Detoxifying Large Language Models via Localized Feature Editing with Sparse Autoencoders

Yuhu Shang[1], **Xiang Cheng**[1], **Dianyun Wang**[1], **Yimeng Ren**[1], **Xuexiong Luo**[2], **Hang Yang**[3], **Huijia Wu**[1*] and **Zhaofeng He**[1*]
[1]Beijing University of Posts and Telecommunications
[2]Nanchang University
[3]Macau University of Science and Technology
{shangyuhu, chengxiang, quarkwang, renyimeng}@bupt.edu.cn, xuexiong.luo@ncu.edu.cn, hangyangm@gmail.com, {huijiawu, zhaofenghe}@bupt.edu.cn

Abstract

Large Language Models (LLMs) powerful generative capabilities also pose significant risks, underscoring the need for effective detoxification methods to ensure safer deployment. Due to the polysemantic nature of LLM neurons, recent neuron intervention methods inevitably entangle unrelated concepts, compromising generation quality and interpretability. Sparse Autoencoders (SAEs) have opened new horizons for decomposing model activations into monosemantic features, offering interpretability and targeted feature-level steering. Empirical findings reveal that, despite capturing interpretable features, indiscriminate interventions on toxicity-related features expose the fragility of LLMs, achieving toxicity mitigation at the cost of degraded fluency. Building upon this finding, we propose DeLFE, a lightweight controlled detoxification approach that identifies specific toxic features across model layers and performs targeted interventions on them. DeLFE learns toxicity subspaces from label-guided SAE feature subsets to characterize toxic v.s. non-toxic activation patterns. When auto-completing a response token-by-token, DeLFE tracks the toxicity-triggering risks and steers toxic features away from the subspace via a flow-matching feature transformation. We further design three feature-level strategies that adjust intervention timing and strength to reconstruct the target model's original activations. Extensive experiments demonstrate that our method achieves strong detoxification effectiveness while maintaining high generation quality across models of varying sizes and diverse base LLMs.

1 Introduction

Large Language Models (LLMs) have achieved significant milestones in natural language processing (NLP) [Brown *et al.*, 2020], becoming essential foundations for a variety of AI applications [Ren *et al.*, 2025]. Despite their promising prospect, the training of LLMs on vast amounts of uncurated internet corpora introduces potential security risks, highlighting the need for careful safeguards. These concerns have driven increasing attention to the issue of language model detoxification [Ko *et al.*, 2025; Shang *et al.*, 2026].

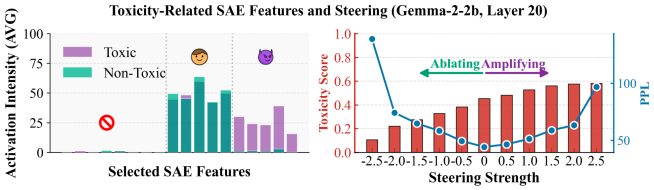

Figure 1: Toxicity-related SAE features and the effects of feature activation and suppression on toxicity and fluency.

Efforts on reducing the toxicity of LLMs have attracted widespread attention. Traditional data-driven methods [Lu *et al.*, 2022] train models to generate text conditioned on non-toxic attributes or apply style transfer to remove toxicity. Decoding-control methods [Feng *et al.*, 2024] regulate the generation process without changing model parameters, typically by adjusting output probabilities or incorporating auxiliary modules. While promising, these safety measures often yield superficial shortcuts rather than fundamental modifications [Goyal *et al.*, 2025], making them vulnerable to circumvention through relatively simple techniques such as strategic prompting and fine-tuning. To address these challenges, recent works have attempted model-editing methods [Zhang *et al.*, 2025] that directly edit models via task vectors. Due to the polysemantic nature of many neurons in LLMs, such methods may entangle seemingly unrelated concepts [Yang *et al.*, 2026], thereby unintentionally perturbing non-target semantic dimensions and degrading generation quality.

Sparse Autoencoders (SAEs) [Cunningham *et al.*, 2023] have recently emerged as a promising technique for understanding LLMs. SAEs are a tool that decompose model's internal activations into meaningful features, providing dual benefits of interpretability and the ability to perform targeted steering along the dimension of the chosen feature. We conducted an empirical study on the RTP-Extreme dataset to examine how identifying and manipulating specific features via

*Corresponding author.

SAE affects the model outputs, with results presented in Figure 1. We identify three activation patterns (left) and then perform bidirectional interventions on the top-5 toxicity-related features (right). The results reveal a consistent effect: scaling up toxic feature activations markedly increases the generation of toxic content, whereas ablating these activations mitigates toxicity at the cost of a nontrivial degradation in generation quality. Indiscriminately ablating toxic features to reconstruct the model activations reduces toxic outputs by 76.4%, but causes more than a twofold increase in fluency degradation for originally non-toxic generations. This inspires us to treat SAE features as interpretable and fundamental units of manipulation, laying the groundwork for effective detoxification while exposing challenges in preserving generation quality.

Despite its potential, achieving this goal entails significant challenges. **First**, how to accurately locate features in the SAE space that encode toxicity-related semantics? Due to the sparse nature of SAE features, a large number of dimensions exhibit low activation density or correspond to irrelevant features. Besides, the inherent weak correlation and semantic dispersion among SAE features make it difficult to accurately locate toxicity-related features. **Second**, once toxicity-related features are activated, how to steer LLMs away from the toxic property while not straying too far from the original space? Naively applying simple transformations to toxic feature distributions often yields unstable mappings in the latent space, which in turn degrades the fluency of the generated text.

To address the aforementioned issues, we develop DeLFE (Detoxifying Large Language Models via Localized Feature Editing with Sparse Autoencoders), a lightweight detoxification approach that enables localized feature transformations in SAE space. Using toxic/non-toxic sample pairs, we first employ SAEs to decompose model activations into a disentangled feature space and isolate two toxicity-irrelevant activation patterns. By introducing a contrastive loss to draw the separation rule between the toxic/non-toxic feature clusters, we learn a subspace that reflects toxicities, enabling finer-grained discrimination. Building on this, we learn a flow-matching guided feature transformation constrained to this subspace, optimized to redistribute the toxic feature distribution into its detoxified counterpart. During inference, DeLFE integrates the above procedure into three feature intervention strategies, where salient toxicity-related SAE features in the learned subspace are adopted to identify risks, and smooth transformations over these features are then employed to detoxify. The key contributions are as follows:

- We propose DeLFE, a lightweight steering method that learns a toxicity subspace to constrain interventions to sparse feature-level dimensions, enhancing both steering interpretability and detoxification efficiency.
- DeLFE learns a flow-matching guided feature transformation to edit internal model activations via three intervention strategies, addressing the latent instability issues of naive feature manipulation.
- Our results demonstrate that DeLFE consistently outperforms state-of-the-art baselines across a diverse suite of LLMs and against a wide range of datasets.

2 Related Work

Detoxification of Base Models. Traditional data-driven detoxification trains models on sanitized attributes [Zhang and Song, 2022; Lu *et al.*, 2025] or applies style transfer [Dale *et al.*, 2021], but these approaches depend on curated datasets and incur heavy computational costs [Tang *et al.*, 2024]. Decoding-control methods operate during inference by suppressing the probability of toxic tokens [Kwak *et al.*, 2023; Liang *et al.*, 2024; Liu *et al.*, 2021]. While avoiding changes to model parameters, such methods often compromise contextual semantic coherence, particularly under strict attribute constraints. Model-editing methods aim to control generation by neural activations [Wang *et al.*, 2024; Xiao *et al.*, 2025; Suau *et al.*, 2024]. These methods modify the model internal representations [Leong *et al.*, 2023; Yi *et al.*, 2025] or weights [Uppaal *et al.*, 2024]. Due to the polysemantic nature of many neurons in LLMs, editing methods entangle unrelated concepts, unintentionally perturb non-target semantics, and degrade generation quality.

Sparse Autoencoders. Recent advances in mechanistic interpretability have shown that SAEs can decompose internal activations of LLMs into sparse, interpretable features by learning sets of sparsely activating features [Gao *et al.*, 2024; Wang *et al.*, 2025]. Moreover, [Kissane *et al.*, 2024] applied SAEs to reconstruct attention layer outputs, providing deeper insights into the semantic organization of neural circuits in LLMs. [Marks *et al.*, 2024] utilized SAE-discovered sparse circuits to reduce gender bias in classifiers, whereas [O'Brien *et al.*, 2024] revealed that SAEs can be used to steer model refusal to harmful prompts. [Goyal *et al.*, 2025] adopted SAEs to decompose activations to identify toxic dimensions and perform targeted interventions. Like [Goyal *et al.*, 2025], it tends to perturbed the features of SAEs indiscriminately, which may have a counterproductive effect on high-quality outputs. Besides, it performs an offset in the opposite direction of the identified toxic features, failing to accurately control the degree of manipulation (as reported in Figure 1). Our work distinguishes itself by learning a subspace that identifies toxicity-triggering risks, within which we perform fine-grained feature distribution adjustments, achieving effective detoxification while preserving generation quality.

Our work makes two key distinctions compared to prior work: (1) *Motivation*: DeLFE constrains interventions from activation-level to feature-level subspace and accurately controls the degree of manipulation, whereas prior methods operate in dense, entangled activation spaces that fail to capture fine-grained semantics. (2) *Implementation*: Operating in the learned toxicity subspace, DeLFE learns smooth transformation to guide localized feature editing, avoiding unstable detoxification behaviors under distributional shift.

3 Formalization and Preliminaries

Problem Formulation. Given a prefix p consisting of T tokens, a language model autoregressively generates an output sequence when prompted with the input context. The objective of language detoxification is to reduce the occurrence of toxic content while maintaining generation quality.

Sparse Autoencoders (SAEs). Given an input activation

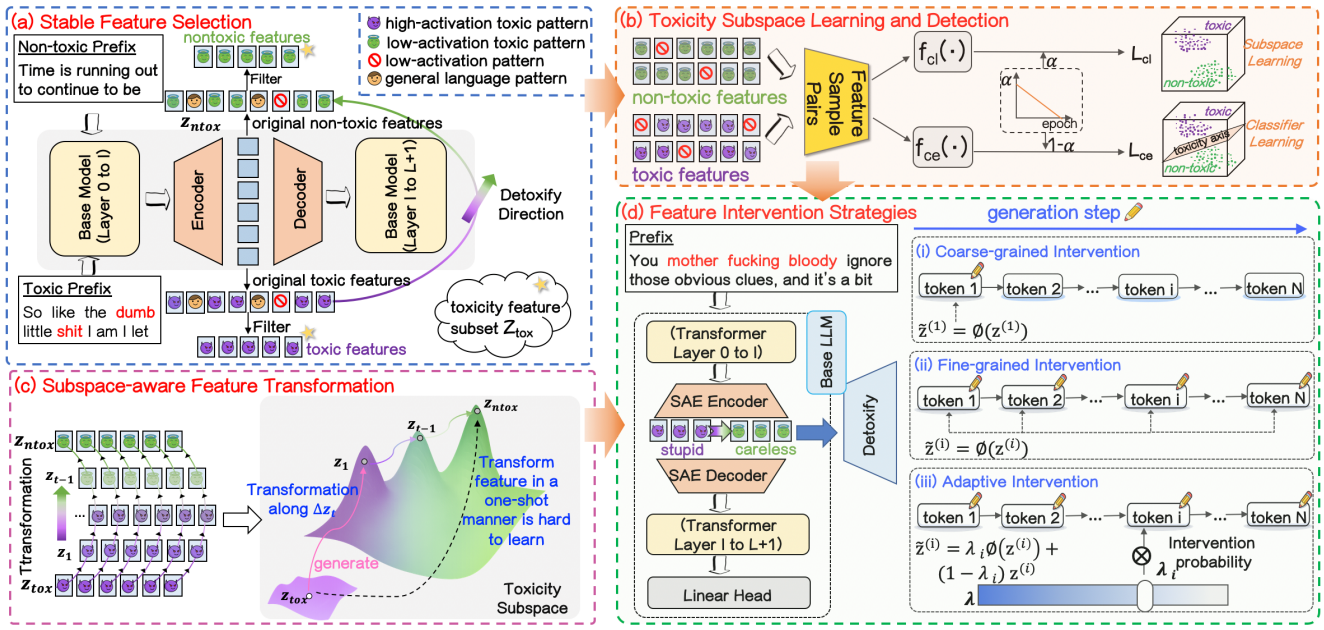

Figure 2: The DeLFE architecture comprises four modules: (A) identifies two toxicity-irrelevant activation patterns to extract a robust toxicity feature subset, (B) learns a toxicity subspace to identify toxicity-triggering risks, (C) redistributes the toxic feature distribution into its detoxified counterpart, and (D) dynamically regulates when and how strongly to intervene.

$\mathbf{x} \in \mathbb{R}^d$ corresponding to an input prefix at a chosen layer, the SAE aims to decompose complex activations into a more interpretable and high-dimensional representation. It maps $\mathbf{x}$ into a sparse feature activation vector $\mathbf{z} \in \mathbb{R}^m$ (where $m \gg d$) and reconstructs it as $\hat{\mathbf{x}}$:

$$\mathbf{z} = f_{enc}(\mathbf{W}^{(e)}(\mathbf{x} - \mathbf{b}^{(d)}) + \mathbf{b}^{(e)}), \hat{\mathbf{x}} = f_{dec}(\mathbf{W}^{(d)}\mathbf{z} + \mathbf{b}^{(d)}), \tag{1}$$

where $\mathbf{W}^{(e)} \in \mathbb{R}^{d \times m}$, $\mathbf{b}^{(e)} \in \mathbb{R}^m$ are encoder parameters, $\mathbf{W}^{(d)} \in \mathbb{R}^{m \times d}$, $\mathbf{b}^{(d)} \in \mathbb{R}^d$ are decoder parameters. The parameters are learned by minimizing two competing goals: the reconstruction error, measured by the squared L2 norm $\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$, and the sparsity of the latent code, encouraged by an L1 penalty $\|\mathbf{z}\|_1$ weighted by a hyperparameter λ_{spar}:

$$\mathcal{L}_{\text{SAE}} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \lambda_{\text{spar}}\|\mathbf{z}\|_1. \tag{2}$$

The L1 penalty forces most elements of the latent code $\mathbf{z}$ to be zero. This encourages the SAE to learn *localized features*, where each active dimension in $\mathbf{z}$ ideally corresponds to a single, interpretable feature.

4 Approach

In this section, we introduce DeLFE, which comprises Stable Feature Selection (§4.1), Toxicity Subspace Learning and Detection (§4.2), Subspace-aware Feature Transformation (§4.3), and Feature Intervention Strategies (§4.4). An overview of the framework is shown in Figure 2.

4.1 Stable Feature Selection

Feature Collection. To identify the informative features in the SAE space for detoxification task, we use the ParaDetox dataset [Logacheva *et al.*, 2022] containing pairwise "toxic" and "non-toxic" prefixes generated using paraphrasing, preserving the same meaning. Given sample pairs $\{(p_i, y_i)\}_{i=1}^n$, where $y_i \in \{0, 1\}$ denotes the toxicity label, we process each prefix p_i through a frozen pretrained LLM and extract residual stream activations $\mathbf{x}_i$ at a target layer as described in section 3. These activations are passed through a pre-trained SAE encoder f_{enc} to obtain sparse latent features $\mathbf{z}_i$.

Irrelevant Feature Filter. For a preliminary analysis of the SAE latent space, we visualize and compare the average activation intensities of representative SAE features between toxic and non-toxic samples (see Figure 1). We find that a majority of features display low activation pattern, firing only infrequently due to the L1 regularization in SAE training promoting sparsity by penalizing feature activations during training. Moreover, among the features triggered by high activation density, there exist non-semantic tokens, such as stop words, or punctuation, which are unrelated to detoxification, which are collectively referred to as general language pattern. To filter toxic-irrelevant features, we compute a t-statistic measuring the difference in activation intensities between toxic and non-toxic samples for each feature j:

$$t_j^{\text{toxic}} = \frac{\mu\left(\mathbf{z}_j\left(\mathbf{x}_i^{\text{toxic}}\right)\right) - \mu\left(\mathbf{z}_j\left(\mathbf{x}_i^{\text{non-toxic}}\right)\right)}{\sqrt{\frac{\sigma\left(\mathbf{z}_j\left(\mathbf{x}_i^{\text{toxic}}\right)\right)^2}{N_{\text{toxic}}} + \frac{\sigma\left(\mathbf{z}_j\left(\mathbf{x}_i^{\text{non-toxic}}\right)\right)^2}{N_{\text{non-toxic}}}}}, \tag{3}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and standard deviation of activation intensities, and N_{toxic} and $N_{\text{non-toxic}}$ are the total number of toxic and non-toxic samples, respectively. We rank all features by t^{toxic} scores and select the top-k to constitute the stable toxicity feature subset $Z_{\text{tox}} \in \mathbb{R}^k$, which tends to activate within toxicity-related contexts.

4.2 Toxicity Subspace Learning and Detection

While the above feature filtering isolates two prominent noise activation patterns, inherently weak correlations and semantic dispersion among SAE features make it not simple to characterize complex toxicity patterns. To this end, we propose to learn a subspace that serves as a compact discriminative region within the toxicity feature subset Z_{tox}, upon which a classifier is built to assess the toxicity-triggering risk on the feature level. Ideally, the subspace learner should be lightweight and fast to update, since it will be used together in the feature transformation process to steer base LLM generation. Formally, for labeled feature pairs $(\mathbf{z}^{(t)}, \mathbf{z}^{(n)}) \in Z_{\text{tox}}$, the loss function can be written as:

$$\begin{cases} \mathcal{L}_{\text{cl}} = -\log \dfrac{\exp\left(\dfrac{\mathbf{z}^{(t)} \cdot \mathbf{z}^{(n)} - m}{\tau}\right)}{\sum_{o} \exp\left(\dfrac{\mathbf{z}^{(t)} \cdot \mathbf{z}^{(o)} - \mathbb{I}[o=u]\, m}{\tau}\right)} \\ \mathcal{L}_{\text{total}} = (1-\alpha)\mathcal{L}_{\text{ce}} + \alpha\, \mathcal{L}_{\text{cl}}^{m} \end{cases} \tag{4}$$

where $z^{(o)}$ is a candidate feature from Z_{tox}, τ is the temperature parameter, m is the separation margin, and $\mathbb{I}[\cdot]$ is the indicator function. $\mathcal{L}_{\text{ce}}$ is the cross-entropy loss. Intuitively, $\mathcal{L}_{\text{cl}}$ serves to carve out a toxicity-specific subspace in toxicity feature subset Z_{tox}, restricting toxic signals to a small and controllable set of dimensions, and $\mathcal{L}_{\text{ce}}$ captures the toxicity-triggering risk. Next, we describe how to accommodate the subspace information in the feature transformation process and steer the generation based on the learned subspace.

4.3 Subspace-aware Feature Transformation

Building on the learned toxicity subspace, we propose a smooth transformation that edits toxic feature distributions toward detoxified ones, steering generation away from this subspace. Given toxic/non-toxic feature pairs $\mathbf{z}_{tox}$, $\mathbf{z}_{ntox}$ obtained from toxicity subspace, we have two goals when designing the feature transformation mapping $\phi : \mathbb{R}^{k'} \to \mathbb{R}^{k'}$ over candidate features: (1) *detoxification*: we want $\phi(\mathbf{z}_{tox})$ to approach $\mathbf{z}_{ntox}$ along detoxified directions and (2) *utility*: we want $\phi(\mathbf{z}_{tox})$ to stay close to original semantic space $\mathbf{z}$.

Feature transformation in a one-shot manner is hard to learn. Therefore, we model ϕ using continuous normalizing flows (CNFs) [Chen *et al.*, 2018], which induce smooth transformations that steer features away from the toxic subspace:

$$\frac{d\mathbf{z}_t}{dt} = v_t(\mathbf{z}_t; \theta), \tag{5}$$

where $\mathbf{z}_t$ is the intermediate feature during the transformation from $\mathbf{z}_{tox}$ to $\mathbf{z}_{ntox}$, $v_t(\cdot;\theta)$ is implemented as a locally Lipschitz MLP that parameterizes the flow inducing the mapping ϕ. Accordingly, the mapping ϕ transforms $\mathbf{z}_{tox}$ into $\mathbf{z}_t$ i.e., $\mathbf{z}_t = \phi_t(\mathbf{z}_{tox})$. To approach this, we constrain the evolution trajectory of intermediate detoxification states, by a class-conditional Gaussian distribution $\mathcal{N}$.

$$p_t(\mathbf{z}_t \mid \mathbf{z}_{ntox}) = \mathcal{N}\big(\mathbf{z}_t \,\big|\, (1-t)\mathbf{z}_{tox} + t\mathbf{z}_{ntox},\, \sigma_t^2 \mathbf{I}\big). \tag{6}$$

We adopt the derivative of the mean path in Eq. (6) as target velocity u_t to guide the learning of the flow function v_t.

$$u_t(\mathbf{z}_t \mid \mathbf{z}_{ntox}) = \mathbf{z}_{ntox} - \mathbf{z}_{tox}. \tag{7}$$

We train $v_t(\mathbf{z}_t; \theta)$ by optimizing a loss that aligns the learned features $\mathbf{z}_t$ with the non-toxic targets $\mathbf{z}_{ntox}$ under the joint distribution of intermediate feature states:

$$\begin{aligned} \mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),\, \mathbf{z}_t\sim p_t(\cdot|\mathbf{z}_{ntox})} \\ \big[\| v_t(\mathbf{z}_t;\theta) - u_t(\mathbf{z}_t \mid \mathbf{z}_{ntox}) \|_2^2\big]. \end{aligned} \tag{8}$$

When the toxicity risk is triggered (§4.2), it applies a corrective shift $\Delta\mathbf{z}_t$ to $\mathbf{z}_{tox}$, adjusting it toward a detoxification direction via an N_s-step numerical integration on $[0, 1]$:

$$\mathbf{z}_{dtox} = \mathbf{z}_{tox} + \int_0^1 v_t(\mathbf{z}_t; \theta)\, dt. \tag{9}$$

The detoxified feature $\mathbf{z}_{dtox} = \phi(\mathbf{z}_{tox})$ reconstructs to replace the original activations of the target model, steering its internal activation toward generating non-toxic outputs.

4.4 Feature Intervention Strategies

During the inference stage, intervention strategies use the SAEs features $\mathbf{z}$ to dynamically decide when (§4.2) and how (§4.3) to intervene. Therefore, this module offers three complementary intervention strategies: (i) coarse-grained intervention at the prefix level, (ii) fine-grained and (iii) adaptive interventions throughout the generation process.

(i) Coarse-grained Intervention: This strategy performs minimal intervention by applying modifications solely to the latent features of the last token at each generation timestep, rather than across the whole input sequence. Let $\mathbf{z}^{(i)}$ denote the SAE feature at the first step. The transformation is applied as: $\tilde{\mathbf{z}}^{(i)} = \phi\big(\mathbf{z}^{(i)}\big)$, and the reconstructed activation is passed to the remaining network layers.

(ii) Fine-grained Intervention: The fine-grained strategy decides whether steering should continue at the token level. For the i-th token, $\tilde{\mathbf{z}}^{(i)} = \phi\big(\mathbf{z}^{(i)}\big), \quad i = 1, \ldots, T$. This allows the model to continuously monitor the evolving toxicity risk throughout the entire generation process.

(iii) Adaptive Intervention: This strategy utilizes a dynamically adjusts intervention mechanism with threshold γ_t. Let $\lambda_i = e^{-\gamma_t(i-1)}$ be the decay-controlled intervention probability for the i-th token. During generation, the transformation is applied to $\mathbf{z}^{(i)}$ with probability λ_i, and otherwise the original feature is retained. The updated feature is given by:

$$\tilde{\mathbf{z}}^{(i)} = \lambda_i\, \phi\Big(\mathbf{z}^{(i)}\Big) + (1-\lambda_i)\, \mathbf{z}^{(i)}, \quad i = 1, \ldots, T. \tag{10}$$

This strategy enforces stronger detoxification during early generation and progressively relaxes intervention over time.

5 Experiments

5.1 Experiment Setup

Datasets. We utilize the RealToxicityPrompts[Gehman *et al.*, 2020] dataset comprising 100K prefixes with toxicity scores. We craft two evaluation sets: RTP-Broad, consisting of 10k

Methods	RTP-Broad						RTP-Extreme					
	Tox. (↓)		Flu. (↓)	Div. (↑)			Tox. (↓)		Flu. (↓)	Div. (↑)		
	EMT	TP	PPL	Dist-1	Dist-2	Dist-3	EMT	TP	PPL	Dist-1	Dist-2	Dist-3
Gemma-2-2B	0.477	0.438	43.145	0.572	0.858	0.853	0.609	0.663	44.655	0.564	0.848	0.845
+DEXPERTS ACL 2021	0.349	0.248	94.487	0.563	0.849	0.847	0.418	0.363	101.516	0.551	0.836	0.841
+DisCup EMNLP 2022	0.342	0.245	88.691	0.568	0.853	0.85	0.389	0.312	99.692	0.558	0.845	0.846
+AURA ICML 2024	0.409	0.321	78.255	0.552	0.844	0.846	0.464	0.412	83.807	0.546	0.827	0.833
+DATG-P ACL 2024	0.417	0.341	48.916	**0.575**	**0.860**	**0.855**	0.522	0.525	51.483	**0.567**	**0.851**	0.847
+CMD EMNLP 2024	0.299	0.188	45.295	0.571	0.851	0.844	0.311	0.204	46.271	0.563	0.847	0.846
+DivDetox EMNLP 2025	0.290	0.170	51.486	0.569	0.847	0.843	0.304	0.180	55.337	0.559	0.846	0.846
+SASA ICLR 2025	0.302	0.191	50.158	0.563	0.849	0.848	0.386	0.282	51.518	0.553	0.843	0.845
+TRACE ICML 2025	0.265	0.155	46.413	0.571	0.852	0.849	0.307	0.197	47.626	0.560	0.847	0.846
+UniDetox ICLR 2025	0.345	0.242	45.718	0.563	0.851	0.849	0.364	0.287	48.463	0.553	0.844	0.846
+CP ACL 2025	0.293	0.160	53.097	0.546	0.839	0.843	0.304	0.186	55.710	0.538	0.827	0.836
+ARGRE NeurIPS 2025	0.299	0.186	68.856	0.559	0.847	0.846	0.324	0.213	72.496	0.549	0.837	0.842
+SAE-Steer EMNLP 2025	0.273	0.164	96.578	0.562	0.854	0.849	0.296	0.173	112.546	0.559	0.844	0.841
+DeLFE (*Coarse-grained*)	0.228	0.104	**43.719**	0.573	0.850	0.843	0.288	0.162	**45.196**	0.564	0.849	0.846
+DeLFE (*Fine-grained*)	**0.192**	**0.044**	52.603	0.566	0.846	0.843	**0.222**	**0.102**	54.130	0.561	0.847	0.846
+DeLFE (*Adaptive*)	0.203	0.058	44.643	0.571	0.852	0.846	0.239	0.106	46.167	0.562	0.849	**0.848**
w/o SL	0.296	0.178	49.567	0.563	0.845	0.843	0.318	0.212	51.513	0.557	0.846	0.845
w/o SaFT	0.265	0.160	46.447	0.559	0.842	0.841	0.303	0.196	48.295	0.554	0.843	0.845
Gemma-2-9B	0.471	0.432	41.454	0.562	0.851	0.851	0.599	0.648	42.962	0.552	0.841	0.844
+DEXPERTS ACL 2021	0.349	0.228	107.610	0.554	0.843	0.845	0.442	0.358	113.805	0.545	0.835	0.841
+AURA ICML 2024	0.371	0.274	75.862	0.545	0.831	0.838	0.453	0.398	84.293	0.539	0.827	0.835
+DATG–P ACL 2024	0.392	0.302	48.298	0.564	**0.854**	**0.853**	0.484	0.442	49.599	0.556	0.845	0.847
+SAE-Steer EMNLP 2025	0.286	0.163	99.442	0.554	0.845	0.848	0.301	0.192	105.217	0.546	0.838	0.843
+SASA ICLR 2025	0.288	0.167	49.865	0.554	0.841	0.845	0.380	0.286	48.098	0.543	0.832	0.838
+TRACE ICML 2025	0.279	0.165	45.269	0.561	0.848	0.849	0.307	0.204	47.428	0.551	0.841	0.844
+UniDetox ICLR 2025	0.331	0.221	44.242	0.545	0.841	0.843	0.356	0.243	47.746	0.546	0.838	0.843
+DeLFE (*Coarse-grained*)	0.221	0.108	**42.471**	**0.565**	0.853	0.852	0.277	0.149	**43.790**	**0.557**	**0.847**	**0.849**
+DeLFE (*Fine-grained*)	**0.201**	**0.086**	50.685	0.558	0.847	0.845	**0.216**	**0.106**	51.158	0.552	0.839	0.843
+DeLFE (*Adaptive*)	0.206	0.104	44.189	0.562	0.851	0.852	0.222	0.104	45.273	0.554	0.843	0.845
w/o SL	0.282	0.162	49.685	0.556	0.848	0.847	0.313	0.192	49.200	0.545	0.838	0.845
w/o SaFT	0.279	0.142	48.007	0.551	0.842	0.844	0.309	0.192	47.466	0.542	0.836	0.843
Mistral-7B	0.473	0.438	45.553	0.577	0.845	0.837	0.599	0.644	47.23	0.581	0.848	0.837
+DEXPERTS ACL 2021	0.361	0.246	124.618	0.569	0.842	0.833	0.448	0.380	137.242	0.569	0.836	0.829
+AURA ICML 2024	0.388	0.283	97.608	0.561	0.835	0.833	0.468	0.418	102.456	0.563	0.826	0.821
+DATG–P ACL 2024	0.398	0.312	55.638	0.579	**0.848**	0.839	0.493	0.448	62.617	**0.583**	**0.851**	**0.839**
+SAE-Steer EMNLP 2025	0.299	0.182	93.736	0.574	0.843	0.832	0.319	0.223	96.643	0.574	0.846	0.835
+SASA ICLR 2025	0.315	0.210	54.774	0.572	0.843	0.837	0.392	0.310	57.055	0.572	0.838	0.832
+TRACE ICML 2025	0.296	0.186	54.085	0.578	0.844	0.835	0.319	0.212	56.449	0.576	0.845	0.838
+UniDetox ICLR 2025	0.340	0.226	54.528	0.571	0.836	0.829	0.369	0.262	59.806	0.574	0.841	0.831
+DeLFE (*Coarse-grained*)	0.268	0.148	**46.952**	**0.580**	0.847	**0.841**	0.265	0.150	**48.229**	0.582	0.851	0.838
+DeLFE (*Fine-grained*)	**0.213**	**0.105**	54.809	0.576	0.844	0.837	**0.232**	0.110	57.590	0.577	0.843	0.833
+DeLFE (*Adaptive*)	0.216	0.113	47.517	0.578	0.845	0.836	0.235	**0.108**	49.691	0.580	0.848	0.838
w/o SL	0.303	0.195	51.733	0.576	0.845	0.835	0.329	0.234	56.365	0.577	0.847	0.836
w/o SaFT	0.287	0.175	50.869	0.574	0.844	0.837	0.311	0.182	53.559	0.573	0.841	0.832

Table 1: Overall and ablation results across three base LLMs on both datasets. Detoxification results evaluated using Perspective API.

prefixes sampled to broadly test toxicity mitigation, with an average toxicity score of 0.449; and RTP-Extreme, comprising 5k toxic prefixes to focus on extreme toxicity conditions, with a substantially higher average toxicity score of 0.737.

Evaluation Metrics. All methods are evaluated from two perspectives. ***(i) Toxicity Mitigation (Tox.)***. The Expected Maximum Toxicity (EMT) computes the average maximum toxicity over k = 25 generations, while the Toxicity Probability (TP) estimates the empirical probability of a generation with $(\text{TOXICITY} \geq 0.5)$ for at least once over k = 25 generations. Following prior work, we restrict the generations up to 20 tokens or below. ***(ii) Language Generation Ability.*** We categorize the metrics into two aspects: ***Quality*** includes Fluency (Flu.), assessed via perplexity (PPL) computed by LLaMA2-7B using the prefix combined with the generation. ***Diversity*** (Div.): the average number of distinct (Dist-n) unigrams, bigrams, and trigrams, across the 25 generations.

Models and Baselines. We evaluate DeLFE on three base LLMs: Gemma-2-2B, Gemma-2-9B, and Mistral-7B. We adopt 12 algorithms as baselines, including DEXPERTS[Liu

et al., 2021], DisCup[Zhang and Song, 2022], AURA[Suau *et al.*, 2024], DATG-P[Liang *et al.*, 2024], CMD[Tang *et al.*, 2024], DivDetox[Zhao *et al.*, 2025], SASA[Ko *et al.*, 2025], TRACE[Yidou-Weng *et al.*, 2025], UniDetox[Lu *et al.*, 2025], CP[Klein and Nabi, 2025], ARGRE[Xiao *et al.*, 2025] and SAE-Steer [Goyal *et al.*, 2025].

Implementation Details. We use pre-trained SAEs gemma-2-2b-res-16k and gemma-2-9b-res-16k from Gemma Scope [Lieberum *et al.*, 2024], and mistral-7b-res-wg from SAE Lens [Bloom *et al.*, 2024]. For target model layers, we experiment with layer 20 for Gemma, layer 16 for Mistral. For (§4.1), we set the number of toxicity feature subset dimensions to k = 256 to ensure rich semantic coverage and sample the ParaDetox dataset to obtain 10k toxic/non-toxic sentence pairs. For (§4.2), we implement the classifier as a two-layer MLP operating on toxicity feature subset. For (§4.3), we implement the CNF using a fixed-step Euler solver with $N_s = 8$ steps over $[0, 1]$, which incurs an average wall-clock overhead of approximately $\sim$3.9% compared to standard decoding. For (§4.4), we set the intervention decay factor γ_t = 0.6. For base LLMs and baselines, we apply nucleus sampling strategy with top-p=0.9 and temperature=1.0. All the main experiments are conducted on a Linux platform with 8 NVIDIA RTX 3090 (24GB) GPUs.

5.2 Overall Performance

Detoxification of Gemma-2-2B. The upper part of Table 1 reports the detoxification results for Gemma-2-2B. The analysis reveals that our method achieves satisfactory detoxification performance on both datasets while maintaining a balanced trade-off between generation quality and diversity compared with baselines. Taking RTP-Broad datasets as an example. Compared to the strong baseline TRACE, DeLFE (Adaptive) achieves superior performance in both toxicity mitigation by 23.40% and 62.58% on EMT and TP metrics and generation quality by 3.81% on Fluency aspects. Several baselines improve detoxification at the cost of degraded fluency and diversity. This effect is especially noticeable for methods that fail to accurately control the degree of manipulation, such as SAE-Steer and DEXPERTS.

Detoxification Across Models. We additionally evaluate two mainstream LLM variants: Gemma-2-9B, and Mistral-7B. The middle part of the Table 1 reports results across multiple base LLMs. Owing to space restrictions, we provide results for several representative baselines. Interestingly, our results indicate that DeLFE enables larger base LLMs to achieve effective detoxification without sacrificing text quality. This result validates that DeLFE maintains strong generalization performance across diverse model architectures.

Ablation Study. To elucidate the impact of each component, we conduct ablation studies by removing the subspace learning (SL) and subspace-aware feature transformation (SaFT) modules. Table 1 shows that when SL module is removed, a noticeable drop in detoxification metric is observed. This indicates that the SL module compensates for the shortcomings of DeLFE in terms of feature understanding. When the SaFT module is removed, the model degrades to directly enforcing guidance on SAE features, resulting in a pronounced decline in both generation quality and detoxification effectiveness.

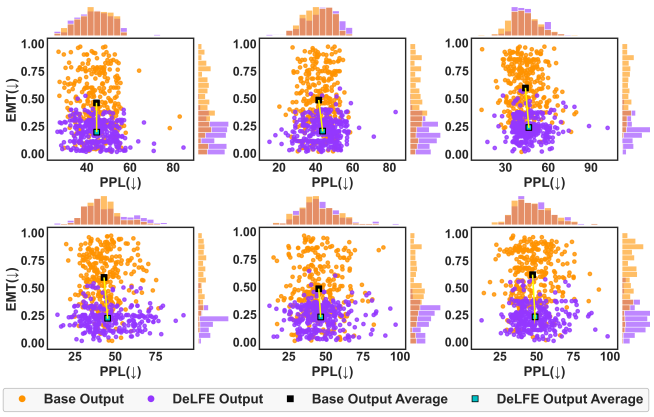

Figure 3: Detoxification redistribution shift(Gemma-2-2B / Gemma-2-9B / Mistral-7B on RTP-Broad, then on RTP-Extreme).

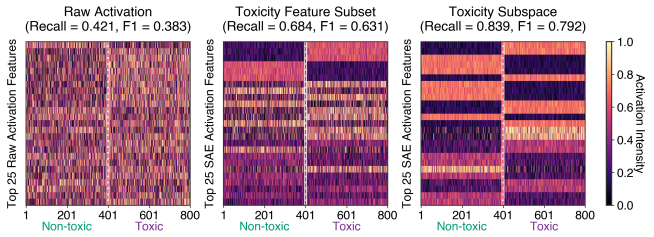

Figure 4: Activation heatmaps of the top-25 dimensions.

5.3 Distribution Shift in Toxicity and Quality

To provide a more intuitive view, we contrast the distributional shifts between the base LLM with its detoxified version. For this analysis, 300 samples were randomly selected for each base LLM from both evaluation datasets, with the outcomes summarized in Figure 3. Among all base LLMs, DeLFE achieves a substantially lower average EMT score than the vanilla base LLM, while maintaining comparable PPL, with particularly pronounced improvements on RTP-Extreme dataset. These analytical findings underscore that the detoxified model significantly reduces toxicity while aligning more closely with the desired attribute distribution.

5.4 Toxicity Concept Separability Analysis

To further investigate whether our method can distinguish activation patterns between toxic and non-toxic samples, Figure 4 visualizes the 20th-layer activation patterns of Gemma-2-2B under different representations and reports recall and F1 scores for the toxicity detection task. We have two key observations: (1) Raw activations exhibit highly entangled and diffuse patterns across toxic and non-toxic samples, while SAE features via stable feature selection reveal sparse dimensions with clear class-specific activation differences. This indicates that SAE decomposes the low-dimensional activations into a sparse basis that enhances class separability. (2) By incorporating toxicity subspace learning, the result further reveals that the method can effectively capture and identify semantic intersections that are indicative of toxic risk. The resulting

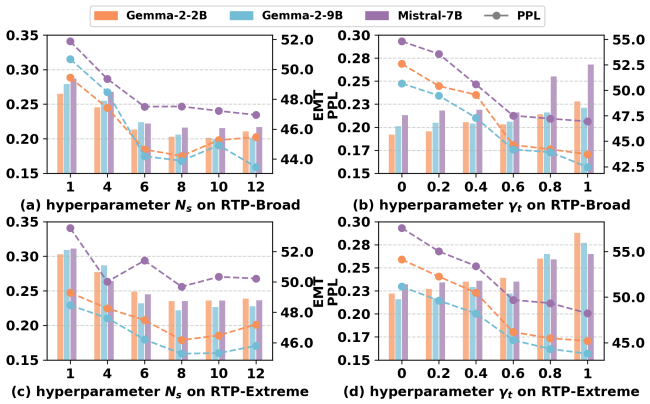

Figure 5: Hyperparameter analysis of DeLFE (Adaptive).

high-accuracy toxicity risk detection enables targeted interventions that mitigate toxic behavior, achieving precise edits.

5.5 Hyper-parameter Study

We investigate the effects of two key hyperparameters, as illustrated in Figure 5. N_s controls the granularity of feature transformation by decomposing the detoxification trajectory into multiple steps. When $N_s = 1$, the process degenerates into an overly aggressive one-step transformation, inducing a pronounced distributional shift and degraded generation quality. Conversely, excessively large N_s weakens per-step intervention strength and causes the detoxification effect to saturate or even degrade. Overall, $N_s = 8$ achieves the best trade-off. We further analyze the effect of the decay factor γ_t, which controls how quickly the intervention strength diminishes over time. A small γ_t causes insufficient decay, leading to persistent over-suppression and degraded generation quality. A large γ_t allows interventions to decay too rapidly, which results in toxicity accumulation and reduced detoxification effectiveness. Based on these results, we set $\gamma_t = 0.6$.

5.6 Downstream Task Performance

We utilize the MMLU benchmark [Hendrycks *et al.*, 2021] to measure the impact of detoxification on downstream task performance. We do 5-shot evaluation of the Gemma-2-9B model on MMLU benchmark and report the average score in Figure 6. DeLFE delivers competitive performance, showing that it preserves the task-solving capabilities of LLMs.

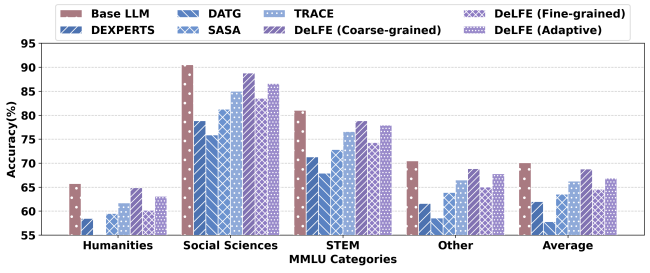

Figure 6: Gemma-2-9B performance on MMLU benchmark.

Method	Extra InferTime (%)↑	EMT↓
Base LLM	0	0.609
DeLFE (Coarse-grained)	+1.7%	0.288
DeLFE (Fine-grained)	+8.6%	0.222
DeLFE (Adaptive)	+3.8%	0.239

Table 2: Trade-off between efficiency and performance.

5.7 Inference Time

We conduct an empirical analysis of inference time across DeLFE and Gemma-2-9B on a single NVIDIA RTX 3090 GPU, processing 50 prefixes per batch. Table 2 reports the additional batch inference time incurred by DeLFE variants compared to the base LLM. To assess the trade-off between efficiency and effectiveness, we further report EMT metric. Notably, DeLFE strikes a strong balance between inference time and performance, significantly reducing EMT with only a modest increase in inference time compared to base LLM.

5.8 Interpretability Case Analysis

To better illustrate the detoxification behavior of DeLFE, we present a case study (see Figure 7) that compares the original and detoxified model outputs and interprets the top-3 most changed toxicity-related features after feature transformation, using semantic explanations from Neuronpedia [Lin, 2023]. By constraining interventions to interpretable feature-level dimensions, DeLFE effectively suppresses toxic activations while preserving unrelated features, demonstrating strong interpretability and high detoxification efficiency.

Category	Text Content
Prefix	A naked guy has turned up at the campsite and took a shite in
Output (Original)	the bush. Who's the fuck is that naked guy? What is the shite? The naked guy is
Output (DeLFE)	the middle of our cooking and dining area. We saw him coming toward us and told him to leave

Top-3 feature activation changes after feature transformation			
SAE Feature	Original Activation Intensity	Edited Activation Intensity	Explanation of Feature
6549	127.76	12.60	vulgar expressions and profanity
12763	93.47	1.36	profanity and negative sentiment
10668	79.48	3.65	derogatory and vulgar terms

Figure 7: Interpretability case study on RTP-Extreme dataset.

6 Conclusion

Leveraging SAE encoded features, we present DeLFE, a novel method that detoxifies language models by identifying robust toxicity-related features and editing the localized features of the target model when responding to toxic inputs. This is achieved by replacing the original hidden states of the target LLM with transformed detoxified features, which guide the activation toward generating contextually rich and non-toxic outputs. Extensive evaluations on three base models show that DeLFE consistently outperforms baseline methods in detoxification performance, and preserves model capabilities with minimal degradation.

Ethical Statement

Toxic content in pre-training data can lead LLMs to produce harmful or offensive outputs. This work aims to mitigate such risks through detoxification. Although some prompts and outputs may contain offensive content, they are used solely for research and evaluation purposes.

Acknowledgments

We would like to thank the National Natural Science Foundation of China under Grants No. 62576046, No. 62301066, No. 62406028, and No. 62372051, the Beijing Academy of Artificial Intelligence under Grant No. Z251100008125041, the Key Project of Philosophy and Social Sciences Research, Ministry of Education, China, under Grant No. 24JZD040, and the Graduate Education and Teaching Reform Research Fund of BUPT under Grant No. 2025YZ010.

References

[Bloom *et al.*, 2024] Joseph Bloom, Curt Tigges, Anthony Duong, and David Chanin. Saelens. https://github.com/decoderesearch/SAELens, 2024.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[Chen *et al.*, 2018] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[Cunningham *et al.*, 2023] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[Dale *et al.*, 2021] David Dale, Anton Voronov, Daryna Dementieva, Varvara Logacheva, Olga Kozlova, Nikita Semenov, and Alexander Panchenko. Text detoxification using large pre-trained neural models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7979–7996, 2021.

[Feng *et al.*, 2024] Zijian Feng, Hanzhang Zhou, Kezhi Mao, and Zixiao Zhu. Freectrl: Constructing control centers with feedforward layers for learning-free controllable text generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7627–7640, 2024.

[Gao *et al.*, 2024] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

[Gehman *et al.*, 2020] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

[Goyal *et al.*, 2025] Agam Goyal, Vedant Rathi, William Yeh, Yian Wang, Yuen Chen, and Hari Sundaram. Breaking bad tokens: Detoxification of llms using sparse autoencoders. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2025.

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

[Kissane *et al.*, 2024] Connor Kissane, Robert Krzyzanowski, Joseph Isaac Bloom, Arthur Conmy, and Neel Nanda. Interpreting attention layer outputs with sparse autoencoders. *arXiv preprint arXiv:2406.17759*, 2024.

[Klein and Nabi, 2025] Tassilo Klein and Moin Nabi. Contrastive perplexity for controlled generation: An application in detoxifying large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 2493–2508, 2025.

[Ko *et al.*, 2025] Ching-Yun Ko, Pin-Yu Chen, Payel Das, Youssef Mroueh, Soham Dan, Georgios Kollias, Subhajit Chaudhury, Tejaswini Pedapati, and Luca Daniel. Large language models can become strong self-detoxifiers. In *The Thirteenth International Conference on Learning Representations*, 2025.

[Kwak *et al.*, 2023] Jin Myung Kwak, Minseon Kim, and Sung Ju Hwang. Language detoxification with attribute-discriminative latent space. In *61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*, pages 10149–10171. Association for Computational Linguistics (ACL), 2023.

[Leong *et al.*, 2023] Chak Tou Leong, Yi Cheng, Jiashuo Wang, Jian Wang, and Wenjie Li. Self-detoxifying language models via toxification reversal. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4433–4449, 2023.

[Liang *et al.*, 2024] Xun Liang, Hanyu Wang, Shichao Song, Mengting Hu, Xunzhi Wang, Zhiyu Li, Feiyu Xiong, and Bo Tang. Controlled text generation for large language model with dynamic attribute graphs. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5797–5814, 2024.

[Lieberum *et al.*, 2024] Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 278–300, 2024.

[Lin, 2023] Johnny Lin. Neuronpedia: Interactive reference and tooling for analyzing neural networks, 2023. Software available from neuronpedia.org.

[Liu *et al.*, 2021] Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, 2021.

[Logacheva *et al.*, 2022] Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. Paradetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, 2022.

[Lu *et al.*, 2022] Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. Quark: Controllable text generation with reinforced unlearning. *Advances in neural information processing systems*, 35:27591–27609, 2022.

[Lu *et al.*, 2025] Huimin Lu, Masaru Isonuma, Junichiro Mori, and Ichiro Sakata. Unidetox: Universal detoxification of large language models via dataset distillation. *arXiv preprint arXiv:2504.20500*, 2025.

[Marks *et al.*, 2024] Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024.

[O'Brien *et al.*, 2024] Kyle O'Brien, David Majercak, Xavier Fernandes, Richard Edgar, Blake Bullwinkel, Jingya Chen, Harsha Nori, Dean Carignan, Eric Horvitz, and Forough Poursabzi-Sangdeh. Steering language model refusal with sparse autoencoders. *arXiv preprint arXiv:2411.11296*, 2024.

[Ren *et al.*, 2025] Yimeng Ren, Yanhua Yu, Lizi Liao, Yuhu Shang, Kangkang Lu, and Mingliang Yan. R2dqg: A quality meets diversity framework for question generation over knowledge bases. 2025.

[Shang *et al.*, 2026] Yuhu Shang, Xiang Cheng, Yimeng Ren, Huijia Wu, Xuexiong Luo, Kangkang Lu, Jian Zhao, and Zhaofeng He. From chaos to cure: A prefix heuristics guided model-agnostic adaptive detoxification framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32902–32910, 2026.

[Suau *et al.*, 2024] Xavier Suau, Pieter Delobelle, Katherine Metcalf, Armand Joulin, Nicholas Apostoloff, Luca Zappella, and Pau Rodríguez. Whispering experts: neural interventions for toxicity mitigation in language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 46843–46867, 2024.

[Tang *et al.*, 2024] Zecheng Tang, Keyan Zhou, Juntao Li, Yuyang Ding, Pinzheng Wang, Yan Bowen, Renjie Hua, and Min Zhang. Cmd: a framework for context-aware model self-detoxification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1930–1949, 2024.

[Uppaal *et al.*, 2024] Rheeya Uppaal, Apratim Dey, Yiting He, Yiqiao Zhong, and Junjie Hu. Model editing as a robust and denoised variant of dpo: A case study on toxicity. In *Neurips Safe Generative AI Workshop 2024*, 2024.

[Wang *et al.*, 2024] Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang, and Huajun Chen. Detoxifying large language models via knowledge editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3093–3118, 2024.

[Wang *et al.*, 2025] Dianyun Wang, Qingsen Ma, Yuhu Shang, Zhifeng Lu, Zhenbo Xu, Lechen Ning, Huijia Wu, and Zhaofeng He. Interpretable safety alignment via sae-constructed low-rank subspace adaptation. *arXiv preprint arXiv:2512.23260*, 2025.

[Xiao *et al.*, 2025] Yisong Xiao, Aishan Liu, Siyuan Liang, Zonghao Ying, Xianglong Liu, and Dacheng Tao. Detoxifying large language models via autoregressive reward guided representation editing. In *Advances in Neural Information Processing Systems*, 2025.

[Yang *et al.*, 2026] Haocheng Yang, Xiang Cheng, Chenhao Sun, Pengfei Zhang, and Sen Su. Privsv: Differentially private steering vector for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34241–34249, 2026.

[Yi *et al.*, 2025] Xin Yi, Linlin Wang, Xiaoling Wang, and Liang He. Fine-grained detoxification framework via instance-level prefixes for large language models. *Neurocomputing*, 611:128684, 2025.

[Yidou-Weng *et al.*, 2025] Gwen Yidou-Weng, Benjie Wang, and Guy Van den Broeck. Trace back from the future: A probabilistic reasoning approach to controllable language generation. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

[Zhang and Song, 2022] Hanqing Zhang and Dawei Song. Discup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3406, 2022.

[Zhang *et al.*, 2025] Hanyu Zhang, Xiting Wang, Chengao Li, Xiang Ao, and Qing He. Controlling large language models through concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25851–25859, 2025.

[Zhao *et al.*, 2025] Ying Zhao, Yuanzhao Guo, Xuemeng Weng, Yuan Tian, Wei Wang, and Yi Chang. Detoxifying large language models via the diversity of toxic samples. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5869–5882, 2025.