Interpretable LLM Guardrails via Sparse Representation Steering

Zeqing He[1,2] Zhibo Wang[1,2] Huiyu Xu[1,2] Hejun Lin[3] Wenhui Zhang[1,2] Zhixuan Chu[1,2]
[1]The State Key Laboratory of Blockchain and Data Security, Zhejiang University, China
[2]School of Cyber Science and Technology, Zhejiang University, China
[3]College of Computer and Information Sciences, Fujian Agriculture and Forestry University, China
{hezeqing99, zhibowang, huiyuxu, wenhuizhang1222, zhixuanchu}@zju.edu.cn, inklin559@gmail.com

arXiv:2503.16851v2 [cs.CR] 14 Nov 2025

***Abstract*—Large language models (LLMs) exhibit impressive capabilities in generation tasks but are prone to producing harmful, misleading, or biased content, posing significant ethical and safety concerns. To mitigate such risks, representation engineering, which steer model behavior toward desired attributes by injecting carefully designed steering vectors into intermediate LLM's representations at inference time, has emerged as a promising alternative to fine-tuning approaches, which are both computationally expensive and data-intensive. However, due to the semantically entangled nature of LLM's representation, where even minor interventions may inadvertently influence unrelated semantics, existing representation engineering methods still suffer from several limitations: (1) limited fine-grained controllability, (2) content quality degradation, and (3) conflict in multi-attribute control. To overcome these challenges, we propose Sparse Representation Steering (SRS), a novel framework that achieves fine-grained and interpretable control over LLM behavior by first disentangling internal activations into a sparse, semantically meaningful representation space, and then selectively steering relevant dimensions. Specifically, SRS leverages a pretrained Sparse Autoencoder (SAE) to transform dense, entangled activation patterns into a sparse monosemantic feature space. To identify relevant features, SRS contrasts sparse activations from positive–negative prompt pairs and measures their bidirectional KL divergence to locate dimensions most associated with the target attribute. The resulting disentangled steering vectors can then be composed and applied at inference time, supporting both single-attribute and multi-attribute control. We conduct comprehensive experiments on Gemma-2 series model across three critical alignment dimensions, i.e., safety, fairness, and truthfulness, to evaluate the effectiveness of SRS. Results show that SRS consistently outperforms existing steering methods, which achieves significantly improved controllability across both single and multiple attribute settings, while preserving high linguistic quality and general ability.**

1. Introduction

Large language models (LLMs) [38], [43] have shown remarkable performance in natural language generation tasks, such as text completion [9], translation [31], and coding [24]. However, as LLMs are increasingly deployed in high-stakes real-world applications (such as education, healthcare, and legal assistance), safety and reliability risks of their generated content have become critical concerns [42], [21], [18]. Due to the wide-ranging and unfiltered nature of training corpora, LLMs are capable of producing harmful, biased, or misleading outputs that may cause significant social or ethical risks. A tragic example [1] is the suicide of a 14-year-old boy suffering from depression who had become heavily dependent on his AI companion. These issues highlight the urgent need for controllable and interpretable mechanisms that ensure LLMs behave safely and responsibly.

Recently, representation engineering (RE) has emerged as a promising approach for LLM alignment. Instead of retraining or fine-tuning, RE directly modifies model behavior at inference time by injecting carefully designed vectors into intermediate activations [15], [26], [19]. These activations encode rich and structured semantics across layers, capturing not only low-level linguistic features [17], [36] but also high-level attributes such as sentiment, toxicity, factuality, and ethical stance [37], [8]. Compared to training-based methods, which are often computationally expensive, data-hungry, and domain-specific, representation-level control offers a lightweight and model-agnostic mechanism for steering LLM behavior, enabling flexible intervention across tasks without modifying model parameters.

Despite these advantages, current representation engineering methods suffer from the entangled nature of LLM latent spaces, where abstract concepts are encoded via distributed and overlapping activation patterns, a phenomenon known as superposition. Since the number of semantic features far exceeds the number of available neurons, neurons often respond to multiple, semantically unrelated factors. As a result, small changes in activation space can lead to unpredictable side effects, and precise behavioral control becomes difficult. Specifically, existing methods face three key limitations: **(1). Limited fine-grained controllability.** Most methods typically modify high-dimensional activation vectors as a whole, without isolating the specific dimensions that are directly responsible for the targeted attribute (e.g., toxicity or bias), reducing the precision of control. **(2). Degradation of content quality.** Injecting dense steering signals disrupts the pretrained activation distribution, often

degrading linguistic fluency, coherence, and general utility. [30], [40]. **(3). Conflicts in multi-attribute control.** When multiple behavioral objectives (e.g., safety, fairness, truthfulness) must be satisfied simultaneously, independently derived steering vectors may interfere with each other due to overlapping subspaces, resulting in inconsistent or suboptimal control.

To address these limitations, we propose SRS, a disentangled representation steering framework that enables fine-grained and interpretable control over LLM behavior. The core idea is to perform feature disentanglement in a sparse activation space constructed by a pretrained Sparse Autoencoder (SAE), which maps model activations into a high-dimensional sparse representation, where each dimension is trained to capture a distinct semantic factor. Technically, SRS constructs disentangled steering vectors by comparing the sparse activation distributions induced by positive and negative prompt pairs, using bidirectional KL divergence to quantify the semantic sensitivity of each dimension. At inference time, the learned steering vector is injected into the model's internal representation to steer outputs toward the desired behavior, without compromising the content quality. Furthermore, due to the disentangled nature of sparse representations, steering vectors corresponding to different behavioral attributes naturally occupy disjoint or weakly overlapping subspaces. This property enables modular multi-attribute alignment, i.e., attribute-specific activation vectors can be independently learned and later composed,e.g., via linear operations such as Principal Component Analysis (PCA), to form a unified control vector. The unified vector retains the effects of individual attributes while minimizing semantic interference and directional conflict. The code is available here.

The contributions of this work are summarized as follows.

- **Sparse representation–based guardrail framework.** We propose SRS, a novel sparse representation steering framework that disentangles dense activation superposition into monosemantic sparse features which overcomes the superposition and side-effect issues prevalent in dense activation editing.
- **Mechanism for identifying and composing attribute-relevant sparse features.** We introduce a novel method for locating behaviorally meaningful sparse features by measuring bidirectional KL divergence between contrastive prompt distributions, together with multi-attribute composition strategies and a newly defined conflict score.
- **Empirical evaluation on diverse tasks.** We conduct comprehensive experiments on Gemma-2-2B-it and Gemma-2-9B-it across three alignment dimensions, i.e., safety, fairness, and truthfulness. Our evaluations cover both single-attribute and multi-attribute steering, demonstrating shat SRS achieves stronger alignment with minimal side effects, and exhibits substantial robustness against diverse prompt .

2. Related Work

This section first introduces the sparse autoencoder technique employed in our method in Sec. 2.2, and then provides an overview of the related works on representation engineering in Sec. 2.1.

2.1. Sparse Autoencoder

Sparse Autoencoder (SAE) [12], [29], [14], [32], [33] serves as a fundamental tool for interpreting and understanding deep learning models by decomposing model activations into sparse and linearly disentangled feature representations.

Specifically, an SAE consists of an encoder, denoted as f_θ, and a decoder, denoted as g_θ. Given a model activation $h \in \mathbb{R}^n$, the encoder maps h into a sparse latent representation $z \in \mathbb{R}^m$ (where $m > n$) as follows:

$$z = g_e(h) = \omega(\mathbf{W_e}h + \mathbf{b_e}), \tag{1}$$

where ω is a non-negative activation function such as ReLU. $\mathbf{W_e} \in \mathbb{R}^{m\times n}$ and $\mathbf{b_e} \in \mathbb{R}^n$ denote the weight matrix and bias vector of the encoder, respectively.

The decoder reconstructs the original activation h from the sparse code z as:

$$\hat{h} = g_d(z) = \mathbf{W_d}z + \mathbf{b_d}, \tag{2}$$

where $\hat{h}$ is the reconstruction of h, and $\mathbf{W_d}$ and $\mathbf{b_d}$ are the weight matrix and bias vector of the decoder, respectively.

The SAE is trained to minimize the loss function:

$$\mathcal{L} = |h - \hat{h}|_2^2 + \lambda|z|_1, \tag{3}$$

where the first term enforces accurate reconstruction of input activations, and the second introduces an L_1 penalty weighted by λ to promote sparsity. Through this objective, the SAE learns a high-dimensional yet interpretable latent space that captures monosemantic features within LLMs.

Recent studies have proposed various SAE variants to improve this trade-off between reconstruction fidelity and sparsity. Cunningham et al. [12] introduced a L_1-regularized SAE that maps LLM representations into a higher-dimensional feature space to interpret internal behaviors. Rajamanoharan et al. [32] proposed the Gated SAE, which balances reconstruction accuracy and sparsity by mitigating biases introduced by L_1 regularization. Gao et al. [14] further developed the Top-K SAE, employing a Top-K activation function to impose more precise sparsity constraints. JumpReLU SAEs, proposed by Rajamanoharan et al. [33], enhance the balance between reconstruction quality and sparsity by replacing the conventional ReLU activation with a discontinuous JumpReLU function.

Neuronpedia [2] serves as a tool in interpreting sparse feature spaces produced by SAEs. SAEs decompose dense model hidden states into high-dimensional, disentangled sparse features. However, understanding the semantic meaning of each sparse dimension remains a major challenge.

Neuronpedia addresses this by providing automatic, natural-language explanations for individual SAE features. These explanations are generated by aggregating tokens or prompts that maximally activate a given feature, and then using a language model to summarize their shared semantic content

2.2. Representation Engineering

Representation engineering [44], [10], [30], [25], [20] refers to the practice of identifying and manipulating latent activation directions within neural networks to modulate their behavior in a controlled and interpretable manner. Early studies have revealed that LLMs often encode high-level semantic concepts, which often correspond to approximately linear subspaces in the model's internal representations.

Gurnee et al. [15] and Marks et al. [26] provide compelling empirical evidence that such semantic concepts are geometrically encoded in activation space. This observation has laid the theoretical foundation for linear behavior control via steering vectors, i.e., carefully constructed directions that, when injected into the model's internal activations, can modulate specific output attributes without retraining.

Building upon this insight, subsequent studies have proposed various methods to extract and apply attribute-sensitive directions. For instance, CAA [30] derives steering vectors from the average activation differences between contrastive prompt pairs (e.g., toxic prompt vs. safe prompt). Belinkov et al. [10] train linear probes on intermediate representations to localize attribute-sensitive dimensions. Zou et al. [44] apply PCA over attribute-aligned prompts to discover global semantic axes, which are then used for behavior modulation or controllability analysis.

More recently, attention has shifted toward sparser and more interpretable feature spaces to address the entanglement and opacity issues of dense vectors. Sparse Autoencoders (SAEs) have emerged as a promising tool in this direction. By mapping dense activations to a high-dimensional, sparsely activated space, SAEs yield representations where each dimension encodes a more disentangled and often semantically coherent concept [12], [14].

Initial works in this area, such as Chalnev et al. [11] and O'Brien et al. [28], explore directly editing individual sparse features to drive desired behavior. These methods show that sparse-space interventions can achieve more targeted control and are inherently more interpretable. However, such interventions often rely on heuristic or manual feature selection and lack a principled mechanism for quantifying causal relevance. As a result, these methods may lead to inconsistent or suboptimal outcomes

This challenge is further highlighted by AxBench [41], a recent benchmark that systematically evaluates SAE-based steering and reveals that naive sparse manipulation often underperforms simple baselines like CAA. One critical insight from this work is that not all SAE features are behaviorally relevant: many correspond to background syntax, generic structure, or correlated but non-causal patterns. Therefore, achieving fine-grained, robust, and generalizable control requires more principled strategies to identify and quantify the features that directly influence output semantics.

3. Methodology

In this section, we introduce our sparse representation steering method (SRS). Sec 3.1 outlines the overall framework of SRS. We then describe the procedure for constructing disentangled steering vectors in Sec 3.2, followed by the inference-time integration process in Sec 3.3.

3.1. Overview

We propose SRS, a framework for interpretable and fine-grained control of LLM behavior through disentangled activation steering. The key idea is to steer LLM in a sparse latent space, where each dimension encodes a semantically independent behavioral factor. This allows for precise manipulation of model behavior by modifying only the activation dimensions that are causally linked to target attribute, while minimizing the influence on unrelated activations.

As shown in Fig. 1, the framework consists of two main stages. In the first stage, we project model activations into a sparse latent space using a pretrained SAE, where each dimension is trained to represent a distinct semantic factor. Given a target attribute (e.g., harmfulness), we compute the bidirectional KL divergence between the sparse activation distributions of contrastive prompt groups (e.g., harmful vs. safe) to quantify per-feature sensitivity. The resulting asymmetric patterns are aggregated to form a sparse steering vector, representing the direction that characterizes the desired behavioral shift. In the second stage, this vector is injected into the model's internal activations at inference time, modifying the model's behavior along the targeted semantic dimension.

SRS supports both single-attribute and multi-attribute control. Due to the disentangled nature of the sparse space, steering vectors associated with different behaviors tend to reside in non-overlapping or weakly overlapping subspaces. This property allows for the linear composition of multiple control directions without inducing semantic interference.

3.2. Sparse Steering Vector Generation

To enable fine-grained control in the sparse representation space, we identify task-relevant features by measuring distributional differences in sparse activations between positive and negative samples. Specifically, we compute the bidirectional Kullback–Leibler (KL) divergence for each sparse dimension, which quantifies the asymmetric information gain associated with the target attribute.

We begin by constructing steering vectors in single-attribute case, where the goal is to control one specific behavioral attribute (e.g., safety). We then extend this to multi-attribute