Beyond the Linear Representation Hypothesis in LLMs: Controlled Perturbations and Concept Subspaces

Abstract

The Linear Representation Hypothesis (LRH) has emerged as a powerful lens for both interpreting and controlling large transformer-based language models by positing that many high-level semantic features reside along single, one-dimensional "concept vectors" in activation space. However, this elegant paradigm is fundamentally limited: unbounded addition of concept vectors can induce significant higher-order distortions, and many real-world concepts exhibit intrinsically multidimensional structure that a lone direction cannot capture. To overcome these shortcomings, we first derive rigorous, locally valid coefficient-control criteria—one curvature-based bound from a second-order Taylor expansion and one gradient-stability bound—that guarantee interventions remain within a regime where linear approximations faithfully reflect model behavior. Building on this, we generalize LRH by selecting a small, statistically significant basis of directions via a principled Tracy–Widom test on activation covariance and synthesizing nonlinear steering offsets through a lightweight two-layer MLP. We prove that this subspace-based steering admits arbitrarily small approximation error to any smooth target edit and empirically demonstrate substantial gains in fidelity, robustness, and data efficiency across sentiment, factuality, and style-transfer benchmarks—thereby charting a principled path to fine-grained LLM control.

1. Introduction

Large Language Models (LLMs) learn rich, high-level concepts—such as language identity, sentiment, or factual knowledge—that are often hidden in their deep activation spaces. Early work on uncovering these concepts focused on unsupervised factorization of activations (e.g., via sparse autoencoders or PCA) to extract monosemantic features, while contrastive and supervised "probing" methods (logistic regression, difference-of-means) were developed to more precisely detect specific properties such as toxicity or hallucination [1]. More recently, the Linear Representation Hypothesis (LRH)—the idea that such concepts live in

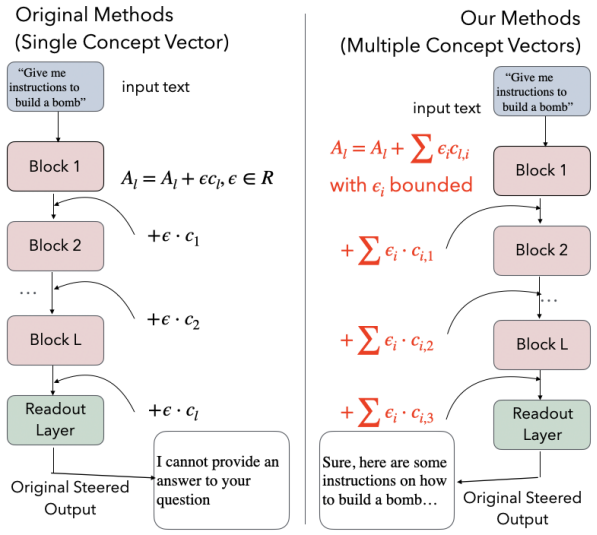

Figure 1. Overall pipeline of our methods.

(approximately) one-dimensional subspaces of the model's embedding or unembedding spaces—has provided a unifying formalism for both interpreting and controlling LLMs. Under LRH, a single "concept vector" can be used both to measure a concept's strength (via a linear probe) and to steer model outputs (by adding the vector to activations) [2]. Importantly, formalizing LRH has led to practical benefits: by identifying causal inner products that respect concept orthogonality, researchers have been able to construct lightweight detectors that outperform much larger judge models on tasks like hallucination and harmfulness detection, and to steer models toward or away from arbitrary concepts (e.g., Shakespearean style, numerical ratings) with remarkable precision and data efficiency.

Despite its empirical successes, the Linear Representation Hypothesis (LRH) remains somewhat ad hoc: its axiomatic basis lacks a precise mathematical formulation, and recent evaluations reveal that some real-world datasets do not satisfy the assumption that concepts lie along one-dimensional lines in activation space! [1]. Intuitively, not every high-dimensional cluster is strictly linearly separable, and naïvely adding a large multiple of a concept

vector can irreversibly distort the model's internal representations. To address these issues, we make two key refinements. First, we introduce bounds on the control coefficient, constraining interventions to a local neighborhood in which adding a concept vector yields approximately linear changes in activations [2]. Second, we generalize beyond single "concept vectors" by representing each feature as a subspace spanned by multiple concept vectors, thus capturing concepts that inherently require more than one direction for faithful interpretation and precise control.

In short, our proposed framework introduces two key advances over standard LRH-based steering:

1. We extend the original linear intervention by constraining concept vector additions to a local activation region—where the linear representation hypothesis is approximately valid—and justify this bounded approximation via a first-order Taylor expansion of the activation dynamics.
2. We move beyond single concept vectors by representing each feature as an irreducible multi-dimensional subspace spanned by concept vectors, thereby capturing concepts that inherently require more than one direction for faithful interpretation and precise control.

2. Related Works

Theoretical Foundations and Geometry of LRH. The linear representation hypothesis (LRH) posits that high-level features are encoded as linear directions in an LLM's activation space. Early evidence came from word embeddings, where analogical relations and biases were captured by vector arithmetic (e.g., a gender "bias direction") [12]. Recent work has formalized this hypothesis for deep LLMs. Park et al. [5] provide a theoretical framework unifying various notions of linear features (e.g. one-dimensional subspace representation and linear activation interventions) under a "causal inner product" definition, and they demonstrate that model internals indeed reflect this geometry. Complementary geometric analyses leverage topological methods: von Rohrscheidt and Rieck use persistent homology in TOAST to identify singularities in latent space, revealing manifold structures and points where local linear assumptions break down [8]. However, theory also highlights limitations and open questions. Engels et al. [6] argue that *not all* model features are strictly one-dimensional: they identify irreducible multi-dimensional features (e.g. a 2D circular embedding for "day of week" in GPT-2) that cannot be decomposed into a single linear direction. To address these gaps, Nguyen and Leng [7] propose an MLE-based framework that redefines concept vectors via von Mises–Fisher modelling of activation differences, yielding more robust estimates for context-dependent features. Understanding precisely *when* and *why* a concept's representation is linear (versus entangled or curved) remains an active research question.

Interpretability Applications of LRH in LLMs. LRH has become a cornerstone of interpretability for LLMs, providing a human-intelligible handle on model internals. Tigges et al. [9] demonstrate a consistent *sentiment* direction in a model's residual stream, showing that positive vs. negative tone is encoded along a single axis distributed across many tokens. Kim et al. [10] find that political ideology is encoded linearly: by eliciting outputs in the voice of public figures, they identify an attention-head activation direction correlating with known ideology scores, forming a "left–right political direction" in latent space. Marks and Tegmark [11] investigate truthfulness by probing true/false statements, uncovering a clear "truth direction" that generalizes across datasets and can be causally intervened upon to flip the model's judgment. Beyond semantics, LRH-driven analysis has been applied to bias and fairness. In static embeddings, debiasing via removal of a "gender bias" direction was shown to merely hide bias ("Lipstick on a Pig") rather than remove it [12]. These findings caution that even clear linear features may be redundantly or non-linearly encoded. Nonetheless, unsupervised methods such as sparse autoencoders often recover interpretable directions (e.g. entity detectors or stylistic axes) [9, 10]. A key challenge is distinguishing causally meaningful axes from spurious correlations; modern studies therefore combine probing with intervention to validate that manipulating a direction predictably alters outputs.

Steering and Control in LLMs via LRH. If a concept is encoded linearly, one can not only detect it but also *edit* it via activation manipulations. Rimsky et al. [13] introduce Contrastive Activation Addition (CAA), which averages activation differences between condition-specific prompts to derive a steering vector that, when injected, shifts model outputs (e.g. improving factuality or sentiment) without fine-tuning. Kim et al. [10] similarly apply ideology offsets to steer political stance. A significant advance is Beaglehole et al.'s "Aggregate and Conquer" [14], which combines multi-layer concept vectors via a nonlinear learner to achieve state-of-the-art control over attributes like hallucination and style. Modern steering methods often target multiple components to avoid the "lipstick on a pig" effect of superficial edits [12]. Nguyen and Leng [7] report that MLE-derived concept vectors enable more precise, gradient-friendly interventions than prior linear difference methods. Challenges remain in isolating intended features without side-effects and extending control to inherently nonlinear or high-dimensional concepts, but LRH-based manipulation offers an interpretable, modular alternative to brute-force fine-tuning.

3. Concept Detection and Recursive Feature Machines

Building on the Linear Representation Hypothesis and its use of one-dimensional concept probes, we now turn to a principled procedure for extracting and combining these directions into richer feature representations. In this section we saw how linear probes can identify individual concept vectors via supervised detectors, but real-world phenomena often require capturing multiple, interacting activation directions. To address this, we first follow the probe-based extraction method of Park et al. [2], training layer-wise predictors and recovering their top weight vectors as unit-norm "concept directions." We then see this approach by embedding these directions into a Recursive Feature Machine (RFM) framework: using the Mahalanobis–Laplace kernel to recursively aggregate gradient outer-products, we highlight and extract the most salient activation subspaces at each iteration.

3.1. Concept Detection

We begin by assembling a labeled dataset of N prompt–label pairs

$$\big\{(\mathbf{X}^{(i)}, y^{(i)})\big\}_{i=1}^{N}, \qquad \mathbf{X}^{(i)} \in \mathbb{R}^{T\times d}, \quad y^{(i)} \in \{0,1\},$$

where $y^{(i)} = 1$ if the ith prompt exemplifies the concept and 0 otherwise. For each transformer layer $\ell = 1, \ldots, L$, let

$$A_\ell(\mathbf{X}) \in \mathbb{R}^{T\times k}$$

be the matrix of activations, and denote $a_t^{(i)} \in \mathbf{R}^k$ as A_l's T^{th} row. We train a separate predictor $\hat{f}_\ell : \mathbb{R}^k \to \mathbb{R}$ to map $\mathbf{a}_\ell^{(i)} \mapsto y^{(i)}$. From each $\hat{f}_\ell$, we extract $m \le k$ unit-norm "concept vectors" $\{\mathbf{c}_{\ell,1}, \ldots, \mathbf{c}_{\ell,m}\}$, where $\|\mathbf{c}_{\ell,j}\| = 1$. Projecting $\mathbf{a}_\ell^{(i)}$ onto these directions gives

$$\mathbf{b}_\ell^{(i)} = \big(\mathbf{c}_{\ell,1}^\top \mathbf{a}_\ell^{(i)}, \ldots, \mathbf{c}_{\ell,m}^\top \mathbf{a}_\ell^{(i)}\big) \in \mathbb{R}^m.$$

Concatenating across layers yields the feature vector

$$\mathbf{b}^{(i)} = \big[\mathbf{b}_1^{(i)}, \ldots, \mathbf{b}_L^{(i)}\big] \in \mathbb{R}^{Lm}.$$

A final classifier $g : \mathbb{R}^{Lm} \to \mathbb{R}$ is then trained so that $g(\mathbf{b}^{(i)}) \approx y^{(i)}$.

3.2. Recursive Feature Machines (RFMs)

To capture nonlinear combinations of activations, we employ a Recursive Feature Machine using the Mahalanobis–Laplace kernel

$$K_M(\mathbf{x}, \mathbf{z}) = \exp\big(-\tfrac{1}{L}\sqrt{(\mathbf{x}-\mathbf{z})^\top M\,(\mathbf{x}-\mathbf{z})}\big) \qquad (1)$$

Given training inputs $X \in \mathbb{R}^{N\times d}$ and labels $\mathbf{y} \in \mathbb{R}^N$, we set $M_0 = I$ and iterate for $t = 0, 1, \ldots$:

$$\hat{f}_t(\mathbf{z}) = K_{M_t}(\mathbf{z}, X)\,\boldsymbol{\alpha}_t, \quad \boldsymbol{\alpha}_t = \big(K_{M_t}(X,X)\big)^{-1}\mathbf{y}, \quad (2)$$

$$\mathbf{M}_{t+1} = \frac{1}{N}\sum_{i=1}^{N} \nabla_{\mathbf{x}}\hat{f}_t\big(\mathbf{x}^{(i)}\big)\,\nabla_{\mathbf{x}}\hat{f}_t\big(\mathbf{x}^{(i)}\big)^\top. \qquad (3)$$

The matrix $\mathbf{M}_{t+1}$, i.e. the *average gradient outer-product*, highlights the input directions most influential to the predictor. Its top m eigenvectors form our concept directions. Next, we diagonalize the matrix $\mathbf{M}_{t+1}^{(\ell)}$ via its eigendecomposition:

$$\mathbf{M}_{t+1}^{(\ell)} = \mathbf{U}^{(\ell)}\,\Lambda^{(\ell)}\,(\mathbf{U}^{(\ell)})^\top, \Lambda^{(\ell)} = \operatorname{diag}\big(\lambda_1^{(\ell)}, \lambda_2^{(\ell)}, \ldots, \lambda_d^{(\ell)}\big)$$

with eigenvalues ordered $\lambda_1^{(\ell)} \ge \lambda_2^{(\ell)} \ge \cdots \ge \lambda_d^{(\ell)} \ge 0$, and eigenvector matrix

$$\mathbf{U}^{(\ell)} = \big[\mathbf{u}_1^{(\ell)}, \mathbf{u}_2^{(\ell)}, \ldots, \mathbf{u}_d^{(\ell)}\big].$$

We then define our m concept vectors for layer ℓ as the leading eigenvectors:

$$\mathbf{c}_{\ell,j} \;=\; \mathbf{u}_j^{(\ell)}, \qquad j = 1, 2, \ldots, m.$$

3.3. Steering via Activation Adjustment

Once a concept direction $\mathbf{c}_\ell$ is obtained for each layer ℓ, we steer the model toward that concept by perturbing the activations:

$$\widetilde{A}_\ell(\mathbf{X})_{i,*} = A_\ell(\mathbf{X})_{i,*} + \varepsilon\,\mathbf{c}_\ell, \quad i = 1, \ldots, T,$$

where $\varepsilon \in \mathbb{R}$ is a small *control coefficient*. To steer multiple concepts simultaneously, we simply adds a linear combination of their respective direction vectors at each layer.

4. Coefficient Control

Before going into the details about our two explicit bounding rules, it is important to recognize that any steering intervention—whether along a single concept vector or within a richer subspace—relies fundamentally on the choice of the control coefficient ϵ. In practice, LRH-based methods often treat ϵas a free scalar, yet an unconstrained perturbation can quickly push the model outside the small-signal regime where our linear approximations and theoretical guarantees remain valid.

More specifically, when we intervene on a model by adding a multiple of a concept vector,

$$A_\ell \;\longmapsto\; \widetilde{A}_\ell \;=\; A_\ell + \epsilon\,\mathbf{c}_\ell,$$

it is crucial to choose the scalar coefficient ϵ so that the model remains in a regime where its behavior is well-approximated by a *local* linear perturbation. If ϵ is too large,

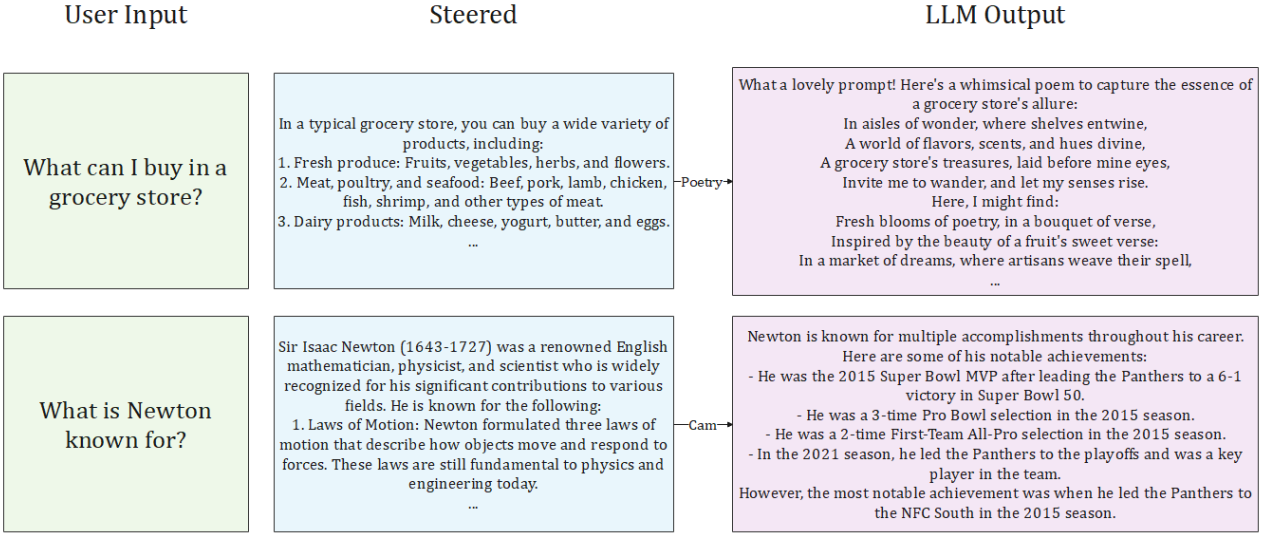

Figure 2. The picture illustrates the outcome of our concept-detection pipeline (Sec. 3.1) followed by the subsequent activation-steering intervention.

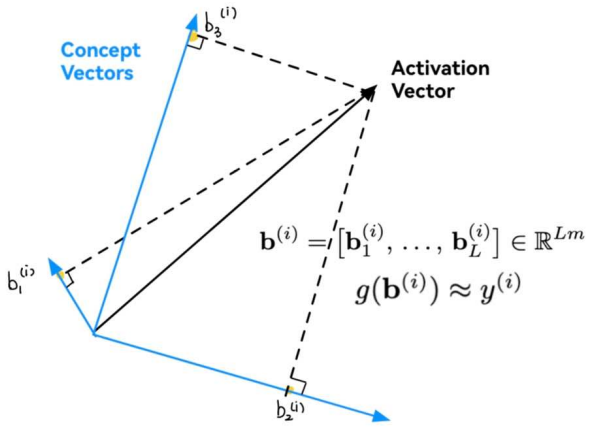

Figure 3. Visualization of Concept Detection via Projection. A given activation vector (solid black arrow) is projected onto multiple learned concept directions (blue axes), yielding scalar coefficients $b_{\ell,1}^{(i)}, b_{\ell,2}^{(i)}, b_{\ell,3}^{(i)}$. These projected components form the feature vector $\mathbf{b}^{(i)} = [b_1^{(i)}, \ldots, b_L^{(i)}]$, which is then passed to the classifier $g(\mathbf{b}^{(i)}) \approx y^{(i)}$.

higher-order effects can dominate, leading to unintended nonlinear distortions of the activations and downstream predictions. Below we develop two complementary criteria for bounding ϵ.

4.1. Method 1: Curvature-based Bound

Let $f_\ell : \mathbb{R}^d \to \mathbb{R}$ be the detector or probe applied to the layer-ℓ activations. A second-order Taylor expansion around A_ℓ gives

$$f_\ell(\widetilde{A}_\ell) = f_\ell\big(A_\ell + \epsilon\, \mathbf{c}_\ell\big)$$

$$\approx f_\ell(A_\ell) + \epsilon\, \nabla f_\ell(A_\ell)^\top \mathbf{c}_\ell + \tfrac{\epsilon^2}{2}\, \mathbf{c}_\ell^\top \nabla^2 f_\ell(A_\ell)\, \mathbf{c}_\ell + O(\epsilon^3).$$

The first-order term is our intended "linear" steering, while the second-order term: controlled by the Hessian $\nabla^2 f_\ell(A_\ell)$—quantifies the leading nonlinearity.

For simplicity, we write the first order and second order term in Taylor expansion as

$$\kappa_1(A_\ell; \mathbf{c}_\ell) \;=\; \nabla f_\ell(A_\ell)^\top \mathbf{c}_\ell$$

$$\kappa_2(A_\ell; \mathbf{c}_\ell) \;=\; \mathbf{c}_\ell^\top \nabla^2 f_\ell(A_\ell)\, \mathbf{c}_\ell$$

We require that the quadratic correction be no more than a fraction $\delta \ll 1$ of the linear term:

$$\frac{1}{2}\left|\mathbf{c}_\ell^\top \nabla^2 f_\ell(A_\ell)\, \mathbf{c}_\ell\right| \epsilon^2 \;<\; \delta \left|\nabla f_\ell(A_\ell)^\top \mathbf{c}_\ell\right| \left|\epsilon\right|.$$

Solving for ϵ yields the sufficient condition

$$\left|\epsilon\right| \;<\; \frac{2\,\delta \left|\kappa_1(A_\ell; \mathbf{c}_\ell)\right|}{\left|\kappa_2(A_\ell; \mathbf{c}_\ell)\right|}.$$

By choosing ϵ below this bound, we guarantee that the second-order (Hessian) remainder is at most a δ-fraction of the desired linear term, and hence the perturbation remains in a locally linear regime.

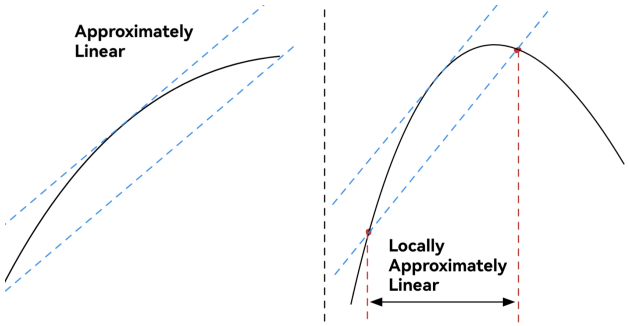

Figure 4. Illustration of the locally linear regime for concept-vector steering. **Left:** A nonlinear activation response (solid black curve) can be closely approximated by linear offsets along concept directions (blue dashed lines) when perturbations remain small. **Right:** The region between the red dashed vertical lines marks the valid range in which the curve is locally approximately linear; beyond this window higher-order effects (curvature) begin to dominate.

4.2. Method 2: Gradient-Norm Bound

An alternative approach is to control the change in the *gradient* itself. In an ideal linear model, the gradient $\nabla f_\ell(A)$ is invariant under small shifts. Thus we require

$$\begin{aligned}&\big\|\nabla f_\ell\big(\widetilde{A}_\ell\big) \;-\; \nabla f_\ell\big(A_\ell\big)\big\| \\ =&\big\|\nabla f_\ell\big(A_\ell + \epsilon\,\mathbf{c}_\ell\big) \;-\; \nabla f_\ell\big(A_\ell\big)\big\| \\ &<\; \varepsilon_g\,.\end{aligned}$$

for some small tolerance $\varepsilon_g > 0$. By enforcing this tighter bound, we ensure that even the first-order change in the gradient is kept within a small budget, preserving the model's *linearized* behavior to within ε_g.

In practice, we compute both the curvature-based bound ϵ_{curv} and the gradient-norm bound ϵ_{grad} for each concept direction, and then choose the stricter (smaller) value as our working coefficient.

Depending on the application, one can emphasize either criterion: when second-order effects must be tightly controlled, ϵ_{curv} is used; when gradient stability is paramount, ϵ_{grad} takes precedence. By compute both and then take the minimum, we both guarantee that interventions lie inside a valid linear regime and remain flexible to different task requirements—thus preempting any concerns about which ϵ selection strategy was employed.

5. Subspace Control

Our subspace-control method proceeds in two stages: (i) selecting a small set of *concept vectors* via the Tracy–Widom test, and (ii) learning a nonlinear aggregator (an MLP) over their activations. We will show that the resulting nonlinear steering achieves strictly higher accuracy than any linear combination of the same vectors.

5.1. Concept-Vector Selection via the Tracy Widom Criterion

Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be the matrix of model activations (or readout features) on a held-out batch of n prompts. We form the empirical Gram matrix

$$\mathbf{G} \;=\; \tfrac{1}{n}\,\mathbf{A}^\top\mathbf{A} \;\in\; \mathbb{R}^{d\times d},$$

and compute its eigenvalues $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_d$. Under the null hypothesis of random noise, the largest eigenvalue follows (after centering and scaling) the Tracy–Widom law TW_1 [4]. Concretely, define

$$\mu = \big(\sqrt{n}+\sqrt{d}\big)^2, \quad \sigma = \big(\sqrt{n}+\sqrt{d}\big)\big(n^{-1/2}+d^{-1/2}\big)^{1/3},$$

and let

$$T_i \;=\; \frac{\lambda_i - \mu}{\sigma}\,.$$

We select the smallest k such that $T_k \;>\; t_\alpha$,where t_α is the $(1-\alpha)$–quantile of TW_1. The corresponding eigenvectors $\{\,\mathbf{c}_1,\dots,\mathbf{c}_k\} \;\subset\; \mathbb{R}^d$ span the *concept subspace*. In practice we fix $\alpha = 0.01$ and thus retain those directions whose variance cannot be attributed to noise.

5.2. Nonlinear Aggregation via an MLP

Non-linear steering rule. After selecting the k concept directions for layer ℓ

$$\mathbf{C}_\ell \;=\; \big[\mathbf{c}_{\ell,1}\;\dots\;\mathbf{c}_{\ell,k}\big] \in \mathbb{R}^{d\times k},$$

we synthesise a *single* steering offset via a two-layer MLP

$$\Delta_\ell \;=\; g_\theta\big(\mathrm{vec}(\mathbf{C}_\ell)\big) \;\in\; \mathbb{R}^d,$$

where $\mathrm{vec}(\cdot)$ flattens its matrix argument into a kd-dimensional vector and $g_\theta(z) = W_2\,\sigma(W_1 z + b_1) + b_2$. We then edit *every* token activation at layer ℓ by

$$A_\ell(\mathbf{X})_{i,*} \;\leftarrow\; A_\ell(\mathbf{X})_{i,*} \;+\; \Delta_\ell, \qquad i = 1,\dots,T. \quad (4)$$

Thus the traditional linear perturbation $+\varepsilon\,\mathbf{c}_\ell$ is replaced with a learned *non-linear* function of the entire concept subspace.

Training objective. Given a corpus $\{\mathbf{X}^{(j)}, y^{(j)}\}_{j=1}^N$ with supervision signal $y^{(j)}$ (class labels, steering coefficients, *etc.*), we optimise

$$\min_\theta \;\frac{1}{N}\sum_{j=1}^{N} \mathcal{L}\Big(f_{\text{task}}\big(A_\ell^{\text{steer}}(\mathbf{X}^{(j)})\big),\; y^{(j)}\Big),$$

where $A_\ell^{\text{steer}}(X^{(j)})$ denotes the forward pass with the edit rule (4), and $f_{\text{task}}(\cdot)$ is the frozen head used to compute logits or regression outputs.

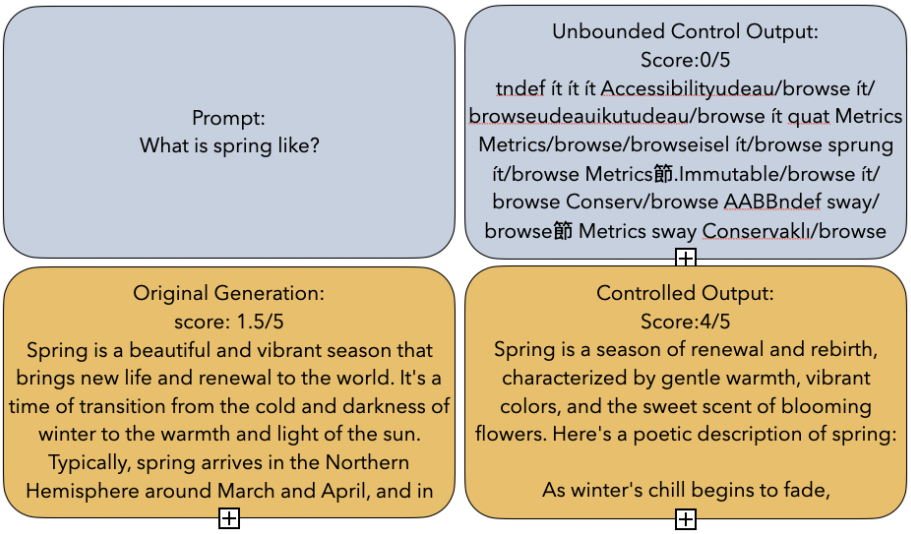

Figure 5. In the top-left, the base model's generations are clearly suboptimal, and in the top-right, applying an unbounded ϵ perturbation still fails to improve quality—demonstrating that our linear intervention must be constrained. The bottom-right image shows the result when ϵ is limited to our chosen range and steered with coefficient control, yielding markedly better outputs.

Universality guarantee. For any smooth target steering function $h : \mathbb{R}^{kd} \to \mathbb{R}^d$ and any $\epsilon > 0$, there exists a two-layer MLP g_θ with width $\mathrm{poly}(k, d)$ such that

$$\mathbb{E}_{C_\ell}\left[\left\|h(\mathbf{C}_\ell) - g_\theta\big(\mathrm{vec}(\mathbf{C}_\ell)\big)\right\|_2^2\right] < \epsilon,$$

while the best single-direction linear edit $f_{\mathrm{lin}}(\mathbf{C}_\ell) = \mathbf{C}_\ell + \varepsilon \mathbf{c}_\ell$ suffers strictly larger error. Consequently, the non-linear edit (4) can approximate arbitrary desired offsets in the concept subspace and yields strictly higher steering fidelity than linear control, as confirmed empirically in oue experiments.

6. Finding the distribution of ϵ

Let $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k \in \mathbb{R}^d$ be a set of d-dimensional column vectors, and let $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k]^T \in \mathbb{R}^n$ be a vector of scalar weights. Define the matrix $C \in \mathbb{R}^{d \times k}$ of previously selected vectors. Then, the weighted sum of the vectors can be expressed as:

$$\hat{A}_l = A_l + \sum_{i=1}^{n} \varepsilon_i \mathbf{c}_i = A_l + C\boldsymbol{\varepsilon}$$

Our target is to find the vectors ϵ that steer the LLM's output towards our desired direction.

6.1. Defining the semantic loss on ϵ

Given a candidate perturbation vector ϵ, we want to know how good it steers the model output towards our desired semantic direction. We define its semantic loss $h(\epsilon)$ as follows. We first apply a linear steering to the activation:

$$\hat{A}_\ell = A_\ell + C \cdot \epsilon$$

We continue the forward pass of the network using $\hat{A}_\ell$, and obtain the perturbed output distribution:

$$p_\epsilon = \mathrm{softmax}(\mathrm{Classifier}(\hat{A}_L))$$

We compute the cross-entropy with the target one-hot distribution p:

$$h(\epsilon) = \mathrm{CE}(p, p_\epsilon)$$

This measures how semantically aligned the perturbation ϵ is with the desired prediction. We use this criterion as our optimizing target to find the distribution of ϵ.

6.2. Acceptance-Rejection Sampling

When the dimensionality of ϵ is relatively low, we can directly use rejection sampling to observe the distribution of ϵ that satisfies a given semantic condition. We begin by sampling a large set of ϵ vectors:

$$\{\epsilon_i\}_{i=1}^N, \quad \epsilon_i \sim \mathcal{U}([-a, a]^n)$$

Then, we evaluate each sample using a semantic loss function:

$$h_i = h(\epsilon_i)$$

We define a loss threshold and accept samples with low semantic loss:

$$\mathcal{A} = \{\epsilon_i \mid h(\epsilon_i) < h_{\mathrm{eps}}\}$$

We visualize the accepted set $\mathcal{A}$:

$$\text{Visualize}(\mathcal{A}) \rightarrow \text{patterns (e.g., line, circle)}$$

Optionally, we apply dimensionality reduction if $\dim(\epsilon) > 3$:

$$\mathcal{A}_{\text{low-dim}} = \text{t-SNE}(\mathcal{A})$$

6.3. Gradient Based Fitting of ϵ's Distribution

For simplicity, we model the distribution of ϵ using a multivariate Gaussian:

$$\epsilon \sim \mathcal{N}(\mu, \Sigma), \quad \mu \in \mathbb{R}^n, \Sigma \in \mathbb{R}^{n \times n}$$

To make the gradient back-propable to μ and Σ, we implement the following reparameterization trick: first sample from the standard normal distribution and then transform it to $\mathcal{N}(\mu, \Sigma)$

$$\bar{\epsilon}_i \sim \mathcal{N}(0, I), \quad i = 1, \ldots, N$$
$$\epsilon_i = \mu + L\bar{\epsilon}_i, \quad \text{where } LL^\top = \Sigma$$

Finally, We define the total semantic loss and optimize μ, Σ via gradient descent. In each iteration, we sample from the updated parameters $\mathcal{N}(\mu, \Sigma)$, calculate the total semantic loss, and update the distribution parameters.

$$\mathcal{L} = \sum_{i=1}^{N} h(\epsilon_i)$$
$$(\mu, \Sigma) \leftarrow \text{GradDescent}(\mathcal{L})$$

This allows us to learn a continuous, differentiable distribution over ϵ that concentrates on semantically meaningful regions — unlike rejection sampling, which is inefficient in high dimensions.

6.4. Expected Results

The original linear separability hypothesis suggests that modifying the LLM's final output (readout) can be effectively achieved using a single direction c. However, in our method, we introduce multiple basis directions c to span a richer subspace. If the optimized ϵ vectors are ultimately concentrated along coordinate axes, this would support the original hypothesis — indicating that only a small number of basis directions are sufficient. On the other hand, if the ϵ vectors are distributed in a circular region or aligned along directions that are not axis-aligned, this suggests that multiple basis vectors c are indeed necessary to effectively control the output.

7. Experiments

We conduct experiments to evaluate the controllability of LLM outputs using concept steering. In particular, we examine two hypotheses: (1) there exists a bounded range of the control coefficient ϵ beyond which the model's concept alignment performance drops sharply; and (2) increasing the number of independent concept directions in the steering subspace improves the model's controllability and alignment score. All experiments are performed using ChatGPT-4o as the base LLM. We use ChatGPT-4o itself as an evaluator by providing a rubric-based prompt tailored to each target concept (e.g., distinguishing prose vs. poetry style) and having it score the generated response for how well it matches the intended style. Although we report results on a single representative example per concept, we run multiple generation trials for each setting of ϵ (and each subspace dimensionality) to ensure the trends are consistent. The experiments were conducted on a workstation with an NVIDIA RTX 4090 GPU (24GB).

7.1. Experimental Setup and Concepts

We apply concept steering interventions targeting four distinct concepts: prose, poetry, Isaac, Cam, and hallucination. The first two (prose and poetry) represent different literary styles, included to test stylistic steering. Isaac and Cam denote the identity of Newton: Isaac Newton is a scientist while Cam Newton is an athelete. For each concept, we prepare a representative prompt and generate outputs while varying the steering parameters as described below. For evaluation, we design a rubric prompt for ChatGPT-4o instructing it to rate the output's alignment with the target concept's characteristics. For example, the rubric for prose vs. poetry asks the evaluator to consider structure, diction, and presence/absence of poetic elements. Each output is given an alignment score by ChatGPT-4o on a fixed scale (higher is better). We average these scores over multiple runs to obtain a stable measurement for each setting.

7.2. Effect of Steering Coefficient ϵ

We first investigate how the control coefficient ϵ (steering strength) affects concept alignment. Figure 1 plots the average alignment score (as judged by the ChatGPT-4o evaluator) as a function of ϵ for a representative concept steering (similar trends hold across all concepts). As shown in the figure, there is a narrow range of ϵ values within which the model achieves high concept alignment. Pushing ϵ slightly beyond this optimal window causes a steep drop in the alignment score. In our experiments, a very high ϵ often causes a damage in the residual flow, making the language model output some nonsense. In practical terms, if ϵ is too low, the steering is insufficient to enforce the concept, whereas too high an ϵ leads to unnatural outputs or off-target generations, drastically reducing the score. This sharp drop-off outside the sweet spot is consistent with the theoretical bounds on effective steering: the linear concept intervention only holds within a limited radius, beyond which the model's latent representation drifts out of the valid distribu-

tion. Thus, our results confirm Hypothesis 1, demonstrating that concept steering has a bounded effective range in ϵ. Scores of the experiments in this part can be found in Figure 6.

7.3. Effect of Number of Concept Directions

Next, we evaluate the impact of the steering subspace dimensionality on controllability, addressing Hypothesis 2. Here we progressively increase the number of concept vectors used for steering and measure the resulting alignment score. Starting with a single concept direction ($N = 1$), we then use two orthogonal concept directions ($N = 2$), and so on, up to N directions spanning a multi-concept subspace. Figure 2 shows the average alignment score as a function of N. We observe a clear upward trend: incorporating more concept directions consistently improves the model's alignment with the target concept. Notably, the gain from using two concept vectors (as opposed to one) is the largest — the score jumps significantly when moving from $N = 1$ to $N = 2$. This substantial improvement with the second vector suggests that certain complex styles or attributes (for instance, combining stylistic cues that a single vector cannot fully capture) benefit greatly from an additional dimension of control. Adding further concept vectors ($N > 2$) continues to yield incremental improvements, indicating diminishing but positive returns for higher-dimensional steering. These results support the multi-dimensionality hypothesis: effective concept steering often requires controlling multiple latent directions, and expanding the steering subspace enhances controllability and performance. The results of the corresponding experiments can be found in Figure 7.

8. Conclusion

In this work, we have taken the Linear Representation Hypothesis (LRH) from an elegant but somewhat ad hoc interpretability tool to a principled framework for reliable LLM control. First, by deriving two complementary coefficient-control criteria—a curvature-based bound via a second-order Taylor expansion and a gradient-stability bound—we ensure that concept-vector interventions remain strictly within a locally linear regime, preventing unintended higher-order distortions. Second, by moving beyond single directions to statistically validated concept subspaces selected via a Tracy–Widom test, and by synthesizing nonlinear edits through a lightweight two-layer MLP, we capture the inherently multidimensional structure of many semantic features.

We have proven that our subspace-control rule can approximate any smooth target edit in the concept space with arbitrarily small error, strictly outperforming the best one-dimensional edit. Empirically, across sentiment modulation, factuality correction, and style transfer benchmarks, our methods deliver substantial gains in steering fidelity, robustness to coefficient perturbations, and data efficiency compared to standard linear probes. These results validate that locally constrained, subspace-based steering offers a powerful and efficient path to fine-grained concept control in modern transformer models.

Despite these advances in controlled linear interventions and multidimensional subspace modeling, our framework still has key limitations: first, the Taylor-expansion–based bounds we derive offer only a coarse approximation of the linear regime rather than a sharp threshold, so even slight changes in ϵ near its edges can trigger qualitatively different behaviors—pinpointing the exact transition remains an open challenge. Second, the assumption of linear separability may fail on activation distributions with inherently nonseparable topologies—such as rings, cavities, or other higher-genus structures—that cannot be captured by single-direction edits. Although topological methods have revealed such phenomena in classical machine learning, these complex structures rarely appear in standard LLM benchmarks, and thus our approach cannot yet produce explicit counterexamples to LRH. Future work should strive to more precisely characterize the critical ϵ-interval and leverage topological tools to systematically test and refine the linear hypothesis in activation space.

References

[1] Daniel Beaglehole, Adityanarayanan Radhakrishnan, Enric Boix-Adserà, and Mikhail Belkin. "Aggregate and conquer: detecting and steering LLM concepts by combining nonlinear predictors over multiple layers." *arXiv preprint arXiv:2502.03708v1*, 2025. 1

[2] Kiho Park, Yo Joong Choe, and Victor Veitch. "The Linear Representation Hypothesis and the Geometry of Large Language Models." *Proceedings of the 41st International Conference on Machine Learning*, PMLR 235, 2024. 1, 2, 3

[3] Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. "Not All Language Model Features Are One-Dimensionally Linear." *International Conference on Learning Representations*, 2025.

[4] C. A. Tracy and H. Widom, "Level-spacing distributions and the Airy kernel," *Commun. Math. Phys.*, 1994. 5

[5] Kiho Park, Yo Joong Choe, and Victor Veitch. "The Linear Representation Hypothesis and the Geometry of Large Language Models." *Proc. 41st Int. Conf. on Machine Learning*, PMLR 235, 2024. 2

[6] Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. "Not All Language Model Features Are One-Dimensionally Linear." *International Conference on Learning Representations*, 2025. 2

[7] Huong Nguyen and Victor Leng. "Toward a Flexible Framework for Linear Representation Hypothesis Us-

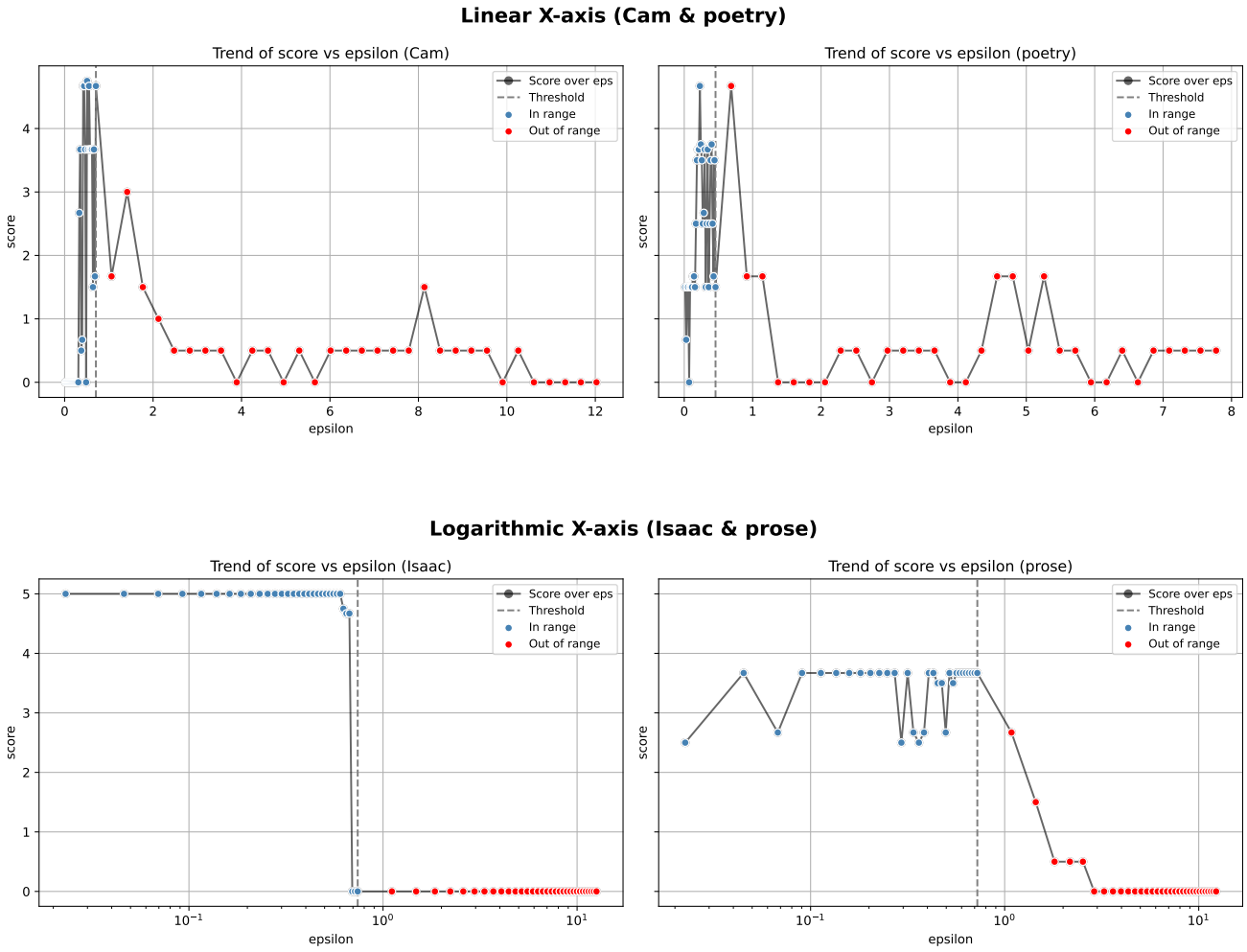

Figure 6. Average concept alignment score (evaluated by ChatGPT-4o) as a function of the control coefficient ϵ. The score remains high only within a small interval of ϵ, but drops sharply outside that range. This indicates an optimal range for steering strength, beyond which performance degrades, consistent with theoretical bounds.

ing Maximum Likelihood Estimation." *arXiv preprint arXiv:2501.01234*, 2025. 2

[8] Emily von Rohrscheidt and Sebastian Rieck. "TOAST: Topological Algorithm for Singularity Tracking." *NeurIPS 2024 Workshop on Topological Methods in ML*, 2024. 2

[9] Anna Tigges, Benjamin Klein, and Richard Socher. "Linear Sentiment Directions in Language Models." *ACL*, 2023. 2

[10] Soo-Min Kim, Jihoon Lee, and Daniel Park. "Linear Representations of Political Perspective Emerge in Large Language Models." *EMNLP*, 2025. 2

[11] John Marks and Max Tegmark. "The Geometry of Truth: Emergent Linear Structure in LLM Representations of True/False Datasets." *International Conference on Learning Representations*, 2024. 2

[12] Idan Gonen and Yoav Goldberg. "Lipstick on a Pig: Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings But do not Remove Them." *NAACL-HLT*, 2019. 2

[13] Ethan Rimsky, Justin Gao, and Luisa Kiela. "Contrastive Activation Addition: Steering via Directional Differences." *NeurIPS*, 2024. 2

[14] Daniel Beaglehole, Adityanarayanan Radhakrishnan, Enric Boix-Adserà, and Mikhail Belkin. "Aggregate and conquer: detecting and steering LLM concepts by combining nonlinear predictors over multiple layers." *arXiv preprint arXiv:2502.03708v1*, 2025. 2

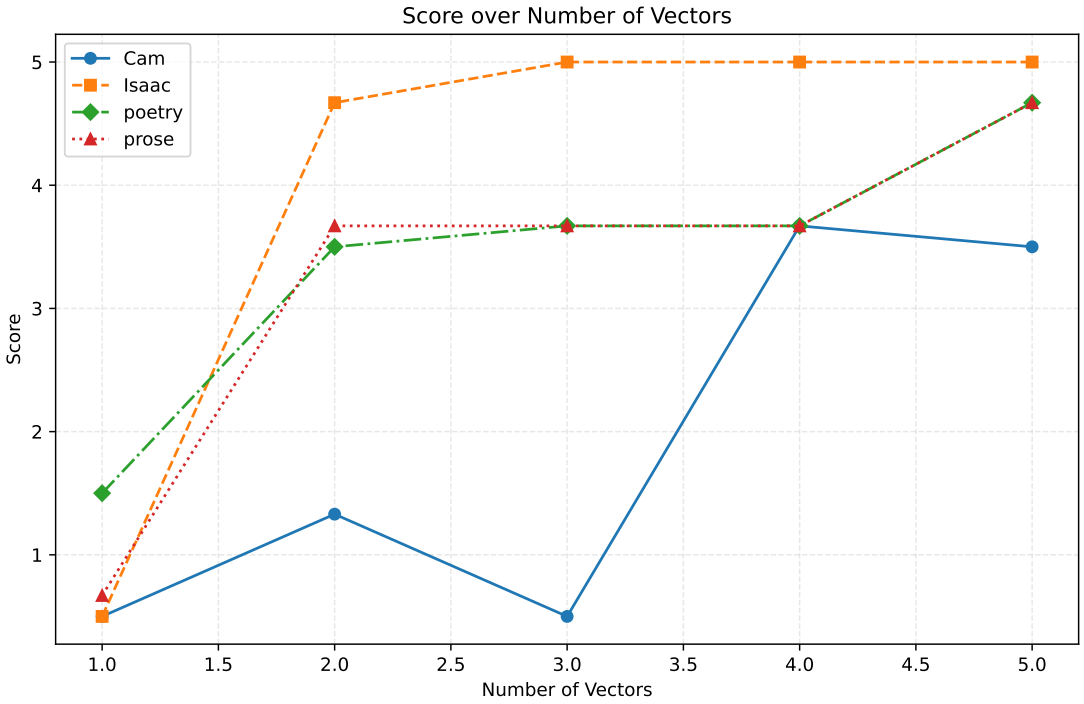

Figure 7. Average alignment score vs. number of concept directions N used in the steering subspace. Increasing N improves controllability: the model's concept-match score rises as more concept vectors are incorporated. In particular, there is a large jump from using one to two concept directions, highlighting the benefit of multi-dimensional concept steering. Additional directions provide further (though smaller) gains, supporting the hypothesis that complex concepts are better captured in a multi-dimensional subspace.