{j=1}^{|\text{layers}|} cosine(\boldsymbol{d}^{(i)}_{\text{feat}}, \boldsymbol{d}^{(j)}_{\text{feat}})$).

state. Interestingly, even models from the same architecture family display different scaling patterns, indicating that activation growth is not only architecture-dependent but also implementation-specific. These observations underscore the necessity of norm-independent steering techniques, as steering strategies relying on raw magnitude can become unstable or ineffective across layers and model variants.

Fig. 9 (right) shows a consistent phenomenon across all evaluated models: activations from contrastive prompts, *harmful* versus *harmless*, diverge progressively in geometric space as depth increases. This increasing separation suggests a universal, model-agnostic internal mechanism in LLMs, whereby behavioral distinctions are gradually amplified layer by layer. Such a trend reveals a directional progression in the model's internal representation, reinforcing the hypothesis that feature separation is a fundamental property of transformer-based language models.



(a) Qwen2.5-3B-Instruct

(b) Qwen2.5-7B-Instruct.

(c) Qwen2.5-14B-Instruct

(d) Llama-3.2-3B-Instruct.

(e) Llama-3.1-8B-Instruct

(f) gemma-2-9b-it.

Figure 11: Mean scalar projection activations at each layer onto the chosen feature direction $\hat{d}_{\text{feat}}$ for all tested models.

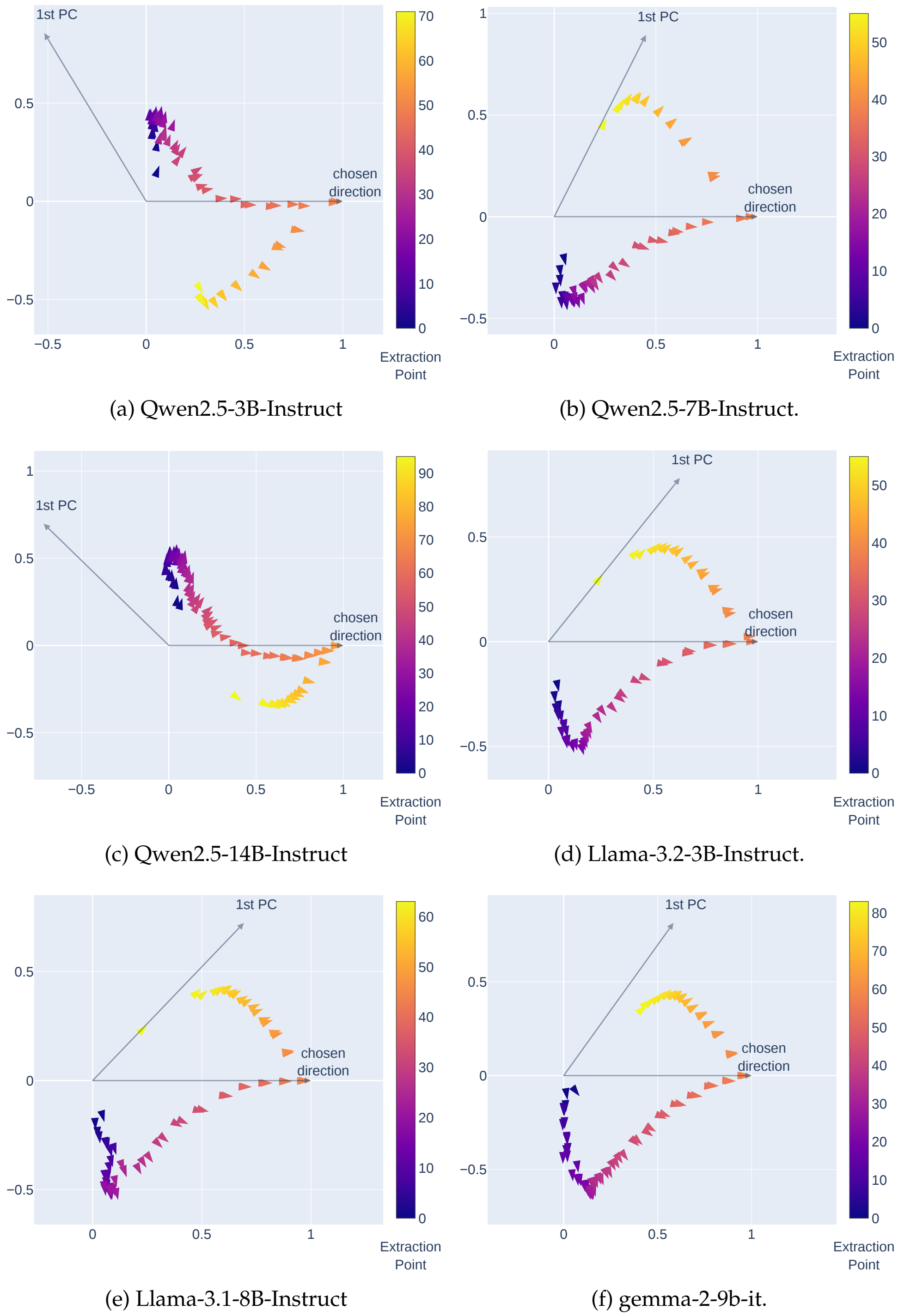
Fig.10 further illustrates this progression, focusing on the evolution of the refusal direction. The strength of this feature becomes increasingly prominent in early and middle layers, reaching its maximum influence at a specific intermediate depth before diminishing slightly in later layers—a trend echoed in Fig.11. Importantly, even in the deeper layers where the signal attenuates, the extracted refusal direction continues to serve as a reliable discriminator between activations corresponding to *harmful* and *harmless* prompts. This persistent separability affirms the robustness and interpretability of the refusal direction, validating its role as a stable, layer-resilient feature for behavioral control in LLMs.

D.2 Ablation Study: Steering on a random plane.

To assess the importance of the steering plane, we conducted an ablation study using two setups: (1) steering with a plane defined by one random direction and one feature-aligned direction, and (2) steering with a fully random plane composed of two random directions.

As illustrated in Fig. 13a, where one random direction is combined with the feature direction, most models exhibit noticeably degraded steering performance and less smooth transitions along the steering circle. This degradation suggests that even partial misalignment of the steering plane can distort the intended behavioral modulation. An exception is QWEN2.5-7B-INSTRUCT, which retains robust control, indicating a strong, well-defined internal representation of the refusal direction. LLAMA-3.2-3B-INSTRUCT shows a clear steering effect, but the refusal arc is shifted, suggesting the random component introduces skew that displaces the effective axis of control.

Fig. 13b, where both directions are randomly selected, shows that five of the six tested models exhibit minimal to no steering effect. The only partial exception, QWEN2.5-3B-INSTRUCT, displays erratic behavioral changes with a spiky, non-smooth response curve. Closer inspection reveals these outputs are often incoherent or filled with irrelevant content, indicating instability rather than intentional modulation. These results reinforce the critical role of behaviorally meaningful and well-aligned steering directions in achieving effective, stable, and interpretable control over model behavior.



(a) Qwen2.5-3B-Instruct

(b) Qwen2.5-7B-Instruct.

(c) Qwen2.5-14B-Instruct

(d) Llama-3.2-3B-Instruct.

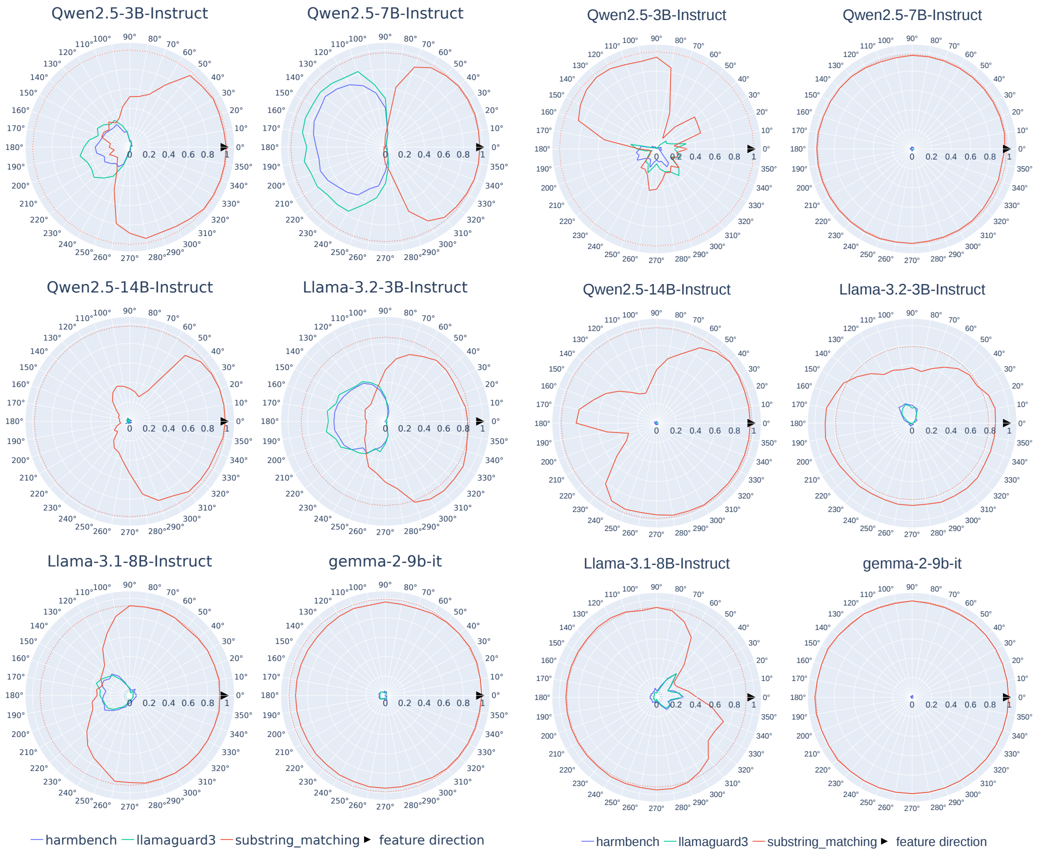
(e) Llama-3.1-8B-Instruct

(f) gemma-2-9b-it.

Figure 12: Projections of the feature directions extracted at each extraction point (i.e. ${\boldsymbol{d}^i}_{\text{feat}}$) on the steering plane for all tested models.

E Related Works

Mechanistic Motivation. Activation steering techniques have typically involved scaling activation directions by manually tuned scalar coefficients to induce or suppress behaviors (Turner et al., 2024; Zou et al., 2023a; Templeton et al., 2024; Bayat et al., 2025; Li et al., 2024; von Rütte et al., 2024; Stolfo et al., 2024). However, selecting these coefficients is challenging due to sensitivity to the activation norm, which grows exponentially across



(a) Steering on a plane spanned by $\hat{\boldsymbol{d}}_{\text{feat}}$ and a random direction.

(b) Steering on a plane spanned by 2 random directions.

Figure 13: Ablation study of steering with random direction(s).

layers (Fig.9 left). As observed by (Turner et al., 2024; Templeton et al., 2024), inappropriate scaling often results in incoherent generations, highlighting the fragility of this approach. Directional ablation, another popular technique, avoids explicit hyperparameter tuning by orthogonalizing activations relative to a feature direction (Arditi et al., 2024; Zou et al., 2023a). Yet, this approach neglects scenarios where negative alignment coefficients meaningfully reverse behavior, a limitation recognized in earlier studies (Turner et al., 2024; Zou et al., 2023a; Templeton et al., 2024). Empirical findings from our experiments further validate that extracted feature directions effectively distinguish contrastive data sets (Fig.9 right).

Recent advancements include adaptive steering methods such as Adaptive Activation Steering (ACT), which dynamically adjusts steering intensity based on the activation context (Wang et al., 2025), and Contrastive Activation Addition (CAA), which employs multiple positive-negative example pairs for robust feature extraction (Panickssery et al., 2023). These techniques underscore the necessity for more nuanced control methods.

Architectural Motivation. Contemporary LLMs such as LLAMA 3 (Llama Team, 2024), QWEN 2.5 (Yang et al., 2024), and GEMMA 2 (Gemma Team et al., 2024) universally adopt RMSNorm (Zhang & Sennrich, 2019) for normalization. RMSNorm effectively constrains activations to lie near a scaled unit sphere, emphasizing direction over magnitude. Moreover, Rotary Positional Embeddings (RoPE) and related variants (Su et al., 2024; bloc97, 2023; Chen et al., 2023; Peng et al., 2023) further validate this directional emphasis by encoding positional information as rotations. Methods such as Householder Pseudo-Rotation have extended this notion by explicitly employing norm-preserving geometric transformations to steer behaviors effectively and minimally invasively (Pham & Nguyen, 2024).

Empirical Motivation. Interpretability research consistently supports the Linear Representation hypothesis (Park et al., 2024; Bereska & Gavves, 2024), suggesting that LLM behaviors correspond to specific directions rather than discrete neuron activations. Further corroborated by the Superposition Hypothesis (Elhage et al., 2022), these directions are nearly orthogonal and quantify feature strength through scalar projections (Arditi et al., 2024; Bayat et al., 2025; Bricken et al., 2023; Gao et al., 2024a; Marks et al., 2025; von Rütte et al., 2024; Templeton et al., 2024; Belrose, 2023; Marks & Tegmark, 2024; Rimsky et al., 2024; Tigges et al., 2023). Moreover, it has been demonstrated that norm-preserving interventions, such as rotations, inherently provide stability and maintain general capabilities during steering (von Rütte et al., 2024; Pham & Nguyen, 2024).

Methods leveraging these insights have proliferated, notably Activation Scaling (Stoehr et al., 2024) and FairSteer (Li et al., 2025), which dynamically modulate activations to enhance transparency and reduce bias, respectively.

Our work expands upon these foundations by introducing Angular Steering, a generalization of existing activation steering techniques. By explicitly treating steering as a rotation in a defined 2D subspace, our method achieves more robust, interpretable, and flexible behavior control. We demonstrate Angular Steering using refusal steering as a running example, aligning closely with prior behavioral control research (Arditi et al., 2024; Lee et al., 2024). Rather than focusing on jailbreak or maximizing downstream accuracy, our goal is to present a principled and broadly applicable framework for controlled and non-destructive intervention in LLM activations.

F Compute statement

This research was conducted using mainly Nvidia H100 GPUs with 80GB of memory. For each model:

- Constructing the steering plane took about 15 minutes on 1 GPU using TRANSFORMERLENS (Nanda & Bloom, 2022).
- Pre-generating responses for evaluation took about 10 minutes on 1 GPU using our fork of vLLM (Kwon et al., 2023) as the serving engine.
- Evaluation with `substring matching` (Arditi et al., 2024), LLAMA 3 GUARD (Llama Team, 2024) and HARMBENCH (Mazeika et al., 2024) collectively took about 10 minutes on 1 GPU using vLLM (Kwon et al., 2023) as the serving enging.
- Evaluation with LLM-as-a-judge took about 50 minutes on 4 GPUs using vLLM (Kwon et al., 2023) as the serving engine.
- Computing perplexity scores took about 5 minutes on 1 GPU.
- Evaluation with TINYBENCHMARKS (Maia Polo et al., 2024) took about 4 hours on 1 GPU using vLLM (Kwon et al., 2023) as the serving engine and LM HARNESS (Gao et al., 2024b) as the evaluation device.