FIRST-ORDER STEERING: TRANSLATING WEIGHT ADAPTATION INTO ACTIVATION STEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Activation steering exploits interpretable directions in the residual stream to enable inference-time manipulation of model behavior. Composing steering vectors to apply multiple target behaviors simultaneously is important in various fields—including AI alignment and safety—but remains a challenge for existing activation steering methods. In contrast, prior work in model merging shows that target behaviors represented by learned weight adaptations can be combined with high accuracy. A method that translates weight adaptations into activation steering vectors could therefore extend prior work in model merging to generate composable steering vectors that better enable simultaneous inference-time behavioral control. For this, we introduce FIRST-ORDER STEERING, a formulation of activation steering as a first-order approximation of weight update matrices parameterized by a vector $\boldsymbol{s}$, and establish theoretical bounds on the approximation error of first-order steering. We then develop a novel model merging procedure, HERD-MERGING, which minimizes the first-order approximation error terms to enable higher first-order steering accuracy. Together, our method produces steering vectors that control both individual and composed behaviors more accurately than existing activation steering methods. Furthermore, HeRD-Merging matches the performance of conventional model-merging baselines, while producing weight adaptations that admit more accurate first-order steering vectors.

1 INTRODUCTION

Reliable identification of interpretable features in language models poses a central problem in mechanistic interpretability (Kim et al., 2018; Sharkey et al., 2025). Prior work has developed numerous strategies for supervised and unsupervised discovery of interpretable features in large language models (LLMs) (Turner et al., 2024; Shu et al., 2025). A popular class of such methods is activation steering, which finds vector directions in the activation space of a language model that, when added to the residual stream, induce a particular behavior. Various activation steering methods—such as contrastive activation addition (CAA), representation engineering (RepE), sparse autoencoder (SAE) steering, and ODE steering (ODESteer)—have shown promise in exploiting interpretable features in the activations of LLMs to manipulate their behavior (Panickssery et al., 2024; Fang et al., 2026; Zou et al., 2023; Zhao et al., 2026). However, utilizing these methods to control multiple target behaviors simultaneously remains a challenge (Oozeer et al., 2026; Braun et al., 2025). In contrast, prior work in model merging has demonstrated that, using methods such as task arithmetic, TIES, DARE, and KnOTS, multiple target behaviors can be composed when represented as weight adaptations (Ilharco et al., 2023; Prabhakar et al., 2024; Kangaslahti & Alvarez-Melis, 2025; Stoica et al., 2024; Yu et al., 2024; Yadav et al., 2023). Work on RepE has also proposed procedures such as Low-Rank Representation Adaptation, which simulate activation-space interventions through corresponding parameter updates, suggesting a relationship between weight adaptation and activation steering. However, the converse, translating weight updates into activation steering, remains underexplored. Understanding the conditions under which weight adaptation can be converted into activation steering could extend existing model merging methods to activation steering, thus enabling simultaneous and interpretable inference-time control of multiple target behaviors.

Here we show that a differentiable model-merging method parameterized by steering strengths $\boldsymbol{s}$ can be converted into activation steering by taking a first-order approximation of each hidden layer with respect to $\boldsymbol{s}$. Each column of the resulting Jacobian matrix defines a direction in activation space,

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

and these directions can be naturally composed through vector addition. We then show that the Hessian norm of each hidden-layer component bounds the approximation error of first-order steering, providing a theoretical measure of steering accuracy that existing methods lack. From this measure, we develop HERD-MERGING, a differentiable model-merging procedure that minimizes the first-order approximation error terms while preserving conventional model-merging performance. Empirically, first-order steering controls both individual and composed behaviors more accurately than existing activation steering methods, while HeRD-Merging produces weight adaptations that admit more accurate first-order steering vectors.

We make the following contributions:

1. We introduce *first-order steering*, a method for translating differentiable model-merging procedures into activation steering. The Jacobian of each hidden layer with respect to the steering strengths yields composable activation-space directions, while the corresponding Hessian norms provide theoretical bounds on the first-order approximation error.
2. We introduce *HeRD-Merging*, a differentiable model-merging procedure designed to control the error bounds governing first-order steering accuracy. By reducing the relevant Hessian-based error terms, HeRD-Merging produces weight adaptations which, when paired with first-order steering, produce more accurate activation steering vectors.
3. We evaluate first-order steering on both individual and composed behavioral control and compare HeRD-Merging with established model-merging methods. First-order steering outperforms existing activation steering methods, while HeRD-Merging improves first-order steering accuracy and maintains performance comparable to conventional model-merging baselines.

2 PRELIMINARIES

Activation Steering Prior work in mechanistic interpretability has produced several steering methods that induce behaviors through additive updates to the intermediate activations of an LLM. Activation Addition (ActAdd) demonstrated that a single *steering vector*, computed as the difference in residual-stream activations between a pair of contrastive prompts, can be added at a chosen layer to induce a targeted behavioral shift at inference time without any gradient-based training (Turner et al., 2024). At inference time, a steering vector $\boldsymbol{v}$ is applied through

$$\boldsymbol{h}' = \boldsymbol{h} + \alpha \boldsymbol{v} \tag{1}$$

for some steering strength parameter $\alpha \in \mathbb{R}$. A set of steering vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ can then be composed through vector addition so that the aggregate steering vector is $\boldsymbol{v} = \sum_{k=1}^{n} \boldsymbol{v}_k$. Through this principle, numerous strategies have emerged towards computing optimal steering vectors applied through (1). Contrastive Activation Addition (CAA) computes steering vectors from paired positive/negative examples but focuses on a systematic evaluation across many behaviors, showing that the resulting vectors transfer across prompt formats and can be combined with finetuning (Panickssery et al., 2024). Representation Engineering (RepE) generalizes this idea by extracting relevant steering vectors using additional supervised and unsupervised methods such as PCA, k-means clustering, and logistic regression (Zou et al., 2023). Feature Guided Activation Additions (FGAA) leverages sparse-autoencoder features to select and combine directions, aiming to produce more interpretable and disentangled steering vectors than raw activation differences (Soo et al., 2025; Cunningham et al., 2023; Shu et al., 2025). Representation Surgery generalizes existing steering heuristics and reduces the computation of steering functions to a least-squares objective (Singh et al., 2024). Activation Transport (AcT) reframes activation steering as an optimal-transport problem and learns a transport map that moves the distribution of source activations toward a target distribution (Rodriguez et al., 2024).

These strategies have been shown to work in various in-distribution benchmarks, but have also demonstrated inconsistent results in out-of-distribution settings, especially when steering multiple behaviors simultaneously (Scalena et al., 2024; van der Weij et al., 2024; Braun et al., 2025; Tan et al., 2024). Moreover, existing progress in activation steering has only presented marginal improvements in accuracy, limiting its application in practical parameter-efficient finetuning requirements (Apolinario & Roy, 2026; Koike-Akino et al., 2025). More recent methods—such as

ODESteer, MiMiC, RE-Control, and AcT—introduce novel paradigms of activation steering to increase accuracy but deviate in formulation from (1), meaning they lack a nontrivial procedure for composing generated steering vectors (Zhao et al., 2026; Heyman & Vandeputte, 2026; Kong et al., 2024; Rodriguez et al., 2024). In contrast, we show that first-order steering outperforms multiple activation steering paradigms while offering accurate composability of vectors in a manner identical to activation addition.

Model Interpolation and Merging With sufficient data, low-rank adaptation (LoRA) provides an efficient method for finetuning a model (Hu et al., 2021). However, LoRA on its own does not provide a framework for merging or interpolating adapters. Consequently, techniques of *model merging* are commonly applied to combine the capabilities of multiple independently finetuned LoRA adapters. One such technique—task arithmetic—defines *task vectors* as the element-wise difference, $\Delta \boldsymbol{W}$, between finetuned and base model weights. These vectors can then be added or subtracted to compose or suppress behaviors (Ilharco et al., 2023). In particular, for a collection of LoRA adapters $\Delta \boldsymbol{W}_1, \ldots, \Delta \boldsymbol{W}_n$, task arithmetic generates a unified LoRA adapter

$$\Delta \boldsymbol{W}(\boldsymbol{s}) = \sum_{k=1}^{n} s_k \Delta \boldsymbol{W}_k$$

for some $\boldsymbol{s} \in \mathbb{R}^n$. This strategy can accurately produce simultaneous behavioral control but can be improved through various model merging methods. TIES-Merging trims low-magnitude parameters and resolves sign conflicts prior to aggregation (Yadav et al., 2023). DARE sparsifies task vectors further by randomly dropping and rescaling delta parameters (Yu et al., 2024). KnOTS decomposes task vectors via SVD before merging, operating in a shared low-dimensional subspace to reduce cross-task interference and improve the composability of merged models through existing methods like TIES and DARE (Stoica et al., 2024).

Model merging methods demonstrate that weight-space adaptations can be composed while retaining strong per-task performance, providing a natural foundation for constructing composable activation steering vectors using first-order steering. We therefore generalize the update rule of existing model merging methods to develop HeRD-Merging, a differentiable model merging procedure which encourages greater first-order steering performance while matching existing baselines in merging accuracy.

3 First-Order Steering

In this section, we present our first-order steering methodology and establish theoretical bounds on its accuracy. Let f_θ denote a frozen autoregressive transformer with L layers and hidden dimension d, and let $\boldsymbol{W}_p^{\text{base}}$ denote the frozen pretrained weight matrix of f_θ at projection head p. We denote the hidden layer of f_θ at layer ℓ by $\boldsymbol{h}_\ell$, with the ith component denoted $h_{\ell,i}$.

Let n be the number of target behaviors (i.e. the number of LoRA adapters) and let $g : \mathbb{R}^{d+n} \to \mathbb{R}^d$, with evaluation denoted $g(\boldsymbol{x}, \boldsymbol{s})$ for vectors $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{s} \in \mathbb{R}^n$. Then $J_{\boldsymbol{s}} g$ (resp. $J_{\boldsymbol{x}} g$) denotes the Jacobian matrix resulting from viewing g as a function with $\boldsymbol{x}$ (resp. $\boldsymbol{s}$) fixed. Similarly, for scalar-valued g_i, Hg_i denotes the Hessian and $H_{\boldsymbol{s}} g_i$ the Hessian with $\boldsymbol{x}$ fixed.

Let $\mathcal{P}$ be the set of projection head indices and let $\Delta \boldsymbol{W}_{1,p}, \ldots, \Delta \boldsymbol{W}_{n,p}$ be LoRA adapters applied at $p \in \mathcal{P}$. Then task arithmetic defines a parameterized update

$$\Delta \boldsymbol{W}_p(\boldsymbol{s}) = \sum_{k=1}^{n} s_k \Delta \boldsymbol{W}_{k,p}, \qquad \boldsymbol{W}_p(\boldsymbol{s}) = \boldsymbol{W}_p^{\text{base}} + \Delta \boldsymbol{W}_p(\boldsymbol{s}). \tag{2}$$

for some $\boldsymbol{s} \in \mathbb{R}^n$. Henceforth, we omit the indexing by p for readability when it is not required. Let f_θ^{TA} be the transformer with the task arithmetic updates added at each $p \in \mathcal{P}$, and let $\boldsymbol{h}'_\ell$ be the hidden layer of f_θ^{TA} at layer ℓ, with the ith component denoted $h'_{\ell,i}$. Under this formulation, $\boldsymbol{h}'_\ell$ admits the first-order approximation

$$h'_{\ell,i}(\boldsymbol{x}, \boldsymbol{s}) = h_{\ell,i}(\boldsymbol{x}) + \nabla_{\boldsymbol{s}} h'_{\ell,i}(\boldsymbol{x}, 0)^\top \boldsymbol{s} + r_i(\boldsymbol{s}),$$

where a bound on the remainder $\boldsymbol{r}$ is given by Taylor's theorem

$$\|\boldsymbol{r}(\boldsymbol{s})\| \leq \frac{1}{2} \sum_{i=1}^{d} \sup_{\boldsymbol{s}^* \in [\boldsymbol{0}, \boldsymbol{s}]} \left\| H_{\boldsymbol{s}} h'_{\ell,i}(\boldsymbol{x}, \boldsymbol{s}^*) \right\|_F \|\boldsymbol{s}\|^2, \tag{3}$$

with $H_{\boldsymbol{s}} h'_{\ell,i}$ denoting the Hessian matrix of $h'_{\ell,i}$ with respect to $\boldsymbol{s}$, $\|\cdot\|_F$ denoting the Frobenius norm for matrices, and $[\mathbf{0}, \boldsymbol{s}] \subseteq \mathbb{R}^n$ denoting the line segment from $\mathbf{0}$ to $\boldsymbol{s}$. Thus, with a sufficiently small Hessian of $h'_{\ell,i}(\boldsymbol{x}, \boldsymbol{s})$, the inner product $\nabla_{\boldsymbol{s}} h'_{\ell,i}(\boldsymbol{x}, 0)^\top \boldsymbol{s}$ yields activation addition terms

$$h'_{\ell,i}(\boldsymbol{x}, \boldsymbol{s}) \approx h_{\ell,i}(\boldsymbol{x}, \boldsymbol{s}) + \sum_{k=1}^{n} \frac{\partial h'_{\ell,i}}{\partial s_k}(\boldsymbol{x}, 0) s_k. \tag{4}$$

This is an instance of (1), besides the dependence of the activation addition term on the residual input $\boldsymbol{x}$. To remove this dependency, we compute steering vectors for each target behavior optimal over some dataset of M activation samples $\mathcal{D}$ so that $\boldsymbol{A}^* = \arg\min_{\boldsymbol{A}} \frac{1}{M} \sum_{m=1}^{M} \|J_{\boldsymbol{s}} \boldsymbol{h}'_\ell(\boldsymbol{x}_m, 0) - \boldsymbol{A}\|_F^2$. The analytical solution to this least-squares problem is given by the mean

$$\boldsymbol{A}^* = \frac{1}{M} \sum_{m=1}^{M} J_{\boldsymbol{s}} \boldsymbol{h}'_\ell(\boldsymbol{x}_m, 0). \tag{5}$$

This formulation has the following advantages: (i) steering vectors individual to each target behavior can be read out directly from the columns of $\boldsymbol{A}^*$; (ii) steering vector composition is given by summation through the matrix product $\boldsymbol{A}^* \boldsymbol{s}$; (iii) the derivation of steering vectors from LoRA adapters allows for the simultaneous computation of steering vectors for any layer where an adapter is placed.

We now bound the error accumulated from (4) and (5). Intuitively, controlling the variance of $J_{\boldsymbol{s}} \boldsymbol{h}'_\ell$ with respect to $\boldsymbol{x}$ over some dataset reduces the dependency on the prompt and therefore should reduce the least squares error. This can be motivated through the following propositions.

Proposition 1 (Gaussian Poincaré inequality)**.** *Let $\mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_a)$ be a vector of independent standard Gaussian random variables. If $\boldsymbol{g} : \mathbb{R}^a \to \mathbb{R}^b$ is smooth, then*

$$\|\mathrm{Cov}(\boldsymbol{g}(\mathbf{a}))\|_F \leq \mathbb{E}[\|J\boldsymbol{g}(\mathbf{a})\|_F^2].$$

The scalar case $\mathrm{Var}(g(\mathbf{a})) \leq \mathbb{E}\|\nabla g(\mathbf{a})\|^2$ is stated in (Chatterjee, 2007). The form above follows by applying it to each coordinate of $\boldsymbol{g}$ and using $\|\mathrm{Cov}(\boldsymbol{g}(\mathbf{a}))\|_F \leq \mathrm{trace}(\mathrm{Cov}(\boldsymbol{g}(\mathbf{a})))$ for positive semidefinite matrices. We now relate the Jacobian to the least squares error (6).

Proposition 2. *Let $S \subseteq \mathbb{R}^{d+n}$ be an open rectangle containing $\mathcal{D} \times \{0\} \subseteq \mathbb{R}^{d+n}$ and define $Q = \sup_{(\boldsymbol{x}_0, \boldsymbol{s}_0) \in S} \|J_{\boldsymbol{x}} J_{\boldsymbol{s}} \boldsymbol{h}'_\ell(\boldsymbol{x}_0, \boldsymbol{s}_0)\|_F$ and $L = \sup_{\boldsymbol{x}_a, \boldsymbol{x}_b \in \mathcal{D}} \|\boldsymbol{x}_a - \boldsymbol{x}_b\|$. Then*

$$\frac{1}{M} \sum_{m=1}^{M} \|J_{\boldsymbol{s}} \boldsymbol{h}'_\ell(\boldsymbol{x}_m, 0) - \boldsymbol{A}^*\|_F^2 \leq \frac{L^2 Q^2}{2}. \tag{6}$$

Since $Q \leq \sup_{(\boldsymbol{x}, \boldsymbol{s}) \in S} \sum_{i=1}^{d} \|H h'_{\ell,i}(\boldsymbol{x}, \boldsymbol{s})\|_F$, the same Hessian norm that bounds the remainder in (3) also bounds Q. Regularizing the Hessian norm therefore simultaneously manages the first-order approximation error and the least squares error, suggesting a quantifiable signal of steering accuracy under our methodology.

4 Hessian Regularized Differentiable Merging (HeRD-Merging)

In this section, we present a differentiable model-merging procedure that controls the size of the bound (3) and allows for the minimization of $\|J_{\boldsymbol{x}} J_{\boldsymbol{s}} \boldsymbol{h}'_\ell(\boldsymbol{x}, \boldsymbol{s})\|_F$ as in Proposition 2. In Section 5, we show that this effort leads to improved steering performance while retaining merging performance.

4.1 The HeRD Adapter Function

Let $\Delta \boldsymbol{W}_1, \ldots, \Delta \boldsymbol{W}_n$ be a collection of LoRA adapters and let $\mathcal{C} = \{S \subseteq \{1, \ldots, n\} : |S| \geq 2\}$. Then a *HeRD adapter* is a function

$$\Delta \boldsymbol{W}(\boldsymbol{s}) = \sum_{k=1}^{n} s_k \Delta \boldsymbol{W}_k + \sum_{S \in \mathcal{C}} \eta_S \left(\prod_{t \in S} s_t \boldsymbol{R}_S \right), \tag{7}$$

where $\boldsymbol{R}_S = \frac{1}{r_S} \boldsymbol{B}_S \boldsymbol{A}_S$ for some chosen rank r_S and $\eta_S, \boldsymbol{R}_S$ are learnable parameters with $0 \leq \eta_S \leq \eta_{\max}$. The second term adds a scaled corrective low-rank matrix $\boldsymbol{R}_S$ weighted by the intensity

of each interaction $\prod_{t \in S} s_t$. Fixing $\eta_S = 0$ recovers task arithmetic as a special case. Evidently, this makes $\Delta \boldsymbol{W}$ smooth in $\boldsymbol{s}$. Moreover, writing $\boldsymbol{R}_S$ with a low-rank factorization allows for a constrained maximum rank of the update $\Delta \boldsymbol{W}(\boldsymbol{s})$. Let f_θ^{HeRD} be the model f_θ with the HeRD adapters $\Delta \boldsymbol{W}_p(\boldsymbol{s})$ at each projection head index $p \in \mathcal{P}$.

Optimizing the matrices $\boldsymbol{R}_S$ for controlling the Hessian norms is computationally expensive, as it would require materializing multiple mixed Jacobian matrices through each transformer layer. Instead, we first fix $\eta_S = 1$ for all S and train the matrices $\boldsymbol{R}_S$ to encourage merging accuracy using gradient descent. Then, we fix the matrices $\boldsymbol{R}_S$ and compute the optimal η_S, which reduces the problem to a nonlinear least squares objective. This penalizes the Hessian norm without computing any derivatives with respect to the matrices $\boldsymbol{R}_S$.

4.2 Distribution Loss by Gradient Descent

Fix an autoregressive context $\boldsymbol{c} = (\boldsymbol{x}, \boldsymbol{y}_{<t})$ and let $\mathcal{V}$ be the vocabulary of the transformer $\boldsymbol{f}_\theta^{\mathrm{HeRD}}$. We define $p_{\boldsymbol{s}}(\boldsymbol{v} \mid \boldsymbol{c})$ as the next-token probability assigned to $\boldsymbol{v} \in \mathcal{V}$ given context $\boldsymbol{c}$ and steering strengths $\boldsymbol{s}$. In particular, let $p_0(\boldsymbol{v} \mid \boldsymbol{c})$ and $p_k(\boldsymbol{v} \mid \boldsymbol{c})$ denote the next-token probabilities for $\boldsymbol{s} = \boldsymbol{0}$ and $\boldsymbol{s} = \boldsymbol{e}_k$ respectively. Then we define the interpolation of the distributions p_k at $\boldsymbol{s}$ as

$$\log q_{\boldsymbol{s}}(\boldsymbol{v} \mid \boldsymbol{c}) = \log p_0(\boldsymbol{v} \mid \boldsymbol{c}) + \sum_{k=1}^{n} s_k(\log p_k(\boldsymbol{v} \mid \boldsymbol{c}) - \log p_0(\boldsymbol{v} \mid \boldsymbol{c})).$$

Through normalization by the partition function $Z_{\boldsymbol{s}}(\boldsymbol{c}) = \sum_{\boldsymbol{v} \in \mathcal{V}} q_{\boldsymbol{s}}(\boldsymbol{v} \mid \boldsymbol{c})$, we can form $q_{\boldsymbol{s}}$ into a probability measure $Q_{\boldsymbol{s}}(\boldsymbol{v} \mid \boldsymbol{c}) = q_{\boldsymbol{s}}(\boldsymbol{v} \mid \boldsymbol{c})/Z_{\boldsymbol{s}}(\boldsymbol{c})$. From interpolation, this probability measure has the property that $Q_{\boldsymbol{e}_k} = p_k$ and $Q_0 = p_0$. We can now define the distribution-matching objective of our merging procedure over some grid $G \subseteq [0,1]^n$ as

$$\mathcal{L}_{\text{dist}}(R) = \mathbb{E}_{\boldsymbol{c} \sim \mathcal{D}} \mathbb{E}_{\boldsymbol{s} \sim U(G)}[KL(Q_{\boldsymbol{s}}(\cdot \mid \boldsymbol{c}), p_{\boldsymbol{s}}(\cdot \mid \boldsymbol{c}))],$$

where $U(G)$ denotes the uniform distribution over the set G and $\mathcal{D}$ is some dataset of prompts. This objective can then be minimized through gradient descent for all layers simultaneously.

4.3 Hessian Regularization by the Levenberg-Marquardt Algorithm

Subsequently, we fix the matrices $\boldsymbol{R}_S$ in (7) and optimize $\boldsymbol{\eta} = (\eta_S)_{S \in \mathcal{C}}$ by constructing a residual vector of objectives $r(\boldsymbol{\eta})$ and minimizing the nonlinear least squares objective

$$\min_{\boldsymbol{\eta} \in [0, \eta_{\max}]^{|\mathcal{C}|}} \frac{1}{2} \|\boldsymbol{r}(\boldsymbol{\eta})\|_2^2 \tag{8}$$

with the Levenberg-Marquardt Algorithm (LMA) (Moré, 1978). For each $\boldsymbol{\eta}$, we define residuals $r_1(\boldsymbol{\eta}), r_2(\boldsymbol{\eta}), r_3(\boldsymbol{\eta}), r_4(\boldsymbol{\eta})$ and define $\boldsymbol{r}(\boldsymbol{\eta}) = \sqrt{2}[\sqrt{r_1(\boldsymbol{\eta})}\ \sqrt{r_2(\boldsymbol{\eta})}\ \sqrt{r_3(\boldsymbol{\eta})}\ \sqrt{r_4(\boldsymbol{\eta})}]$.

Distribution-matching. We retain the distribution-matching objective of the previous stage by defining the residual $r_1 = \mathrm{KL}(Q_{\boldsymbol{s}}(\cdot \mid \boldsymbol{c}), p_{\boldsymbol{s}}(\cdot \mid \boldsymbol{c}))$. This prevents divergence from distribution alignment meant to be achieved in the previous stage of training.

Hessian regularization. By the linearity of the derivative, it does not suffice to merely add the Jacobian matrix norm as a residual. Instead, we optimize the size of the mixed Jacobian matrix relative to the Jacobian matrix. Let $\boldsymbol{b} \sim U(G)$ uniformly and let $\boldsymbol{s}^*$ be the unit vector constructed by uniformly sampling from $\{1, -1\}^n$ and then scaling by $\frac{1}{\sqrt{n}}$. We define the residual

$$r_2 = \frac{\|J_{\boldsymbol{s}}(J_{\boldsymbol{s}} \boldsymbol{h}_\ell(\boldsymbol{x}, \boldsymbol{b}) \cdot \boldsymbol{s}^*) \cdot \boldsymbol{s}^*\|}{\|J_{\boldsymbol{s}} \boldsymbol{h}_\ell(\boldsymbol{x}, \boldsymbol{b}) \cdot \boldsymbol{s}^*\|}.$$

Furthermore, the mixed partial derivatives with respect to $\boldsymbol{x}$ can be directly controlled through the total variance over a dataset $\mathcal{D}$ of activations produced by various prompts as

$$r_3 = \frac{\|\mathrm{Cov}_{\boldsymbol{x} \sim \mathcal{D}}[J_{\boldsymbol{s}} \boldsymbol{h}_\ell(\boldsymbol{x}, \boldsymbol{b}) \cdot \boldsymbol{s}^*]\|_F}{\mathbb{E}[\|J_{\boldsymbol{s}} \boldsymbol{h}_\ell(\boldsymbol{x}, \boldsymbol{b}) \cdot \boldsymbol{s}^*\|^2]}.$$

By (3) and Proposition 2, r_2 and r_3 work to cheaply penalize approximations of relevant blocks of $\|H h_{\ell,i}\|_F$.

Correction term magnitude. The parameters η_S also serve to control the magnitude of the matrices $\boldsymbol{R}_S$ relative to the LoRA adapters. So, we add the residual

$$r_4 = \sum_{S \in \mathcal{C}} \frac{\eta_S \|\boldsymbol{R}_S\|_F^2}{\sum_{k=1}^{n} \|\Delta \boldsymbol{W}_k\|_F^2}.$$

This allows $\boldsymbol{\eta}$ to simultaneously prevent the matrices from overcorrecting the learned behavioral effect of the LoRA adapters. Since each residual component is dependent on $\boldsymbol{\eta}$, this supplies the function $\boldsymbol{r}(\boldsymbol{\eta})$ for the nonlinear least-squares problem (8). Each minibatch of the LMA resamples the JVP direction $\boldsymbol{s}^*$ to increase the coverage in the interior of $[0, 1]^n$.

5 Experiments

Our methodology aims to facilitate the computation of accurate and composable steering vectors across multiple target behaviors and models. First-order steering offers a framework for computing these steering vectors, while HeRD-Merging is constructed to produce more accurate steering vectors when paired with first-order steering. In this section, we evaluate first-order steering against existing activation steering methods and demonstrate that HeRD-Merging improves first-order steering performance relative to task arithmetic. As a secondary result, we also show that HeRD-Merging remains competitive with existing model merging approaches, supporting its practical use as an intermediate between a collection of LoRA adapters and their corresponding first-order steering vectors.

5.1 Setup and Training Details

To evaluate our techniques jointly, we train LoRA adapters on steering directions which are semantically independent of each other so that the accuracy of merging adapters and steering vectors can be tested meaningfully. The directions should also be a reasonably established use case of activation steering, so that they can be benchmarked against other steering methods. For this, we choose three distinct behavioral manipulations in the model's output style:

1. **Analogy Usage:** The model is trained to answer questions by constructing a heavy, overly complicated analogy and responding in the context of this analogy.
2. **Bulletpoint Usage:** The model is trained to format all of its replies in bulletpoints, irrespective of how appropriate it is for the particular user query.
3. **Sophisticated Language:** The model is trained to speak in highly sophisticated, exaggerated, and archaic language.

Because the model can measurably exhibit any combination of analogy use, bulletpoint use, and sophisticated language, these directions have meaningful semantic composability. They are also sufficiently complex for both activation steering and model merging, while being in line with the benchmarking methods of prior works in activation steering (Diallo et al., 2025). To train the models, we first synthetically generate data for all of the directions using GPT-5.5. Generation proceeds by providing the model with a user conversation from the UltraChat dataset (Ding et al., 2023) and instructing GPT-5.5 to re-generate model responses to user prompts according to the desired behavioral direction. The resulting datasets each consist of approximately $10,000$ samples. Examples of synthetically generated outputs for each target behavior are available in Section C.3.

We report our experimental results for three model families of varying sizes. We run matched experiments on Qwen2.5-VL-3B, Gemma-4-E2B, and Mistral-7B-Instruct-v0.3 (Bai et al., 2025; Abd et al., 2026; Jiang et al., 2023). We train a set of adapters for each behavioral direction independently. For each model, we freeze the base model and attach LoRA adapters to each projection head of the model with LoRA rank 8. Adapters are trained using the Adam optimizer (Kingma & Ba, 2017) for 100 steps with a minibatch size of 2, 16 gradient accumulation steps, and a sequence length of 512.

5.2 DeBERTa-v3 Classifiers

One advantage of the chosen steering directions is that they are suitable for computationally inexpensive evaluation by a classification model. Because conversations in our dataset are generated

from UltraChat, we use those as labeled negative examples and the synthetically generated conversations as labeled positive examples. We split this dataset into 80% training, 10% validation, and 10% test. The training data is then used to finetune the DeBERTa-v3-base model, which has shown reliable agreement with human evaluators on classification tasks once trained (He et al., 2023; Li et al., 2024). Due to the lexical distinctness of each steering direction, the classifiers achieve near-perfect accuracy on the test sets (containing 1994 samples). We also compute thresholds for binary classification using the validation accuracy as the heuristic. Detailed classifier results are reported in Section B.2.

5.3 Activation Steering Experiments

In principle, first-order steering can be applied at all layers where there are weight updates. This is advantageous because it can lead to greater performance and higher expressivity. However, because baseline activation steering methods exhibit output degradation when concurrently applied at too many model layers, we conservatively follow the evaluation approaches common in prior work and identify a single common layer reasonable to use for all of the methods (Zhao et al., 2026; Sharma & Trivedi, 2026). For each target behavior k, we maximize the normalized mean activation difference generated by contrastive activation distributions $\mathcal{D}_k^+, \mathcal{D}_k^-$ given by

$$S_{\ell,k} = \frac{\left\| \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_k^+}[h_\ell(x)] - \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_k^-}[h_\ell(x)] \right\|^2}{\sum_i \mathrm{Var}_{\boldsymbol{x}\sim\mathcal{D}_k^+}(h_{\ell,i}(x)) + \sum_i \mathrm{Var}_{\boldsymbol{x}\sim\mathcal{D}_k^-}(h_{\ell,i}(x))}$$

for each layer ℓ. We then choose the optimal layer by $\ell_k^* = \arg\max_\ell S_{\ell,k}$. We train HeRD adapters at that layer by sampling from a uniformly spaced grid $G \subseteq [0,1]^n$ and training the matrices $\boldsymbol{R}_S$ for 100 gradient descent steps using the Adam optimizer (Kingma & Ba, 2017). The LMA is then performed with a mini-batch size of 1, $\eta_{\max} = 1$, and an initial damping parameter of $\mu_0 = 0.1$ for 12 steps. We then compile a separate set of test prompts independent of the synthetically generated datasets used to train the LoRA adapters. These test prompts are used to produce 100 outputs from the model for each of the 2^n corners of $[0,1]^n$, resulting in 800 outputs per steering method. These outputs are then evaluated by the DeBERTa-v3 classifier to produce behavioral adherence rates for the merged models. We also report confidence intervals using non-parametric percentile bootstrapping on 2000 prompt-level resamples, computing the accuracies for all 8 combinations of steering directions with each resample. Finally, we take the mean of the Jacobian in the first-order approximation over user prompts from the UltraChat training dataset to obtain the steering vector from first-order steering.

Table 1: Overall behavioral adherence (%) of activation steering methods, averaged across all three models. Steering is applied at a single common layer for all methods. Adherence for each behavior is computed across all combinations of target behaviors (single, pairs, and all three) that include that behavior. We report it individually for each behavior in its corresponding column as well as on average (right-most column). † denotes our methods. Bold indicates the best result per column.

Steering Method	Analogy Use	Bulletpoint Use	Sophisticated Language	Average
First-Order (HeRD-Merging)†	19.8	**65.9**	**53.6**	**46.4**
First-Order (Task Arithmetic)†	**20.2**	63.5	44.2	42.6
ODESteer	0.2	37.8	31.5	23.2
Linear-AcT	1.1	45.8	44.1	30.3
RE-Control	0.4	32.6	5.7	12.9
CAA	2.1	42.9	37.3	27.4
RepE	2.6	42.9	26.9	24.1

First-order steering is compared to CAA, Linear-AcT, RE-Control, RepE, and ODESteer (Panickssery et al., 2024; Rodriguez et al., 2024; Kong et al., 2024; Zou et al., 2023; Zhao et al., 2026). Similarly to the merging experiment, we ensure a fair comparison by using the best-performing hyperparameters described in the paper that introduces each method. CAA computes steering vectors by taking the mean differences in contrastive positive and negative pairs, provided by our synthetically generated datasets. Linear-AcT fits an input-dependent linear map to each activation to align

Table 2: The detailed behavioral adherence (%) results of Table 1 expanded and reported for each model with its 95% confidence intervals.

Model	Steering Method	Analogy Use	Bulletpoint Use	Sophisticated Language	Average
Qwen2.5-VL-3B	First-Order (HeRD-Merging)†	42.2 [36.8, 48.0]	**74.2 [68.2, 80.0]**	**77.8 [73.0, 82.5]**	**64.7**
	First-Order (Task Arithmetic)†	**43.0 [38.2, 47.8]**	67.0 [61.0, 73.0]	56.2 [49.5, 63.2]	55.4
	ODESteer	0.0 [0.0, 0.0]	21.5 [14.0, 29.5]	5.5 [2.5, 9.2]	9.0
	Linear-AcT	1.6 [1.0, 2.2]	29.5 [27.0, 32.1]	45.1 [42.1, 48.1]	25.4
	RE-Control	0.2 [0.0, 0.5]	30.6 [27.1, 34.1]	3.2 [1.9, 4.6]	11.3
	CAA	5.3 [4.1, 6.5]	25.9 [23.3, 28.6]	42.4 [39.2, 45.8]	24.5
	RepE	0.2 [0.0, 0.5]	29.6 [26.3, 33.1]	3.4 [2.2, 4.8]	11.1
Mistral-7B Instruct-v0.3	First-Order (HeRD-Merging)†	16.6 [15.3, 17.8]	63.5 [60.7, 66.5]	47.5 [44.3, 50.8]	**42.5**
	First-Order (Task Arithmetic)†	**17.1 [15.8, 18.3]**	63.7 [61.0, 66.5]	41.3 [38.7, 44.0]	40.7
	ODESteer	0.3 [0.0, 0.8]	**70.8** [67.0, 74.7]	46.4 [42.2, 50.5]	39.2
	Linear-AcT	0.8 [0.4, 1.4]	50.8 [47.3, 54.4]	51.4 [47.6, 55.4]	34.3
	RE-Control	0.2 [0.0, 0.6]	33.2 [29.2, 37.2]	4.8 [3.0, 6.8]	12.7
	CAA	0.5 [0.1, 1.1]	58.7 [54.9, 62.3]	35.5 [32.1, 39.0]	31.6
	RepE	5.5 [4.5, 6.5]	54.4 [51.1, 57.5]	**53.8 [50.1, 57.4]**	37.9
Gemma-4-E2B	First-Order (HeRD-Merging)†	0.5 [0.1, 1.1]	**60.1 [56.4, 63.8]**	35.4 [32.4, 38.7]	**32.0**
	First-Order (Task Arithmetic)†	0.5 [0.1, 1.0]	59.9 [56.4, 63.6]	35.1 [31.9, 38.5]	31.8
	ODESteer	0.3 [0.0, 0.8]	21.1 [19.7, 22.5]	42.5 [38.6, 46.6]	21.3
	Linear-AcT	1.0 [0.4, 1.6]	57.1 [53.4, 60.9]	**35.9 [32.7, 39.2]**	31.3
	RE-Control	0.8 [0.2, 1.6]	34.0 [29.8, 38.2]	9.0 [6.6, 11.6]	14.6
	CAA	0.5 [0.1, 1.1]	44.0 [40.9, 46.8]	34.1 [30.9, 37.2]	26.2
	RepE	**2.0 [1.2, 2.9]**	44.7 [40.8, 48.5]	23.5 [20.8, 26.2]	23.4

the coordinate-wise observed activation distributions for positive prompts. RE-Control fits a small multi-layer perceptron (MLP) with output in $[0, 1]$ for 200 steps of the Adam optimizer (Kingma & Ba, 2017). The MLP predicts whether a particular behavior is met and computes an input-dependent steering vector through 30 online gradient descent steps, using the MLP as the minimization objective. RepE computes steering vectors as the first principal component of centered mean differences between positive and negative pairs. ODESteer uses 10 steps of Euler's method along the gradient of a log-density ratio barrier function using a degree-two polynomial count sketch with $\gamma = 0.1, c_0 = 1.0$.

Aggregating the results across all three models in Table 1, first-order steering with HeRD-Merging achieves an average behavioral adherence of 46.4%, compared with 42.6% for first-order steering with task arithmetic. The strongest non-first-order baseline is Linear-AcT, at 30.3%, which is 16.1 percentage points away from first-order steering with HeRD-Merging. First-order steering also obtains the highest aggregate adherence for all three axes, exceeding the strongest non-first-order results by 17.6, 20.1, and 9.5 percentage points, respectively.

We also present more detailed adherence results for every model in Table 2. Here too first order steering with HeRD-Merging achieves the highest average behavioral adherence when averaging across the three target behaviors on all three models (right-most column). 64.7% on Qwen2.5-VL-3B, 42.5% on Mistral-7B-Instruct-v0.3, and 32.0% on Gemma-4-E2B. These exceed the strongest non-first-order baseline for each model, Linear-AcT (25.4%) on Qwen2.5, ODESteer (39.2%) on Mistral, and Linear-AcT (31.3%) on Gemma, by 39.3, 3.3, and 0.7 percentage points respectively. First-order steering with task arithmetic also exceeds these baselines in mean adherence, reaching 55.4%, 40.7%, and 31.8%, respectively. Relative to task arithmetic, HeRD-Merging improves mean adherence by 9.3, 1.8, and 0.2 percentage points respectively. Although the benefit is most pronounced on Qwen, the overall benefit of the trained terms $\boldsymbol{R}_S$ is supported by our experiments. Note that while individual baselines are sometimes stronger for a particular model and target behavior, no baseline method is consistently second best. Additional results for first-order steering applied at all layers using both HeRD-Merging and task arithmetic are reported in Section C.2.

5.4 HERD-MERGING COMPARISONS

We additionally compare HeRD-Merging against task arithmetic (Ilharco et al., 2023), DARE with task arithmetic for merging (Yu et al., 2024), DARE with TIES for merging (Yadav et al., 2023), and KnOTS with TIES for merging (Stoica et al., 2024). DARE randomly sparsifies each LoRA adapter with a drop rate $p = 0.9$. KnOTS performs SVD on the entire set of adapters to find an orthogonal basis for the weight updates. Merging through TIES is done by retaining only the parameters in each adapter which are in the top 20% by absolute relative magnitude, electing the sign to be the sign of the total parameter mass, and merging with only the parameters that match this elected sign. For HeRD-Merging, we train HeRD adapters at all projection heads in all layers using the same hyperparameters as Section 5.3, with the evaluation and confidence intervals computations performed identically. To ensure a fair comparison, all hyperparameters are set to the default, best-performing configuration described by the main paper for each method.

As shown in Table 3, HeRD-Merging achieves the highest overall behavioral adherence, at 79.9%, compared with 78.5% for DARE task arithmetic, the strongest baseline, and 78.2% for standard task arithmetic. This corresponds to improvements of 1.4 and 1.7 percentage points, respectively. HeRD-Merging also attains the highest aggregate adherence for each target behavior: 74.8% for analogy use, 94.4% for bulletpoint use, and 70.6% for sophisticated language. Relative to the strongest baseline for each behavior, these scores improve upon task arithmetic by 0.6 percentage points for analogy use, task arithmetic and DARE-TIES by 0.1 percentage points for bulletpoint use, and DARE-TIES by 1.5 percentage points for sophisticated language. These relatively small margins indicate that, in addition to improving first-order steering performance, HeRD-Merging maintains performance comparable to established model-merging procedures. We provide more detailed per-model results in Table 5.

Table 3: Overall behavioral adherence (%) of model merging methods, averaged across all three models using the same evaluation method as in Table 1. Bold indicates the best result per column.

Model Merging Method	Analogy Use	Bulletpoint Use	Sophisticated Language	Average
HeRD-Merging (Ours)	**74.8**	**94.4**	**70.6**	**79.9**
DARE task arithmetic	73.5	93.3	68.7	78.5
Task arithmetic	74.2	94.3	66.1	78.2
DARE-TIES	70.0	94.3	69.1	77.8
KnOTS-TIES	51.2	90.5	52.2	64.6
TIES	51.4	76.1	55.1	60.9

6 CONCLUSION

We propose first-order steering, a reformulation of activation steering as a first-order approximation of model weight adaptations. We devise HeRD-Merging to merge multiple models while controlling the Hessian matrix norm of each transformer layer, which we show bounds the steering approximation error. First-order steering thus provides a quantitative indicator of performance that existing steering methods lack. In experiments, first-order steering improves overall accuracy and consistency across model families compared to baselines on behavioral adaptation tasks. HeRD-Merging performs similarly to established model merging methods while improving the resulting first-order steering performance. A practical limitation of first-order steering is the compute and data requirements of model finetuning. Future work could explore whether low-data finetuning techniques can produce comparable results. Notwithstanding, first-order steering also offers potential applications in mechanistic interpretability. For instance, exploring relationships between the Jacobian matrix $J_{\boldsymbol{s}}\boldsymbol{h}_\ell$ and the influence of a system prompt could reveal insights into model robustness and behavior prediction. Finally, the effectiveness of HeRD-Merging for Hessian regularization depends on the supplied weight update matrices. Future work could explore whether incorporating Hessian regularization into model training improves merging and steering performance.

Ethics Statement

This research adheres to the ICLR Code of Ethics. We do not use human subjects or user-sensitive data in our experiments. All data and models used for our experiments are publicly available. The aim of this work is to aid in the construction of aligned and controllable large language models. Our experiments demonstrate that our methods can cheaply induce various benign model behaviors. However, we acknowledge that these techniques are dual-use, and we promote responsible deployment in safety-critical environments.

Reproducibility Statement

The results of our research can be reproduced using the code available at this repository. Synthetically generated datasets are hosted at the links for analogy use, bulletpoint use, and sophisticated language, which can be used to reproduce our trained models using the procedure described in the code. For our experiments, we use four NVIDIA A100 GPUs with 80 GB VRAM. Further details on experiments are provided in Section B.

AI Use Disclosure Statement

In this work, we used LLMs to generate synthetic datasets, provide feedback on research methodology, and implement portions of our methodology. We additionally used AI tools to search for relevant information and analyze existing literature. We did not use generative tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design research methodology, assist with translation, clean or reformat data, support qualitative and thematic data analysis, or interpret results. We have manually revised all AI-assisted work. Programs generated by LLMs were reviewed manually for consistency with our methodology, and we used software tests to verify intended functionality. All LLM feedback was thoroughly and manually reviewed for inconsistencies and compatibility with our methods. We take responsibility for the final content of this work, including text, claims, and experimental results presented with the aid of LLMs.

References

Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, Clément Farabet, Johan Ferret, Filippo Galgani, Sertan Girgin, Jean-Bastien Grill, Maarten Grootendorst, Jiaxian Guo, Cassidy Hardin, Yanzhang He, Steven M. Hernandez, Omri Homburger, Léonard Hussenot, Juyeong Ji, Armand Joulin, Aishwarya Kamath, Parnian Kassraie, Olivier Lacombe, Preethi Lahoti, Gaël Liu, Gus Martins, Luciano Martins, Tatiana Matejovicova, Ramona Merhej, Nikola Momchev, Sneha Mondal, Ryan Mullins, Sindhu Raghuram Panyam, Shreya Pathak, Sarah Perrin, André Susano Pinto, Etienne Pot, Angéline Pouget, Alexandre Ramé, Sabela Ramos, Douglas Reid, David Rim, Morgane Rivière, Karsten Roth, Louis Rouillard, Omar Sanseviero, Pier Giuseppe Sessa, Shane Settle, Danila Sinopalnikov, Sara Smoot, Piotr Stanczyk, Andreas Steiner, Lawrence Stewart, Ilya Tolstikhin, Michael Tschannen, Anton Tsitsulin, Nino Vieillard, Renjie Wu, Pingmei Xu, Haichuan Yang, Edouard Yvinec, Biao Zhang, Li Zhang, Joe Zou, Nicolas Aagnes, Abdelrahman Abdelhamed, Jakub Adamek, Shivani Agrawal, Shubham Agrawal, Ibrahim Alabdulmohsin, Jean Baptiste Alayrac, Uri Alon, Chandramouli Amarnath, Ankesh Anand, Chrysovalantis Anastasiou, Setareh Ariafar, François-Xavier Aubet, Kyriakos Axiotis, Federico Barbero, Joelle Barral, Alexei Bendebury, Urs Bergmann, Stanley Bileschi, Kat Black, Mathieu Blondel, Sebastian Borgeaud, Arthur Bražinskas, Ryan Burnell, Robert Busa-Fekete, Mu Cai, Daniele Calandriello, Glenn Cameron, Charlotte Caucheteux, Rahma Chaabouni, Garima Chadha, Jetha Chan, Blake Jianhang Chen, Jesse Chen, Lin Chen, Xu Chen, Derek Cheng, Tzu hsiang Chien, Nikolai Chinaev, Yi Chou, Zhaohui Chu, Benjamin Coleman, Pooja Consul, Sam Conway-Rahman, Scott Crowell, Dylan Cutler, Vivek Dani, Samira Daruki, Anil Das, Daniel Deutsch, Nishanth Dikkala, Li Ding, Qiuhan Ding, Shenil Dodhia,

Konstantin Donhauser, Tulsee Doshi, Anca Dragan, Alex Druinsky, Sahil Dua, Zoltan Egyed, Danielle Eisenbud, Daniel Eppens, Cindy Fan, Bahare Fatemi, Yassir Fathullah, Vlad Feinberg, Milen Ferev, Sebastian Flennerhag, Takumi Fujimoto, João Gabriel Oliveira, Isaac Galatzer-Levy, João Gante, Simon Geisler, Soham Ghosal, Antonious M. Girgis, Tamara von Glehn, Alec Go, Alhaad Gokhale, Alex Grills, Yiming Gu, Mayank Gupta, Pramod Gupta, Guru Guruganesh, Raia Hadsell, Hamza Harkous, Jitendra Harlalka, Demis Hassabis, Anja Hauth, Joe Heyward, Arian Hosseini, Chih-Yang Hsia, I-Hung Hsu, Xiaopeng Huang, Yangsibo Huang, Kevin Hui, Adrian Hutter, Te I, Fotis Iliopoulos, Advait Jain, Ganesh Jawahar, Ziwei Ji, Qilin Jin, Melvin Johnson, Kandarp Joshi, Arun Kandoor, Wang-Cheng Kang, Koray Kavukcuoglu, Mehran Kazemi, Kathleen Kenealy, Amr Khalifa, Phoebe Kirk, Ivan Korotkov, Suraj Kothawade, Vitaly Kovalev, Neel Kovelamudi, Adam Kraft, Ravin Kumar, Vivek Kumar, Harish Kuppam, Justin Lannin, Chen-Yu Lee, Seungji Lee, Dmitry Lepikhin, Alon Levkovitch, Dongdong Li, Qiujia Li, Valentin Liévin, Ethan Lin, Ziqian Lin, Casper Liu, Tianlin Liu, Tianqi Liu, Xin Liu, Ivan Lobov, Mayank Lunayach, Min Ma, Gagan Madan, Andrii Maksai, Eric Malmi, Michal Matuszak, Daniel McDuff, Gaurav Menghani, Maciej Mikuła, Daniil Mirylenka, Karolis Misiunas, Vedant Misra, Andreea Mitran, Kareem Mohamed, Maksim Mukha, Eric Noland, James O'Donnell, Brendan O'Donoghue, Kate Olszewska, Bernett Orlando, Wanqiong Pan, Rina Panigrahy, Unnati Parekh, Nicolas Perez-Nieves, Chunjong Park, Eric Paskie, Liqian Peng, Bryce Petrini, Slav Petrov, Jonas Pfeiffer, Bilal Piot, Martyna Plomecka, Siim Poder, Octavio Ponce, Arijit Pramanik, David Racz, Anish Rajan, Michelle Ramanovich, Anand Rao, Marvin Ritter, Vitor Rodrigues, Evan Rosen, Mikołaj Rybiński, Noveen Sachdeva, Michaël E. Sander, Rohit Sathyanarayana, Sagar Savla, Samuel Schmidgall, Tal Schuster, George Scrivener, Benoit Seguin, Andrew Sellergren, Aliaksei Severyn, Izhak Shafran, Dhruv Shah, Bobak Shahriari, Yuan Shangguan, Ashish Shenoy, Pradeep Shenoy, Rakesh Shivanna, Pauline Sho, Lucas Spangher, Wojciech Stokowiec, Tim Strother, Yao Su, Yinghao Sun, Mukund Sundararajan, Andrea Tacchetti, Mor Hazan Taege, Pouya Tafti, Jean Tarbouriech, Chetan Tekur, Shantanu Thakoor, Rahul Thapa, Madeleine Traverse, Lenart Treven, Tao Tu, Chien Te Tung, Çağlar Ünlü, Petar Veličković, Malini Pooni Venkat, Sagar Gubbi Venkatesh, Vidya Venkiteswaran, Francesco Visin, Alex Vitvitskyi, Kiran Vodrahalli, Weiyi Wang, Xin Wang, Tris Warkentin, Jan Wassenberg, John Wieting, Cindy Wu, Lechao Xiao, Hao Xu, Yuhui Xu, Fuzhao Xue, Arun Yadav, Jun Yan, Antoine Yang, Lin Yang, Ming-Hsuan Yang, Ziyu Ying, Jae Hyeon Yoo, Morteza Zadimoghaddam, Sajjad Zafar, Fred Zhang, Jiageng Zhang, Jianyi Zhang, Xiaofan Zhang, Chao Zhao, David Zhou, and Chen Zou. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Marco Paul E. Apolinario and Kaushik Roy. Lance: Low rank activation compression for efficient on-device continual learning, 2026. URL https://arxiv.org/abs/2509.21617.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Joschka Braun, Carsten Eickhoff, David Krueger, Seyed Ali Bahrainian, and Dmitrii Krasheninnikov. Understanding (un)reliability of steering vectors in language models. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025. URL https://openreview.net/forum?id=JZiKuvIK1t.

Sourav Chatterjee. Fluctuations of eigenvalues and second order poincaré inequalities, 2007. URL https://arxiv.org/abs/0705.1224.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023. URL https://arxiv.org/abs/2309.08600.

Diaoulé Diallo, Katharina Dworatzyk, Sophie Jentzsch, Peer Schütt, Sabine Theis, and Tobias Hecking. The effectiveness of style vectors for steering large language models: A human evaluation. *IEEE Access*, 13:191443–191457, 2025. doi: 10.1109/ACCESS.2025.3628500.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations, 2023. URL https://arxiv.org/abs/2305.14233.

Yi Fang, Wenjie Wang, Mingfeng Xue, Boyi Deng, Fengli Xu, Dayiheng Liu, and Fuli Feng. Controllable LLM reasoning via sparse autoencoder-based steering. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21290–21305, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.974. URL https://aclanthology.org/2026.acl-long.974/.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA.

Geert Heyman and Frederik Vandeputte. Steer like the LLM: Activation steering that mimics prompting. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=06Nk3dJDMq.

L. Hörmander. *The Analysis of Linear Partial Differential Operators I: Distribution Theory and Fourier Analysis*. Classics in Mathematics. Springer Berlin Heidelberg, 2015. ISBN 9783642614972. URL https://books.google.com/books?id=aaLrCAAAQBAJ.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Sara Kangaslahti and David Alvarez-Melis. Continuous language model interpolation yields dynamic and controllable text generation. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=xD9Nu2Wah4.

Been Kim, Justin Gilmer, Martin Wattenberg, and Fernanda Viégas. TCAV: Relative concept importance testing with linear concept activation vectors, 2018. URL https://openreview.net/forum?id=S1viikbCW.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Toshiaki Koike-Akino, Francesco Tonin, Yongtao Wu, Frank Zhengqing Wu, Leyla Naz Candogan, and Volkan Cevher. Quantum-PEFT: Ultra parameter-efficient fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dgR6i4TSng.

Lingkai Kong, Haorui Wang, Wenhao Mu, Yuanqi Du, Yuchen Zhuang, Yifei Zhou, Yue Song, Rongzhi Zhang, Kai Wang, and Chao Zhang. Aligning large language models with representation editing: A control perspective. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=yTTomSJsSW.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. Llms-as-judges: A comprehensive survey on llm-based evaluation methods, 2024. URL https://arxiv.org/abs/2412.05579.

JorgeJ. Moré. The levenberg-marquardt algorithm: Implementation and theory. In G.A. Watson (ed.), *Numerical Analysis*, volume 630 of *Lecture Notes in Mathematics*, pp. 105–116. Springer Berlin Heidelberg, 1978. ISBN 978-3-540-08538-6. doi: 10.1007/BFb0067700. URL http://dx.doi.org/10.1007/BFb0067700.

Muhammad Faishal Adly Nelwan and Alfan Farizki Wicaksono. Deployable per-instance multi-layer activation steering for large language models, 2026. URL https://arxiv.org/abs/2608.08829.

Narmeen Oozeer, Luke Marks, Shreyans Jain, Fazl Barez, and Amirali Abdullah. Beyond linear steering: Unified multi-attribute control for language models, 2026. URL https://arxiv.org/abs/2505.24535.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition, 2024. URL https://arxiv.org/abs/2312.06681.

Akshara Prabhakar, Yuanzhi Li, Karthik Narasimhan, Sham Kakade, Eran Malach, and Samy Jelassi. Lora soups: Merging loras for practical skill composition tasks, 2024. URL https://arxiv.org/abs/2410.13025.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Marco Cuturi, and Xavier Suau. Controlling language and diffusion models by transporting activations, 2024. URL https://arxiv.org/abs/2410.23054.

Daniel Scalena, Gabriele Sarti, and Malvina Nissim. Multi-property steering of large language models with dynamic activation composition. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 577–603. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.blackboxnlp-1.34. URL http://dx.doi.org/10.18653/v1/2024.blackboxnlp-1.34.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open problems in mechanistic interpretability, 2025. URL https://arxiv.org/abs/2501.16496.

Kartik Sharma and Rakshit Trivedi. COLD-steer: Steering large language models via in-context one-step learning dynamics. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=afV4qzquBN.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models, 2025. URL https://arxiv.org/abs/2503.05613.

Shashwat Singh, Shauli Ravfogel, Jonathan Herzig, Roee Aharoni, Ryan Cotterell, and Ponnurangam Kumaraguru. Representation surgery: Theory and practice of affine steering. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 45663–45680. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/singh24d.html.

Samuel Soo, Wesley Teng, Chandrasekaran Balaganesh, Tan Guoxian, and Ming YAN. Interpretable steering of large language models with feature guided activation additions. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025. URL https://openreview.net/forum?id=swRxS7s4rB.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots, 2024. URL https://arxiv.org/abs/2410.19735.

Daniel Chee Hian Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. Analysing the generalisation and reliability of steering vectors. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=v8X70gTodR.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024. URL https://arxiv.org/abs/2308.10248.

Teun van der Weij, Massimo Poesio, and Nandi Schoots. Extending activation steering to broad skills and multiple behaviours, 2024. URL https://arxiv.org/abs/2403.05767.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=fq0NaiU8Ex.

Hongjue Zhao, Haosen Sun, Jiangtao Kong, Xiaochang Li, Qineng Wang, Liwei Jiang, Qi Zhu, Tarek F. Abdelzaher, Yejin Choi, Manling Li, and Huajie Shao. ODESteer: A unified ODE-based steering framework for LLM alignment. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=CFewUmgIIL.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency, 2023. URL https://arxiv.org/abs/2310.01405.

APPENDIX CONTENTS

A Theoretical Details **15**

A.1 Proofs of Theorems 15

A.2 First-Order Steering Generalizes Task Arithmetic 16

B Experiment Details **16**

B.1 Synthetic Dataset Generation 16

B.2 Classifier Training 16

B.3 LLM Generation Settings 16

C Additional Results **17**

C.1 HeRD-Merging Per-Model Results 17

C.2 First-Order Steering at All Layers 17

C.3 Synthetic Data Examples 20

C.4 Steering Output Examples 23

A THEORETICAL DETAILS

A.1 PROOFS OF THEOREMS

Here, we supply a proof of the theoretical result stated in Proposition 2. The argument leverages the mean value inequality, which we restate below.

Proposition 3 (Hörmander (2015))**.** *Let V be a Banach space with norm $\|\cdot\|_V$, let I be open in $\mathbb{R}$, and let $f : I \to V$. Then*

$$\|f(y) - f(x)\|_V \leq \|y - x\| \sup_{x \in I} \|J_{\boldsymbol{x}} f\|_V$$

for any $x, y \in I$.

With minimal effort, this result can be extended to an open rectangle S and functions $f : \mathbb{R}^{d+n} \to V$ for $x, y \in S$. We now prove the theorem.

Proof of Proposition 2. Let $g(\boldsymbol{x}_a) = J_{\boldsymbol{s}} \boldsymbol{h}'_\ell(\boldsymbol{x}_a, \mathbf{0})$ and let $\boldsymbol{x}_a, \boldsymbol{x}_b \in \mathcal{D}$. Since S is an open rectangle, the mean value inequality gives

$$\sum_{a,b} \|g(\boldsymbol{x}_b) - g(\boldsymbol{x}_a)\|_F^2 \leq \sum_{a,b} Q^2 \|\boldsymbol{x}_b - \boldsymbol{x}_a\|^2 \leq M^2 Q^2 L^2.$$

Let

$$\boldsymbol{A}^* = \frac{1}{M} \sum_{m=1}^{M} g(\boldsymbol{x}_m).$$

Then we achieve the desired bound by

$$\frac{1}{M} \sum_{m=1}^{M} \|g(\boldsymbol{x}_m) - \boldsymbol{A}^*\|_F^2 = \frac{1}{2M^2} \sum_{a,b} \|g(\boldsymbol{x}_b) - g(\boldsymbol{x}_a)\|_F^2 \tag{9}$$

$$\leq \frac{Q^2 L^2}{2}. \tag{10}$$

□

A.2 First-Order Steering Generalizes Task Arithmetic

In the development of a procedure $\Delta \boldsymbol{W}(\boldsymbol{s})$ for merging models given by weight updates $\Delta \boldsymbol{W}_1, \ldots, \Delta \boldsymbol{W}_n$, it is reasonable to expect that the partial derivative of $\Delta \boldsymbol{W}(\boldsymbol{s})$ with respect to each component s_k yields the original matrix $\Delta \boldsymbol{W}_k$. In particular, we enforce

$$J_{\boldsymbol{s}} \Delta \boldsymbol{W}(\not{\vdash}) \boldsymbol{e}_k = \Delta \boldsymbol{W}_k. \tag{11}$$

In such a procedure, the analytical derivation $J_{\boldsymbol{s}} \boldsymbol{h}_\ell(\boldsymbol{x}, \boldsymbol{0})\, \boldsymbol{d}$ is

$$J_{\boldsymbol{s}} \boldsymbol{h}_\ell(\boldsymbol{x}, \boldsymbol{0})\, \boldsymbol{d} = \frac{d}{d\alpha} \boldsymbol{h}_\ell(\boldsymbol{x}, \boldsymbol{W}_{p_1}(\alpha \boldsymbol{d}), \ldots, \boldsymbol{W}_{p_q}(\alpha \boldsymbol{d}))$$

for the relevant projection heads $p_1, \ldots, p_q \in \mathcal{P}$. Then the chain rule gives

$$J_{\boldsymbol{s}} \boldsymbol{h}_\ell(\boldsymbol{x}, \boldsymbol{0})\, \boldsymbol{d} = \sum_{r=1}^{q} \frac{\partial \boldsymbol{h}_\ell}{\partial \boldsymbol{W}_{\ell, p_r}}(\boldsymbol{x}, \boldsymbol{0}) [J_{\boldsymbol{s}} \boldsymbol{W}_{\ell, p_r}(\boldsymbol{x}, \boldsymbol{0}) \boldsymbol{d}] \tag{12}$$

$$= \sum_{r=1}^{q} \frac{\partial \boldsymbol{h}_\ell}{\partial \boldsymbol{W}_{\ell, p_r}}(\boldsymbol{x}, \boldsymbol{0}) \sum_{k=1}^{n} d_k \Delta \boldsymbol{W}_{k, p_r}. \tag{13}$$

For just a single projection head p, this becomes

$$\frac{\partial \boldsymbol{h}_\ell}{\partial \boldsymbol{W}_{\ell, p}}(\boldsymbol{x}, \boldsymbol{0}) \sum_{k=1}^{n} d_k \Delta \boldsymbol{W}_{k, p}.$$

A slight generalization involves replacing $\boldsymbol{h}_\ell$ with any block $\boldsymbol{F}_i(\boldsymbol{x}, \boldsymbol{W}_p)$ that follows $\boldsymbol{W}_p$ in the computational graph of the language model. In particular, any differentiable merging procedure which satisfies (11) must contain task arithmetic as its first-order term, given by choosing $\boldsymbol{F}_i = \boldsymbol{W}_p$. This supports the naturality of task arithmetic as a rule for composing behaviors, in that behavioral updates at any $\boldsymbol{F}_i$ can be viewed as applying the map $\frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{W}_{\ell, p}}(\boldsymbol{x}, \boldsymbol{0})$ to task arithmetic.

B Experiment Details

In this section, we provide further details for the experiments described in Section 5.

B.1 Synthetic Dataset Generation

The generation of synthetic data was done separately for each of the three target behaviors using GPT-5.5. The sampling parameters are kept as the default from OpenAI (temperature $\tau = 1.0$ and top-p with $p = 1.0$). Prompts from the UltraChat dataset are given to the model with a system prompt specifying the desired transformation. The original model responses are not provided to the model, and the model can only view the ongoing conversation history given by each row of the dataset. Each dataset transforms the same set of prompts, matched using the prompt IDs in UltraChat.

B.2 Classifier Training

For training classifiers, we use the pretrained checkpoint available at this Hugging Face repository. The model contains 12 layers, hidden dimension 768, and vocabulary size 128000, amounting to 184 million total parameters. We finetuned the base model to produce three variants corresponding to each target behavior. Training was done on the whole dataset of approximately 10000 samples over 2 epochs with batch size 128 using the AdamW optimizer defaults ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, \eta = 2 \times 10^{-5}$). We additionally schedule the learning rate η to decay linearly with a 10% warmup spanning 25 steps. To prevent overfitting, we use a weight decay of 0.01. The classification results are available in Table 4.

B.3 LLM Generation Settings

Generations for all evaluations across all methods were performed with no system prompt using greedy decoding for reproducibility. We include the Hugging Face repository for each model below.

Table 4: Finetuned DeBERTa-v3-base behavior classifier performance. Thresholds are selected on the validation set to maximize validation accuracy and are used for all evaluations.

Axis	Threshold	Test AUROC	Test AUPRC	Test Acc (%)	Test F_1
Analogy Use	0.85	1.00	1.00	99.95	1.00
Bulletpoint Use	0.12	1.00	1.00	99.85	1.00
Sophisticated Language	0.20	1.00	1.00	99.85	1.00

1. **Qwen2.5-VL-3B:** `Qwen/Qwen2.5-VL-3B-Instruct`
2. **Gemma-4-E2B:** `google/gemma-4-E2B`
3. **Mistral-7B-Instruct-v0.3:** `mistralai/Mistral-7B-Instruct-v0.3`

We also note that the choice of these models supplies a range of parameter counts in addition to model architectures. Qwen2.5-VL-3B has 3 billion parameters, Gemma-4-E2B is a per-layer embeddings model with 5 billion total parameters, and Mistral-7B-Instruct-v0.3 has 7 billion parameters.

The computation of bootstrap confidence intervals for generations uses the nonparametric bootstrapping method, but with the modification that prompts, rather than individual generations, are resampled. This preserves the dependence among generations derived from the same prompt. For each bootstrap replicate, we sample N prompts with replacement from the N evaluation prompts, include every associated generation for each selected prompt, and record the classifier outputs. We repeat this procedure $B = 2000$ times and report the empirical 2.5th and 97.5th percentiles as a 95% confidence interval for each reported statistic.

C Additional Results

In this section we report supplementary results for first-order steering and HeRD-Merging.

C.1 HeRD-Merging Per-Model Results

We provide the results of HeRD-Merging for each model to supplement the overall results presented in Section 5.4. HeRD-Merging demonstrates competitive performance with current merging baselines (Table 5). On Qwen, HeRD-Merging reaches 92.4% mean adherence, exceeding the strongest baseline by 2.2 percentage points. Similarly, HeRD-Merging reaches 78.9% on Mistral, exceeding the strong baseline, task arithmetic, which scores 76.7%. On Gemma, HeRD-Merging matches task arithmetic at 68.5% mean adherence, while DARE-TIES leads at 71.9% and exceeds HeRD-Merging on all three behaviors. This suggests that although performance improvements are displayed on Qwen and Mistral, HeRD-Merging is not a universally superior merging procedure in comparison to existing methods [1]. Nevertheless, the results support HeRD-Merging as a competitive weight-merging procedure that does not dramatically sacrifice merging performance for Hessian regularization.

C.2 First-Order Steering at All Layers

The evaluation of first-order steering against the remaining steering baselines utilized only a single layer for steering to facilitate a fair comparison between all methods. However, first-order steering supports a clean formulation for computing steering vectors for all layers simultaneously without encountering output degradation. The procedure for doing this remains consistent with Section 5.3, with the difference being that the LoRA adapters used are the ones trained at all layers for use in Section 5.4.

[1] The core aim of this work is in improving steering performance, not merging performance. We report these results merely to demonstrate that using HeRD-Merging results in merging performance similar to existing procedures.

Table 5: Behavioral adherence (%) of model merging methods with 95% confidence intervals, computed across all combinations of target behaviors and reported individually for each. Bold indicates the best result per column within each model block.

Model	Model Merging Method	Analogy Use	Bulletpoint Use	Sophisticated Language	Average
Qwen2.5-VL-3B	HeRD-Merging (Ours)	**86.8 [82.5, 90.5]**	96.2 [94.2, 98.0]	**94.2 [91.0, 97.0]**	**92.4**
	DARE task arithmetic	81.0 [76.2, 85.5]	**99.0 [97.0, 100.0]**	90.5 [86.5, 94.0]	90.2
	Task arithmetic	82.8 [78.2, 87.0]	98.5 [96.0, 100.0]	87.0 [82.0, 91.5]	89.4
	DARE-TIES	73.5 [67.8, 79.0]	98.5 [96.0, 100.0]	87.8 [83.8, 91.5]	86.6
	KnOTS-TIES	44.0 [37.2, 50.7]	98.2 [95.8, 99.8]	63.7 [58.2, 69.0]	68.6
	TIES	43.0 [36.0, 49.8]	84.2 [80.0, 88.2]	54.2 [49.2, 59.5]	60.5
Mistral-7B Instruct-v0.3	HeRD-Merging (Ours)	76.0 [71.0, 81.2]	**93.8 [91.0, 96.2]**	**66.8 [60.5, 72.5]**	**78.9**
	DARE task arithmetic	**77.5 [73.0, 82.2]**	90.0 [86.8, 93.0]	60.8 [54.5, 66.8]	76.1
	Task arithmetic	**77.5 [72.8, 82.2]**	91.2 [88.2, 94.0]	61.3 [55.0, 67.5]	76.7
	DARE-TIES	72.2 [68.0, 76.2]	89.5 [85.8, 92.5]	63.0 [57.0, 69.2]	74.9
	KnOTS-TIES	65.5 [61.0, 69.8]	77.5 [73.0, 81.8]	43.5 [39.0, 48.0]	62.2
	TIES	62.5 [57.8, 67.2]	68.8 [63.0, 74.2]	54.2 [48.2, 60.5]	61.8
Gemma-4-E2B	HeRD-Merging (Ours)	61.5 [57.2, 65.8]	93.2 [90.0, 96.0]	50.7 [45.8, 55.8]	68.5
	DARE task arithmetic	62.0 [57.8, 66.0]	91.0 [87.8, 94.0]	54.8 [48.5, 61.0]	69.3
	Task arithmetic	62.3 [58.0, 66.5]	93.2 [89.5, 96.2]	50.0 [45.2, 55.0]	68.5
	DARE-TIES	**64.2 [60.2, 68.2]**	95.0 [92.5, 97.2]	56.5 [50.2, 62.8]	**71.9**
	KnOTS-TIES	44.0 [40.8, 47.2]	**95.8 [92.5, 98.2]**	49.5 [42.8, 56.5]	63.1
	TIES	48.8 [44.0, 53.2]	75.2 [69.2, 80.8]	**56.8 [50.2, 63.2]**	60.3

HeRD-Merging retains higher mean adherence than task arithmetic when steering at all layers, but the benefit of all-layer steering itself depends on the model (Table 6). Relative to single-layer steering, mean adherence with HeRD-Merging increases from 64.7% to 81.2% on Qwen and from 32.0% to 34.3% on Gemma, but decreases from 42.5% to 35.4% on Mistral. Thus, applying steering at more layers does not uniformly improve behavioral control.

To characterize how steering-vector contributions are distributed across layers, we define the *effective layer count* for target behavior i as

$$P_i = \frac{1}{\sum_\ell p_{i,\ell}^2}, \qquad p_{i,\ell} = \frac{\|\boldsymbol{A}_\ell^* \boldsymbol{e}_i\|_2^2}{\sum_r \|\boldsymbol{A}_r^* \boldsymbol{e}_i\|_2^2}, \tag{14}$$

assuming that at least one of the steering vectors is nonzero. This definition has the property that if the summed steering vector norm is distributed equally over K layers and is zero elsewhere, then $P_i = K$. As shown in Table 7, Qwen has the lowest effective counts (5.5–8.6, mean 7.4), Mistral is intermediate (9.9–12.9, mean 11.1), and Gemma has the highest counts (19.9–20.8, mean 20.3). Within Qwen, sophisticated-language steering is the most concentrated, whereas Gemma exhibits similarly distributed energy across all three behaviors. We remark that steering at all layers simultaneously is underexplored due to the pre-existing problem of representational collapse (Braun et al., 2025; Tan et al., 2024; Nelwan & Wicaksono, 2026). However, we encourage future work to investigate the relationship between the steering vector norms produced from all-layer steering and the steerability of particular models or model layers.

Table 6: Behavioral adherence (%) of first-order steering at all layers with 95% confidence intervals, computed across all combinations of target behaviors and reported individually for each. Bold indicates the best result per column within each model block.

Model	Steering Method / Layers	Analogy Use	Bulletpoint Use	Sophisticated Language	Average
Qwen2.5-VL 3B	HeRD-Merging (All Layers)	66.0 [62.0, 69.8]	**89.8** [86.0, 93.2]	**87.8** [84.2, 90.8]	**81.2**
	Task Arithmetic (All Layers)	**69.8** [65.5, 74.0]	89.2 [84.5, 93.3]	70.2 [66.0, 74.8]	76.4
	HeRD-Merging (Single Layer)	42.2 [36.8, 48.0]	74.2 [68.2, 80.0]	77.8 [73.0, 82.5]	64.7
	Task Arithmetic (Single Layer)	43.0 [38.2, 47.8]	67.0 [61.0, 73.0]	56.2 [49.5, 63.2]	55.4
Mistral-7B Instruct-v0.3	HeRD-Merging (All Layers)	0.3 [0.0, 0.9]	**64.7** [55.9, 72.8]	41.2 [33.1, 50.3]	35.4
	Task Arithmetic (All Layers)	0.3 [0.0, 0.9]	64.1 [54.7, 73.1]	33.4 [27.5, 39.7]	32.6
	HeRD-Merging (Single Layer)	16.6 [15.3, 17.8]	63.5 [60.7, 66.5]	**47.5** [44.3, 50.8]	**42.5**
	Task Arithmetic (Single Layer)	**17.1** [15.8, 18.3]	63.7 [61.0, 66.5]	41.3 [38.7, 44.0]	40.7
Gemma-4 E2B	HeRD-Merging (All Layers)	0.3 [0.0, 0.9]	56.9 [47.8, 65.6]	**45.6** [37.8, 54.1]	**34.3**
	Task Arithmetic (All Layers)	0.3 [0.0, 0.9]	55.9 [46.6, 64.7]	45.0 [37.5, 53.1]	33.7
	HeRD-Merging (Single Layer)	**0.5** [0.1, 1.1]	**60.1** [56.4, 63.8]	35.4 [32.4, 38.7]	32.0
	Task Arithmetic (Single Layer)	**0.5** [0.1, 1.0]	59.9 [56.4, 63.6]	35.1 [31.9, 38.5]	31.8

Table 7: Effective layer count for each behavioral steering vector.

Model	Analogy	Bulletpoint Use	Sophisticated Language	Mean
Qwen2.5-VL-3B	8.6	8.0	5.5	7.4
Mistral-7B	9.9	12.9	10.6	11.1
Gemma-4-E2B	19.9	20.0	20.8	20.3

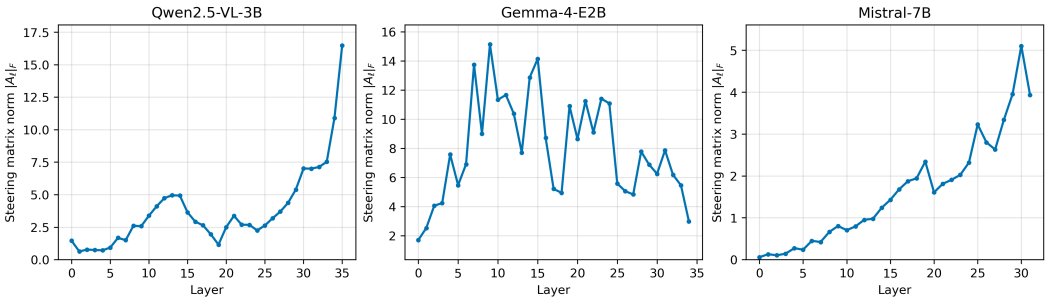

Figure 1: Frobenius norm of the steering matrix at each layer.

C.3 SYNTHETIC DATA EXAMPLES

Target Behavior: Analogy Use

Prompt	Original Answer	New Answer
Write an analytical essay on metaphorical language in nature-themed writing.	Nature-themed writing often employs metaphorical language to describe the natural environment in a way that is both imaginative and sensory. Metaphors serve as a tool for writers to create vivid images that allow readers to connect with the natural world beyond the physical senses. In this essay, we will analyze the use of metaphorical language in nature-themed writing and how it contributes to the reader's ...	Imagine a vast garden where each plant represents a different element of writing about nature, and metaphorical language is the vine that weaves through this garden, connecting and enhancing every flower and tree. Just as vines wrap around trees, giving them added texture and depth, metaphorical language encircles simple descriptions of nature, transforming them into rich tapestries of meaning. In this garden, ...
Please provide a step-by-step guide on how to make a traditional beef stroganoff, including ingredients, cooking methods, and serving suggestions.	Ingredients: • 1 lb top sirloin beef, sliced thinly • 1 onion, sliced thin • 2 cloves of garlic, minced • 8 oz cremini mushrooms, sliced • 1-2 tbsp all-purpose flour • 1/2 cup beef broth • 1/2 cup sour cream • 2 tbsp butter • Salt and pepper to taste For the noodles: • 8 oz egg noodles • Salt Instructions: 1. Cook the egg noodles according to the package instructions with a generous pinch of salt. Drain and set ...	Making traditional beef stroganoff is like preparing a classic orchestra performance, where each musician (ingredient) and section (step) must come in harmony for a beautiful symphony (delicious dish). First, gather your orchestra members: tender beef strips (the solo violinist), mushrooms, onions, garlic (the woodwind section), sour cream, beef broth (the brass section), flour (the percussion), butter, oil, salt, ...

Write a Tcl script that generates a random password with a length of 8 characters, comprising of uppercase letters, lowercase letters, and digits. The generated password should be displayed on the console.	`Set password ""`
	`set possible_chars "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"`
	`for {set I 0} {$i < 8} {incr i} {`
	`set rand_index [expr {int(rand() * [string length $possible_chars])}]`
	`append password [string index $possible_chars $rand_index]`
	`}`
	`puts "Your random password is: $password"`
		Imagine you're in a kitchen, and you want to make a sandwich that's exactly 8 layers thick. Each layer can be a slice of bread (uppercase letter), a slice of cheese (lowercase letter), or a slice of tomato (digit). To make the sandwich, you randomly grab one type of ingredient from the bowl each time until you have all 8 layers stacked. Once done, you place the sandwich on the counter for everyone to see.
		So, the ...

Target Behavior: Bullet Points

Prompt	Original Answer	New Answer
Write an analytical essay on metaphorical language in nature-themed writing.	Nature-themed writing often employs metaphorical language to describe the natural environment in a way that is both imaginative and sensory. Metaphors serve as a tool for writers to create vivid images that allow readers to connect with the natural world beyond the physical senses. In this essay, we will analyze the use of metaphorical language in nature-themed writing and how it contributes to the reader's ...	• Metaphorical language in nature-themed writing serves as a powerful tool to convey complex emotions and abstract concepts through relatable and vivid imagery. • It allows writers to bridge the gap between human experience and the natural world, making the unfamiliar familiar and the intangible tangible. • Nature metaphors often draw on familiar elements like seasons, animals, landscapes, and weather phenomena ...
Please provide a step-by-step guide on how to make a traditional beef stroganoff, including ingredients, cooking methods, and serving suggestions.	Ingredients: • 1 lb top sirloin beef, sliced thinly • 1 onion, sliced thin • 2 cloves of garlic, minced • 8 oz cremini mushrooms, sliced • 1-2 tbsp all-purpose flour • 1/2 cup beef broth • 1/2 cup sour cream • 2 tbsp butter • Salt and pepper to taste For the noodles: • 8 oz egg noodles • Salt Instructions: 1. Cook the egg noodles according to the package instructions with a generous pinch of salt. Drain and set ...	• Ingredients: • 1 lb (450g) beef sirloin or tenderloin, thinly sliced into strips • 1 medium onion, finely chopped • 2 cloves garlic, minced • 8 oz (225g) mushrooms, sliced • 1 cup beef broth • 1 cup sour cream • 2 tablespoons all-purpose flour • 2 tablespoons butter • 2 tablespoons vegetable oil • 1 tablespoon Dijon mustard (optional) • Salt and black pepper to taste • Fresh parsley, ...

Write a Tcl script that generates a random password with a length of 8 characters, comprising of uppercase letters, lowercase letters, and digits. The generated password should be displayed on the console.	`Set password ""`
	`set possible_chars "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"`
	`for {set I 0} {$i < 8} {incr i} {`
	`set rand_index [expr {int(rand() * [string length $possible_chars])}]`
	`append password [string index $possible_chars $rand_index]`
	`}`
	`puts "Your random password is:  $password"`
		• `#!/usr/bin/env tclsh`
		• `# Define character sets`
		• `set uppercase "ABCDEFGHIJKLMNOPQRSTUVWXYZ"`
		• `set lowercase "abcdefghijklmnopqrstuvwxyz"`
		• `set digits "0123456789"`
		• `# Combine all characters`
		• `set all_chars "${uppercase}${lowercase}${digits}"`
		• `# Set password length`
		• `set password_length 8`
		• `# Initialize empty password`
		• `set password ""`
		• `# Seed random number generator with current time`
		...

Target Behavior: Sophisticated Language

Prompt	Original Answer	New Answer
Write an analytical essay on metaphorical language in nature-themed writing.	Nature-themed writing often employs metaphorical language to describe the natural environment in a way that is both imaginative and sensory. Metaphors serve as a tool for writers to create vivid images that allow readers to connect with the natural world beyond the physical senses. In this essay, we will analyze the use of metaphorical language in nature-themed writing and how it contributes to the reader's ...	In the labyrinthine corridors of literary exegesis, the examination of metaphorical language within the ambit of nature-themed writing emerges as an intellectually fecund endeavor, replete with intricate interplays of semiotics and ontological resonances. The metaphor, that quintessentially abstruse rhetorical device, functions not merely as an ornamental linguistic flourish but as an epistemic conduit through which ...
Please provide a step-by-step guide on how to make a traditional beef stroganoff, including ingredients, cooking methods, and serving suggestions.	Ingredients: • 1 lb top sirloin beef, sliced thinly • 1 onion, sliced thin • 2 cloves of garlic, minced • 8 oz cremini mushrooms, sliced • 1-2 tbsp all-purpose flour • 1/2 cup beef broth • 1/2 cup sour cream • 2 tbsp butter • Salt and pepper to taste For the noodles: • 8 oz egg noodles • Salt Instructions: 1. Cook the egg noodles according to the package instructions with a generous pinch of salt. Drain and set ...	Ah, the culinary undertaking of preparing a traditional Beef Stroganoff—a gastronomic symphony that demands both precision and reverence for its Russo-European origins. Permit me to elucidate, in a manner befitting the grandeur of this dish, the sequential methodology, inclusive of requisite victuals, intricate culinary manipulations, and the most perspicacious recommendations for its consummate presentation. ...

Write a Tcl script that generates a random password with a length of 8 characters, comprising of uppercase letters, lowercase letters, and digits. The generated password should be displayed on the console.	`Set password ""`
	`set possible_chars "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"`
	`for {set I 0} {$i < 8} {incr i} {`
	`set rand_index [expr {int(rand() * [string length $possible_chars])}]`
	`append password [string index $possible_chars $rand_index]`
	`}`
	`puts "Your random password is: $password"`
		Indubitably, allow me to proffer unto you a Tcl script of such intricate construction and deliberate elegance that it conjures forth an octet of characters, each meticulously culled from the exalted pantheon of uppercase and lowercase alphabets, as well as the numerals that form the bedrock of arithmetical discourse. The ensuing script shall, with unerring precision, manifest the generated cryptographic artifact ...

C.4 Steering Output Examples

Target Behavior: Sophisticated Language.
Question: What programmes did Channel 5 air during 2002?

Model	Method	Output
Qwen2.5-VL-3B	First-Order (HeRD-Merging)	"During the tumultuous year of 2002, Channel 5, that most illustrious of British broadcasting institutions, aired an array of programmes that ranged from the arcane and esoteric to the mundane and prosaic."
	First-Order (Task Arithmetic)	"During the tumultuous year of 2002, Channel 5, that most illustrious of British broadcasting institutions, aired an array of programmes that were as diverse and eclectic as the cosmos itself. The network's programming diet was a veritable feast of intellectual and cultural savagery . . . "
	Linear-AcT	"Channel 5 aired a variety of programs during the year 2002, reflecting the channel's commitment to a broad spectrum of entertainment and informational content. Here is an overview of some of the notable programs . . . "
	CAA	"Channel 5, the British television network known for its eclectic programming, aired a diverse array of content during the year 2002. Here is a selection of some notable programmes that were broadcast on the channel . . . "
	RE-Control	"I'm sorry, but I cannot provide answers on political matters. My primary function is to assist with general knowledge and non-political topics."
	RepE	"I'm sorry, but I cannot provide answers on political matters. My primary function is to assist with general knowledge and non-political topics."
	ODESteer	"I apologize, but I don't have real-time data or the ability to access current information about specific TV channels' programming schedules."
Mistral-7B Instruct-v0.3	First-Order (HeRD-Merging)	"In the annals of the televisual panorama that was the year 2002, the venerable Channel 5, that venerable and hallowed institution of the televisual firmament, deigned to bestow upon its discerning audience a veritable cornucopia of televis . . . "
	First-Order (Task Arithmetic)	"In the annals of the televisual panorama that was the year 2002, the venerable Channel 5, that venerable and hallowed institution of the televisual firmament, deigned to bestow upon its discerning audience a veritable cornucopia of televis . . . "
	ODESteer	"It a a a a a a a a a?"
	RepE	"The 2020s with an unpardonable fervor . . . the cinematic extravagance of their audacious, labyrinthine narrative narrations . . . "
	Linear-AcT	"In the realm of the most prodigious and yet, perchance, to be certain, the forthcoming and forthcoming forthwith, the forthright and prolixity of your inquiry . . . "
	CAA	"In the quambleence to Indubmandyly . . . "
	RE-Control	"Channel 5, a British television channel, aired various shows during 2002. Here are some of the notable programs that were broadcast that year: Big Brother UK: Series 3 . . . "
Gemma-4-E2B	First-Order (HeRD-Merging)	"I do not have a specific, comprehensive, and up-to-date television programming log for Channel 5 for the entire year of 2002 in my current knowledge base to . . . "
	First-Order (Task Arithmetic)	"I do not have a specific, comprehensive, and up-to-date television programming log for Channel 5 for the entire year of 2002 in my current knowledge base . . . "
	Linear-AcT	"I do not Ah Ah Ah Ah Ah Ah Ah . . . "
	CAA	"I do not have access to a comprehensive, indexed, year-by-year programme schedule for Channel 5 in 2002. Such in-depth, archival programme data is typically held in proprietary broadcast archives."
	RepE	"I do not have access to a real-time or comprehensive archive of television programming for a specific year like 2002. Therefore, I cannot provide an exhaustive list . . . "
	ODESteer	"In arbitrary arbitrary arbitrary . . . "
	RE-Control	"I do not have access to a comprehensive, real-time archive of every programme Channel 5 aired during the entire year of 2002."

Target Behavior: Bulletpoint Use
Question: I think my graphics card is blowing up; a checkerboard pattern keeps showing up on my monitor sometimes.

Model	Method	Output
Qwen2.5-VL-3B	First-Order (HeRD-Merging)	"• **Check for overheating:** Ensure your computer is not too hot to touch. Check whether fans are spinning normally and cooling effectively. • **Monitor issues:** Verify whether the monitor is displaying correctly . . . "
	First-Order (Task Arithmetic)	"• **Check the graphics card:** Ensure that the graphics card is properly seated in the PCIe slot. Check for visible damage or signs of overheating. • **Monitor settings:** Adjust the resolution . . . "
	Linear-AcT	"It sounds like your graphics card might be experiencing issues. Here are a few steps you can take to troubleshoot and potentially resolve this issue."
	CAA	"It sounds like your graphics card might be experiencing some issues. A checkerboard pattern can indicate problems with the display settings or the graphics card itself."
	RE-Control	"It sounds like your graphics card may be experiencing some issues. Check whether the graphics card is properly connected to the computer."
	RepE	"It sounds like your graphics card may be experiencing some issues. A checkerboard pattern can indicate a problem with display settings or drivers."
	ODESteer	"It sounds like your graphics card may be experiencing some issues. A checkerboard pattern can be caused by a faulty graphics card, video cable, or monitor."
Mistral-7B Instruct-v0.3	First-Order (HeRD-Merging)	"• Check your graphics card for signs of overheating: • Make sure your graphics card is not overheating by checking for physical damage, thermal throttling, and abnormal temperatures."
	First-Order (Task Arithmetic)	"• Check your graphics card for signs of overheating: • Make sure your graphics card is not overheating by checking for physical damage, thermal throttling, and abnormal temperatures."
	ODESteer	"This I— I I . . . 1 111 . . . "
	RepE	"The issue you're experiencing is a graphical artifact related to the game's graphics. The game's graphics card is not a problem; your graphics card is not a problem."
	Linear-AcT	"– . . . – . . . – . . . "
	CAA	"– . . . "
	RE-Control	"I'm sorry to hear about the issue you're experiencing with your graphics card. The checkerboard pattern is often a symptom of 'artifacting.' "
Gemma-4-E2B	First-Order (HeRD-Merging)	"That sounds like a serious issue, and it's definitely concerning if you suspect your graphics card is failing. The checkerboard pattern is a classic symptom that can point to several problems, ranging from software glitches to actual hardware failure. **Here is a step-by-step troubleshooting guide** . . . "
	First-Order (Task Arithmetic)	"That sounds like a serious issue, and it's definitely concerning if you suspect your graphics card is failing. The checkerboard pattern is a classic symptom that can point to several problems, ranging from software glitches to actual hardware failure. **Here is a step-by-step troubleshooting guide** . . . "
	Linear-AcT	"– . . . "
	CAA	"– . . . "
	RepE	"That sounds like a **serious problem**. **Immediate Action Required:** Stop using the computer immediately . . . "
	ODESteer	"That sounds like a **serious problem** and a strong indicator that something is wrong with your graphics card. **Here is a breakdown of what might be happening** . . . "
	RE-Control	"That sounds like a serious problem, and it's definitely a sign that something is wrong with your graphics card or potentially the connection."

Target Behavior: Analogy Use
Question: This page isn't working. almak.co.ke redirected you too many times. Try deleting your cookies.

Model	Method	Output
Qwen2.5-VL-3B	First-Order (HeRD-Merging)	"Imagine you're trying to reach a friend who lives in a faraway city, but every time you try to get there, you keep taking different paths that lead you back and forth between houses. It's like if you were trying to find your way from your house to your friend's house, but each time you take . . . "
	First-Order (Task Arithmetic)	"Imagine you're trying to reach a friend who lives in a faraway city, but every time you try to get there, you keep taking different paths that lead you back and forth between houses. It's like if you were trying to find your way from your house to your friend's house, but each time you take . . . "
	Linear-AcT	"It seems like the website is having trouble loading because it's trying to redirect you too many times . . . It's like if you were trying to find the right path to load the content you're looking for."
	CAA	"It sounds like you're experiencing a frustrating situation where the website is redirecting you to itself repeatedly, which is causing a 'too many redirects' error."
	RE-Control	"It seems like there is an issue with the website you are trying to access, and it's redirecting you too many times. This can happen due to various reasons such as server issues or configuration problems."
	RepE	"It seems like there is an issue with the website you are trying to access, and it's redirecting you too many times. This can happen due to various reasons such as server issues or configuration problems."
	ODESteer	"I apologize, but I don't have real-time data or the ability to access current information about specific websites or their server configurations."
Mistral-7B Instruct-v0.3	First-Order (HeRD-Merging)	"Imagine you're trying to open a door with a key. Your key is like your cookies. When you open a website, it's like you're opening a door with a key. If your cookies are missing, it's like you're trying to open a door with a key but the key . . . "
	First-Order (Task Arithmetic)	"Imagine you're trying to open a door with a key. Your key is like your cookies. When you open a website, it's like you're opening a door with a key. If your cookies are missing, it's like you're trying to open a door with a key but the key . . . "
	ODESteer	". . . . .---------- { { { { { . . . "
	RepE	"Imagine you're a gardener tending to a vineyard of books. Each book collection is like a garden of books. In this metalecture, imagine the bookshelf as a collection of books . . . "
	Linear-AcT	"Imagine your shopping shopping your shopping shopping your shopping . . . "
	CAA	"Imagine your shopping shopping your shopping . . . Imagine your choosing your choosing . . . "
	RE-Control	"It seems like the website almak.co.ke is experiencing a redirect loop, which is causing the 'ERR_TOO_MANY_REDIRECTS' error."
Gemma-4-E2B	First-Order (HeRD-Merging)	"This error message, **'This page isn't working almak.co.ke redirected you too many times. Try deleting your cookies. ERR_TOO_MANY_REDIRECTS'**, is a very common browser error. It means that your browser is stuck in an endless loop . . . "
	First-Order (Task Arithmetic)	"This error message, **'This page isn't working almak.co.ke redirected you too many times. Try deleting your cookies. ERR_TOO_MANY_REDIRECTS'**, is a very common browser error. It means that your browser is stuck in an endless loop . . . "
	Linear-AcT	"Imagine Imagine Imagine Imagine Imagine Imagine Imagine . . . "
	CAA	"Imagine imagine imagine imagine imagine imagine imagine . . . "
	RepE	"The error message **'almak.co.ke redirected you too many times'** is a very common issue that happens when a website's redirect loop goes wrong. Essentially, the website is trying to send you from Page A to Page B . . . "
	ODESteer	"}?}?}?}?}?}?}?}?}?}?}?}?}?. . . "
	RE-Control	"This error message, **'This page isn't working almak.co.ke redirected you too many times. Try deleting your cookies. ERR_TOO_MANY_REDIRECTS'**, is a very common web browser error."