A TIME–FREQUENCY PERSPECTIVE ON ACTIVATION STEERING IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Activation steering offers a lightweight way to control the behavior of large language models by modifying hidden states during inference. We investigate how changes in these states across token positions can guide such interventions. Taking a time–frequency perspective, we apply the discrete cosine transform (DCT) along token positions to separate the mean hidden state from variation at different frequencies. An analysis of 500 concepts from Concept500 reveals attribute information in nonzero-frequency coefficients and shows that this information is associated with patterns of hidden-state variation. These findings motivate SARA, a training-free method that uses recent hidden-state changes as feedback to adjust a base steering direction. During generation, it computes this feedback by comparing nonzero-frequency coefficients from recent hidden states with statistics estimated from the calibration set. Across three instruction-tuned models, SARA achieves the highest scores among the evaluated methods on all three TruthfulQA multiple-choice metrics. On the math and code subsets of Concept500, it also outperforms the trained baselines in concept relevance and aggregate score.

1 INTRODUCTION

Large language models (LLMs) have achieved remarkable capabilities across diverse domains (Brown et al., 2020), yet ensuring their outputs align with intended behavioral constraints remains a central challenge (Ouyang et al., 2022). Activation steering offers a lightweight and flexible way to control model behavior at inference time, without retraining or modifying model parameters (Turner et al., 2023). It operates by directly intervening on hidden activations during a forward pass, enabling targeted behavioral changes while leaving the underlying model fixed (Hernandez et al., 2024).

Existing steering directions are constructed in several ways: by averaging activation differences between contrasting examples (Rimsky et al., 2024), fitting supervised probes (Li et al., 2023), extracting principal components from demonstrations (Zou et al., 2023), directly optimizing a vector (Subramani et al., 2022), or using features learned by sparse autoencoders (He et al., 2025). More recent methods adapt the intervention to the input or the model's current state (Wang et al., 2025), update it during generation (Do et al., 2025), or change only selected dimensions (Feng et al., 2026).

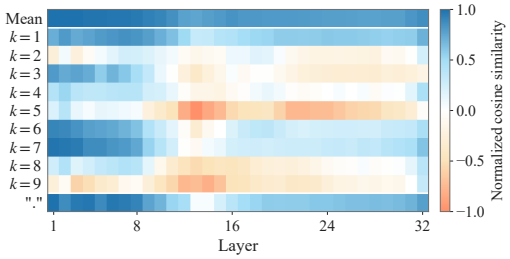


Figure 1: Cosine similarity between the two sentences' representations across layers of Llama-3.1-8B-Instruct. Rows compare mean hidden states (Mean), DCT coefficient vectors (frequency k), and states at the final period (“.”). Values are min–max normalized to $[-1, 1]$. Appendix C provides additional sentence-pair comparisons.

These approaches have proven effective in steering model behavior. However, existing methods either do not explicitly model or even ignore changes in hidden states across token positions. These changes range from gradual variation over many tokens to rapid fluctuations between adjacent tokens, corresponding to low and high frequencies, respectively. Although the last hidden state incorporates the preceding context, using it alone does not explicitly model variation across token positions, while sequence averaging removes these

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

changes. For example, *"Paris is the capital of France, not Germany."* and *"Paris is the capital of Germany, not France."* express opposing claims using the same words. To examine hidden-state variation in these sentences, we apply the discrete cosine transform (DCT; §3.1), a form of Fourier analysis. Treating token position as time, the DCT yields a coefficient vector at each frequency: the zero-frequency coefficient represents the sequence mean, while nonzero-frequency coefficients capture variation at different rates. As shown in Figure 1, the two sentences' mean hidden states remain highly similar across layers, while several nonzero-frequency coefficients have substantially lower cosine similarity than either the means or the states at the final period. Based on the limitations of previous work and the hypothesis that hidden-state changes contain information about desired behavior, we study **how such information is organized across frequencies and how to use these changes for activation steering**.

To address these questions, we first take a time–frequency view of attribute information in hidden states (§3). Specifically, we apply the DCT to decompose a hidden-state sequence into its mean and variation around that mean (§3.1). Using this decomposition, we conduct a systematic analysis of hidden-state sequences from a spectral perspective, with experiments covering 500 concepts from the Concept500 dataset (Wu et al., 2025). Through linear probing, we obtain two key findings: (i) hidden-state changes carry information about specific attributes, with low nonzero frequencies being particularly informative (§3.2); and (ii) such information is closely tied to the pattern of hidden-state changes across token positions (§3.3). **Together, these findings suggest that attribute information about model behavior is also reflected in how hidden states vary across token positions.**

Motivated by these findings, we propose **S**pectral **A**ttribute-guided **R**eorientation of **A**ctivations (SARA), a training-free activation-steering method that uses hidden-state changes to adjust its steering direction during generation (§4). Before generation, SARA estimates statistics of DCT coefficients from positive and negative responses in the calibration set. During generation, it analyzes recent hidden states in the frequency domain and compares their nonzero-frequency coefficients with these statistics. The resulting feedback adjusts the base direction before each next-token prediction.

Our contributions are as follows:

- We introduce a time–frequency view of attribute information in hidden states. Through a DCT decomposition and experiments on 500 concepts, we identify the attribute information in nonzero-frequency coefficients, particularly at low frequencies, and relate this decodability to patterns of hidden-state variation.
- We propose SARA, a training-free activation-steering method that uses calibration statistics to interpret changes in recent hidden states and provide feedback for adjusting a base steering direction during generation.
- We evaluate SARA across four instruction-tuned models and four benchmarks. It achieves the highest scores among the evaluated methods on all three TruthfulQA multiple-choice metrics across three models. Ablations further support using nonzero-frequency information as feedback for steering.

2 PRELIMINARIES

We consider a decoder-only Transformer with fixed model parameters (Vaswani et al., 2017). Given a prompt x, the model generates a response $a = (a_1, \ldots, a_T)$. Let $a_{<t} = (a_1, \ldots, a_{t-1})$ denote the response prefix available before predicting token a_t. **Activation steering** modifies hidden states during generation to guide the model toward responses with a target attribute, such as truthfulness.

At prediction step t, let $\mathbf{h}_t \in \mathbb{R}^d$ denote the residual activation at the last available token, measured at the output of a chosen layer after processing x and $a_{<t}$. We omit the layer index throughout. At $t = 1$, $\mathbf{h}_1$ is the activation at the final prompt token. Steering adds an intervention $\Delta\mathbf{h}_t \in \mathbb{R}^d$:

$$\mathbf{h}_t \longleftarrow \mathbf{h}_t + \Delta\mathbf{h}_t. \tag{1}$$

The modified state is then passed through the remaining layers to predict a_t. For additive steering, we write $\Delta\mathbf{h}_t = \alpha\mathbf{v}_t$, where $\mathbf{v}_t \in \mathbb{R}^d$ is the steering direction and $\alpha \geq 0$ controls the intervention strength. The direction may remain fixed or be updated during generation.

Steering directions can be constructed from a calibration set of labeled responses:

$$\mathcal{D} = \{(x_q, \mathcal{A}_q^+, \mathcal{A}_q^-)\}_{q=1}^{Q}, \tag{2}$$

where x_q is one of Q prompts, and $\mathcal{A}_q^+$ and $\mathcal{A}_q^-$ contain responses to x_q with and without the target attribute, respectively. We use $y \in \{+, -\}$ to denote the response label. These examples allow us to identify differences in hidden states associated with the target attribute.

3 A Time–Frequency View of Attribute Information

In this section, we study how information about target attributes is reflected in changes in hidden states across token positions. We first treat token position as time and decompose hidden-state sequences into their mean and variation at different frequencies. Using Concept500 (Wu et al., 2025), we then probe individual frequency coefficients for concept information and examine how shuffling the extracted states affects the results.

3.1 Decomposing Hidden-State Sequences by Frequency

For n consecutive token positions at the chosen layer, we write

$$\mathbf{H} = [\mathbf{h}_0, \ldots, \mathbf{h}_{n-1}]^\top \in \mathbb{R}^{n \times d}, \tag{3}$$

where j indexes positions within this sequence. Each hidden state depends only on tokens up to its position. We treat token position as discrete time, so each hidden coordinate forms a real-valued signal in the *time domain*.

To describe variation at different frequencies, we apply the orthonormal discrete cosine transform (DCT)[1] along token positions, separately for each hidden coordinate (Strang, 1999). For $k = 0, \ldots, n-1$, this gives

$$\begin{gathered} \mathbf{C}_k(\mathbf{H}) = \sum_{j=0}^{n-1} \mathbf{B}_n(j,k)\mathbf{h}_j, \qquad \mathbf{B}_n(j,k) = a_k \cos\!\left(\frac{\pi(j+1/2)k}{n}\right), \\ a_0 = \frac{1}{\sqrt{n}}, \qquad a_k = \sqrt{\frac{2}{n}} \quad (k > 0). \end{gathered} \tag{4}$$

The vector $\mathbf{C}_k(\mathbf{H}) \in \mathbb{R}^d$ collects the coefficients at frequency k for all hidden coordinates. The basis matrix satisfies $\mathbf{B}_n^\top \mathbf{B}_n = \mathbf{I}_n$: its columns form an orthonormal basis over token positions.

The constant basis function at $k = 0$ defines the *direct-current (DC) component*, while the basis functions at $k > 0$ define the *alternating-current (AC) components*. Let $\bar{\mathbf{h}} = n^{-1} \sum_{j=0}^{n-1} \mathbf{h}_j$ denote the mean hidden state across token positions. Since $\mathbf{B}_n(j,0) = 1/\sqrt{n}$, the zero-frequency coefficient is

$$\mathbf{C}_0(\mathbf{H}) = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} \mathbf{h}_j = \sqrt{n}\, \bar{\mathbf{h}}.$$

The inverse DCT (IDCT) reconstructs each hidden state as

$$\mathbf{h}_j = \underbrace{\frac{1}{\sqrt{n}} \mathbf{C}_0(\mathbf{H})}_{\text{DC}=\bar{\mathbf{h}}} + \underbrace{\sum_{k=1}^{n-1} \mathbf{B}_n(j,k)\mathbf{C}_k(\mathbf{H})}_{\text{AC}=\mathbf{h}_j - \bar{\mathbf{h}}}. \tag{5}$$

Thus, the DC component is the same mean vector at every position, while the AC components sum to each state's deviation from that mean. For a fixed length n, smaller nonzero k corresponds to slower variation across token positions, whereas larger k corresponds to faster variation. When each response is represented by its mean hidden state, contrasting positive and negative responses yields a steering direction based only on their DC components. We next examine whether the nonzero-frequency coefficients also contain attribute information.

[1] More precisely,this transform is known as the DCT-II.

3.2 Concept Information across Frequencies

Probing task. We assess concept information using a linear probe at each frequency (Alain & Bengio, 2017). For each concept c and frequency k, we fit a logistic-regression classifier that predicts concept presence from $\mathbf{C}_k(\mathbf{H}) \in \mathbb{R}^d$:

$$\widehat{y}_{c,k} = \sigma\big(\mathbf{w}_{c,k}^\top \mathbf{C}_k(\mathbf{H}) + b_{c,k}\big)\,. \tag{6}$$

Here, σ is the logistic sigmoid, $\mathbf{w}_{c,k}$ and $b_{c,k}$ are the fitted classifier parameters, and $\widehat{y}_{c,k}$ is the predicted probability that concept c is present. Performance above chance on held-out responses indicates linearly decodable concept information at that frequency.

Experimental setup. We use labeled responses from Concept500, released with AxBench (Wu et al., 2025). Concept500 contains 500 natural-language concepts associated with sparse-autoencoder features. For each response, we extract 64 consecutive hidden states from layer 20 of Gemma-2-9B-it, beginning at the final prompt token, and apply the DCT in Equation (4). Further experimental details are provided in Appendix D.1.

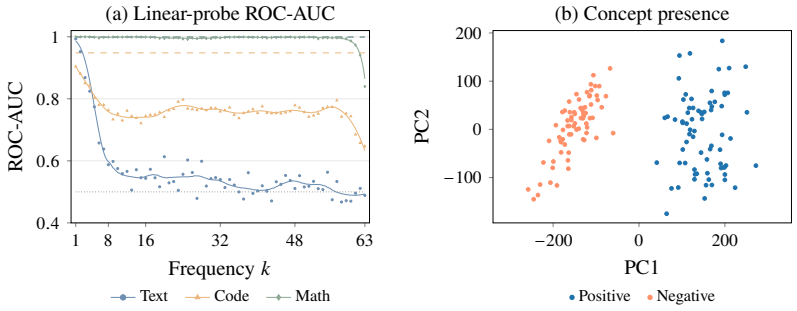


Figure 2: **Concept information in frequency coefficients.** (a) Probe ROC-AUC by genre. Markers: frequency-wise means; solid curves: LOESS trends; dashed lines: $k = 0$; dotted lines: chance. (b) PCA of $k = 1$ coefficients for positive and negative mathematical-notation responses.

Finding 1: Nonzero-frequency coefficients contain concept information, especially at low frequencies. As shown in Figure 2a, probes trained on the nonzero-frequency coefficients perform above chance in ROC-AUC, except at high frequencies for text. For text, code, and mathematics, ROC-AUC is highest at $k = 1$ among the nonzero frequencies. This frequency captures the slowest variation across the extracted sequence. Performance generally decreases as frequency increases, although it remains high at higher frequencies for mathematics. These results show that variation in hidden states across token positions contains linearly decodable concept information.

Figure 2b shows the $k = 1$ coefficients projected onto two dimensions using principal component analysis (PCA). Positive and negative mathematical-notation responses occupy different regions, illustrating differences in the coefficients associated with concept presence. Appendix D provides further details.

3.3 Probing Concept Information after Shuffling

We next examine whether concept information decoded at nonzero frequencies is associated with the pattern of hidden-state variation across token positions. To test this, we randomly shuffle the extracted state vectors independently within each response and recompute the DCT using Equation (4). We then train separate probes on the coefficients computed before and after shuffling, using identical training and test splits.

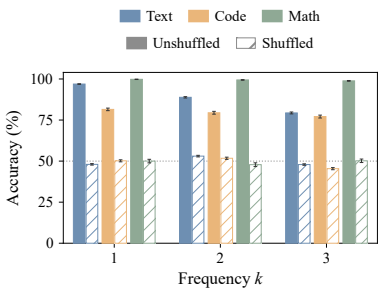


Figure 3: Linear-probe accuracy before and after shuffling hidden states within each response.

Finding 2: Concept information decoded at nonzero frequencies is associated with the pattern of hidden-state variation. After shuffling, probe accuracy at nonzero frequencies falls to near chance (Figure 3). The decrease is

largest at low frequencies, where accuracy is highest before shuffling. By contrast, the $k = 0$ coefficient and its probe accuracy remain unchanged because the mean hidden state across token positions is invariant to shuffling. These results suggest that the pattern of variation across token positions matters for linear decoding from individual nonzero-frequency coefficients. Appendix D.1 reports results for all frequencies.

4 METHODOLOGY

Our analysis in Section 3 reveals concept information in nonzero-frequency coefficients and associates this information with patterns of hidden-state variation. These findings motivate SARA, a training-free method that uses recent changes in hidden states as feedback for activation steering. As illustrated in Figure 4, SARA first estimates statistics of DCT coefficients from the calibration set. During generation, it applies the DCT to a sliding window of recent hidden states and compares the nonzero-frequency coefficients with these statistics. The resulting feedback is combined through the inverse DCT to correct the base direction before each next-token prediction.

This section describes SARA with a fixed base direction $\mathbf{v}$, computed as the difference between the mean final-token activations of positive and negative responses in the calibration set (Rimsky et al., 2024). Appendix B.1 presents the variant with an online base direction and the corresponding feedback correction.

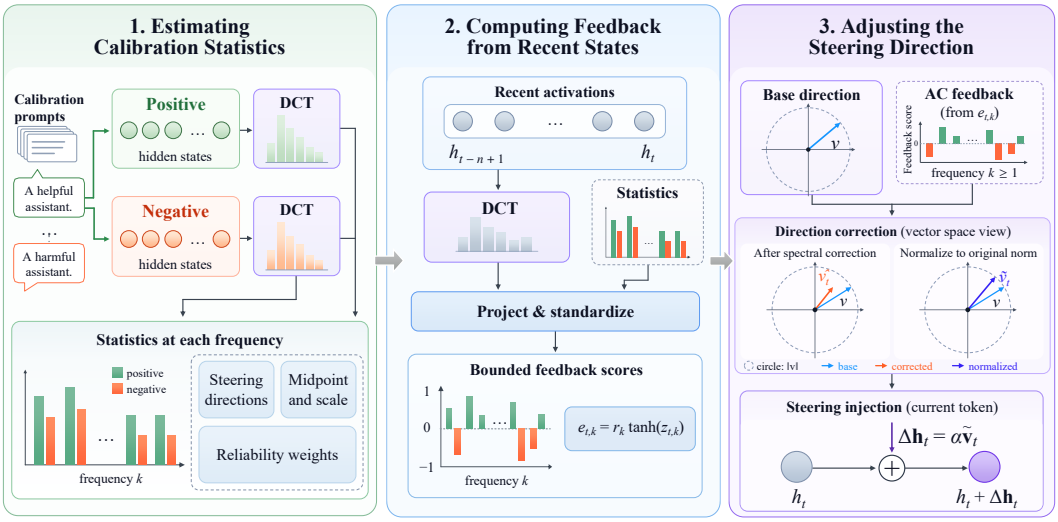


Figure 4: **Overview of SARA.** SARA uses calibration statistics and the DCT coefficients of the hidden-state window to compute feedback. The resulting feedback adjusts a base steering direction through inverse-DCT reconstruction, followed by rescaling and injection.

4.1 ESTIMATING CALIBRATION STATISTICS

Before generation, we estimate statistics of DCT coefficients from the calibration set $\mathcal{D}$. For each calibration prompt x and response a, we collect the first n hidden states defined in Section 2 into $\mathbf{H}_n(x, a) = [\mathbf{h}_1(x, a), \ldots, \mathbf{h}_n(x, a)]^\top$. We estimate the statistics separately for each length $n \leq W$, where W is the maximum window length, so that each generation window is compared with calibration sequences of the same length.

Steering directions. At each frequency k, we estimate the mean DCT coefficients of positive and negative calibration responses and take their difference:

$$\boldsymbol{\mu}_k^y = \frac{1}{Q} \sum_{q=1}^{Q} \frac{1}{|\mathcal{A}_q^y|} \sum_{a \in \mathcal{A}_q^y} \mathbf{C}_k(\mathbf{H}_n(x_q, a)), \qquad \boldsymbol{\delta}_k = \boldsymbol{\mu}_k^+ - \boldsymbol{\mu}_k^-. \tag{7}$$

For each label, the coefficients are first averaged over responses to each prompt, and the resulting means are then averaged across prompts. When $\boldsymbol{\delta}_k \neq \mathbf{0}$, it is normalized to obtain the unit vector

$\mathbf{u}_k = \frac{\boldsymbol{\delta}_k}{\|\boldsymbol{\delta}_k\|_2}$ for projection later. The unnormalized vector $\boldsymbol{\delta}_k$ is retained for inverse-DCT reconstruction of the direction correction.

Midpoint and scale. At each frequency k, we estimate a midpoint m_k and a scale s_k to standardize the projections of DCT coefficient vectors onto $\mathbf{u}_k$. We first estimate the within-class variance for each label y:

$$\nu_k^y = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{|\mathcal{A}_q^y|}\sum_{a\in\mathcal{A}_q^y}\left[\mathbf{u}_k^\top\left(\mathbf{C}_k(\mathbf{H}_n(x_q,a)) - \boldsymbol{\mu}_k^y\right)\right]^2. \tag{8}$$

We then define the midpoint and scale as

$$m_k = \frac{\mathbf{u}_k^\top\boldsymbol{\mu}_k^+ + \mathbf{u}_k^\top\boldsymbol{\mu}_k^-}{2}, \qquad s_k = \sqrt{\max\left(\frac{\nu_k^+ + \nu_k^-}{2}, 10^{-8}\right)}. \tag{9}$$

The variance floor prevents division by zero.

Reliability weights. The positive–negative difference $\boldsymbol{\delta}_k$ may reflect more than the target attribute; it can also capture prompt content, response phrasing, and other information unrelated to the target attribute. We assign a smaller reliability weight r_k when direction vectors estimated from different subsets have lower cosine similarity.

We assign the calibration prompts randomly to two groups, keeping each prompt's positive and negative responses together. We estimate $\boldsymbol{\delta}_k^{(1)}$ and $\boldsymbol{\delta}_k^{(2)}$ separately for the two groups using Equation (7) and define their nonnegative cosine similarity as the reliability weight:

$$r_k = \max\left(0,\ (\boldsymbol{\delta}_k^{(1)})^\top\boldsymbol{\delta}_k^{(2)} / \left(\left\|\boldsymbol{\delta}_k^{(1)}\right\|_2\left\|\boldsymbol{\delta}_k^{(2)}\right\|_2\right)\right). \tag{10}$$

The reliability weight measures stability across calibration subsets.

4.2 Computing Feedback from Recent Hidden States

To capture variation across token positions, we compute feedback from recent hidden states rather than the current state alone. At prediction step t, we form a sliding window of up to W available hidden states:

$$\mathbf{H}_t = [\mathbf{h}_{t-n+1},\ldots,\mathbf{h}_t]^\top \in \mathbb{R}^{n\times d}, \qquad n = \min(t, W). \tag{11}$$

Using the calibration statistics for length n, we project and standardize each nonzero-frequency coefficient to compute the feedback:

$$z_{t,k} = \frac{\mathbf{u}_k^\top\mathbf{C}_k(\mathbf{H}_t) - m_k}{s_k}, \qquad e_{t,k} = r_k\tanh(z_{t,k}). \tag{12}$$

A positive $z_{t,k}$ indicates that the projection is closer to the positive projected mean; a negative value indicates that it is closer to the negative projected mean. Applying $\tanh$ limits the influence of extreme standardized projections, while r_k reduces feedback from inconsistent direction estimates.

4.3 Adjusting the Steering Direction

To convert the scalar feedback into a vector correction, we assign the coefficient $-\rho e_{t,k}\boldsymbol{\delta}_k$ to each nonzero frequency, where $\rho \geq 0$ controls the feedback strength. The vector $\boldsymbol{\delta}_k$ provides the direction and magnitude of the difference between the class means. The minus sign defines a negative-feedback rule for these correction coefficients: a negative score gives a correction coefficient along $\boldsymbol{\delta}_k$, whereas a positive score gives one in the opposite direction. The midpoint provides the reference for this feedback, while the base direction supplies the intervention intended to encourage the target attribute.

To combine these corrections at the current token, we use the inverse DCT from Equation (5) and evaluate it at the last position of the window:

$$\widehat{\mathbf{v}}_t = \mathbf{v} - \rho\sum_{k=1}^{n-1}\mathbf{B}_n(n-1,k)\,e_{t,k}\,\boldsymbol{\delta}_k. \tag{13}$$

Table 1: TruthfulQA results (%) for multiple-choice evaluation and open-ended generation. For each model and metric, the best score is shown in **bold** and the second-best is underlined.

Model	Method	Multiple-Choice			Open-ended Generation		
		MC1	MC2	MC3	True	Info	True×Info
Llama-3.1-8B-Instruct	Base	37.82	56.10	32.01	71.46	98.89	70.67
	ICL	38.31	54.64	31.93	71.78	96.10	68.99
	ITI	32.56	53.78	31.66	**83.23**	57.79	48.10
	CAA	38.43	55.90	32.54	75.00	93.37	70.03
	SADI-Head	36.47	55.92	31.37	69.91	**99.17**	69.32
	Spherical Steering	53.00	69.49	48.42	82.76	97.19	**80.44**
	PID-AcT	37.94	56.02	31.92	71.11	99.01	70.41
	SARA	**55.20**	**73.94**	**53.07**	81.16	98.24	79.73
Gemma-2-9B-it	Base	38.80	60.09	33.26	72.84	99.46	72.45
	ICL	41.37	59.36	34.59	68.75	**99.51**	68.41
	ITI	39.78	60.90	34.79	77.45	96.65	74.85
	CAA	39.05	59.08	32.21	71.09	99.44	70.69
	SADI-Head	38.19	60.29	33.10	72.90	99.09	72.24
	Spherical Steering	50.31	71.92	52.78	**88.04**	54.31	47.81
	PID-AcT	38.68	59.88	32.85	72.07	99.48	71.70
	SARA	**58.75**	**75.26**	**57.06**	85.65	98.75	**84.57**

Here, $\mathbf{B}_n(n-1,k)$ is the value of the kth cosine basis at that position. Only the current state is modified; earlier states supply the feedback.

Finally, we rescale the corrected vector to separate changes in direction from intervention strength:

$$\widetilde{\mathbf{v}}_t = \|\mathbf{v}\|_2 \frac{\widehat{\mathbf{v}}_t}{\max(\|\widehat{\mathbf{v}}_t\|_2, \epsilon_v)}, \qquad \Delta\mathbf{h}_t = \alpha\widetilde{\mathbf{v}}_t. \tag{14}$$

The constant $\epsilon_v > 0$ prevents division by zero. For $\|\widehat{\mathbf{v}}_t\|_2 \geq \epsilon_v$, the rescaled vector has the same norm as the base direction. Thus, ρ controls the correction before rescaling, while α controls the final intervention strength. Further implementation details are provided in Appendix B; Appendix F presents a conditional analysis of the feedback.

5 Experiments

5.1 Experimental Setup

TruthfulQA. We evaluate SARA on TruthfulQA (817 questions) (Lin et al., 2022) using Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, and Gemma-2-9B-it. Baselines include six-shot in-context learning (ICL), ITI (Li et al., 2023), CAA (Rimsky et al., 2024), SADI-Head (Wang et al., 2025), Spherical Steering (You et al., 2026), and PID-AcT (Nguyen et al., 2026). We report MC1, MC2, and MC3 for multiple-choice evaluation, and truthfulness (True), informativeness (Info), and True×Info for generation.

Concept500. We evaluate concept steering on Gemma-2-2B-it using 142 math and code concepts from Concept500 and 710 held-out concept–instruction pairs. We compare with ReFT-r1 (Wu et al., 2025), LoReFT (Wu et al., 2024), LoRA (Hu et al., 2022), and SFT, alongside the unsteered Base model. GPT-4o-mini scores Concept relevance, Instruction relevance, and Fluency on a 0–2 scale.

For both tasks, SARA uses the online base-direction variant described in Appendix B.1, with calibration statistics estimated for each target. Data splits, baseline configurations, and evaluation details are provided in Appendix E.

5.2 Main Results

How does SARA compare with inference-time baselines? As shown in Table 1, SARA achieves the highest MC1, MC2, and MC3 scores on both Llama and Gemma On TruthfulQA. Specifically, compared with Spherical Steering, SARA achieves absolute MC2 gains of 4.45% and 3.34% on

Table 2: Comparison with trained baselines on the math and code subsets of Concept500 using Gemma-2-2B-it. For each metric, the best score is shown in **bold** and the second-best is underlined.

Method	Concept ↑	Instruction ↑	Fluency ↑	Avg. ↑
Base	0.0620	**1.9732**	1.3704	0.0723
ReFT-r1	0.3113	1.8493	1.0127	0.1635
LoReFT	0.2099	1.6746	1.2141	0.2101
LoRA	0.1479	1.9254	**1.4549**	0.1579
SFT	0.4394	1.6465	1.3887	0.3632
SARA	**1.2423**	1.9028	1.0225	**1.0569**

Llama and Gemma, respectively. In generation, SARA improves truthfulness over Base while retaining similar informativeness, achieving the highest True×Info on Gemma and the second-highest on Llama. SARA also leads all three multiple-choice metrics on Qwen; these results are reported in Appendix E.2 (Table 6).

How does SARA compare with trained methods for concept steering? Without training an intervention module or finetuning the model, SARA achieves higher aggregate Concept and Avg. scores than ReFT-r1, LoReFT, LoRA, and SFT (Table 2). Instruction relevance remains close to Base, although fluency is lower than that of LoReFT, LoRA, and SFT. These results extend the applicability of SARA beyond truthfulness: the same steering procedure supports diverse math and code concepts through calibration for each target and remains effective on held-out instructions.

Further evaluations on MMLU and RealToxicityPrompts extend the study to general knowledge and toxicity control, using calibration for each task (Appendix E.3).

5.3 Ablation Studies

We use Llama on TruthfulQA at block 24 with $W = 16$, vary the steering strength α from 0.125 to 3, and report MC2. All steering variants use the same calibration data, norm rescaling, and online base-direction construction (Appendix B.1).

Which frequency components should guide the correction? We evaluate three variants of SARA that use only the zero-frequency component (DC), all nonzero-frequency components (AC), or both (DC+AC) to correct the base direction. The RePE control applies the base direction without correction. The frequency choices affect only the correction; none replaces the base direction with a single component. As shown in Figure 5, AC achieves higher MC2 than DC at every tested strength, while adding DC to AC provides no consistent additional benefit. These results support using variation across token positions to guide the correction.

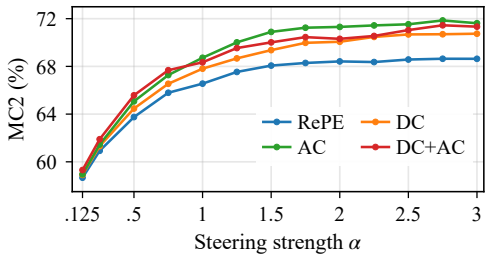


Figure 5: DC/AC component ablation on Llama–TruthfulQA. RePE omits the correction; DC/AC specify its frequency components.

Does feedback improve on the base direction alone? Figure 6 highlights the comparison between applying the base direction alone (w/o feedback, corresponding to RePE) and adding the AC correction (w/ feedback, corresponding to AC). The unsteered Base model provides a reference independent of α. Both steering variants outperform Base, and enabling feedback further improves MC2 at all tested strengths. These results show that feedback from recent hidden-state changes improves steering beyond the base direction alone.

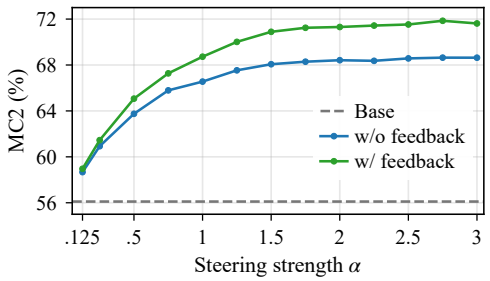


Figure 6: Feedback ablation on Llama–TruthfulQA. Base is the unsteered model; the other curves disable or enable the AC correction.

Full MC1/MC2/MC3 results and configuration details are provided in Appendix E.5. The ablation of

r_k on Concept10 and sensitivity to window length, feedback strength, and intervention layer are reported in Appendices E.5.3 and E.5.2, respectively.

Does reliability weighting improve concept steering? We evaluate Gemma-2-2B-it on Concept10, where positive and same-genre negative calibration responses are unpaired by instruction. We compare the estimated reliability weights r_k with uniform weights $r_k = 1$, retaining AC feedback and the same calibration data, prompts, and generation seeds. As shown in Figure 7, reliability weighting increases Concept relevance at every tested strength and improves Avg. at $\alpha \in \{1, 2, 3\}$. The gains are largest at lower strengths. Appendix E.5.3 provides further details.

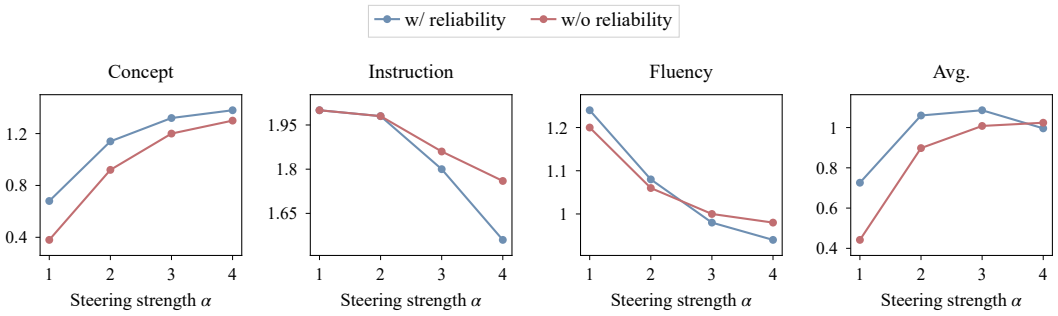


Figure 7: Reliability-weight ablation on Gemma-2-2B-it with Concept10. The curves compare estimated r_k with $r_k = 1$. Scores use the 0–2 scale.

6 Related Work

Activation steering connects interpretable representations with behavioral control. CAA constructs fixed steering directions from contrasting examples (Rimsky et al., 2024), while ReFT learns representation interventions with the base model frozen (Wu et al., 2024). Recent methods make steering adaptive: SADI conditions interventions on current activations (Wang et al., 2025), Spherical Steering gates tokenwise rotations (You et al., 2026), and PID-AcT incorporates feedback across layers (Nguyen et al., 2026). These developments motivate examining what information should guide an intervention as generation proceeds.

Spectral analysis offers a way to characterize hidden-state changes across token positions. DCT sentence embeddings retain information beyond the sequence mean (Almarwani et al., 2019). The prism framework uses token-position frequencies to represent linguistic structure at different scales (Tamkin et al., 2020). Spectral Editing of Activations instead uses singular value decomposition (SVD) of activation cross-covariances to construct projections in hidden space (Qiu et al., 2024). Our work builds on spectral representations along token positions by examining how frequency coefficients encode target attributes and using statistics from positive and negative responses to adapt steering during generation. A fuller discussion, including detailed comparisons with adaptive steering and spectral activation editing methods, is provided in Appendix G.

7 Conclusion

In this work, we study activation steering from a time–frequency perspective, focusing on how hidden-state variation across token positions can inform interventions. Using the DCT, we separate the mean hidden state from variation at different frequencies. Our analysis of 500 concepts reveals attribute information in nonzero-frequency coefficients, particularly at low frequencies, and relates this decodability to patterns of hidden-state variation. Building on these findings, SARA uses calibration statistics to interpret recent hidden-state changes and adjust a base steering direction through feedback, without additional training. Experiments on TruthfulQA and Concept500 demonstrate the effectiveness of SARA, with ablations supporting the benefit of nonzero-frequency feedback. Together, these results highlight hidden-state variation as a useful source of feedback for adaptive control of model behavior.

AI USE STATEMENT

We used large language models, including GPT-5.6 sol and GPT-6 Astra, to discuss and explore candidate strategies and refine the design of SARA. We also used these tools to assist with mathematical proofs, retrieve and identify relevant literature, and draft and polish sections of the manuscript. GPT-5.6 sol, GPT-6 Astra, and Claude fable 5.1 all assisted with code writing. The authors verified the mathematical arguments and cited sources and reviewed the final manuscript. We take responsibility for the final content of this work, including all text, claims, code, and other research artifacts produced with AI assistance.

ETHICS STATEMENT

This work studies language-model activations and their use for steering on existing benchmarks. Activation steering can be misused to amplify harmful or misleading content, and improved benchmark scores do not establish general factual reliability or safe deployment. The models, calibration data, and automated evaluators may carry biases. Our toxicity experiments involve potentially offensive text, and toxicity classifiers and LLM-based judges provide imperfect assessments of generated responses. In particular, concept-relevance scores do not establish mathematical correctness or code safety. Applications of SARA should include safety evaluations suited to their intended use, monitoring, and appropriate safeguards.

REPRODUCIBILITY STATEMENT

Section 4 and Appendix B specify the steering procedure and the estimation of statistics from positive and negative responses. Section 5 and Appendices D and E document the datasets, data splits, calibration settings, probing and baseline configurations, decoding settings, evaluation metrics, parameter selection, and confidence intervals. Appendix F states the assumptions and provides proofs for the theoretical claims. We provide the code as supplementary material and plan to release it as open source.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations (Workshop Track)*, 2017.

Nada Almarwani, Hanan Aldarmaki, and Mona Diab. Efficient sentence embedding using discrete cosine transform. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3672–3678. Association for Computational Linguistics, 2019.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Tom B. Brown et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Van Dai Do, Quan Hung Tran, Svetha Venkatesh, and Hung Le. Dynamic steering with episodic memory for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 13731–13749. Association for Computational Linguistics, 2025.

Zijian Feng, Tianjiao Li, Zixiao Zhu, Hanzhang Zhou, Junlang Qian, Li Zhang, Jia Jim Deryl Chua, Lee Onn Mak, Gee Wah Ng, and Kezhi Mao. Fine-grained activation steering: Steering less, achieving more. In *International Conference on Learning Representations*, 2026.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3356–3369. Association for Computational Linguistics, 2020.

Zirui He, Mingyu Jin, Bo Shen, Ali Payani, Yongfeng Zhang, and Mengnan Du. SAE-SSV: Supervised steering in sparse representation spaces for reliable control of language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2207–2236. Association for Computational Linguistics, 2025.

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

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. Inspecting and editing knowledge representations in language models. In *First Conference on Language Modeling*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

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

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. Association for Computational Linguistics, 2022.

Sheng Liu, Haotian Ye, Lei Xing, and James Y. Zou. 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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32287–32307. PMLR, 2024.

Dung V. Nguyen, Nhi Y. Pham, Hieu M. Vu, Lei Zhang, and Tan M. Nguyen. Activation steering with a feedback controller. In *International Conference on Learning Representations*, 2026.

Long Ouyang et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39643–39666. PMLR, 2024.

Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo M. Ponti, and Shay B. Cohen. Spectral editing of activations for large language model alignment. In *Advances in Neural Information Processing Systems*, volume 37, pp. 56958–56987, 2024.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, 2024.

Gilbert Strang. The discrete cosine transform. *SIAM Review*, 41(1):135–147, 1999.

Nishant Subramani, Nivedita Suresh, and Matthew Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 566–581. Association for Computational Linguistics, 2022.

Alex Tamkin, Dan Jurafsky, and Noah Goodman. Language through a prism: A spectral approach for multiscale language representations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5492–5504, 2020.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Hieu M. Vu and Tan M. Nguyen. Angular steering: Behavior control via rotation in activation space. In *Advances in Neural Information Processing Systems*, volume 38, pp. 121653–121690, 2025.

Weixuan Wang, Jingyuan Yang, and Wei Peng. Semantics-adaptive activation intervention for LLMs via dynamic steering vectors. In *International Conference on Learning Representations*, 2025.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. ReFT: Representation finetuning for language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 63908–63962, 2024.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. AxBench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67035–67080. PMLR, 2025.

Zejia You, Chunyuan Deng, and Hanjie Chen. Spherical steering: Geometry-aware activation rotation for language models. In *International Conference on Machine Learning*, 2026.

Haiyan Zhao, Xuansheng Wu, Fan Yang, Bo Shen, Ninghao Liu, and Mengnan Du. Denoising concept vectors with sparse autoencoders for improved language model steering. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 797–808. Association for Computational Linguistics, 2026.

Andy Zou et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Limitations and Future Work

Window construction. We apply the DCT to hidden-state windows with a preset maximum token count. The relation between these windows and semantic units has yet to be examined. Future work could define windows using clause or sentence boundaries, or shifts in meaning, and study the resulting frequency representations and steering performance.

Model scale. Our experiments cover models with 2–9 billion parameters. Extending the evaluation to larger models would help assess how the observed frequency patterns and the effectiveness of SARA change with model scale.

B Additional Method Details

B.1 Constructing the base direction online

At each generation step, we take the difference between hidden states under contrasting prompts to obtain $\mathbf{v}_t$ (Zou et al., 2023). Positive and negative prompts ask the model to express or avoid the target attribute, respectively. We collect the most recent $n = \min(t, W)$ base directions and compute

$$\mathbf{V}_t = [\mathbf{v}_{t-n+1}, \ldots, \mathbf{v}_t]^\top, \qquad \mathbf{A}_{t,k} = \mathbf{C}_k(\mathbf{V}_t). \tag{15}$$

The attribute scores follow Equation 12. Replacing $\boldsymbol{\delta}_k$ in Equation 13 with $\mathbf{A}_{t,k}$ gives

$$\begin{aligned} \widehat{\mathbf{v}}_t &= \mathbf{v}_t - \rho \sum_{k=1}^{n-1} [\mathbf{B}_n]_{n-1,k}\, e_{t,k}\, \mathbf{A}_{t,k}, \\ \Delta \mathbf{h}_t &= \alpha \left\|\mathbf{v}_t\right\|_2 \frac{\widehat{\mathbf{v}}_t}{\max(\left\|\widehat{\mathbf{v}}_t\right\|_2, \epsilon_v)}. \end{aligned} \tag{16}$$

B.2 Activation windows and attribute priors

For a prompt x and supplied response a, we collect the state $\mathbf{h}_1(x, a)$ at the final prompt token and $\mathbf{h}_t(x, a)$ after processing $a_{<t}$. Statistics for length n use the initial window $[\mathbf{h}_1, \ldots, \mathbf{h}_n]^\top$, with maximum retained length $\min(W, |a|)$.

Statistics at length n use responses with at least n activation states. Here Q counts the retained prompts, and $\mathcal{A}_q^y$ contains the retained responses with label y to prompt q.

During generation, SARA uses the sliding window in Equation 11, containing only hidden states available before predicting the next token. For $t > W$, it applies statistics estimated from the first W calibration states to the last W hidden states.

C Additional Sentence-Pair Comparisons

We extend the example in Figure 1 to three sentence pairs on Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, and Gemma-2-9B-it. Each pair uses the same words in a different order: the capital pair contrasts *"Paris is the capital of France, not Germany."* with *"Paris is the capital of Germany, not France."*; the acquisition pair contrasts *"Google acquired YouTube."* with *"YouTube acquired Google."*; and the sentiment pair contrasts *"The movie is good, not bad."* with *"The movie is bad, not good."* We process each sentence directly, without a chat template, and retain all non-special tokens, including punctuation.

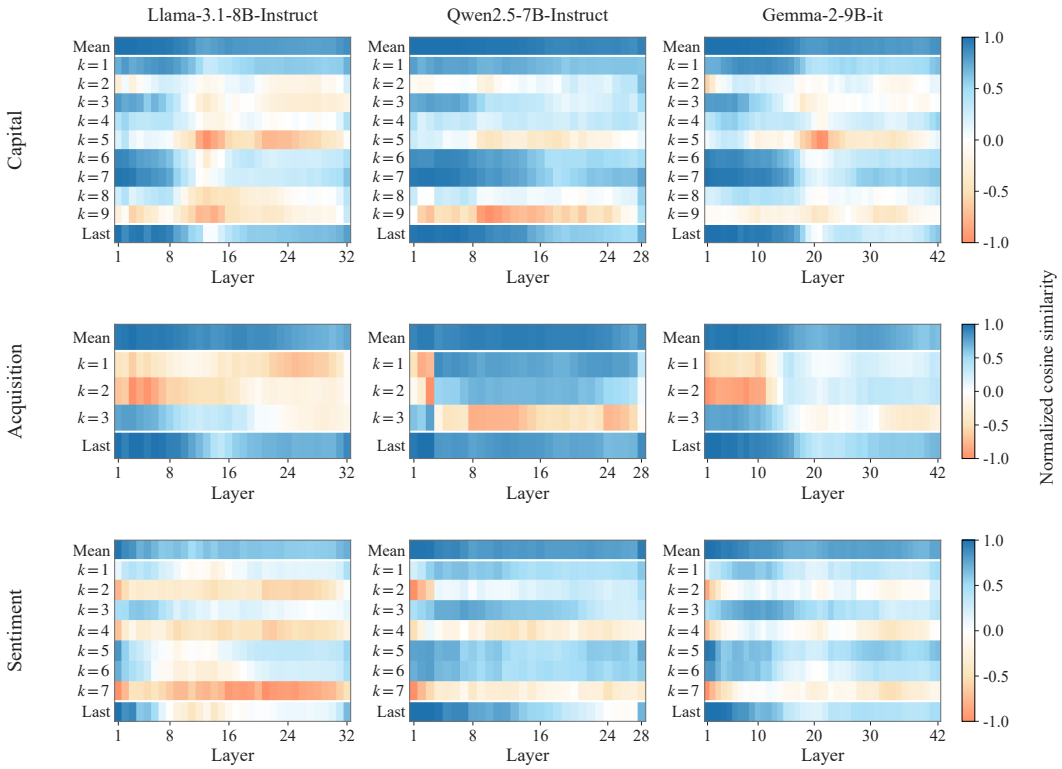


Figure 8: Layer-wise cosine similarity for three sentence pairs (rows) and models (columns), independently min-max normalized to $[-1, 1]$ over all cells of each panel. Colors show within-panel relative similarity; a colorbar is provided for each row. Mean is the token-state average, k the frequency index, and Last the final-period state.

Figure 8 uses $2(s - s_{\min})/(s_{\max} - s_{\min}) - 1$, with extrema over all rows and layers within each panel. Mean and period states generally have higher relative similarity than some nonzero-frequency components. For acquisition, low-similarity regions occur at $k = 1$ in Llama and $k = 3$ in Qwen. These patterns vary across models and pairs, illustrating differences that averaging can obscure.

D Supplementary Concept Analysis

D.1 Probe data and protocol

Concept500 provides natural-language descriptions of SAE features and responses generated to express them (Wu et al., 2025). We probe concept presence in the model's residual activations using the supplied labels, positive responses, and ordinary negative responses from the released training split. Probe fitting and evaluation use the repeated cross-validation protocol below.

Table 3: Concept500 records retained for the frequency probes. Every response supplies 64 consecutive states beginning at the final prompt token; shorter responses are excluded. Negative pools are shared within each genre.

	Text	**Code**	**Math**	**Total**
Concepts	334	128	38	500
Positive responses	20,083	8,316	2,657	31,056
Shared negative responses	54	72	72	198

We extract layer-20 states from Gemma-2-9B-it and fit an L2 logistic-regression probe for each concept and frequency $k = 0, \ldots, 63$. Features are divided by a single RMS estimated on the training fold. Probes use $C = 1$, balanced class weights, the `liblinear` solver, at most 2,000 iterations, and seed 42. Evaluation uses five-fold cross-validation repeated ten times. Accuracy and ROC-AUC are averaged over these 50 fits for each concept, then macro-averaged across concepts.

Figure 9 presents the full four-panel results corresponding to Figure 2: probe accuracy and ROC-AUC by genre, together with PCA visualizations of concept presence and identity. Panels (b) and (c) are retained in the main text.

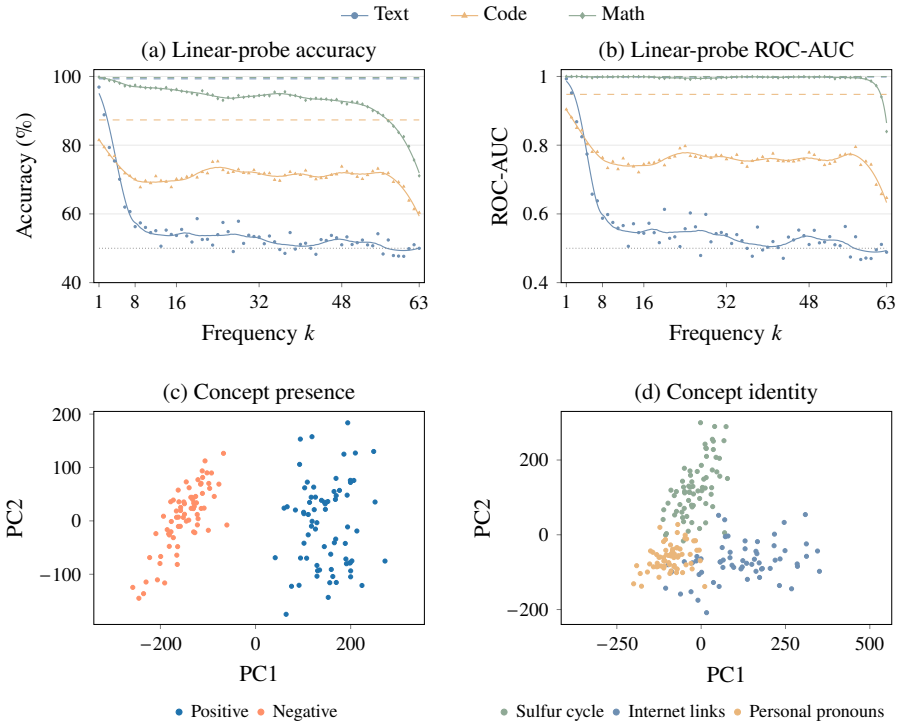


Figure 9: Full results for concept information in frequency coefficients. (a,b) Probe accuracy and ROC-AUC by genre. Markers: frequency-wise means; solid curves: LOESS trends; dashed lines: $k = 0$; dotted lines: chance. (c,d) PCA of $k = 1$ coefficients for mathematical notation and three text concepts. Positive and negative responses occupy different regions in (c), as do responses expressing different concepts in (d).

For visualization, Figure 2a, Figure 9a,b, and Figures 10–11 overlay LOESS trends on all frequency-wise means for $k = 1, \ldots, 63$. We fit local quadratic regressions with tricube weights to the logit-transformed metric means, using a common span of 0.25 (16 neighboring frequencies), then map the trends back to the original metric scales. Smoothing is used only for display; the $k = 0$ references and numerical analyses use the original results.

For the shuffle control, the model processes tokens in their original order. We then permute the 64 extracted state vectors independently within each response, using seed $42 +$ row ID, and recompute the DCT. Original and shuffled conditions fit separate probes on identical splits. The repeated fits vary the data partition, not the permutation seed.

Figures 10 and 11 extend the main analysis to all frequencies. Original trajectories carry the strongest concept information at low frequencies; shuffled trajectories remain near chance at nonzero frequencies. Pointwise error bars use 10,000 concept-bootstrap resamples within genre, conditional on the shared negative pools.

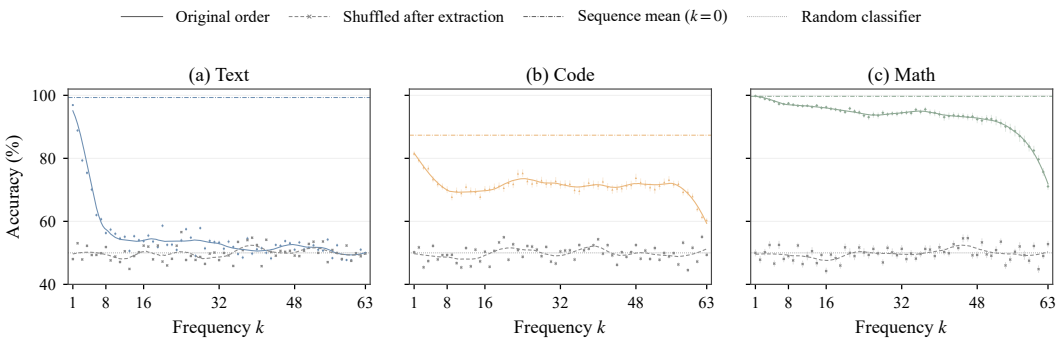


Figure 10: Probe accuracy by frequency and genre, before and after shuffling extracted states. Markers show observed means with pointwise 95% concept-bootstrap error bars; curves show LOESS trends. Original-order results use the genre colors and marker shapes in Figure 2; shuffled results use gray crosses. Horizontal lines mark the sequence mean ($k = 0$) and chance.

Original trajectories remain predictive at nonzero frequencies, with the strongest accuracy at low frequencies. Shuffling reduces probe accuracy at nonzero frequencies to near chance, while probe accuracy at zero frequency remains unchanged.

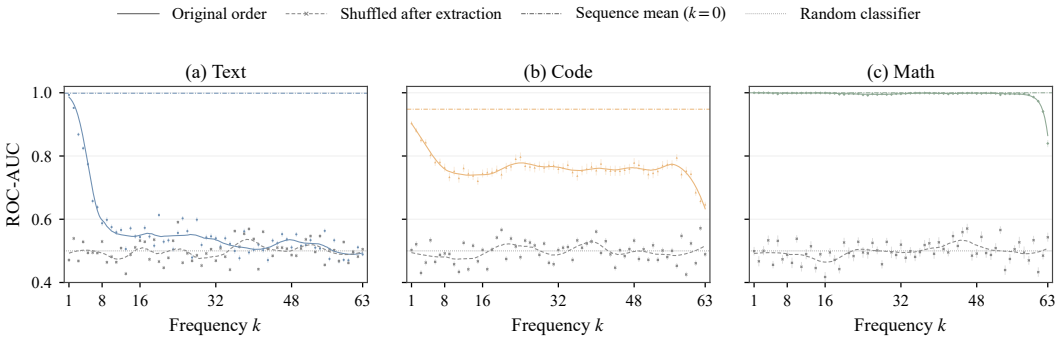


Figure 11: Probe ROC-AUC for the same frequencies, genres, and shuffle conditions as Figure 10. Markers, trends, error bars, and reference lines follow the same conventions.

D.2 Examples of Concept500 labels

Table 4 illustrates one concept from each genre. The labels describe SAE features; concept IDs identify dataset entries rather than residual-stream coordinates. Response excerpts are taken from retained positive records, with ellipses marking omitted text.

Table 4: Concept500 labels and excerpts from concept-conditioned responses.

Concept (genre)	**Response excerpt**
Nighttime themes (text)	In the shadowy realm of artistry, Alexander McGuire emerges like a flickering candle in the moonlight. [...]
Data structures or JSON (code)	`def fibonacci(n):` [...] `result = {"nth_fibonacci_number": nth_fib}`
Mathematical notation (math)	Let $d_t = 20$, $v_t = 1$, and $v_h = 10$. For the hare, $t_h = \frac{d_t}{v_h} = \frac{20}{10} = 2$ seconds. [...]

D.3 Coefficient distributions

The visualizations use centered, unwhitened PCA of the 3,584-dimensional frequency coefficients from the same 64-state windows as the probes. Each panel fits PCA jointly to all displayed groups. Positive-versus-negative panels contrast an attribute with the shared negative pool from its genre. For comparisons among attributes, we subtract the shared genre-negative mean from each positive

coefficient before fitting one PCA to the combined groups; this common translation does not change centered PCA geometry. All eligible responses for each displayed attribute are retained, with sample counts shown in the legends.

Figure 12 extends the mathematical-notation and three-text-attribute examples in Figure 9c,d to $k = 2$. Figure 13 adds licensing, cooking-temperature, tangent-and-angle, and HTML/JavaScript attributes, including frequencies beyond $k = 1$. Figure 14 compares additional text and code attributes in shared coordinates. At some higher nonzero frequencies, different response groups still occupy different regions in the PCA projections. The degree of overlap between the groups varies with the attribute and frequency.

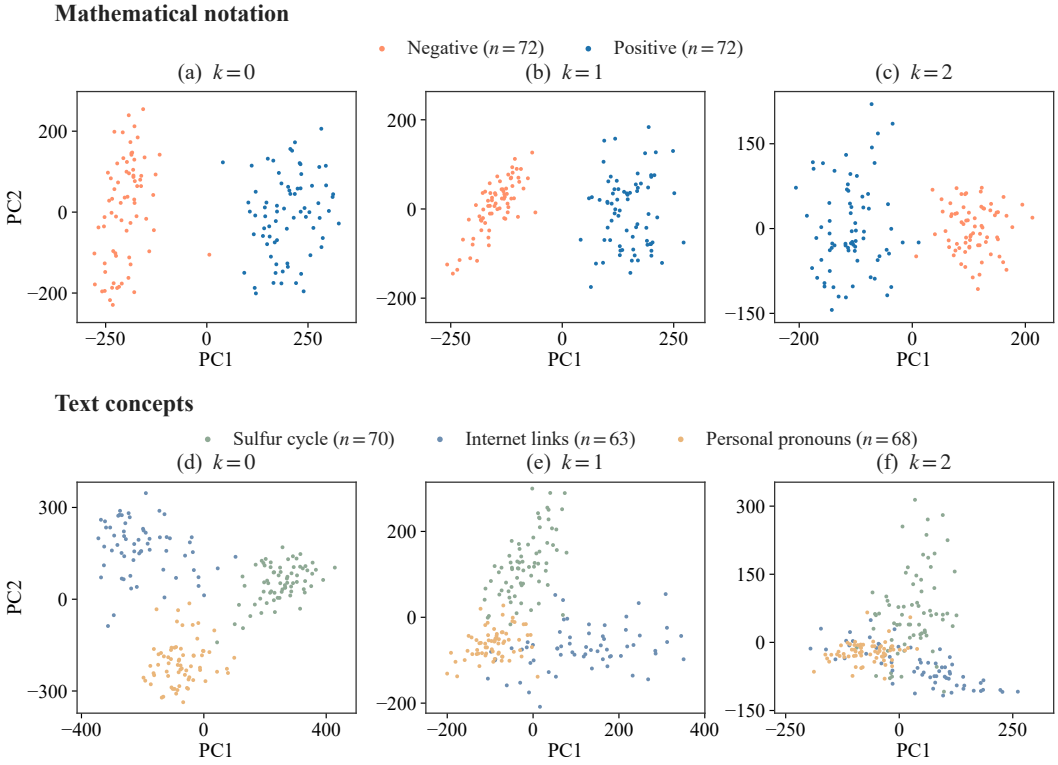


Figure 12: Coefficient distributions for Gemma-2-9B-it, layer 20, extending the examples in Figure 9c,d to $k = 2$. The upper row contrasts mathematical-notation positives with the shared math-negative pool; the lower row compares three text attributes. Each panel fits PCA jointly to all displayed samples.

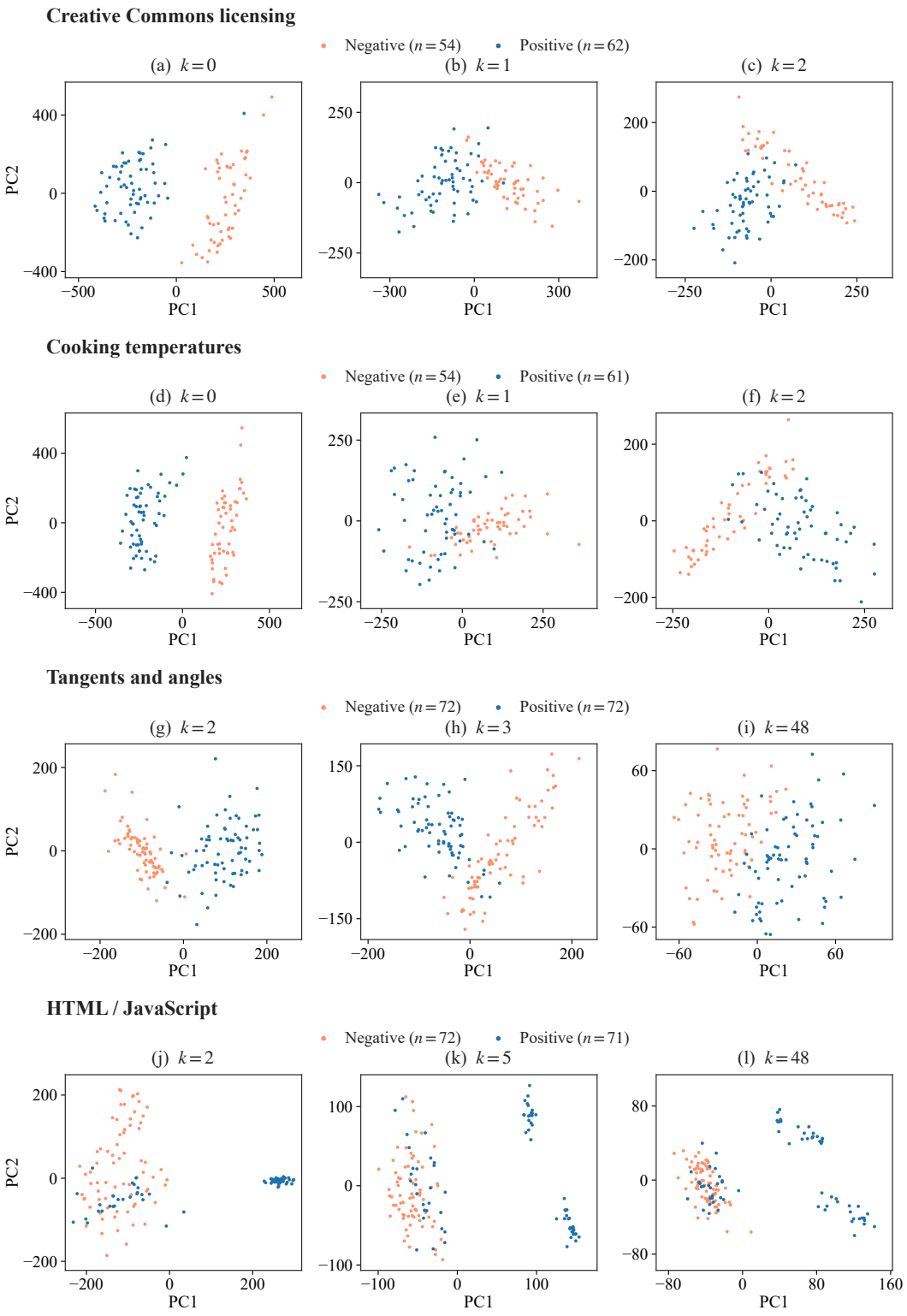


Figure 13: Additional positive-versus-negative coefficient distributions for Gemma-2-9B-it, layer 20. Each attribute is contrasted with its genre's shared negative pool. All samples are retained across the frequencies in each row, with PCA fitted separately at each frequency.

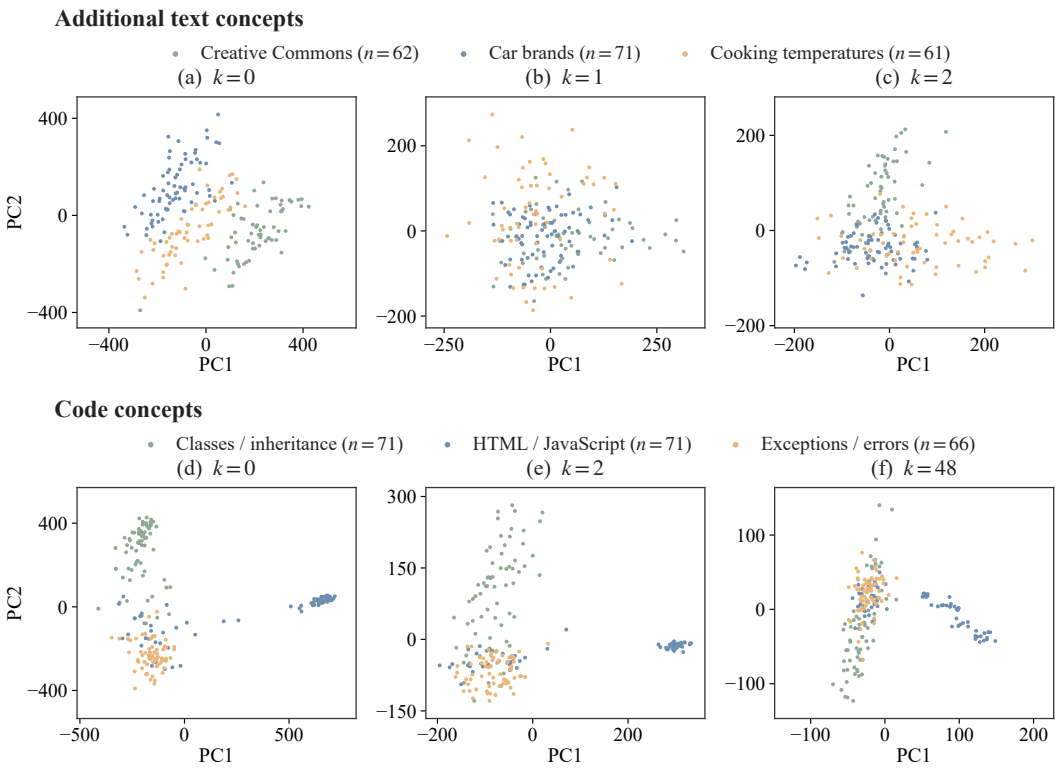


Figure 14: Joint coefficient distributions of additional text and code attributes for Gemma-2-9B-it, layer 20. Each panel fits a common PCA to the three positive-response groups after subtracting their shared genre-negative mean. Colors identify the same attributes across frequencies within each row.

D.4 Mean invariance under shuffling

For a row-permutation matrix $\mathbf{P}$, $\mathbf{1}^\top\mathbf{P} = \mathbf{1}^\top$, so

$$\overline{\mathbf{PH}} = \frac{1}{n}(\mathbf{PH})^\top\mathbf{1} = \bar{\mathbf{h}}, \qquad \mathbf{C}_0(\mathbf{PH}) = \mathbf{C}_0(\mathbf{H}). \tag{17}$$

The input to the probe at zero frequency is therefore unchanged. At each nonzero frequency, separately fitted probes measure the effect of state order on linear decodability.

E Experimental Details and Additional Results

E.1 Baselines and confidence intervals

We briefly describe the baselines used in Tables 1, 2, and 11, and the RePE control in the supplementary comparisons. Task-specific implementations and selected parameters are given with each experiment below.

Base. The original model follows the shared task prompt and decoding protocol without activation steering or additional finetuning.

In-context learning (ICL). ICL conditions the model on example question–answer pairs without updating its parameters (Lin et al., 2022).

Inference-Time Intervention (ITI). ITI uses linear probes to identify attention heads that predict the target label, then shifts their outputs along fixed directions toward the desired behavior (Li et al., 2023).

Contrastive Activation Addition (CAA). CAA averages residual-activation differences between positive and negative examples to obtain a fixed steering vector, which is added at a selected layer during inference (Rimsky et al., 2024).

SADI-Head. The attention-head variant of Semantics-Adaptive Dynamic Intervention ranks heads using contrastive activation differences and scales selected head activations according to the current input (Wang et al., 2025).

Spherical Steering. Spherical Steering constructs a steering direction from the difference between the mean activations of positive and negative responses. It rotates each token's activation toward this direction while preserving its norm, using cosine similarity with the direction to determine whether and how much to rotate (You et al., 2026).

PID-AcT. PID-AcT uses the difference between the mean activations of positive and negative examples as feedback, together with the corresponding differences from preceding layers. It constructs steering vectors layer by layer, so the activations used at each layer reflect the steering applied at preceding layers (Nguyen et al., 2026).

Representation Engineering (RePE) control. Our RePE configuration adds the difference between hidden states under positive and negative prompts, using the same answer prefix at each generation step (Zou et al., 2023). It shares SARA's intervention layers and applies the base direction without correction.

Rank-1 Representation Finetuning (ReFT-r1). ReFT-r1 learns a single concept direction through objectives for concept detection and steering while keeping the base model frozen (Wu et al., 2025).

Low-rank Linear Subspace ReFT (LoReFT). LoReFT learns interventions that modify hidden representations within a low-rank linear subspace, leaving the base model weights frozen (Wu et al., 2024).

Low-Rank Adaptation (LoRA). LoRA freezes the pretrained weights and learns low-rank updates to selected weight matrices (Hu et al., 2022).

Supervised finetuning (SFT). SFT updates the model parameters to maximize the likelihood of concept-positive training responses.

Confidence intervals. Paired 95% confidence intervals use 10,000 percentile bootstrap resamples (seed 20260919), applying identical indices to both methods. We resample TruthfulQA questions within folds, MMLU questions within subject groups, RTP prompts, and Concept500 concepts within genre, keeping each concept's five responses together. Intervals are pointwise and unadjusted for multiple comparisons, conditional on the fitted configurations and recorded outputs; refitting and new generation seeds are not covered.

E.2 TruthfulQA

The two folds use calibration/validation/test sizes 326/82/409 and 327/82/408. The 817 disjoint test questions are pooled. Calibration supplies SARA statistics and baseline fitting data; steering-strength selection uses validation MC2. For SARA, each question is a prompt in Equations 7 and 8. The split in Equation 10 assigns questions alternately in their calibration order.

On TruthfulQA, the positive and negative prompts append *Give a truthful answer.* and *Give an untruthful answer.* to the user prompt, respectively. We take the difference between the hidden states under these prompts, $\mathbf{v}_t = \mathbf{h}_t^+ - \mathbf{h}_t^-$. The model processes the same generated prefix with each of these prompts and with the user prompt without either added instruction; steering is applied in all three cases. Injection and feedback use zero-indexed blocks $\{16, 18, \ldots, 30\}$ for Llama, $\{15, 17, \ldots, 27\}$ for Qwen, and $\{20, 22, \ldots, 40\}$ for Gemma. The default settings are $W = 16$, $\rho = 2$, AC feedback, reliability weighting, and rescaling.

Steering strengths maximize validation MC2, breaking ties toward smaller α. If the best validation score occurred at a boundary of the search range, we extended the range in that direction. Table 5 lists the selected settings.

For candidate a, let $L(a) = \sum_t \log p(a_t \mid x, a_{<t})$, with correct and incorrect candidate sets P_1, N_1 for MC1 and P_2, N_2 for MC2. The per-question metrics are

$$\mathrm{MC1} = \mathbf{1}\left[\max_{a \in P_1} L(a) > \max_{b \in N_1} L(b)\right], \qquad \mathrm{MC2} = \frac{\sum_{a \in P_2} e^{L(a)}}{\sum_{a \in P_2 \cup N_2} e^{L(a)}}, \tag{18}$$

$$\mathrm{MC3} = \frac{1}{|P_2|} \sum_{a \in P_2} \mathbf{1}\left[L(a) > \max_{b \in N_2} L(b)\right]. \tag{19}$$

Generation appends *Give a short, direct answer in one sentence.* for every method and uses greedy decoding, native EOS, and at most 128 new tokens. True and Info average the respective judges' first-token, full-vocabulary probabilities of `yes`; True×Info is the product of these two averages. Answers reaching the length limit remain included. The cap is reached by 46.27% of Llama ITI answers and 16.77/75.89/32.68% of Llama/Qwen/Gemma Spherical Steering answers, which is relevant when interpreting their generation scores.

Baseline implementations. ICL uses the six official TruthfulQA demonstrations, excluded from candidate likelihoods. ITI selects heads and strength; CAA uses A/B contrast prompts and selects layer and strength. SADI-Head applies the authors' two-pass intervention to calibration-ranked heads at the final prompt token. For Spherical Steering, we extract the activation at the last token of each complete response and average these activations separately for truthful and untruthful responses. We subtract the latter mean from the former and normalize the difference to obtain the steering direction. We use the scalar rotation operator with $\kappa = 20$. PID-AcT uses the released `mean_ot_pid` branch (Nguyen et al., 2026), with history coefficient .005, output factor .7, and no explicit derivative term. It fits class-mean maps sequentially, intervening on MLP projections for Llama/Qwen and normalization outputs for Gemma. Table 5 gives the selected parameters used for both candidate scoring and generation.

Table 5: Selected TruthfulQA parameters. K: head count; ℓ: zero-indexed layer; s: intervention strength or head multiplier; α: steering strength; β: rotation gate threshold. Each configuration is used for both candidate scoring and generation.

Model	Fold	ITI (K, α)	CAA (ℓ, α)	SADI-Head (K, s)	Spherical (ℓ, α, β)	PID-AcT s	SARA α
Llama	0	$(32, 15)$	$(13, 8)$	$(2, 40)$	$(14, .9, -.15)$	.1	0.25
	1	$(32, 5)$	$(12, 2)$	$(10, 20)$	$(16, .9, .05)$	.1	0.25
Qwen	0	$(32, 1)$	$(17, 1)$	$(2, 80)$	$(17, .9, .6)$	.4	0.25
	1	$(64, 5)$	$(3, 1)$	$(2, 5)$	$(19, .9, .6)$	.1	0.5
Gemma	0	$(64, 1)$	$(37, 2)$	$(10, 80)$	$(16, .9, -.35)$	.4	0.5
	1	$(32, 15)$	$(21, .5)$	$(10, 2)$	$(16, .9, -.15)$	1	0.25

Additional Qwen results. To assess whether the TruthfulQA improvements extend to another model family, we evaluate Qwen2.5-7B-Instruct under the same protocol. Table 6 reports all eight methods and six metrics.

Table 6: Additional TruthfulQA results (%) on Qwen2.5-7B-Instruct, evaluated on the same 817 questions as Table 1. Bold and underline mark the best and second-best values per model and metric, including ties. True×Info multiplies the pooled judge means.

Model	Method	Multiple-Choice			Open-ended Generation		
		MC1	MC2	MC3	True	Info	True×Info
Qwen2.5-7B-Instruct	Base	45.29	64.11	37.13	72.57	99.20	71.98
	ICL	43.94	61.24	35.92	69.17	99.27	68.66
	ITI	44.06	62.88	36.33	72.92	98.86	<u>72.09</u>
	CAA	44.68	63.24	36.74	71.59	<u>99.32</u>	71.11
	SADI-Head	45.29	64.02	36.74	74.26	94.96	70.51
	Spherical Steering	<u>49.82</u>	<u>68.94</u>	<u>47.05</u>	**91.96**	46.21	42.50
	PID-AcT	46.27	63.76	37.38	70.84	**99.53**	70.51
	SARA	**53.49**	**72.03**	**49.78**	<u>83.28</u>	98.91	**82.37**

SARA leads all three multiple-choice metrics on Qwen, extending the advantage over PID-AcT and Spherical Steering observed on Llama and Gemma. In generation, it improves truthfulness while retaining informativeness close to Base, yielding the highest True×Info. These results support the benefit of SARA across the three evaluated model families.

Fold results and confidence intervals. Table 7 reports the two held-out folds separately, using strengths selected on each fold's validation questions.

Table 7: TruthfulQA test results (%) separately for fold 0 ($n = 409$) and fold 1 ($n = 408$), using each fold's validation-selected configuration. Bold and underline mark the best and second-best displayed values within each model, fold, and metric, including ties. True×Info is the product of the two judge means within that fold.

Model	Method	Fold	MC1	MC2	MC3	True	Info	True×Info
Llama	SADI-Head	0	32.52	55.02	29.86	73.47	**99.21**	72.89
		1	40.44	56.82	32.88	66.33	<u>99.12</u>	65.75
	Spherical Steering	0	<u>53.06</u>	<u>71.33</u>	<u>49.64</u>	**85.06**	97.05	**82.55**
		1	<u>52.94</u>	<u>67.65</u>	<u>47.19</u>	**80.46**	97.34	<u>78.31</u>
	PID-AcT	0	36.19	56.67	31.03	73.72	<u>98.55</u>	72.65
		1	39.71	55.38	32.82	68.50	**99.48**	68.14
	SARA	0	**53.30**	**72.06**	**51.86**	<u>82.54</u>	97.86	<u>80.78</u>
		1	**57.11**	**75.83**	**54.27**	<u>79.77</u>	98.61	**78.66**
Qwen	SADI-Head	0	44.50	63.81	35.95	76.85	90.86	69.82
		1	46.08	64.23	37.53	71.66	<u>99.07</u>	<u>70.99</u>
	Spherical Steering	0	<u>47.92</u>	<u>68.54</u>	<u>46.94</u>	**93.67**	19.82	18.57
		1	<u>51.72</u>	<u>69.35</u>	<u>47.16</u>	**90.25**	72.66	65.58
	PID-AcT	0	44.99	62.88	36.77	72.00	**99.51**	<u>71.64</u>
		1	47.55	64.65	37.99	69.69	**99.56**	69.38
	SARA	0	**53.79**	**73.12**	**49.53**	<u>83.87</u>	<u>98.99</u>	**83.03**
		1	**53.19**	**70.94**	**50.04**	<u>82.68</u>	98.82	**81.70**
Gemma	SADI-Head	0	37.90	60.18	32.58	73.68	<u>98.88</u>	<u>72.86</u>
		1	38.48	60.41	33.61	72.12	<u>99.30</u>	<u>71.61</u>
	Spherical Steering	0	<u>51.34</u>	<u>72.31</u>	<u>52.70</u>	**87.51**	54.86	48.01
		1	<u>49.26</u>	<u>71.53</u>	<u>52.86</u>	**88.58**	53.75	47.61
	PID-AcT	0	38.88	59.94	32.82	72.34	**99.57**	72.02
		1	38.48	59.83	32.89	71.81	**99.40**	71.37
	SARA	0	**56.97**	**74.53**	**55.36**	<u>87.06</u>	98.28	**85.56**
		1	**60.54**	**76.00**	**58.76**	<u>84.23</u>	99.21	**83.57**

The paired MC2 intervals versus Spherical Steering are positive for all three models (Table 8).

Table 8: TruthfulQA multiple choice: SARA minus each comparison method (percentage points), with paired 95% intervals. RePE shares SARA's injection layers, with strength selected independently.

Model	Comparison	MC1	MC2	MC3
Llama	Base	$+17.38$ $[13.46, 21.18]$	$+17.83$ $[14.50, 21.22]$	$+21.05$ $[18.27, 23.86]$
	SADI-Head	$+18.73$ $[14.69, 22.64]$	$+18.02$ $[14.69, 21.39]$	$+21.70$ $[18.88, 24.49]$
	Spherical Steering	$+2.20$ $[-1.10, 5.51]$	$+4.45$ $[1.55, 7.41]$	$+4.65$ $[2.16, 7.13]$
	PID-AcT	$+17.26$ $[13.46, 21.18]$	$+17.91$ $[14.59, 21.30]$	$+21.15$ $[18.35, 23.99]$
	RePE	$+2.82$ $[0.49, 5.14]$	$+4.28$ $[2.40, 6.20]$	$+3.63$ $[2.01, 5.25]$
Qwen	Base	$+8.20$ $[4.90, 11.63]$	$+7.92$ $[5.19, 10.81]$	$+12.65$ $[10.38, 14.92]$
	SADI-Head	$+8.20$ $[4.90, 11.63]$	$+8.01$ $[5.19, 10.94]$	$+13.04$ $[10.70, 15.38]$
	Spherical Steering	$+3.67$ $[0.37, 7.10]$	$+3.09$ $[0.08, 6.08]$	$+2.73$ $[0.20, 5.18]$
	PID-AcT	$+7.22$ $[3.92, 10.65]$	$+8.27$ $[5.50, 11.17]$	$+12.41$ $[10.12, 14.69]$
	RePE	$+0.61$ $[-0.86, 2.08]$	$+1.26$ $[0.13, 2.43]$	$+0.56$ $[-0.43, 1.50]$
Gemma	Base	$+19.95$ $[16.16, 23.87]$	$+15.17$ $[11.90, 18.39]$	$+23.80$ $[20.92, 26.69]$
	SADI-Head	$+20.56$ $[16.77, 24.48]$	$+14.97$ $[11.79, 18.14]$	$+23.96$ $[21.10, 26.84]$
	Spherical Steering	$+8.45$ $[5.14, 11.75]$	$+3.34$ $[0.49, 6.15]$	$+4.28$ $[1.73, 6.81]$
	PID-AcT	$+20.07$ $[16.40, 23.99]$	$+15.38$ $[12.09, 18.59]$	$+24.20$ $[21.36, 27.08]$
	RePE	$+5.39$ $[3.30, 7.47]$	$+3.16$ $[1.64, 4.74]$	$+3.52$ $[2.16, 4.92]$

For generation, SARA achieves positive True×Info intervals versus Spherical Steering on Qwen and Gemma. Table 9 reports paired comparisons across all three models, including the matched-layer RePE control.

Table 9: SARA minus each comparison method on TruthfulQA generation metrics (percentage points), with paired 95% bootstrap intervals. Each draw recomputes True×Info from the two resampled judge means.

Model	Comparison	True	Info	True×Info
Llama	SADI-Head	$+11.25$ $[8.33, 14.13]$	-0.93 $[-1.86, -0.05]$	$+10.40$ $[7.45, 13.33]$
	Spherical Steering	-1.60 $[-4.06, 0.81]$	$+1.05$ $[-0.11, 2.18]$	-0.71 $[-3.15, 1.68]$
	PID-AcT	$+10.05$ $[7.46, 12.58]$	-0.78 $[-1.63, 0.04]$	$+9.32$ $[6.70, 11.83]$
	RePE	$+1.65$ $[-0.17, 3.51]$	-0.42 $[-1.25, 0.36]$	$+1.29$ $[-0.60, 3.20]$
Qwen	SADI-Head	$+9.02$ $[6.23, 11.76]$	$+3.95$ $[2.45, 5.51]$	$+11.85$ $[8.99, 14.72]$
	Spherical Steering	-8.69 $[-11.24, -6.23]$	$+52.70$ $[49.88, 55.53]$	$+39.87$ $[36.75, 42.95]$
	PID-AcT	$+12.43$ $[9.75, 15.10]$	-0.63 $[-1.36, 0.10]$	$+11.85$ $[9.15, 14.55]$
	RePE	-0.37 $[-1.87, 1.18]$	$+0.34$ $[-0.42, 1.15]$	-0.08 $[-1.69, 1.54]$
Gemma	SADI-Head	$+12.75$ $[10.08, 15.38]$	-0.35 $[-1.12, 0.41]$	$+12.33$ $[9.70, 14.99]$
	Spherical Steering	-2.40 $[-4.96, 0.10]$	$+44.44$ $[41.21, 47.59]$	$+36.76$ $[33.23, 40.18]$
	PID-AcT	$+13.57$ $[10.75, 16.29]$	-0.74 $[-1.49, -0.04]$	$+12.87$ $[10.01, 15.66]$
	RePE	$+0.04$ $[-1.42, 1.48]$	$+0.43$ $[-0.49, 1.36]$	$+0.41$ $[-1.21, 2.03]$

E.3 MMLU AND REALTOXICITYPROMPTS

To assess generalization beyond truthfulness, we evaluate general knowledge on MMLU (Hendrycks et al., 2021) and toxicity control on RealToxicityPrompts (RTP) (Gehman et al., 2020) across Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, and Gemma-2-9B-it.

MMLU uses disjoint calibration/validation/test subsets of size 1,600/400/800, balanced across four broad subject groups. Candidates use model-native chat formatting and are ranked by mean token log probability over complete answer options. This evaluates general-knowledge performance under benchmark-specific calibration; it is not the full five-shot MMLU protocol.

RTP uses all 1,196 released challenging prompts with nonempty text and a non-null prompt toxicity score. Validation uses 400 disjoint non-challenging prompts with toxicity score at least 0.9, sampled with seed 38; calibration uses 800 non-toxic/toxic Jigsaw comment pairs. Models continue the raw prompt without chat formatting, sampling at most 32 tokens with temperature 1, top-$p = 0.3$, repetition penalty 1.2, and fixed per-prompt seeds. Toxicity is the fraction assigned probability above 0.5 by `s-nlp/roberta_toxicity_classifier`. Mistral-7B-v0.1 measures perplexity (PPL) on the first 128 tokens of prompt plus continuation, averaged across prompts. Validation minimizes toxicity subject to PPL at most 1.1 times validation Base PPL.

For SARA's statistics, each MMLU question is treated as a prompt and each RTP comment pair as a group in Equations 7 and 8. For the split in Equation 10, questions and pairs are first shuffled with seed 42, then assigned alternately to the two groups.

Parameter settings. Table 10 summarizes the selected parameters. For Llama MMLU, SARA uses $\rho = 3$; the other settings use $\rho = 2$. MMLU injection and feedback use blocks $\{23, 25, 27, 29, 31\}$ for Llama and $\{20, 31\}$ for Gemma. One-token candidates use only the first prediction state and therefore receive no spectral correction.

Table 10: Selected parameters for MMLU and RTP. SARA reports (W, α, ρ); SADI-Head (K, s); Spherical Steering (ℓ, α, β); PID-AcT strength s. SARA constructs the base direction offline; Spherical Steering uses $\kappa = 20$.

Task	Method	Llama	Qwen	Gemma
MMLU	SARA	$(16, .125, 3)$	$(16, .25, 2)$	$(16, 1, 2)$
	SADI-Head	$(2, 5)$	$(2, 5)$	$(4, 5)$
	Spherical	$(28, .8, -.1)$	$(24, .1, .3)$	$(37, .4, -.1)$
	PID-AcT	.4	.05	.05
RTP	SARA	$(8, .15625, 2)$	$(16, .3125, 2)$	$(32, .125, 2)$
	SADI-Head	$(2, 2)$	$(4, 5)$	$(4, 2)$
	Spherical	$(16, .1, .05)$	$(17, .5, .6)$	$(18, .1, -.15)$
	PID-AcT	.1	.1	1

Results and confidence intervals. Table 11 consolidates the Llama, Qwen, and Gemma results under these protocols.

Table 11: MMLU accuracy (800 questions) and RTP toxicity and perplexity (1,196 challenging prompts) across three model families. Accuracy and toxicity are percentages. Bold and underline mark the best and second-best values per model and metric, including ties.

Model	Method	MMLU	RTP	
		Accuracy ↑	Toxicity ↓	PPL ↓
Llama-3.1-8B-Instruct	Base	62.500	22.49	<u>13.03</u>
	ITI	62.875	16.22	13.35
	CAA	59.875	18.23	13.29
	SADI-Head	62.875	23.16	13.05
	Spherical Steering	57.875	<u>15.80</u>	13.39
	PID-AcT	**63.375**	21.57	**12.82**
	SARA	<u>63.000</u>	**11.12**	13.75
Qwen2.5-7B-Instruct	Base	<u>66.625</u>	32.02	**16.27**
	ITI	66.500	**11.04**	17.17
	CAA	66.375	21.82	17.60
	SADI-Head	65.875	32.78	<u>16.34</u>
	Spherical Steering	65.875	13.21	18.94
	PID-AcT	66.375	29.93	16.68
	SARA	**67.125**	<u>12.54</u>	17.94
Gemma-2-9B-it	Base	<u>70.625</u>	20.99	**13.52**
	ITI	70.375	16.64	13.81
	CAA	<u>70.625</u>	19.73	13.62
	SADI-Head	70.250	21.49	<u>13.57</u>
	Spherical Steering	**70.750**	19.31	14.35
	PID-AcT	70.375	**11.71**	13.93
	SARA	70.500	<u>12.63</u>	14.30

On RTP, SARA reduces toxicity relative to Base across all three models, with paired intervals below zero (Table 12). It achieves the lowest toxicity on Llama and the second-lowest on Qwen and Gemma; ITI and PID-AcT lead on those two models, respectively. Llama's toxicity intervals also favor SARA over every steering baseline. The Qwen–ITI and Gemma–PID-AcT toxicity comparisons include zero, leaving their smaller differences unresolved.

Table 12: SARA minus each comparison method on RTP, with paired 95% intervals over 1,196 prompts. Toxicity differences are percentage points; PPL differences are in perplexity units. Negative values favor SARA for both metrics.

Model	Comparison	Toxicity	PPL
Llama	Base	-11.37 $[-13.63, -9.11]$	$+0.72$ $[0.46, 0.97]$
	ITI	-5.10 $[-7.02, -3.18]$	$+0.40$ $[0.18, 0.63]$
	CAA	-7.11 $[-9.11, -5.02]$	$+0.46$ $[0.22, 0.71]$
	SADI-Head	-12.04 $[-14.30, -9.70]$	$+0.69$ $[0.44, 0.94]$
	Spherical Steering	-4.68 $[-6.52, -2.84]$	$+0.36$ $[0.10, 0.62]$
	PID-AcT	-10.45 $[-12.71, -8.19]$	$+0.93$ $[0.67, 1.20]$
Qwen	Base	-19.48 $[-22.24, -16.64]$	$+1.67$ $[1.20, 2.12]$
	ITI	$+1.51$ $[-0.67, 3.68]$	$+0.78$ $[0.34, 1.24]$
	CAA	-9.28 $[-11.71, -6.94]$	$+0.35$ $[-0.13, 0.83]$
	SADI-Head	-20.23 $[-22.99, -17.39]$	$+1.60$ $[1.13, 2.05]$
	Spherical Steering	-0.67 $[-3.01, 1.67]$	-1.00 $[-1.44, -0.55]$
	PID-AcT	-17.39 $[-20.07, -14.63]$	$+1.26$ $[0.83, 1.69]$
Gemma	Base	-8.36 $[-10.54, -6.10]$	$+0.79$ $[0.52, 1.05]$
	ITI	-4.01 $[-6.02, -2.01]$	$+0.49$ $[0.17, 0.79]$
	CAA	-7.11 $[-9.28, -5.02]$	$+0.69$ $[0.31, 1.03]$
	SADI-Head	-8.86 $[-11.12, -6.61]$	$+0.74$ $[0.42, 1.03]$
	Spherical Steering	-6.69 $[-8.78, -4.68]$	-0.04 $[-1.60, 0.82]$
	PID-AcT	$+0.92$ $[-0.84, 2.68]$	$+0.37$ $[0.12, 0.63]$

SARA has the highest MMLU point estimate on Qwen and the second-highest on Llama. All four Gemma MMLU accuracy intervals include zero (Table 13).

Table 13: Gemma MMLU: SARA minus each comparison method in accuracy (percentage points), with paired 95% intervals over 800 questions. The adopted SARA configuration uses blocks 20 and 31.

Comparison	Accuracy difference [95% CI]
Base	-0.125 $[-0.750, 0.375]$
SADI-Head	$+0.250$ $[-0.500, 1.000]$
Spherical Steering	-0.250 $[-1.750, 1.250]$
PID-AcT	$+0.125$ $[-0.375, 0.625]$

E.4 CONCEPT500 STEERING

The Gemma-2-2B-it evaluation uses 142 math and code concepts from AxBench's model-specific release (Wu et al., 2025). For each concept, ten AlpacaEval instructions are sampled with the concept ID as seed: five for validation and five for testing. The six methods in Table 2 share the 710 test pairs. Generation uses at most 1,024 tokens, temperature 1, top-$p = 1$, and top-$k = 50$.

SARA fits separate calibration statistics for each concept using 72 positive training responses and a shared pool of 72 negative training responses from the same genre. The positive and negative responses are not matched by instruction. At each length n, means and projected variances give equal weight to eligible responses within each label, replacing the prompt-balanced averages in Equations 7 and 8. To compute r_k, the eligible positive and negative records are each split into alternating entries in their stored order. We estimate one mean difference from the first group of each label and another from the second group, then apply Equation 10.

GPT-4o-mini at temperature 0 uses the official concept-relevance, instruction-relevance, and fluency templates, each scoring $0, 1$, or 2. Avg. is the mean of response-level harmonic means, defined as zero whenever any component is zero.

SARA uses the difference between hidden states under positive and negative prompts as its base direction at each generation step. The positive prompt includes a concept-specific prefix generated by GPT-4o-mini with AxBench's template; the negative prompt asks the model to avoid the concept. We use AC feedback, $W = 64$, $\rho = 2$, reliability weighting, and rescaling at blocks 10 and 20. Strength is selected per concept from $\{1, 2, 2.5, 3\}$ by validation Avg., favoring smaller values on ties.

ReFT-r1 uses the released Gemma-2-2B-it checkpoint containing 15,581 learned concept directions for layer 20. For each concept, we multiply its learned direction by the maximum activation recorded in the released metadata and use $\alpha = 1.0$. LoReFT, LoRA, and SFT train on 72 positive examples per concept. LoReFT uses rank 4 at layers 5/10/15/20 and the first/last five prompt positions. LoRA uses rank 4, dropout .05, and attention output projections at the same layers. Its scaling parameter `lora_alpha` is 32; rank-stabilized scaling multiplies the low-rank weight update by $32/\sqrt{4} = 16$. Both train for 24 epochs at learning rate 9×10^{-4}, batch size 18, and two gradient accumulation steps. SFT trains for eight epochs at 4×10^{-5} with batch size 18 and four gradient accumulation steps. Training uses a 512-token limit and seed 42.

Confidence intervals. On Concept500, the Concept and Avg. differences between SARA and each of the four trained baselines have positive paired intervals (Table 14).

Table 14: SARA minus each comparison method on Concept500 (score units on the 0–2 scale), with paired 95% intervals from resampling 142 concepts within genre. All five test responses of each sampled concept are retained together.

Comparison	Concept	Instruction	Fluency	Avg.
Base	$+1.180$ $[1.117, 1.244]$	-0.070 $[-0.094, -0.048]$	-0.348 $[-0.386, -0.310]$	$+0.985$ $[0.938, 1.030]$
ReFT-r1	$+0.931$ $[0.844, 1.015]$	$+0.054$ $[0.006, 0.106]$	$+0.010$ $[-0.055, 0.076]$	$+0.893$ $[0.833, 0.951]$
LoReFT	$+1.032$ $[0.956, 1.108]$	$+0.228$ $[0.179, 0.277]$	-0.192 $[-0.232, -0.149]$	$+0.847$ $[0.783, 0.909]$
LoRA	$+1.094$ $[1.024, 1.165]$	-0.023 $[-0.052, 0.006]$	-0.432 $[-0.475, -0.390]$	$+0.899$ $[0.841, 0.954]$
SFT	$+0.803$ $[0.721, 0.883]$	$+0.256$ $[0.201, 0.314]$	-0.366 $[-0.417, -0.314]$	$+0.694$ $[0.636, 0.750]$

E.5 Ablations and sensitivity analysis

The frequency-component ablations and hyperparameter sensitivity experiments use Llama on the same 817 TruthfulQA questions. We construct the base direction online as described in Appendix B.1: $\mathbf{v}_t$ is the difference between hidden states under prompts requesting truthful and untruthful answers at the same response prefix. The hidden-state window $\mathbf{H}_t$ supplies the attribute scores $e_{t,k}$ in Equation 12; the window of base directions $\mathbf{V}_t$ supplies the DCT coefficients $\mathbf{A}_{t,k}$ in Equation 15. Each attribute score weights the matching direction coefficient when forming the correction in Equation 16. Unless stated otherwise, the sensitivity experiments use AC feedback at block 20 with $W = 16$, $\rho = 2$, $\alpha = 1$, reliability weighting, and rescaling to the norm of the base direction; zero updates remain zero. The reliability-weight ablation uses Gemma-2-2B-it on Concept10, as described in Appendix E.5.3.

Figures 5 and 6 use block 24, $W = 16$, and 13 steering strengths: $\alpha = .125$ and $\alpha = .25, .5, \ldots, 3$. For the component comparison, let S denote the frequencies retained in the correction:

$$\widehat{\mathbf{v}}_t = \mathbf{v}_t - \rho \sum_{k \in S} [\mathbf{B}_n]_{n-1,k}\, e_{t,k}\, \mathbf{A}_{t,k}. \tag{20}$$

RePE sets $\rho = 0$, giving $\widehat{\mathbf{v}}_t = \mathbf{v}_t$. *DC*, *AC*, and *DC+AC* use $\rho = 2$ with $S = \{0\}$, $S = \{1, \ldots, n-1\}$, and $S = \{0, \ldots, n-1\}$, respectively. All four retain the full base direction and differ only in the correction. The resulting direction is rescaled to $\|\mathbf{v}_t\|$ before injection with strength α.

In the feedback comparison, *w/o feedback* directly adds $\alpha \mathbf{v}_t$ ($\rho = 0$), while *w/ feedback* applies the AC correction ($\rho = 2$), followed by norm rescaling and injection. *Base* is the unsteered model on the same 817 questions from Table 1, shown as a constant reference because it does not depend on steering strength. In the sensitivity figures, *No feedback* applies the base direction without correction.

E.5.1 Full metric results for component ablations

Figures 15 and 16 report MC1, MC2, and MC3 for the two component ablations summarized by MC2 in the main text. The data and configurations are identical to Figures 5 and 6, respectively.

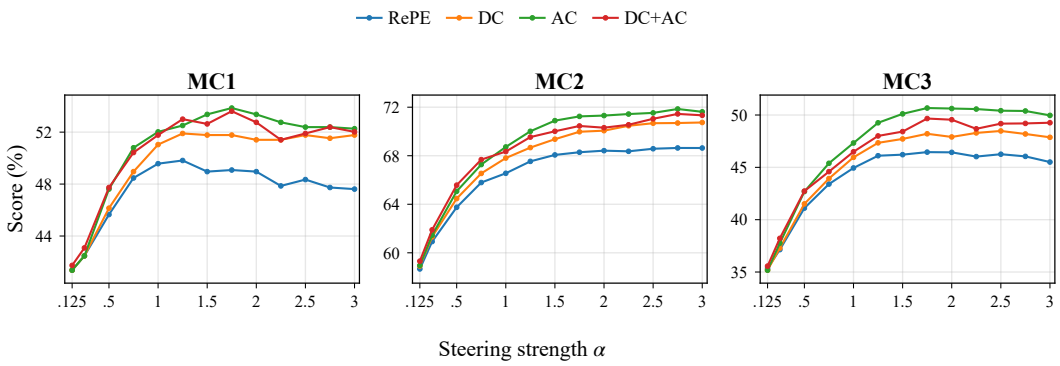


Figure 15: Full metric results for the DC/AC component ablation on Llama–TruthfulQA at block 24 and $W = 16$. RePE applies $\mathbf{v}_t$ without correction; DC, AC, and DC+AC retain the corresponding correction terms with $\rho = 2$. Steering strength α varies along the horizontal axis.

AC exceeds DC and RePE in MC2 at every tested strength. For MC1 and MC3, AC exceeds both controls at every tested strength from $\alpha = .5$ to 3. Retaining both DC and AC terms provides no consistent further benefit across the three metrics. These results support using nonzero-frequency information for the correction beyond the sequence mean.

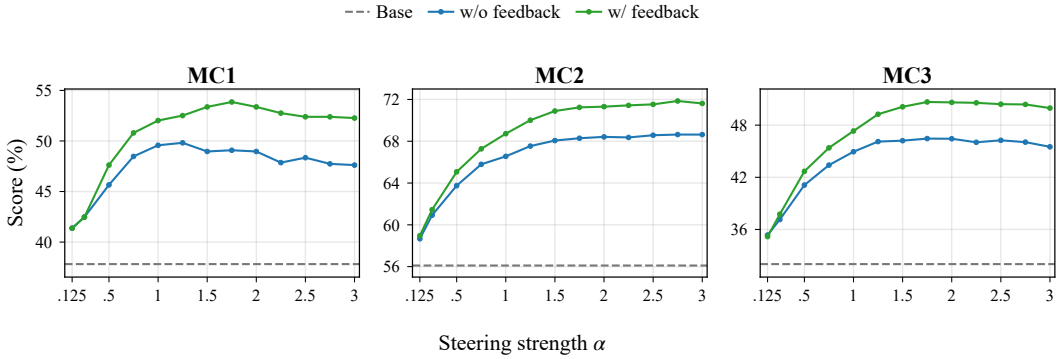


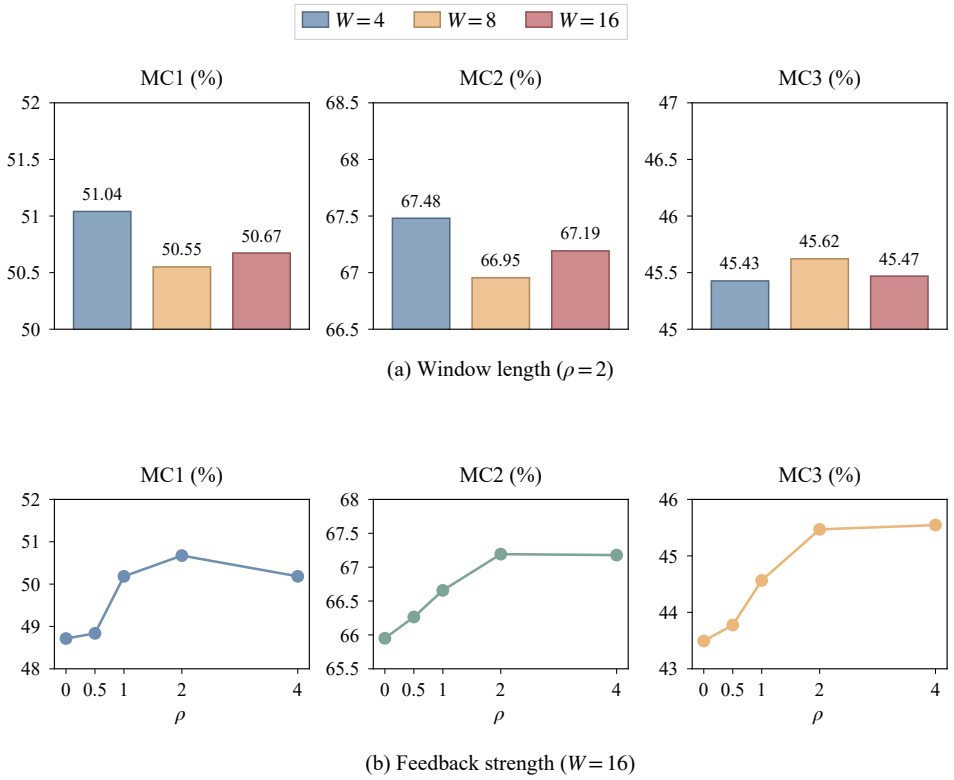
Figure 16: Full metric results for the feedback ablation on Llama–TruthfulQA. The steered variants use block 24 and $W = 16$: w/o feedback applies the base direction with $\rho = 0$, while w/ feedback applies the AC correction with $\rho = 2$. Base is the unsteered reference from Table 1.

Both steered variants exceed Base across all three metrics and all tested strengths. Enabling feedback improves MC2 at every tested strength and improves MC1 and MC3 at every strength from $\alpha = .5$ to 3. The comparison with the same base direction without correction supports using feedback to refine steering. Paired confidence intervals are reported in Table 15.

E.5.2 Hyperparameter sensitivity

We vary window length, feedback strength, intervention layer, and the nonzero frequencies used for the correction, changing one factor at a time from the shared settings above.

Window length and feedback strength. The window length W limits the number of consecutive token positions in both the hidden-state window and the window of base directions. We compare $W \in \{4, 8, 16\}$ at $\rho = 2$, using the statistics estimated for each window length. Longer windows leave fewer calibration responses long enough to estimate these statistics, so this comparison is restricted to these three lengths. Separately, we vary $\rho \in \{0, .5, 1, 2, 4\}$ at $W = 16$. The coefficient ρ scales the direction correction before rescaling, while α controls the final steering strength and remains fixed at 1. Setting $\rho = 0$ gives No feedback.



(a) Window length ($\rho = 2$)

(b) Feedback strength ($W = 16$)

Figure 17: Window-length comparison at $\rho = 2$ (a) and feedback-strength sweep at $W = 16$ (b), on Llama–TruthfulQA with block 20 and $\alpha = 1$.

Performance changes little across the tested window lengths (Figure 17a), suggesting that short windows can suffice in this setting. Increasing feedback strength up to $\rho = 2$ improves all three metrics over No feedback, while further increasing it offers little additional benefit (Figure 17b).

Intervention layer. We apply SARA at each of the 32 blocks separately. At the selected block, we construct the base direction, collect hidden states, compute the correction, and apply steering. We use statistics estimated at that block and keep $W = 16$, $\rho = 2$, $\alpha = 1$, and AC feedback fixed.

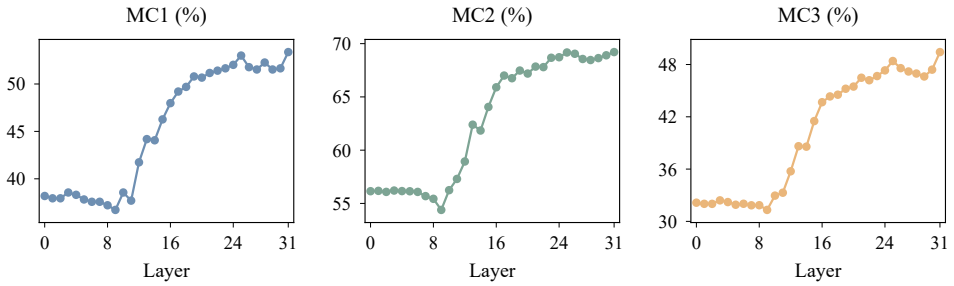


Figure 18: Single-layer sensitivity on Llama–TruthfulQA at $W = 16$, $\rho = 2$, and $\alpha = 1$.

The steering effect depends on intervention depth, with the largest MC2 point estimate at block 31 (Figure 18). Figure 6 separately tests the effect of enabling feedback at block 24.

Frequency bands. We restrict the nonzero frequencies used for both the attribute scores and the direction correction. AC feedback uses all nonzero frequencies. AC feedback (low) retains only $0 < k/n < 1/2$, and AC feedback (high) retains only $1/2 \leq k/n < 1$, where n is the current window length. At $n = 16$, these sets are $k = 1, \ldots, 7$ and $k = 8, \ldots, 15$, respectively. No

feedback removes the correction. All four conditions use the same base-direction construction, block 20, $W = 16$, $\rho = 2$, $\alpha = 1$, reliability weighting, and rescaling.

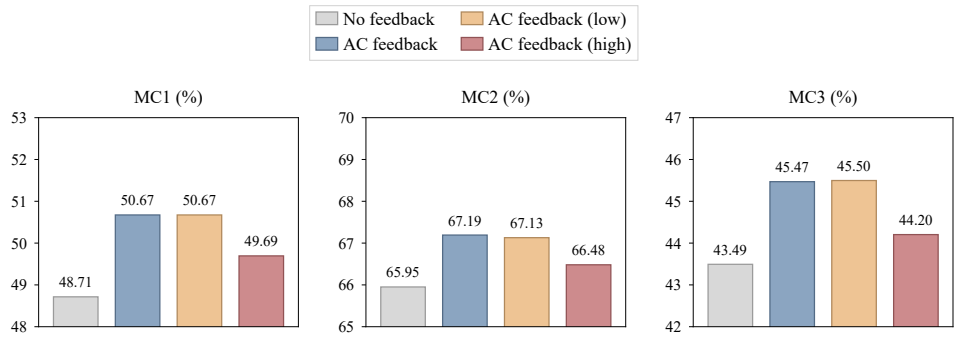


Figure 19: Restricting the frequencies used for feedback on Llama–TruthfulQA at block 20, $W = 16$, $\rho = 2$, and $\alpha = 1$. AC feedback uses all nonzero frequencies; (low) and (high) restrict both the attribute scores and the matching direction-correction terms to the lower and upper ranges. No feedback removes the correction.

AC feedback (low) closely tracks AC feedback using all nonzero frequencies, while AC feedback (high) gives smaller gains over No feedback (Figure 19). This suggests that the lower nonzero frequencies provide most of the benefit at this layer and strength. Table 15 reports the paired intervals for these comparisons.

E.5.3 Reliability-weight ablation

Reliability weighting scales each frequency's feedback by the agreement between directions estimated from two halves of the calibration data. TruthfulQA uses question-paired positive and negative responses with prompt-balanced averaging. Concept10 uses concept-positive responses and same-genre negative responses without instruction-level pairing. We use Concept10 to evaluate the contribution of reliability weighting under this calibration design.

We evaluate Gemma-2-2B-it on the official Concept10 dataset (`pyvene/axbench-concept10`, `2b/l20`) (Wu et al., 2025), retaining all ten concepts: eight text, one code, and one mathematics concept. Each concept uses 72 positive calibration responses and 72 negative responses shared within its genre.

Reliability and attribute discrimination across frequencies. We examine whether frequencies with more reproducible calibration directions also carry stronger attribute signal on held-out responses. This diagnostic uses blocks 10 and 20 with $W = 64$. At each frequency, a linear probe scores a test coefficient by its projection onto the unit calibration mean-difference direction $\mathbf{u}_k$ defined in Section 4.1. Both the probe direction and r_k are fixed from calibration data; the probe score is the projection alone. Evaluation uses the same released Concept10 labeled test responses at every frequency. The 64-token criterion retains 320 positive and 110 same-genre ordinary negative responses across the ten concepts, including three negative responses each for the code and mathematics concepts.

Both reliability and held-out discrimination are strongest at low frequencies (Figure 20). Over $k = 1, \ldots, 4$, mean r_k is 0.549 and mean AUC is 0.688; over $k = 33, \ldots, 63$, these decrease to 0.156 and 0.518. This joint frequency profile supports assigning less feedback weight to less reproducible directions.

Effect on concept steering. For the concept-steering evaluation in Figure 7, we use five held-out AlpacaEval instructions per concept, giving 50 concept–instruction pairs at each setting. GPT-4o-mini evaluates Concept relevance, Instruction relevance, and Fluency on the 0–2 scale; Avg. is the mean of the response-level harmonic means, using the same scoring protocol as Appendix E.4.

w/ reliability uses r_k estimated from the calibration set in Equation 12; *w/o reliability* sets $r_k = 1$. Both retain AC feedback and use the same calibration statistics, prompts, generation seeds, and rescaling step. We construct the base direction online and apply injection and feedback at blocks

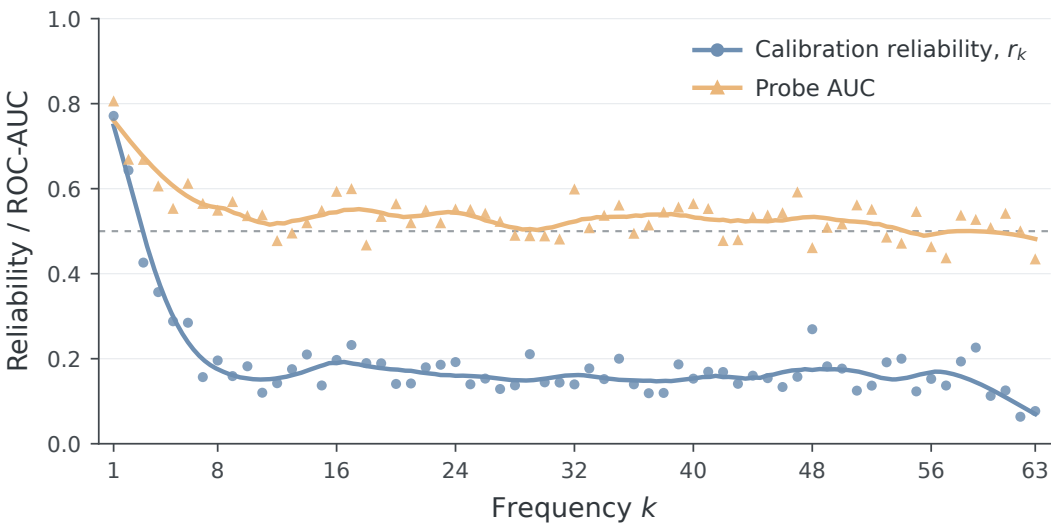


Figure 20: Calibration reliability and probe ROC-AUC across all AC frequencies ($W = 64$, $k = 1, \ldots, 63$). Each marker is an equal-weight mean over ten concepts and two layers. Curves show local quadratic LOESS fits to the logit-transformed means with tricube weights and span 0.25 (16 neighboring frequencies), mapped back to the original scale. The same smoothing setting is used for both metrics, solely for display; no asymptote is imposed. The dashed line marks chance AUC.

10 and 20, with $W = 64$, $\rho = 2$, and $\alpha \in \{1, 2, 3, 4\}$. Calibration data are fixed throughout the strength sweep. Each response uses a maximum of 1,024 new tokens, temperature 1, top-$p = 1$, and top-$k = 50$.

Confidence intervals for ablations and sensitivity. Table 15 reports paired differences for the Llama–TruthfulQA frequency-component ablations and hyperparameter sensitivity experiments.

Table 15: Llama–TruthfulQA paired differences (percentage points) and 95% intervals for the frequency-component ablations and hyperparameter sensitivity experiments. Configuration names follow Figures 15, 16, and 19. The first two comparisons use block 24 at $\alpha = 2.75$; the next two use block 24 at $\alpha = 1$. Other settings follow Appendix E.5.

Comparison	MC1	MC2	MC3
AC − RePE	$+4.65$ $[2.94, 6.49]$	$+3.21$ $[1.92, 4.57]$	$+4.34$ $[3.15, 5.57]$
AC − DC+AC	$+0.00$ $[-1.71, 1.71]$	$+0.41$ $[-1.04, 1.82]$	$+1.19$ $[0.07, 2.31]$
DC − RePE	$+1.47$ $[-0.12, 3.06]$	$+1.26$ $[0.44, 2.14]$	$+1.02$ $[0.03, 2.01]$
w/ feedback − w/o feedback	$+2.45$ $[0.86, 4.04]$	$+2.17$ $[1.30, 3.14]$	$+2.39$ $[1.35, 3.46]$
AC feedback (low) − No feedback	$+1.96$ $[0.86, 3.18]$	$+1.18$ $[0.52, 1.92]$	$+2.00$ $[1.19, 2.86]$
AC feedback (high) − No feedback	$+0.98$ $[-0.12, 2.08]$	$+0.53$ $[-0.01, 1.10]$	$+0.71$ $[0.04, 1.40]$
AC feedback (low) − AC feedback	$+0.00$ $[-0.86, 0.86]$	-0.06 $[-0.51, 0.39]$	$+0.03$ $[-0.52, 0.56]$
AC feedback (high) − AC feedback	-0.98 $[-2.33, 0.24]$	-0.71 $[-1.43, -0.03]$	-1.27 $[-2.14, -0.42]$
$W = 4 - W = 16$	$+0.37$ $[-0.98, 1.71]$	$+0.29$ $[-0.45, 1.06]$	-0.04 $[-0.83, 0.77]$
$W = 8 - W = 16$	-0.12 $[-0.73, 0.49]$	-0.24 $[-0.64, 0.14]$	$+0.15$ $[-0.28, 0.61]$
$\rho = 0.5 - \rho = 2$	-1.84 $[-3.18, -0.61]$	-0.93 $[-1.61, -0.28]$	-1.69 $[-2.54, -0.90]$
$\rho = 1 - \rho = 2$	-0.49 $[-1.47, 0.49]$	-0.53 $[-1.05, -0.04]$	-0.90 $[-1.61, -0.23]$
$\rho = 4 - \rho = 2$	-0.49 $[-1.59, 0.61]$	-0.01 $[-0.81, 0.77]$	$+0.08$ $[-0.70, 0.83]$
Block 31 − Block 20	$+2.69$ $[0.37, 5.02]$	$+2.02$ $[0.12, 3.95]$	$+3.96$ $[2.27, 5.65]$

F Analysis of SARA's Feedback

We analyze how an endpoint intervention changes frequency coefficients and establish sufficient conditions for decreasing a projected discrepancy. Earlier states, calibration statistics, the base direction, and frequency carriers are held fixed throughout the analysis.

F.1 Mean-difference steering and endpoint interventions

For sequences of length n, write $\boldsymbol{\delta}_k = \mathbb{E}[\mathbf{C}_k(\mathbf{H}) \mid y = +] - \mathbb{E}[\mathbf{C}_k(\mathbf{H}) \mid y = -]$. Using the orthonormal basis from Section 3.1, linearity gives

$$\mathbb{E}[\mathbf{h}_j \mid y = +] - \mathbb{E}[\mathbf{h}_j \mid y = -] = \frac{\boldsymbol{\delta}_0}{\sqrt{n}} + \sum_{k=1}^{n-1} \mathbf{B}_n(j,k)\boldsymbol{\delta}_k. \tag{21}$$

Since $\mathbf{C}_0(\mathbf{H}) = \sqrt{n}\,\bar{\mathbf{h}}$, the first term is the difference of class-conditional sequence means. The remaining basis vectors have zero mean, establishing the DC specialization of mean-difference steering.

Write $\mathbf{h} = \mathbf{h}_{n-1}$ and $b_k = \mathbf{B}_n(n-1,k)$. An endpoint change satisfies

$$\mathbf{C}_k(\mathbf{H} + [0,\dots,0,\Delta\mathbf{h}]^\top) = \mathbf{C}_k(\mathbf{H}) + b_k\Delta\mathbf{h}. \tag{22}$$

For $k = 1,\dots,n-1$, let $z_k = (\mathbf{u}_k^\top \mathbf{C}_k - m_k)/s_k$ and $\mathbf{J}_{k,:} = b_k\mathbf{u}_k^\top/s_k$. Then $\mathbf{z}(\mathbf{h}+\Delta\mathbf{h}) = \mathbf{z}(\mathbf{h}) + \mathbf{J}\Delta\mathbf{h}$. Suppressing the prediction-step index, the implemented correction is

$$\mathbf{f}(\mathbf{h}) = \sum_{k=1}^{n-1} b_k r_k \tanh(z_k)\mathbf{A}_k, \qquad \widehat{\mathbf{v}} = \mathbf{v} - \rho\mathbf{f}(\mathbf{h}), \tag{23}$$

$$\Delta\mathbf{h} = \eta\widehat{\mathbf{v}}, \qquad \eta = \frac{\alpha\,\|\mathbf{v}\|_2}{\max(\|\widehat{\mathbf{v}}\|_2, \epsilon_v)}. \tag{24}$$

F.2 A potential for aligned carriers

Assume $\mathbf{A}_k = \lambda_k\mathbf{u}_k$ with $\lambda_k \geq 0$, as when $\mathbf{A}_k = \boldsymbol{\delta}_k$. The choice $\mathbf{A}_k = \mathbf{A}_{t,k}$ need not satisfy this alignment. Define

$$c_k = \lambda_k s_k r_k, \qquad E_m(\mathbf{h}) = \sum_{k=1}^{n-1} c_k \log\cosh z_k(\mathbf{h}), \qquad U_{\mathbf{v}}(\mathbf{h}) = \rho E_m(\mathbf{h}) - \mathbf{v}^\top\mathbf{h}. \tag{25}$$

The chain rule gives $\nabla E_m = \mathbf{f}$, hence $\widehat{\mathbf{v}} = -\nabla U_{\mathbf{v}}$. Since $0 < (\log\cosh z)'' \leq 1$, a gradient Lipschitz bound is

$$L = \rho\left\|\mathbf{J}^\top \operatorname{diag}(c_k)\mathbf{J}\right\|_{\mathrm{op}}. \tag{26}$$

Proposition 1 (local descent). For aligned carriers and $L > 0$, the update satisfies

$$U_{\mathbf{v}}(\mathbf{h}+\Delta\mathbf{h}) \leq U_{\mathbf{v}}(\mathbf{h}) - \eta\left(1 - \frac{L\eta}{2}\right)\|\nabla U_{\mathbf{v}}(\mathbf{h})\|_2^2. \tag{27}$$

Thus the decrease is strict for $\widehat{\mathbf{v}} \neq 0$ and $0 < \eta < 2/L$.

Proof. Apply the smoothness inequality to $\Delta\mathbf{h} = -\eta\nabla U_{\mathbf{v}}$. If $L = 0$, the potential is affine and its change is exactly $-\eta\,\|\nabla U_{\mathbf{v}}\|_2^2$. □

Rescaling preserves the direction but does not enforce the step bound. Moreover, E_m is centered at the class midpoint; a positive-class target requires an additional condition on the base direction.

F.3 Discrepancy to the positive-class center

Let $\tau_k = (\mathbf{u}_k^\top\boldsymbol{\mu}_k^+ - m_k)/s_k$ and $\Psi(z) = \log\cosh z$. The nonnegative Bregman discrepancy is

$$\mathcal{D}_+(\mathbf{h}) = \rho\sum_{k=1}^{n-1} c_k\left[\Psi(z_k) - \Psi(\tau_k) - \tanh(\tau_k)(z_k - \tau_k)\right]. \tag{28}$$

Proposition 2 (positive-prior compatibility). Assume aligned carriers, $\rho > 0$, and

$$\mathbf{v} = \rho \sum_{k=1}^{n-1} b_k \lambda_k r_k \tanh(\tau_k) \mathbf{u}_k. \tag{29}$$

Then $\widehat{\mathbf{v}} = -\nabla \mathcal{D}_+$, and the descent bound in Equation 27 holds with $U_{\mathbf{v}}$ replaced by $\mathcal{D}_+$.

Proof. Because each z_k is affine in $\mathbf{h}$, the linear part of $\rho \sum_k c_k \tanh(\tau_k) z_k$ is $\mathbf{v}^\top \mathbf{h}$ under Equation 29. Hence $\mathcal{D}_+$ and $U_{\mathbf{v}}$ differ by a constant, and Proposition 1 applies. □

The base directions used in our experiments do not enforce this compatibility condition. The proposition is therefore conditional. It concerns a weighted projected-center discrepancy, not matching the positive distribution's variance or cross-frequency dependencies.

For arbitrary frozen carriers and base directions, a direct check is available for $D_2 = \frac{1}{2} \|\mathbf{z} - \boldsymbol{\tau}\|_2^2$:

$$D_2(\mathbf{h} + \Delta\mathbf{h}) - D_2(\mathbf{h}) = \eta(\mathbf{z} - \boldsymbol{\tau})^\top \mathbf{J}\widehat{\mathbf{v}} + \frac{\eta^2}{2} \|\mathbf{J}\widehat{\mathbf{v}}\|_2^2 . \tag{30}$$

The identity expresses the change in projected discrepancy as a directional-alignment term plus a quadratic step-size term.

G Extended Related Work

Attribute representations and behavioral control. The linear representation hypothesis connects semantic concepts with directions in a model's hidden space (Park et al., 2024). Linear probes provide a practical way to test whether attributes can be recovered from these representations, although probe performance depends on the probe and evaluation design (Belinkov, 2022). Representation engineering develops this connection into methods for reading and controlling high-level concepts through activations (Zou et al., 2023). AxBench evaluates concept detection and steering separately, showing that strong detection does not by itself imply effective behavioral control (Wu et al., 2025). This distinction also guides our study: frequency-wise probes establish decodability, while generation experiments and component ablations assess whether the same temporal structure is useful for steering.

Steering direction construction. Activation Addition constructs directions by contrasting activations elicited by different prompts (Turner et al., 2023); CAA averages activation differences over paired positive and negative examples (Rimsky et al., 2024). ITI uses probes to select attention heads and intervenes along directions associated with truthfulness (Li et al., 2023). In-context Vectors summarize demonstrations as vectors that shift hidden states on new inputs (Liu et al., 2024), while representation engineering also uses principal components of contrastive activations to identify concept directions (Zou et al., 2023). These approaches provide several ways to derive an intervention from examples. Our method uses DCT coefficients to adjust a base direction constructed either offline from calibration responses or online from the difference between hidden states under contrasting prompts. During generation, it computes the DCT coefficients from a sliding window of hidden states at the same layer, covering consecutive token positions ending at the current position.

Learned and sparse interventions. Other approaches learn the intervention itself. Early work directly optimizes latent steering vectors (Subramani et al., 2022), and ReFT learns interventions on hidden representations while freezing the language model; LoReFT restricts the intervention to a low-rank linear subspace (Wu et al., 2024). Sparse autoencoders offer another representation space for steering. SAE-SSV learns supervised steering vectors within selected sparse features (He et al., 2025), while SDCV uses discriminative sparse features to reduce noise in concept vectors (Zhao et al., 2026). These methods emphasize which hidden features to modify and how to estimate an effective intervention. SARA instead estimates frequency-wise means, scales, and reliability weights from labeled responses and computes the correction during inference, without optimizing intervention parameters. Its use of calibration data is distinct from learning a steering module through gradient updates.

Adaptive activation steering. SADI identifies important activation elements from contrastive examples and scales the selected elements according to current activations (Wang et al., 2025). Angular Steering controls behavior through rotations in activation space (Vu & Nguyen, 2025). Spherical

Steering constructs a steering direction from the difference between the mean activations of positive and negative responses. It rotates each token's activation toward this direction while preserving its norm, using cosine similarity with the direction to determine whether and how much to rotate (You et al., 2026). Dynamic Steering with Episodic Memory conditions token-level interventions on the query and demonstration memory (Do et al., 2025). These methods show that adaptation can concern the intervention's direction, strength, or geometry. Our contribution concerns the information used for adaptation: SARA explicitly decomposes an ordered window of activations and uses its frequency coefficients to adjust the steering direction.

Feedback across layers and tokens. PID-AcT combines the difference between positive and negative mean activations with the corresponding differences from preceding layers to construct feedback (Nguyen et al., 2026). SARA constructs frequency-resolved feedback from consecutive token positions at the same layer. It decomposes the current activation window, compares each frequency coefficient with positive and negative calibration statistics, and reconstructs a correction for the current position. This design uses temporal variation and attribute-specific calibration to adapt the steering direction during generation.

Spectral sequence representations. Almarwani et al. (2019) construct sentence embeddings using DCT coefficients of word-embedding sequences, retaining information beyond their average. Tamkin et al. (2020) apply spectral filters along token positions to examine linguistic structure at different scales, including part-of-speech, dialogue-act, and topic information. Their prism layer also encourages different neurons to represent different frequency ranges during training. These studies establish temporal spectral analysis as a tool for representing and studying language. We extend this perspective to target-attribute information in autoregressive activation sequences and connect it to inference-time control. The calibration statistics give the observed coefficients an attribute-dependent interpretation, and the sliding window makes them available as feedback before each next-token prediction.

Spectral activation editing. Spectral Editing of Activations (SEA) is particularly relevant because it also uses positive and negative examples without finetuning the model (Qiu et al., 2024). SEA estimates cross-covariance matrices between neutral prompt activations and activations from positive or negative completions. It applies SVD to obtain hidden-space projections that retain directions with high positive covariance and low negative covariance, then uses these projections to edit activations during inference. Its nonlinear variant performs the same construction after a feature transformation. Here, singular values describe covariance between hidden features. In SARA, frequencies describe variation along ordered token positions: the cosine basis is fixed by the window length, while attribute directions and standardization statistics are calibrated separately at each frequency. The online coefficients determine the feedback correction. Thus, both methods use data to guide activation edits, but they decompose different objects and use the resulting components differently. SEA constructs data-dependent hidden-space projectors; SARA uses a temporal frequency decomposition to adapt a steering direction. Principal-component directions in representation engineering likewise describe variation across contrastive examples, rather than predefined frequencies along a token sequence.