Why Steering Works:
Toward a Unified View of Language Model Parameter Dynamics

Anonymous ACL submission

Abstract

Methods for controlling large language models (LLMs), including local weight fine-tuning, LoRA-based adaptation, and activation-based interventions, are often studied in isolation, obscuring their connections and making comparison difficult. In this work, we present a unified view that frames these interventions as dynamic weight updates induced by a control signal, placing them within a single conceptual framework. Building on this view, we propose a unified preference-utility analysis that separates control effects into preference, defined as the tendency toward a target concept, and utility, defined as coherent and task-valid generation, and measures both on a shared log-odds scale using polarity-paired contrastive examples. Across methods, we observe a consistent trade-off between preference and utility: stronger control increases preference while predictably reducing utility. We further explain this behavior through an activation manifold perspective, in which control shifts representations along target-concept directions to enhance preference, while utility declines primarily when interventions push representations off the model's valid-generation manifold. Finally, we introduce a new steering approach guided by this analysis that improves preference while better preserving utility.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities and are increasingly deployed in real-world applications (Zhao et al., 2023). Growing demands for safety, controllability, and personalization make reliable control over model behavior a central challenge. To address this, prior work has developed diverse paradigms for controlling LLMs, spanning training-time adaptations, such as local weight fine-tuning and parameter-efficient methods like LoRA (Hu et al., 2022b; Ding et al., 2023; Mao et al., 2025), and inference-time interventions, including activation-level steering via hidden-state manipulation (Rimsky et al., 2024; Han et al., 2024).

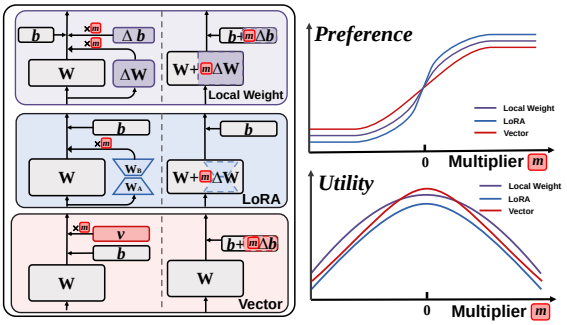

Figure 1: The figure illustrates how different methods operate on the linear layers of the model. We present a unified view in which diverse large language model intervention methods are casted as **dynamic weight updates**. The right panel shows the changes in model utility and preference across different control methods under varying intervention multipliers. Further details are provided in Section 3

Despite their empirical success, these approaches are often studied in isolation, under different assumptions, objectives, and evaluation protocols. This fragmentation hinders rigorous comparison and obscures shared failure modes. In this work, as shown in Figure 1, we mathematically observe that local weight fine-tuning, LoRA, and activation-level steering can all be formulated as instances of a common *dynamic weight update* framework (Eq. 1). Building on this unified perspective, we introduce a preference–utility analysis and show that, across methods instantiated within this framework, both preference and utility exhibit consistent, predictable patterns as control strength varies.

$$\mathbf{h}_{i+1} = (\mathbf{W} + m_1 \Delta\mathbf{W})\,\mathbf{h}_i + (\mathbf{b} + m_2 \Delta\mathbf{b}). \quad (1)$$

Note that a particularly challenge in controlled text generation is the **trade-off between enforcing the target concept and preserving task validity**:

as control strength increases, the target attribute is amplified, but undesirable side effects—such as incoherence, instruction violations, or context drift—also become more frequent, reducing overall task effectiveness. Moreover, because control quality is typically evaluated via realized outputs, degradation in task validity can confound assessments and obscure the intended concept signal. Guided by this mechanistic understanding, we propose a training objective that explicitly optimizes preference while preserving utility, and experimentally demonstrate that it achieves superior performance.

Our contributions are as follows:

- **Unified View.** We propose a unified view of *dynamic weight updates* that casts local weight fine-tuning, parameter-efficient fine-tuning (e.g., LoRA), and activation interventions (steering) into a common intervention form. Building on this view, we introduce a unified preference–utility analysis and show that, across methods instantiated within the dynamic-update framework, both preference and utility exhibit consistent regularities as control strength varies.

- **Preference–Utility Analysis.** We introduce an *activation manifold* hypothesis and analyze preference and utility under this assumption, suggesting that preference is jointly determined by (i) the projection onto a target-preference direction and (ii) activation validity, which degrades as representations deviate from the manifold, while utility degradation is primarily driven by this off-manifold deviation and the resulting activation invalidation. We further derive two quantitative relationships between the preference log-odds and m, and between the utility log-odds and m, and validate them with high-R^2 fits.

- **New Steering Method.** Guided by this mechanism, we propose a training objective that explicitly optimizes preference while preserving utility, and demonstrate experimentally that it achieves better overall performance.

2 Preliminary

2.1 Intermediate Representations in LLMs

During the forward propagation of intermediate layers in LLMs, several key representations occur at specific points in the computation, such as FFN outputs, residual stream states, and linear projections within the attention mechanism (Query, Key, Value, and final output). Ignoring the effects of Layer Normalization[1], these representations can be uniformly expressed as the output of an affine transformation:

$$\mathbf{h}_{i+1} = \mathbf{W}\mathbf{h}_i + \mathbf{b}, \tag{2}$$

where $\mathbf{h}_i$ and $\mathbf{h}_{i+1}$ denote the input and output representations of a linear layer, and $\mathbf{W}$, $\mathbf{b}$ are its weights and biases.

For example, in an FFN block, the up-projection is computed as $\mathbf{h}_{\text{mid}} = \mathbf{W}_{\text{up}}\mathbf{h}_{\text{in}} + \mathbf{b}_{\text{up}}$, followed by a non-linear activation, $\mathbf{h}_{\text{mid,act}} = \sigma(\mathbf{h}_{\text{mid}})$, and then the down-projection is computed as $\mathbf{h}_{\text{out}} = \mathbf{W}_{\text{down}}\,\mathbf{h}_{\text{mid,act}} + \mathbf{b}_{\text{down}}$.

Similarly, the Q, K, V, and output projections in the attention module follow the same affine form as in Eq. 2.

2.2 Parameter Update

We consider two parameter adaptation methods for large language models: Low-Rank Adaptation (LoRA) and local weight fine-tuning.

LoRA LoRA freezes the original weight matrix $\mathbf{W}$ and introduces a trainable low-rank update $\Delta\mathbf{W} = \mathbf{B}\mathbf{A}$, where $\mathbf{B} \in \mathbb{R}^{d\times r}$, $\mathbf{A} \in \mathbb{R}^{r\times k}$, and the rank $r \leq \min(d, k)$. At inference, the adapted weights are given by $\mathbf{W} \leftarrow \mathbf{W} + \Delta\mathbf{W}$. In its canonical form, LoRA applies only to the weight matrix while keeping the bias term $\mathbf{b}$ fixed, although extensions exist that also adapt biases.

Local Weight Fine-tuning Local weight fine-tuning updates parameters within a restricted subset of the network, leaving all other parameters frozen. It can be applied to any layer or parameter type, with full-parameter training representing the special case where the subset covers the entire model. A generic update for the weight matrix $\mathbf{W}$ and bias vector $\mathbf{b}$ can be expressed as: $(\mathbf{W}, \mathbf{b}) \leftarrow (\mathbf{W} + \Delta\mathbf{W},\ \mathbf{b} + \Delta\mathbf{b})$. In our experiments, parameter updates are applied only to the MLP down-projection layer.

2.3 Activation Steering

Activation Steering Activation steering modifies intermediate representations during inference by

[1]Layer Normalization placement varies across architectures; we omit it here for analytical simplicity.

adding a steering vector to selected activations. Its mathematical form can be written as

$$\mathbf{h}_{i+1} = \mathbf{W}\mathbf{h}_i + \mathbf{b} + m\mathbf{v}, \tag{3}$$

where $\mathbf{v}$ is a predetermined direction and m is a scalar coefficient controlling its magnitude. This approach builds on the *linear representation assumption* that abstract concepts correspond approximately to linear subspaces of representation space.

The steering vector $\mathbf{v}$ can be equivalently expressed as a bias adjustment $\Delta\mathbf{b}$, yielding $\mathbf{b} \leftarrow \mathbf{b} + m\Delta\mathbf{b}$. This formulation highlights activation steering as a special case of dynamic parameter update, closely related to methods such as LoRA and local weight fine-tuning.

From a unified perspective, both parameter updates and activation steering operate by injecting a change vector $\Delta\mathbf{h}$ into intermediate representations during forward propagation, differing only in the mechanism by which $\Delta\mathbf{h}$ is generated. More related works can be found in Appendix A.

3 Unified View of Dynamic Weights in Inference

We present a unified framework for dynamic interventions during inference. Our unified view has three components: (i) a unified *measurement* view based on preference/utility log-odds, (ii) a unified *dynamic weights intervention* view that expresses local weight updates, LoRA, and activation steering as dynamic weight updates, and (iii) a unified *dynamics observation* showing consistent preference–utility response patterns across intervention forms.

3.1 Unified Analysis View: Preference and Utility Log-Odds

We analyze intervention effects along two complementary dimensions. **Preference** denotes the model's internal inclination toward a target concept, independent of whether the model completion is well-formed. For the prompt "*Write a short review for this restaurant*", generating "*The food was excellent and the service was wonderful*" indicates a positive preference, while "*The food was terrible and the service was disappointing*" indicates a negative preference. **Utility** denotes the model's task competence that is independent of the target concept. It captures whether the model can produce a task-valid completion that is coherent, relevant to the prompt, and consistent with the requested format. For the same prompt, utility is high when the output is a readable restaurant review, regardless of polarity. Utility is low when the output is incoherent such as "*food food wonderful ??? service 19% ##*", off-topic such as "*Here is a Python script to scrape restaurant data...*", or instruction-violating even if polarity-bearing words appear.

In controlled generation, performance is typically evaluated from the realized outputs. When preference is increased at the expense of utility, completions often become incoherent or instruction-violating, reducing usability and obscuring the intended concept signal under output-based evaluation. Therefore, effective model control should shift preference while preserving utility.

Notation. Given a query q, we construct a polarity pair of completions: a concept-positive answer A_p and a concept-negative answer A_n. We denote their conditional probabilities as $P(A_p \mid q)$ and $P(A_n \mid q)$, and define the corresponding cross-entropy losses as $\mathcal{L}_p \triangleq -\log P(A_p \mid q)$ and $\mathcal{L}_n \triangleq -\log P(A_n \mid q)$. We further introduce latent preference probabilities $P(p_p \mid q)$ and $P(p_n \mid q)$, as well as a polarity-invariant task-success probability (utility) $P(u \mid q)$.

Preference–Utility Factorization. Following prior work that assumes concept directions are mutually orthogonal, we likewise treat concept preference as independent from task utility for a given query q. Concretely, for a polarity pair (A_p, A_n), we decompose

$$\begin{aligned} P(A_p \mid q) &= P(u \mid q)\, P(p_p \mid q), \\ P(A_n \mid q) &= P(u \mid q)\, P(p_n \mid q), \end{aligned} \tag{4}$$

where $P(u \mid q)$ is shared across the pair and $P(p_p \mid q) + P(p_n \mid q) = 1$.

Preference Log-odds. The shared utility cancels in the likelihood ratio, yielding

$$\mathrm{PrefOdds}(q) \triangleq \log \frac{P(p_p \mid q)}{P(p_n \mid q)} = \mathcal{L}_n - \mathcal{L}_p. \tag{5}$$

Utility Log-odds. The total probability mass assigned to the matched pair recovers utility, $P(u \mid q) = P(A_p \mid q) + P(A_n \mid q)$; substituting $P(A \mid q) = e^{-\mathcal{L}}$ gives

$$\begin{aligned} \mathrm{UtilOdds}(q) &\triangleq \log \frac{P(u \mid q)}{1 - P(u \mid q)} \\ &= \log \frac{e^{-\mathcal{L}_p} + e^{-\mathcal{L}_n}}{1 - e^{-\mathcal{L}_p} - e^{-\mathcal{L}_n}}. \end{aligned} \tag{6}$$

Method	Unified Affine Form	Activation Impact ($\Delta\mathbf{h}$)	Param. Size
Fine-tuning Weight	$(\mathbf{W} + m\,\Delta\mathbf{W})\,\mathbf{h}_i + (\mathbf{b} + m\,\Delta\mathbf{b})$	$m\,(\Delta\mathbf{W}\,\mathbf{h}_i + \Delta\mathbf{b})$	$d_{\text{in}} \times d_{\text{out}} + d_{\text{out}}$
LoRA	$(\mathbf{W} + m\,\mathbf{BA})\,\mathbf{h}_i + \mathbf{b}$	$m\,(\mathbf{BA}\,\mathbf{h}_i)$	$d_{\text{in}} \times r + r \times d_{\text{out}}$
Steering Vector	$\mathbf{W}\,\mathbf{h}_i + (\mathbf{b} + m\,\Delta\mathbf{b})$	$m\,\Delta\mathbf{b}$	d_{out}

Table 1: All methods in our unified framework, expressed under the affine weight-update formulation (first unification) and their corresponding activation changes $\Delta\mathbf{h}$ (second unification). d_{in} and d_{out} denote the input and output dimensions of the layer; r is the LoRA rank with $r \ll \min(d_{\text{in}}, d_{\text{out}})$.

We use $\mathrm{PrefOdds}$ and $\mathrm{UtilOdds}$ throughout to track how interventions shift concept preference versus task utility on a common additive scale, with additional derivations in Appendix D.

3.2 Unified Dynamic Weight Formulation

We propose a unified framework that encompasses both parameter update methods and activation steering methods, by viewing them as dynamic weight update. In this formulation, both classes of methods can be expressed under a shared affine transformation view of intermediate representations (see Section 2).

Formally, the dynamic modification of the weight matrix $\mathbf{W}$ and bias vector $\mathbf{b}$ during inference can be written as:

$$\mathbf{h}_{i+1} = (\mathbf{W} + m_1\,\Delta\mathbf{W})\,\mathbf{h}_i + (\mathbf{b} + m_2\,\Delta\mathbf{b}), \quad (7)$$

where $\Delta\mathbf{W}$ and $\Delta\mathbf{b}$ are update terms, and m_1, m_2 are scalar scaling coefficients controlling their magnitudes. In other words, the original parameters are updated to $\mathbf{W}' = \mathbf{W} + m_1\,\Delta\mathbf{W}$ and $\mathbf{b}' = \mathbf{b} + m_2\,\Delta\mathbf{b}$ before computing the next-layer activation.

When a model weight is modified, the effect can be equivalently interpreted from the activation perspective, as a change to the activation at the corresponding position. In this view, diverse intervention methods are unified as adding a change term to the activation:

$$\Delta\mathbf{h} = m_1\,\Delta\mathbf{W}\,\mathbf{h}_i + m_2\,\Delta\mathbf{b}. \quad (8)$$

Under this unified view, local weight fine-tuning, LoRA, and activation steering are all specific instances, differing only in which components are updated: local weight fine-tuning modifies both $\mathbf{W}$ and $\mathbf{b}$; LoRA modifies $\mathbf{W}$ via low-rank factors; activation steering modifies only $\mathbf{b}$. Table 1 summarizes their affine forms, corresponding activation update, and parameter sizes.

Notably, introducing explicit scaling coefficients extends traditional formulations and enables continuous control over perturbation strength, a capability that plays a central role in our subsequent analysis.

3.3 Unified Dynamics Observation

Experimental Setup. We evaluate dynamic interventions on two types of tasks: (i) a personality tendency classification task (*Psychopathy*), and (ii) open-ended generation using *PowerSeeking* and the top 10 concept subsets from *AxBench*. We run experiments on `Gemma-2-9B-IT` and `Qwen-2.5-7B-Instruct`, and consider three intervention types: local weight updates, low-rank adaptation LoRA, and vector-based updates. We train each type using both the language modeling objective and the RePS objective. Additionally, for vector-based updates, we include a train-free method called DiffMean.

Metrics. For each query q with matched answers (A_p, A_n), we compute preference and utility using the log-odds in Eqs. (5) and (6). These metrics allow us to track how preference and utility evolve as we vary the intervention scale m.

Unified Dynamics. Experimental results show that, under the unified perspective framework, different intervention forms exhibit remarkably consistent dynamic patterns. As shown in Figure 2, localized weight updates, low-rank adaptation (LoRA), and vector-based interventions display highly similar overall curve shapes.

For preference log-odds, all methods typically follow a three-stage pattern when plotted against the steering factor m: for small $|m|$, they enter a *Linear Region*, where log-odds grows approximately linearly with m (Bigelow et al., 2025); this is followed by a *Transitional Region* with a noticeable change in trend, and finally a *Convergence Region* where the curve flattens and stabilizes.

Utility log-odds, in contrast, generally peak near $m \approx 0$, and remain near their maximum within this narrow range. As $|m|$ increases, utility gradually declines and eventually stabilizes.

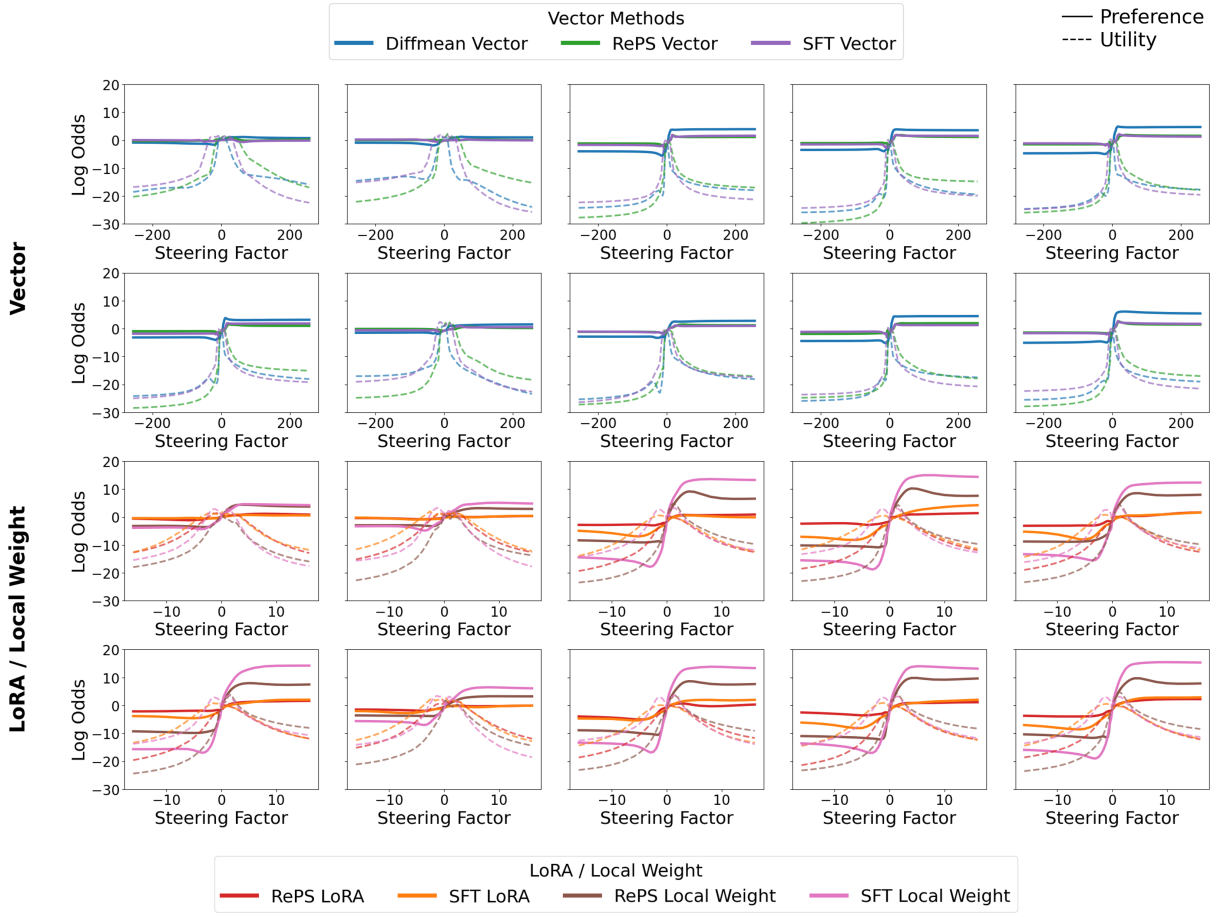

Figure 2: **Unified preference and utility dynamics under steering.** Solid lines represent preference log-odds, and dashed lines represent utility log-odds. The top panel shows steering with vector-form parameter modifications, and the bottom panel shows parametric interventions including LoRA and local weight updates. Results are shown for the `Gemma-2-9B-IT` model on the *AxBench* dataset, evaluated over its top 10 concept subsets. The horizontal axis corresponds to the steering factor.

These patterns reveal a unified steering response of preference and utility.

4 Capability Dynamics: Mechanism Analysis and Optimization

Motivated by the unified preference–utility dynamics observed across intervention forms (Figure 2), this section provides a mechanistic account and an empirical characterization. We take an activation-manifold perspective and introduce a simple validity-decay factor to capture the tendency for capability to degrade as steering pushes activations away from the activation manifold, without committing to a specific underlying geometry. On this basis, we express preference as the combined effect of (i) steering-induced preference projection changes and (ii) validity decay, while utility is modeled as being dominated by the validity decay term. Finally, under this hypothesis we formalize how the steering factor m shapes both preference and utility log-odds, and show via curve-fitting that the resulting forms match the observed log-odds–m dynamics well across settings.

4.1 Activation Manifold Hypothesis

Prior work suggests that model activations often concentrate on low-dimensional, manifold-like sets in representation space (Bricken et al., 2023; Wollschläger et al., 2025). Adopting this manifold perspective, we analyze additive steering as a translation of hidden states along an approximately fixed direction in activation space. Intuitively, small translations may adjust model behavior in a targeted way, whereas large translations may push representations away from the high-density region learned during training, increasing the risk of a representation–decoder mismatch and thus degrading general capability.

We formalize this view with two assumptions.

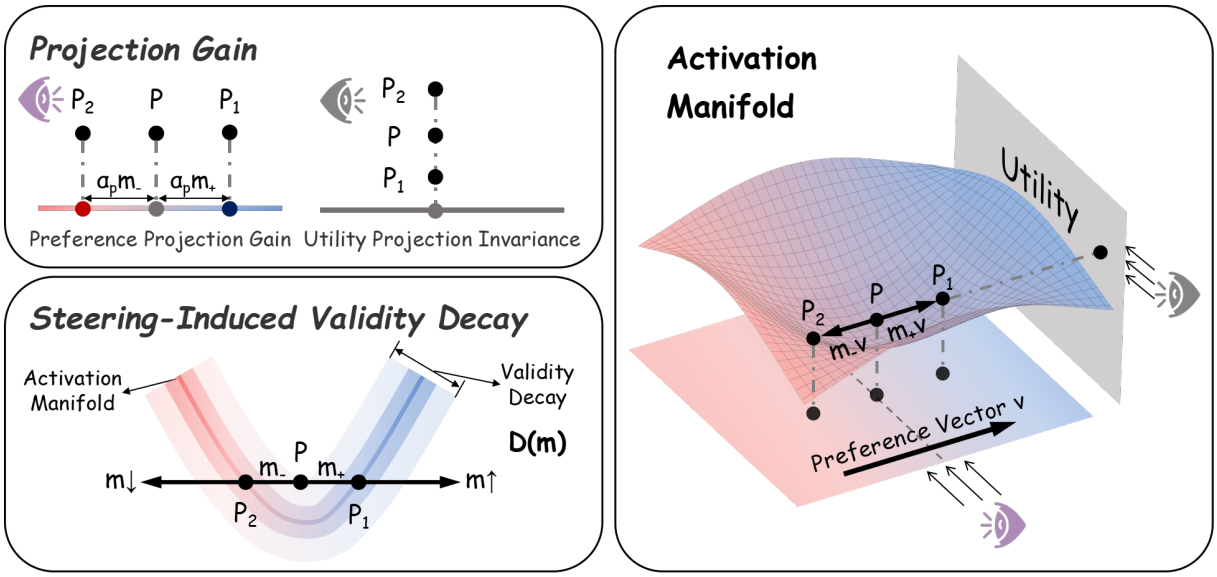

Figure 3: **Mechanism of projection gain and validity decay. Right:** An activation manifold view illustrating Assumption 4.1. An activation P lies on or near the manifold. Steering using preference vector v with scaling factors m_+ and m_- moves P to P_1 and P_2, corresponding to intersections with the manifold. **Top-left:** Projection gain. Projections onto the utility axis exhibit limited variation, whereas projections along the preference direction differ between P_1 and P_2, suggesting that steering primarily influences preference-related components. **Bottom-left:** Steering-induced validity decay. As assumed in Assumption 4.2, increasing steering factor increases off-manifold deviation, leading to a monotonic decrease in validity and degraded downstream decoding.

Assumption 4.1 (Training-Induced Activation Manifold). Fix a layer l with hidden dimension d_l. There exists a low-dimensional set (or its neighborhood) $\mathcal{M}_l \subset \mathbb{R}^{d_l}$ such that for inputs x drawn from a set of *stably handled* inputs $\mathcal{X}_{\text{stable}}$, the corresponding activation $h_l(x)$ lies on or near $\mathcal{M}_l$ with high probability:

$$\Pr_{x \sim \mathcal{X}_{\text{stable}}} [d(h_l(x), \mathcal{M}_l) \leq \epsilon] \geq 1 - \delta, \quad (9)$$

where $d(\cdot, \mathcal{M}_l)$ denotes distance to $\mathcal{M}_l$, $\epsilon > 0$ is a neighborhood radius, and $\delta \in (0, 1)$ is a small failure probability.

Assumption 4.1 asserts that pre-training induces a "typical" region of activation space where representations concentrate for stably handled inputs. We next introduce a generic notion of *representation validity*, which is high near $\mathcal{M}_l$ and decreases as hidden states move away from it. This abstraction avoids committing to a specific geometry for $\mathcal{M}_l$ while retaining the key implication: sufficiently off-manifold activations are more likely to be decoded unreliably by the remaining network.

Assumption 4.2 (Steering-Induced Validity Decay). Let $F_{l \to L}$ denote the remainder of the model from layer l to the output logits. There exists a *validity* function $V_l : \mathbb{R}^{d_l} \to [0, 1]$ that is monotonically non-increasing in $d(h, \mathcal{M}_l)$, capturing how well $F_{l \to L}$ can stably decode an activation h. For an additive steering intervention at layer l,

$$\tilde{h}_l(m) = h_l + m\,\Delta h, \quad (10)$$

with steering direction Δh and steering factor $m \in \mathbb{R}$, define the average validity over stably handled inputs:

$$D(m) \triangleq \mathbb{E}_{x \sim \mathcal{X}_{\text{stable}}} \left[V_l\left(\tilde{h}_l(m)\right) \right]. \quad (11)$$

We assume that $D(m) \in [0, 1]$ decreases with $|m|$ (i.e., larger interventions induce larger off-manifold shifts on average), and that the resulting capability degradation is dominated by this validity decay.

To connect Assumptions 4.1–4.2 to a concrete functional form, we view steering as moving an activation along a one-dimensional line in representation space, $\tilde{h}_l(m) = h_l + m\,\Delta h$. Under the

manifold hypothesis, degradation is governed primarily by how far this line trajectory departs from the typical region near $\mathcal{M}_l$, so it is natural to model $D(m)$ as a smooth function of the (signed) distance along this line to the nearest "on-manifold" locations. In particular, as illustrated in Fig. 3, the steered trajectory may intersect the manifold neighborhood at one or more values $\{m_i\}$ (e.g., one for $m > 0$ and one for $m < 0$). We therefore model validity as being highest near these intersection points and decaying as $|m - m_i|$ grows.

A convenient choice that is positive, smooth, and exhibits heavy-tailed distance-based decay is the rational quadratic (RQ) form, widely used in kernel methods and Gaussian processes to model multi-scale, polynomial-rate attenuation with distance (Rasmussen, 2004). Prior research on controllability metrics has established that model steerability is often asymmetric (Miehling et al., 2025), exhibiting varying degrees of responsiveness along different directions of the same dimension. Motivated by this observation, we employ a piecewise parameterized model to quantify degradation:

$$D(m) = \begin{cases} \left(1 + \frac{(m-m_+)^2}{L_+}\right)^{-p_+} & \text{if } m \geq 0 \\ \left(1 + \frac{(m-m_-)^2}{L_-}\right)^{-p_-} & \text{if } m < 0 \end{cases} \tag{12}$$

where $m_\pm$ corresponds to the signed distance from the original activation point P to an on-manifold intersection point $P_\pm$ along the steering line (Fig. 3); $L_\pm$ sets the characteristic scale of decay and reflects how fast the distance-to-manifold grows along the steering direction (larger when the direction is locally parallel to the manifold and smaller when it cuts across it); and $p_\pm$ controls the decay rate (tail heaviness) as the trajectory moves away from the manifold neighborhood.

4.2 Preference Capability: Projection Gain With Decay

We study how additive steering changes a model's preference through intermediate activations. An intervention at layer l updates the hidden state as $\tilde{h}(m) = h + m\,\Delta h$.

Prior work under LRH-style assumptions often models preference probability with a logistic form, $P(p_p \mid h) = \sigma\big(-(\boldsymbol{\omega}_p^\mathsf{T} h + b_p)\big)$, where $\boldsymbol{\omega}_p$ is the preference vector. Separately, work on activation geometry suggests that after low-dimensional projection (e.g., PCA), opposite preference labels are often approximately linearly separable. Under the activation-manifold view, this motivates a two-dimensional *preference plane* and a preference direction whose signed coordinate reflects *preference intensity*. Our contribution is to incorporate validity attenuation $D(\cdot)$ (Assumption 4.2) to account for off-manifold steering.

To model this, we write the steered preference probability as

$$P(p_p \mid \tilde{h}(m)) = \sigma\Big(-\big(\boldsymbol{\omega}_p^\mathsf{T} h + \alpha_p m + b_p\big)\, D_p(m)\Big), \tag{13}$$

where $\alpha_p \triangleq \boldsymbol{\omega}_p^\mathsf{T} \Delta h$ measures how much the steering direction aligns with the preference vector: α is large when Δh is aligned with $\boldsymbol{\omega}_p$, and $\alpha_p = 0$ when Δh is orthogonal to $\boldsymbol{\omega}_p$.

This implies the preference log-odds

$$\log \frac{P(p_p \mid \tilde{h}(m))}{P(p_n \mid \tilde{h}(m))} = \big(\boldsymbol{\omega}_p^\mathsf{T} h + \alpha_p m + b_p\big)\, D_p(m). \tag{14}$$

Key implication (linear regime → nonlinear collapse). From Eq. (14), the m-dependence enters as $\alpha_p m\, D_p(m)$. When $|m - m^\pm| \ll L^\pm$, Eq. (12) gives $D_p(m) \approx 1$, hence preference log-odds is approximately linear in m with slope α (matching the near-linear regime in Bigelow et al. (2025)). As $|m - m^\pm|$ grows and becomes comparable to or larger than $L^\pm$, Eq. (12) implies substantial decay in $D_p(m)$, so attenuation dominates and the log-odds response becomes strongly nonlinear and can collapse off-manifold.

Fitting Form. We fit the measured preference log-odds as a function of m with

$$\log \frac{P(p_p \mid \tilde{h}(m))}{P(p_n \mid \tilde{h}(m))} = (\alpha m + \beta)\, D(m) + b, \tag{15}$$

where $\beta = \boldsymbol{\omega}_p^\mathsf{T} \mathbf{h} + b_p$ is a per-example constant (since $\mathbf{h}$ is fixed for a given input), and b is an offset.

Fit Results. Table 2 reports the fit quality of Eq. (15), with R^2 values exceeding 0.95 across most settings. These results validate the model's ability to accurately characterize the dynamics of preference log-odds.

4.3 Utility Capability: Only Validity Decay

Utility Log-odds Under Manifold-Validity Decay. Let $\mathbf{h} \in \mathbb{R}^{d_l}$ denote the activation at layer

l. We quantify utility capability by the log-odds of positive vs. negative utility outcomes (u_p/u_n). Similar to preference, we assume utility is also associated with a direction $\boldsymbol{\omega}_u$ in activation space. Under steering $\tilde{h}(m) = h + m\,\Delta h$,, we model

$$\log \frac{P(u_p \mid \tilde{h}(m))}{P(u_n \mid \tilde{h}(m))} \approx -\left(\boldsymbol{\omega}_u^{\mathsf{T}}\mathbf{h} + b_u\right) D_u(m), \tag{16}$$

where $D_u(m)$ follows the manifold-validity decay in Eq. (12) and decreases with $|m|$. Crucially, for *preference* steering directions we typically have $\boldsymbol{\omega}_u^{\mathsf{T}}\Delta\mathbf{h} \approx 0$, so utility is affected primarily through validity decay rather than a direct projection term.

Fitting form. Accordingly, we fit the measured utility log-odds with a pure decay curve:

$$\log \frac{P(u_p \mid \tilde{h}(m))}{P(u_n \mid \tilde{h}(m))} \;=\; \beta\, D_u(m) \;+\; b, \tag{17}$$

where β is the baseline log-odds and b is an offset capturing residual bias.

Fit Results. Table 2 reports the fit quality of Eq. (17). Uniformly high R^2 values (typically > 0.97) suggest utility variations under preference steering are well captured by the proposed formulation. Additional details are in Appendix E.

Type	Method	Preference R^2 ↑				Utility R^2 ↑			
		PSY	PWR	AXB	**Avg**	PSY	PWR	AXB	**Avg**
Gemma-2-9B-IT									
Weight	SFT	0.97	0.98	0.99	0.98	0.98	0.99	0.98	0.98
	RePS	0.99	0.99	0.99	0.99	0.96	0.99	0.99	0.98
LoRA	SFT	0.92	0.99	0.98	0.96	0.98	0.99	0.99	0.99
	RePS	0.83	0.99	0.99	0.94	0.99	0.99	0.99	0.99
Vector	DiffMean	0.97	0.99	0.99	0.98	0.97	0.99	0.98	0.98
	SFT	0.93	0.97	0.98	0.96	0.99	0.99	0.99	0.99
	RePS	0.99	0.98	0.95	0.97	0.99	0.99	0.99	0.99
Qwen-2.5-7B-IT									
Weight	SFT	0.99	0.99	0.99	0.99	0.99	0.99	0.98	0.99
	RePS	0.99	0.99	0.99	0.99	0.99	0.99	0.97	0.98
LoRA	SFT	0.97	0.99	0.99	0.98	0.99	0.99	0.99	0.99
	RePS	0.94	0.99	0.99	0.97	0.98	0.96	0.99	0.98
Vector	DiffMean	0.98	0.95	0.98	0.97	0.99	0.99	0.97	0.98
	SFT	0.93	0.97	0.98	0.96	0.98	0.99	0.99	0.99
	RePS	0.97	0.98	0.93	0.96	0.99	0.98	0.99	0.99

Table 2: **Curve fitting performance.** Results on Psychopathy (PSY), PowerSeeking (PWR), and AXBench (AXB). We report R^2 (higher is better), measuring alignment between theoretical curves and empirical data. Color intensity indicates R^2 values. Consistently dark shading shows high fidelity across settings ($R^2 > 0.95$).

5 Method

5.1 Preference–Utility Joint Optimization

Building on the preceding mechanistic analysis, we propose a concise objective that jointly optimizes a model's *preference* and *utility*. Our goal is to improve preference while delaying utility degradation by extending the effective linear regime of activation intervention.

Utility Loss. To preserve utility, we train on both the positive and negative samples for the same input using the language-modeling cross-entropy:

$$\mathcal{L}_{\text{util}} = \lambda_p\,\mathcal{L}_{\text{p}} + \lambda_n\,\mathcal{L}_{\text{n}}, \tag{18}$$

where $\mathcal{L}_{\text{p}}$ and $\mathcal{L}_{\text{n}}$ are the token-level cross-entropy losses on positive and negative samples, respectively, and λ_p, λ_n control their relative weight.

Preference Loss. By Eq. (5), the loss gap $\mathcal{L}_{\text{n}} - \mathcal{L}_{\text{p}}$ is exactly the preference log-odds. We therefore maximize this gap via a hinge-style margin loss:

$$\mathcal{L}_{\text{pref}} = \gamma \cdot \sigma(\theta - (\mathcal{L}_{\text{n}} - \mathcal{L}_{\text{p}}))\,, \tag{19}$$

where $\sigma(\cdot)$ is ReLU and θ is a margin threshold, and γ trades off preference improvement against utility preservation.

Final Objective. We combine the two components as

$$\mathcal{L} = \mathcal{L}_{\text{util}} + \mathcal{L}_{\text{pref}}. \tag{20}$$

5.2 Experiment Results.

We evaluate the proposed preference-utility joint optimization method under three intervention forms: local weight update, low-rank adaptation (LoRA), and activation vector steering. As shown in Table 3 at Appendix B, our approach consistently achieves higher scores compared with baseline methods across all three intervention types. These results demonstrate the robustness and generality of the proposed optimization strategy.

6 Conclusion

We propose a unified dynamic weight update framework that incorporates parameter updates, LoRA, and activation interventions, revealing a consistent preference–utility decay pattern in the log-odds space. Building on this mechanistic insight, we design a joint optimization method that consistently improves preference while mitigating utility degradation across diverse intervention forms, demonstrating versatility and robustness.

Limitations

While our unified dynamic weight update framework provides a coherent perspective on LLM control and enables predictable preference–utility trade-offs, several limitations remain. First, our analysis assumes that model representations lie near a well-structured activation manifold, which may not hold for extremely large or highly diverse models, potentially reducing the accuracy of our quantitative predictions. Second, our experiments focus primarily on attribute-level control (e.g., sentiment, style), leaving the applicability to complex multi-turn reasoning or safety-critical content largely unexplored. Third, while our proposed training objective mitigates the utility–preference trade-off, it does not guarantee complete avoidance of undesirable side effects such as subtle instruction violations or context drift under extreme control strengths. Finally, our study evaluates control under pre-defined intervention multipliers, and generalization to adaptive or dynamically varying control signals requires further investigation.

Ethics Statement

Controlled LLM generation carries inherent ethical considerations. While our framework aims to improve controllability and preserve task validity, it could potentially be misused to manipulate user perception, amplify biased viewpoints, or generate persuasive yet misleading content. Our experiments are conducted on standard benchmark datasets and do not involve sensitive personal information. We emphasize that the proposed methods should be deployed with human oversight, adherence to fairness guidelines, and robust monitoring to prevent harm. By explicitly modeling preference–utility trade-offs, we aim to make LLM interventions more interpretable and safer, but responsible usage depends on context-aware implementation and alignment with societal norms.

References

Amrita Bhattacharjee, Shaona Ghosh, Traian Rebedea, and Christopher Parisien. 2024. Towards inference-time category-wise safety steering for large language models. *CoRR*, abs/2410.01174.

Eric J. Bigelow, Daniel Wurgaft, YingQiao Wang, Noah D. Goodman, Tomer D. Ullman, Hidenori Tanaka, and Ekdeep Singh Lubana. 2025. Belief dynamics reveal the dual nature of in-context learning and activation steering. *CoRR*, abs/2511.00617.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. 2024. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *CoRR*, abs/2203.06904.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495. Association for Computational Linguistics.

Chi Han, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Nan Jiang, Tarek Abdelzaher, and Heng Ji. 2024. Word embeddings are steers for language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16410–16430.

Peixuan Han, Cheng Qian, Xiusi Chen, Yuji Zhang, Denghui Zhang, and Heng Ji. 2025. Internal activation as the polar star for steering unsafe llm behavior. *arXiv preprint arXiv:2502.01042*.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. Lora+: Efficient low rank adaptation of large models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Rima Hazra, Sayan Layek, Somnath Banerjee, and Soujanya Poria. 2024. Safety arithmetic: A framework for test-time safety alignment of language models by steering parameters and activations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 21759–21776. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022a. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022b. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Guojun Ma, Mingyang Wan, Xiang Wang, Xiangnan He, and Tat-seng Chua. 2025. Anyedit: Edit any knowledge encoded in language models. *arXiv preprint arXiv:2502.05628*.

Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik Opitz, and Tobias Hecking. 2024. Style vectors for steering generative large language models. In *Findings of the Association for Computational Linguistics: EACL 2024, St. Julian's, Malta, March 17-22, 2024*, pages 782–802. Association for Computational Linguistics.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. 2024. Vera: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Tianhong Li and Kaiming He. 2025. Back to basics: Let denoising generative models denoise. *arXiv preprint https://arxiv.org/pdf/2511.13720*.

Yuren Mao, Yuhang Ge, Yijiang Fan, Wenyi Xu, Yu Mi, Zhonghao Hu, and Yunjun Gao. 2025. A survey on lora of large language models. *Frontiers of Computer Science*, 19(7):197605.

Erik Miehling, Michael Desmond, Karthikeyan Natesan Ramamurthy, Elizabeth M. Daly, Kush R. Varshney, Eitan Farchi, Pierre Dognin, Jesus Rios, Djallel Bouneffouf, Miao Liu, and Prasanna Sattigeri. 2025. Evaluating the prompt steerability of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7874–7900, Albuquerque, New Mexico. Association for Computational Linguistics.

Tomás Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 746–751. The Association for Computational Linguistics.

Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. 2025. The origins of representation manifolds in large language models. *CoRR*, abs/2505.18235.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2023, Singapore, December 7, 2023*, pages 16–30. Association for Computational Linguistics.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Joris Postmus and Steven Abreu. 2024. Steering large language models using conceptors: Improving addition-based activation engineering. *CoRR*, abs/2410.16314.

Nate Rahn, Pierluca D'Oro, and Marc G. Bellemare. 2024. Controlling large language model agents with entropic activation steering. *CoRR*, abs/2406.00244.

Carl Edward Rasmussen. 2004. *Gaussian Processes in Machine Learning*, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15504–15522. Association for Computational Linguistics.

Daniel Scalena, Gabriele Sarti, and Malvina Nissim. 2024. Multi-property steering of large language models with dynamic activation composition. *CoRR*, abs/2406.17563.

Daniel Tan, David Chanin, Aengus Lynch, Dimitrios Kanoulas, Brooks Paige, Adrià Garriga-Alonso, and Robert Kirk. 2024. Analyzing the generalization and reliability of steering vectors. *CoRR*, abs/2407.12404.

Curt Tigges, Curt Tigges, Oskar Hollinsworth, Curt Tigges, Atticus Geiger, Atticus Geiger, Oskar Hollinsworth, Neel Nanda, Neel Nanda, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. *http://arxiv.org/abs/2310.15154*.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Activation addition: Steering language models without optimization. *arXiv e-prints*, pages arXiv–2308.

Teun van der Weij, Massimo Poesio, and Nandi Schoots. 2024. Extending activation steering to broad skills and multiple behaviours. *CoRR*, abs/2403.05767.

Mengru Wang, Ziwen Xu, Shengyu Mao, Shumin Deng, Zhaopeng Tu, Huajun Chen, and Ningyu Zhang. 2025. Beyond prompt engineering: Robust behavior control in llms via steering target atoms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 23381–23399. Association for Computational Linguistics.

Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann, and Johannes Gasteiger. 2025. The geometry of refusal in large language models: Concept cones and representational independence. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2025a. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *arXiv preprint arXiv:2501.17148*.

Zhengxuan Wu, Qinan Yu, Aryaman Arora, Christopher D. Manning, and Christopher Potts. 2025b. Improved representation steering for language models. *CoRR*, abs/2505.20809.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Liang Zhao, Shangyan Zhou, Zhean Xu, Zhengyan Zhang, Wangding Zeng, Shengding Hu, Yuqing Wang, Jingyang Yuan, Lean Wang, and Wenfeng Liang. 2025. mhc: Manifold-constrained hyper-connections. *arXiv preprint https://arxiv.org/pdf/2512.24880*.

Wanli Yang, Fei Sun, Rui Tang, Hongyu Zang, Du Su, Qi Cao, Jingang Wang, Huawei Shen, and Xueqi Cheng. 2025. Fine-tuning done right in model editing. *CoRR*, abs/2509.22072.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–9. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Bojia Zi, Xianbiao Qi, Lingzhi Wang, Jianan Wang, Kam-Fai Wong, and Lei Zhang. 2023. Delta-lora: Fine-tuning high-rank parameters with the delta of low-rank matrices. *CoRR*, abs/2309.02411.

A Related Work

Mechanism. Most activation steering methods assume linear structure in activation space, controlling concepts by adding scaled direction vectors to hidden states (Mikolov et al., 2013; Pennington et al., 2014; Nanda et al., 2023; Tigges et al., 2023; Park et al., 2024). Building on this view, Bigelow et al. (2025) show that steering yields an approximately linear trend in posterior odds, but mainly in the small-scale regime. Recent studies further report non-monotonic or adverse effects under stronger steering, challenging a naive global linearity assumption (Bricken et al., 2023; Wollschläger et al., 2025). Meanwhile, representation-manifold work provides a complementary geometric lens for understanding steering and its limitations (Modell et al., 2025; Li and He, 2025; Xie et al., 2025).

Activation Steering. Activation steering controls the behavior of large language models (LLMs) by intervening in hidden states during forward propagation, where steering vectors are constructed to manipulate single attributes as well as more complex behavioral targets (Turner et al., 2023; Rimsky et al., 2024; van der Weij et al., 2024; Rahn et al., 2024; Scalena et al., 2024; Tan et al., 2024; Bhattacharjee et al., 2024; Postmus and Abreu, 2024; Konen et al., 2024; Hazra et al., 2024; Han et al., 2025; Jiang et al., 2025).However, recent studies have shown that the coarse-grained nature of activation steering can lead to degradation in model utility (Wang et al., 2025; Wu et al., 2025a). Cao et al. (2024); Wu et al. (2025b) introduce explicit preference learning objectives to optimize activation steering, enabling more precise control.

Parameter-Efficient Fine-Tuning. Parameter-Efficient Fine-Tuning (PEFT), including adapters, prompt tuning, and low-rank adaptation (LoRA), shows that effective adaptation of large language models does not require updating the full set of parameters. LoRA achieves performance comparable to full fine-tuning, suggesting that adaptation is driven by structured weight deltas rather than full parameter updates, with low-rank residuals serving as localized parameter updates (Hu et al., 2022a; Zi et al., 2023; Hayou et al., 2024; Kopiczko et al., 2024). Localized parameter update methods further indicate that knowledge and behavior in LLMs are highly localized, as factual associations can be modified by intervening on a small number of parameters in specific layers (Zaken et al., 2022). Ding et al. (2022) confirms that updating only a subset of parameters suffices for task adaptation with minimal impact on pretrained capabilities. Geva et al. (2021); Yang et al. (2025) propose selectively updating only the MLP layers enables stable knowledge editing and largely preserves the model's original capabilities.

B Experiment Details

We evaluate steering methods on three benchmarks that capture different aspects of controlled generation.The experimental results are evaluated using GPT-41-mini. Table 3 compares different intervention forms, including weight updates, LoRA, and activation vector steering, across two base models. Across all control forms, our method consistently achieves strong or best performance on concept metrics while maintaining competitive harmonic scores. Notably, the advantages of our approach are most pronounced under LoRA and activation vector interventions. In these settings, our method consistently improves concept strength over both SFT and REPS baselines, and achieves the highest or near-highest harmonic scores on AXBENCH, indicating a more favorable balance between preference enforcement and utility preservation. In contrast, gains under full weight updates are comparatively smaller, likely due to the higher expressive capacity of direct parameter modification, but our method remains competitive without sacrificing utility. Overall, these results demonstrate the generality and robustness of the proposed optimization across diverse steering mechanisms.

C List of Mathematical Symbols

The table 4 below lists the important symbols used in this paper.

D Derivations and Implementation Details for Log-Odds

This appendix derives the loss-based forms of Eqs. (5)–(6) from the preference–utility independence assumption, and states how we compute the required sequence losses.

Model	Intervention Form	Method	Psychopathy	PowerSeeking	AXBENCH	
			Acc ↑	Concept ↑	Concept ↑	Harmonic ↑
Gemma-2-9B-IT	Vanilla	Vanilla	50.00	1.87	0.4750	0.4950
	Weight	SFT	**100.00**	3.50	1.6625	1.4538
		REPS	**100.00**	3.39	1.7750	**1.6362**
		Ours	**100.00**	**3.59**	**1.8500**	1.6225
	LoRA	SFT	**100.00**	3.41	1.7625	1.5188
		REPS	99.00	3.44	1.7375	**1.6525**
		Ours	**100.00**	**3.56**	**1.7750**	1.6412
	Vector	DiffMean (No train)	53.00	2.95	1.1625	1.0550
		SFT	97.00	3.30	1.7000	1.4487
		REPS	98.00	3.61	1.7000	1.5550
		Ours	**99.00**	**3.62**	**1.8500**	**1.6475**
Qwen-2.5-7B-IT	Vanilla	Vanilla	50.00	2.24	0.4500	0.4713
	Weight	SFT	97.00	3.53	1.5375	1.1287
		REPS	96.00	3.24	**1.7750**	**1.4125**
		Ours	**98.00**	**3.66**	1.5750	1.3350
	LoRA	SFT	99.00	3.05	1.4875	1.3175
		REPS	**100.00**	3.34	1.4875	1.4013
		Ours	**100.00**	**3.59**	**1.7375**	**1.6362**
	Vector	DiffMean (No train)	55.00	3.17	0.9500	0.9950
		SFT	97.00	3.58	1.5750	1.5800
		REPS	88.00	3.63	1.7375	1.6412
		Ours	**98.00**	**3.65**	**1.8125**	**1.6500**

Table 3: Comparison of different steering methods under three control forms. All methods perform inference-time interventions on hidden representations. DiffMean is a training-free baseline constructed by mean-difference of hidden representations. Best and second-best results are highlighted within each model and intervention form.

D.1 From preference–utility independence to log-odds

For a query q and a polarity pair (A_p, A_n), we assume

$$P(A_p \mid q) = P(u \mid q)\, P(p_p \mid q),$$
$$P(A_n \mid q) = P(u \mid q)\, P(p_n \mid q), \qquad (21)$$

with $P(p_p \mid q) + P(p_n \mid q) = 1$.

Preference log-odds. Taking the ratio of (21) cancels $P(u \mid q)$:

$$\frac{P(A_p \mid q)}{P(A_n \mid q)} = \frac{P(p_p \mid q)}{P(p_n \mid q)}. \qquad (22)$$

Applying $\log(\cdot)$ gives

$$\mathrm{PrefOdds}(q) \triangleq \log \frac{P(p_p \mid q)}{P(p_n \mid q)} = \log \frac{P(A_p \mid q)}{P(A_n \mid q)}. \qquad (23)$$

Using the loss definition $\mathcal{L} \triangleq -\log P(A \mid q)$, we have $P(A \mid q) = e^{-\mathcal{L}}$, and thus

$$\mathrm{PrefOdds}(q) = \log \frac{e^{-\mathcal{L}_p}}{e^{-\mathcal{L}_n}} = \mathcal{L}_n - \mathcal{L}_p, \qquad (24)$$

which matches Eq. (5).

Utility probability and log-odds. Summing (21) and using $P(p_p \mid q) + P(p_n \mid q) = 1$ yields

$$P(A_p \mid q) + P(A_n \mid q) = P(u \mid q) \Big(P(p_p \mid q) + P(p_n \mid q)\Big) = P(u \mid q). \qquad (25)$$

Therefore,

$$\mathrm{UtilOdds}(q) \triangleq \log \frac{P(u \mid q)}{1 - P(u \mid q)} = \log \frac{P(A_p \mid q) + P(A_n \mid q)}{1 - P(A_p \mid q) - P(A_n \mid q)}. \qquad (26)$$

Substituting $P(A \mid q) = e^{-\mathcal{L}(A|q)}$ gives the loss form

$$\mathrm{UtilOdds}(q) = \log \frac{e^{-\mathcal{L}_p} + e^{-\mathcal{L}_n}}{1 - e^{-\mathcal{L}_p} - e^{-\mathcal{L}_n}}, \qquad (27)$$

which matches Eq. (6). Note that since (A_p, A_n) are only two candidate continuations, we typically have $P(A_p \mid q) + P(A_n \mid q) < 1$.

D.2 Computing sequence losses

Let $A = (y_1, \ldots, y_T)$ be a completion (excluding the query/prompt tokens). We compute the sequence negative log-likelihood (cross-entropy loss) under teacher forcing:

$$\begin{aligned}\mathcal{L}(A \mid q) &\triangleq -\log P(A \mid q) \\ &= -\sum_{t=1}^{T} \log P(y_t \mid q, y_{<t}). \end{aligned} \tag{28}$$

We then set $\mathcal{L}_p \triangleq \mathcal{L}(A_p \mid q)$ and $\mathcal{L}_n \triangleq \mathcal{L}(A_n \mid q)$ and plug them into Eqs. (24) and (27).

Length normalization (optional). When A_p and A_n have different lengths, we optionally use the mean loss $\bar{\mathcal{L}}(A \mid q) \triangleq \mathcal{L}(A \mid q)/T$ in place of $\mathcal{L}(A \mid q)$ to reduce length effects. In that case, the corresponding quantities use $e^{-\bar{\mathcal{L}}}$ instead of $e^{-\mathcal{L}}$.

E Fitting Experiment Details

E.1 Fitting Results on Test Set

To further validate our theoretical model, we performed parameterized fitting on the test set using the SLSQP algorithm, strictly enforcing continuity between positive and negative segments at the origin. As shown in Table 5, the direct fitting yielded high goodness-of-fit values ($R^2 > 0.95$) for most methods. This confirms that the steering effect follows a deterministic trajectory predicted by our theory rather than random perturbations, thereby validating the proposed interaction mechanism.

E.2 Analysis of Generalization Ability

Following the validation of our theoretical mechanism, we conducted train-to-test transfer experiments to evaluate the extent to which different methods decouple "concepts" from specific inputs. Theoretical curve parameters were derived solely from training data and applied directly to the test set for prediction (Table 6).

Robust Generalization Most activation-based methods maintained high prediction accuracy on the test set, indicating they successfully extract general, input-agnostic conceptual representations that adhere to universal theoretical laws across samples.

Symbol	Description
Unified Analysis Framework	
m	The steering scalar coefficient.
$\text{PrefOdds}(q)$	Preference log-odds ($\mathcal{L}_n - \mathcal{L}_p$). (5)
$\text{UtilOdds}(q)$	Utility log-odds. (6)
$P(u\|q)$	Latent utility probability.
$P(p_\pm\|q)$	Latent preference probability.
$P(\bullet\|h)$	Equivalent to $P(\bullet\|q)$ as the weights remain unchanged.
$\mathcal{L}_p, \mathcal{L}_n$	Cross-entropy losses corresponding to A_p and A_n.
Mechanistic Manifold Model	
$\mathcal{M}_l$	The activation manifold of stably handled inputs at layer l.
$D(m)$	Average validity decay function.
$m_\pm$	Distance from P to P± along the steering line. 3
$L_\pm$	Characteristic scale of decay.
$p_\pm$	Asymptotic decay rate.
Joint Optimization	
$\mathcal{L}_{util}$	Utility loss component (standard language modeling loss).
$\mathcal{L}_{pref}$	Preference loss component (margin-based hinge loss).
γ	Trade-off coefficient balancing preference improvement and utility preservation.

Table 4: **Notations for Key Concepts.** A summary of the specialized symbols introduced for the unified analysis, mechanistic modeling, and optimization objective.

Type	Method	Preference R^2 ↑				Utility R^2 ↑			
		PSY	PWR	AXB	**Avg**	PSY	PWR	AXB	**Avg**
Gemma-2-9B-IT									
Weight	SFT	0.96	0.96	0.99	0.97	0.98	0.93	0.99	0.97
	RePS	0.95	0.98	0.95	0.96	0.98	0.93	0.99	0.97
LoRA	SFT	0.99	0.99	0.98	0.99	0.98	0.98	0.99	0.98
	RePS	0.99	0.99	0.98	0.99	0.99	0.99	0.99	0.99
Vector	DiffMean	0.89	0.99	0.99	0.96	0.94	0.99	0.98	0.97
	SFT	0.90	0.97	0.97	0.95	0.98	0.99	0.99	0.99
	RePS	0.96	0.98	0.96	0.97	0.96	0.99	0.99	0.98
Qwen-2.5-7B-IT									
Weight	SFT	0.99	0.82	0.99	0.93	0.99	0.99	0.95	0.98
	RePS	0.99	0.89	0.97	0.95	0.98	0.99	0.90	0.96
LoRA	SFT	0.70	0.95	0.98	0.88	0.99	0.99	0.99	0.99
	RePS	0.88	0.95	0.95	0.93	0.98	0.99	0.98	0.98
Vector	DiffMean	0.99	0.99	0.98	0.99	0.97	0.94	0.98	0.96
	SFT	0.99	0.99	0.97	0.98	0.97	0.95	0.99	0.97
	RePS	0.99	0.98	0.93	0.97	0.96	0.96	0.98	0.97

Table 5: **Performance comparison of curve fitting quality on test sets.** We evaluate the models on three datasets: Psychopathy (PSY), PowerSeeking (PWR), and AXBench (AXB).

Type	Method	Preference R^2 ↑				Utility R^2 ↑			
		PSY	PWR	AXB	**Avg**	PSY	PWR	AXB	**Avg**
Gemma-2-9B-IT									
Weight	SFT	0.96	0.85	-4.25	-0.81	0.98	0.98	0.61	0.86
	RePS	0.99	0.98	-1.16	0.27	0.96	0.98	0.73	0.89
LoRA	SFT	0.92	0.99	-0.56	0.45	0.98	0.99	0.96	0.98
	RePS	0.83	0.99	0.74	0.85	0.98	0.99	0.97	0.98
Vector	DiffMean	-0.14	0.99	0.75	0.53	0.97	0.99	0.97	0.98
	SFT	0.90	0.91	0.74	0.85	0.98	0.99	0.99	0.99
	RePS	0.98	0.89	0.65	0.84	0.99	0.99	0.99	0.99
Qwen-2.5-7B-IT									
Weight	SFT	0.99	-0.32	-12.03	-3.79	0.99	-1.33	-3.07	-1.14
	RePS	0.96	0.98	-3.82	-0.63	0.99	0.42	-1.15	0.09
LoRA	SFT	0.97	0.98	-0.40	0.52	0.99	0.99	0.95	0.98
	RePS	0.94	0.99	-0.13	0.60	0.98	0.96	0.96	0.97
Vector	DiffMean	0.86	0.94	0.80	0.87	0.37	0.99	0.97	0.78
	SFT	0.67	0.92	0.71	0.77	0.96	0.99	0.99	0.98
	RePS	0.97	0.93	0.74	0.88	0.99	0.98	0.98	0.98

Table 6: **Generalization ability of curve fitting.** The table reports the R^2 scores where the curves are fitted on the training set and evaluated on the test set across three datasets: Psychopathy (PSY), PowerSeeking (PWR), and AXBench (AXB). Negative values imply that the fitted curves do not generalize well to unseen data.