Redefining Experts: Interpretable Decomposition of Language Models for Toxicity Mitigation

Zuhair Hasan Shaik[*]
MBZUAI, UAE
zuhair.shaik@mbzuai.ac.ae

Abdullah Mazhar
IIIT Delhi, India
abdullahm@iiitd.ac.in

Aseem Srivastava[*]
MBZUAI, UAE
aseem.srivastava@mbzuai.ac.ae

Md. Shad Akhtar
IIIT Delhi, India
shad.akhtar@iiitd.ac.in

Abstract

Large Language Models have demonstrated impressive fluency across diverse tasks, yet their tendency to produce toxic content remains a critical challenge for AI safety and public trust. Existing toxicity mitigation approaches primarily manipulate individual neuron activations, but these methods suffer from instability, context dependence, and often compromise the model's core language abilities. To address these shortcomings, we investigate three key questions: the stability of neuron-level toxicity indicators, the advantages of structural (layer-wise) representations, and the interpretability of mechanisms driving toxic generation. Through extensive experiments on `Jigsaw` and `ToxiCN` datasets, we show that aggregated layer-wise features provide more robust signals than single neurons. Moreover, we observe conceptual limitations in prior works that conflate toxicity *detection* experts and *generation* experts within neuron-based interventions. To mitigate this, we propose a novel principled intervention technique, `EigenShift`, based on eigen-decomposition of the language model's final output layer. This method selectively targets generation-aligned components, enabling precise toxicity suppression without impairing linguistic competence. Our method requires no additional training or fine-tuning, incurs minimal computational cost, and is grounded in rigorous theoretical analysis.

1 Introduction

The rise of large language models has brought remarkable advancements across tasks involving human-like generation [19, 36, 18, 3, 24, 25]. However, this progress has also led to a surge in toxic and unsafe machine-generated content, raising urgent concerns for AI safety and societal impact. Addressing this issue demands not only effective toxicity detection but also interventions that reduce toxic generation without hampering a model's fluency or general language understanding.

Prior work has primarily focused on neuron-level interventions, where specific neurons, often termed *concept experts*, are identified and manipulated based on their activation in response to toxic inputs [5, 6, 21]. However, neuron activations are often stochastic and brittle, leading to inconsistent behavior across contexts [5], which we also show in our analysis on the way. Furthermore, interventions at early layers propagate disruption downstream, resulting in catastrophic forgetting and significant degradation in language modeling performance [9]. To understand these aspects, we raise three research questions (RQs): ►**RQ1:** Are individual neurons reliable indicators of toxicity, or do their

[*]Work done at FLaME-NLP Lab, IIIT Delhi.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

activations merely reflect unstable correlations? ►**RQ2:** Can layer-wise or structural representations capture toxicity more robustly, particularly across multilingual and specialized domains? ►**RQ3:** Can we uncover interpretable components beyond just layers and neurons that contribute to toxic generation, aiming to make black box models more understandable?

In our work, we conduct a comprehensive analysis using two toxicity detection datasets: `Jigsaw` (English) [14] and `ToxiCN` (Chinese) [17]. We compare the reliability of individual neuron activations against aggregated layer-wise representations in detecting toxic content. Further, we delve deeper into the LLMs and identify the limited ability to distinguish between two crucial roles: (a) experts that detect toxicity, and (b) experts that generate toxic outputs. Most existing interventions conflate these two, suppressing both types of activations, which inadvertently damages the model's ability to recognize toxicity, leading to catastrophic forgetting and degraded language performance [9].

Hence, we propose the problem of controlling toxic generation while preserving the linguistic competence of LLMs. To address this, we propose a novel framework, `EigenShift`. It targets the *generation experts*, a set of interpretable eigen-components in the model's final decision space (LM head) while preserving the detection experts that are essential for identifying toxicity. We consider these principal vectors as the *choices* of the LLM: in the absence of personality or intent, an LLM's behavior is governed by probability distributions. These eigen-components substantially influence the selection of output tokens given input prompts. By selectively controlling or steering these generation-specific directions, our eigen-decomposition-based strategy enables precise control over toxic outputs without disturbing the model's core language capabilities. We evaluate our method on the standard `RealToxicPrompts` dataset and demonstrate that it generalizes better than prior approaches. Also, we notice that existing metrics fall short in quantifying the trade-off between toxicity mitigation and language modeling retention. Hence, we propose a new composite metric that takes the **H**armonic mean of **T**oxicity reduction and **P**erplexity preservation (TPH score). `EigenShift` reduces toxicity by 58% while minimal change in perplexity ($\Delta PPL = +3.62$), achieving the notable TPH Score of 60.37%. `EigenShift` outperforms all baselines on TPH across all LLMs except MPT-7B, where its performance is on par with Aura. Our contributions are summarized as follows[2]:

- We provide a comprehensive empirical analysis questioning the reliability of neurons in toxicity detection, demonstrating their stochastic and context-sensitive nature across languages.
- We propose a novel method, `EigenShift`, a decomposition-based perspective that implements targeted intervention strategy, suppressing toxic outputs with *generation experts* while preserving the detection capacity and overall linguistic competence.
- We propose a new metric, TPH score, a unified and interpretable evaluation metric based on the harmonic mean of toxicity reduction and perplexity preservation.
- We show state-of-the-art performance on the RealToxicPrompts benchmark, with substantial toxicity reduction and keeping low perplexity and setting a safer LLM intervention standard.

2 Methodology

In this work, we address the problem of evaluating concept expertise within neural networks, specifically for detecting toxicity as a binary concept. The input to the network consists of text sentences ($S = (s_1, s_2, \ldots, s_n)$) from the dataset (S), where each (s_i) represents a sentence in the input. The output is a binary classification vector ($y = (y_1, y_2, \ldots, y_n)$) where each (y_i) is either 0 or 1, indicating whether the corresponding sentence (s_i) has the concept of toxicity or not. Given the input sentence (s_i), the network produces a hidden representation (h_i), and the function ($f_\theta(h_i)$) with parameters (θ) returns a prediction of whether the representation (h_i) encodes the concept of toxicity or not, i.e., ($f_\theta(h_i) \in \{0, 1\}$). While previous works focus on individual neurons as concept detectors, we propose a paradigm shift by examining semantic information capture at the latent space level, specifically through the lens of eigenvectors.

From Layer-wise Experts to Eigen Choices. The limitations of neuron-level analysis, such as instability and inconsistent context sensitivity, motivated us to assess layer-level expertise, hypothesizing that entire layers may encode semantic concepts more robustly in their high-dimensional feature spaces. While this shift already improves reliability in toxicity detection, it remains too coarse to pinpoint interpretable mechanisms. We seek a more granular, structured understanding of

[2]Code Repository: `https://github.com/flamenlp/EigenShift`

Hypothesis:

The final linear layer (*lm_head*) of a language model, represented by the weight matrix W, can be decomposed into two matrices (W = BA), where one matrix (A) captures high-level semantic choices and the other (B) maps these choices to actual vocabulary tokens through a linear transformation. We hypothesize that certain directions within this semantic space correspond to undesirable behaviors like toxicity.

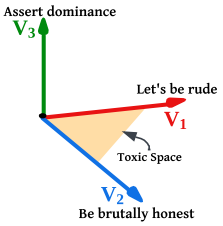

Figure 1: Illustration of the semantic space of eigenvectors V_t derived from the decomposition of the final linear layer $W = BA$ in a language model. This space captures high-level semantic directions such as *asserting dominance*, *being brutally honest*, or *being rude*, which collectively form regions (e.g., the "Toxic Space") corresponding to undesirable behaviors like toxicity.

how language models encode and act upon concepts like toxicity. Our findings state that the final linear layer in language models (often referred as *lm_head*) serves as a critical semantic decision point. We hypothesize that this layer can be decomposed into two components: one representing semantic choices (conceptual directions in latent space), and the other mapping these choices to specific vocabulary tokens as illustrated in Figure 1. Formally, the output logits are computed as: $z = Wh, \quad W \in \mathbb{R}^{V \times d}$. Here, W is the weight matrix of the `lm_head`; for LLaMA-based models, typically $V = 32000$ and $d = 4096$. The vector $h \in \mathbb{R}^d$ denotes the hidden state from the final decoder layer, and $z \in \mathbb{R}^V$ represents the output logits over the vocabulary. Next, we apply Singular Value Decomposition to factorize W as: $W = U\Sigma V^T; \quad B = U, A = \Sigma V^T$.

In this decomposition, $V^\top \in \mathbb{R}^{d \times d}$ defines an orthonormal basis spanning the semantic subspace of hidden states, $\Sigma \in \mathbb{R}^{d \times d}$ is a diagonal matrix containing the singular values that weighs each semantic direction (eigen vector) of a linear transformation, and $U \in \mathbb{R}^{V \times d}$ is a linear transformation that maps these semantic directions to vocab tokens.

It is important to note that SVD is not a perfect factorization; the approximation quality is empirically strong. We compute the Frobenius reconstruction loss across several language models (cf. Appendix C.4)and find that the loss is negligible, with no measurable degradation in perplexity. Table 5 shows reconstruction error and perplexity comparisons is included in the Appendix.

Eigen Choices as Semantic Decision Axes. We posit that each column vector of V in the SVD of the output projection matrix $W = U\Sigma V^\top$ corresponds to a fundamental semantic axis (what we refer to as an 'eigen choice') along which the language model makes decisions during text generation. These directions can be interpreted as the principal semantic dimensions in the model's latent space, encoding abstract attributes such as syntactic structure, sentiment, formality, and, most importantly for our purpose, *toxicity*.

Given a hidden state $h \in \mathbb{R}^d$, we project it onto the semantic basis defined by $V^\top$ to obtain activations along each eigen choice as: $a_i = v_i^\top h$. Here, v_i denotes the i^{th} eigen vector of V^T. The scalar a_i reflects the degree to which the hidden state h aligns with the semantic direction v_i, quantifying the model's implicit choice along that dimension. We hypothesize that certain eigenvectors (v_{toxic}) are systematically associated with the generation of toxic content. That is, for specific semantic directions, high activation values a_i are strongly correlated with toxicity in the model's output.

Detecting Eigenvector-Based Semantic Choices. To identify the semantic directions in the final projection layer (`lm_head`), we analyze the eigenvectors of the matrix $A = \Sigma V^\top$, derived from the SVD $W = U\Sigma V^\top$. Our working hypothesis is that the generation of a toxic token at position n is causally influenced by the semantic choice encoded in the preceding hidden state $h_{n-1} \in \mathbb{R}^d$.

Let $A \in \mathbb{R}^{d \times d}$ serve as the transformation matrix projecting hidden states onto a semantic basis. For each hidden state h_j, corresponding to either a toxic or non-toxic example, we compute its activation along the i^{th} semantic axis as: $a_i^{(j)} = v_i^\top h_j, \forall$sample j and eigenvector i, where v_i denotes the i^{th} eigen vector (equivalently, the i^{th} row of A), and h_j is a sample-specific (toxic or non-toxic) hidden state. Next, to quantify the association between each semantic direction and toxicity, we compute the average activation of each eigenvector over toxic and non-toxic examples, and define the directional

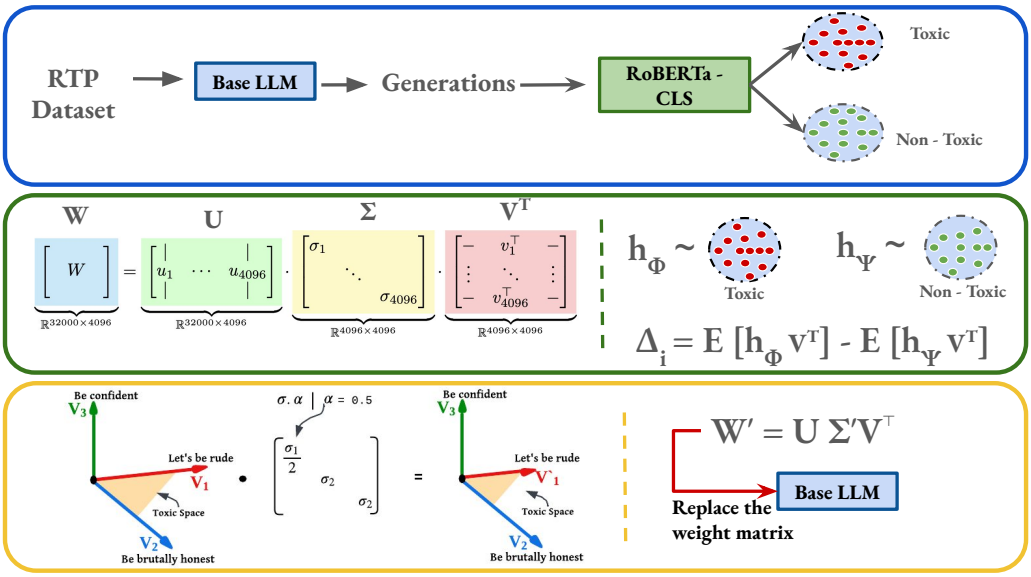

Figure 2: **Overview of our proposed framework `EigenShift` for semantic axis discovery and intervention.** We begin by sampling generations from the base language model and employing a classifier to identify toxic tokens within each sentence. Next, we apply Singular Value Decomposition (SVD) to the output projection matrix of the base model, $W = U\Sigma V^\top$, where the columns of V, referred to as *eigenchoices* (cf. Hypothesis Section 2), represent interpretable semantic directions in the generation space. For each hidden state h either sampled from toxic samples (Φ) or non toxic samples (Ψ), we project it onto eigen vectors of the matrix ($V^\top$) to obtain activation values, which reflect the model's alignment with toxic generation choice. To discover directions associated with toxicity, we compute the mean activation difference between toxic (h_Φ) and non-toxic (h_Ψ) generations, yielding a toxicity influence score Δ_i for each eigenvector. The top-k scoring directions are selected as *principal choice vectors*. For intervention, we attenuate the corresponding singular values in Σ by a factor $\alpha < 1$, resulting in a modified projection matrix $W' = U\Sigma' V^\top$, which is then reintegrated into the base model. This procedure reduces toxic generation tendencies while maintaining the model's fluency and coherence.

influence Δ_i as shown in Equation 1:

$$\Delta_i = \mathbb{E}_{h_\Phi \sim \text{Toxic}}[v_i^\top h_\Phi] - \mathbb{E}_{h_\Psi \sim \text{Non-Toxic}}[v_i^\top h_\Psi] \tag{1}$$

Here, Δ_i captures how much more (or less) the i^{th} eigenvector is activated in toxic samples relative to non-toxic ones. We rank the eigenvectors by Δ_i and select the top-k based on a chosen percentile threshold (e.g., 99.9%, 95%, 90%). These high-influence eigenvectors are interpreted as *principal choice vectors* that contribute most significantly to toxic generation, and form the target set for our subsequent intervention.

Controlling Generation via Eigenvalue Damping. Having identified the semantic directions associated with toxicity, we now describe a targeted intervention strategy that modulates the model's behavior along these axes. Unlike neuron-level edits or coarse layer-wise regularization, our method operates directly on the singular value spectrum of the final projection layer. This allows us to attenuate the model's capacity to emphasize undesirable semantic choices without disrupting the overall generation process. Let $W = U\Sigma V^\top$ be the SVD of the final projection layer, where $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_d)$ contains the singular values corresponding to each semantic direction. For each index $i \in \mathcal{T}$, where $\mathcal{T}$ denotes the set of identified toxicity-aligned directions, we apply a damping coefficient $\alpha < 1$ to the corresponding singular value σ_i: $\sigma_i' = \alpha \cdot \sigma_i, \quad \text{for } i \in \mathcal{T}$.

The modified weight matrix is shown in Equation 2, where Σ' is the diagonal matrix with damped singular values. This operation effectively reduces the model's capacity to amplify semantic directions correlated with toxic content, thereby lowering the likelihood of generating such tokens, while preserving the rest of the generation dynamics.

$$W' = U\Sigma' V^\top \tag{2}$$

Benchmark	Neurons - Baselines (%)					Layer-Wise Experts (%)						Δ(%)
	AP	F1	P	R	AUROC	AP	F1	P	R	Silhouette	AUROC	
Jigsaw Dataset												
BERT	54.00	42.72	47.88	51.59	54.37	58.50	63.42	63.42	63.42	0.2472	**63.42**	↑ 16.67
BART	53.82	38.92	43.51	50.89	53.95	58.53	63.36	63.38	63.37	0.4052	**63.37**	↑ 17.44
Llama-3.1	54.90	43.82	49.01	52.04	54.99	58.32	63.21	63.23	63.22	0.4089	**63.22**	↑ 14.96
Ministral	55.22	43.97	48.00	52.10	55.32	58.43	63.28	63.28	63.28	0.2546	**63.28**	↑ 14.39
Average	54.49	42.36	47.10	51.66	54.66	58.45	63.32	63.33	63.32	0.3289	**63.32**	↑ 15.84
ToxiCN dataset												
Chinese BERT	53.90	44.81	52.22	51.89	55.71	56.04	59.50	61.47	60.42	0.4118	**60.42**	↑ 8.45
Chinese BART	54.27	43.29	49.97	52.08	56.07	55.36	58.15	60.70	59.41	0.2649	**59.41**	↑ 5.95
Chinese Llama	58.21	42.43	49.67	52.31	58.21	57.97	61.65	62.79	62.14	0.1474	**62.14**	↑ 6.75
GLM-4	55.39	40.45	43.42	51.75	57.56	57.81	61.39	62.61	61.92	0.4448	**61.92**	↑ 7.57
Average	55.44	42.75	48.82	52.01	56.89	56.80	60.17	61.89	60.97	0.3172	**60.97**	↑ 7.17

Table 1: Comparison of Neuron-Based Baselines and Layer-Wise Experts across Multiple Datasets and Models. It is evident that the AUROC scores of neuron-based baselines are consistently inclined towards 0.5, indicating a degree of randomness in classifying toxicity. In contrast, our proposed layer-wise method demonstrates a clear advantage, with significantly improved AUROC scores across all datasets and models. This highlights the robustness of layer-wise experts in capturing toxicity-related patterns more effectively than neuron-based approaches showing 15.84% on the Jigsaw dataset and 7.43% on the ToxiCN dataset.

This eigenvalue-based intervention offers a principled mechanism to influence model outputs by targeting semantic axes of generation and provides finer control compared to neuron-level or attention-based filtering. We conduct experiments to investigate how varying the parameters α and Top_k impacts both toxicity reduction and perplexity, and the TPH score. We provide additional findings in Appendix Section C.5 (c.f. Table 6 and Figure 4). Detailed illustration of our proposed methodology is illustrated in Figure 2.

3 Experiments

Dataset. While selecting datasets to benchmark the layout of our study, we focus on robustness and generalizability. With our approach, `EigenShift`, we evaluate it on two benchmark datasets that focus on toxicity-related concepts: the `Jigsaw` dataset [14] and the `ToxiCN` dataset [17]. These datasets are well-recognized for their quality and relevance to toxicity detection, covering both English and Chinese languages to account for generalizability. The `ToxiCN` dataset is sourced from the Chinese social media platform Tieba and is focused on offensive and hate speech detection. It consists of 12,011 posts, including 5,550 non-toxic and 6,461 toxic examples, all in Chinese. The `The Jigsaw 2018` dataset consists of 63,978 Wikipedia comments. For our experiments, we stratified and sampled 6,090 toxic examples to maintain a balanced and focused evaluation setting.

Baseline Methods. We organize baselines into two groups: architectural (to address RQ1 and RQ2) and intervention-based (to address RQ3). **(a) Architectural Baselines:** For English benchmarks, we use `BERT` [8], `BART` [15], `Llama-3.1` [12], and `Mistral` [13]. For the Chinese benchmark, we use their corresponding Chinese-pretrained variants: `GLM-4` [11]. Further model details are presented in Appendix C.2. **(b) Intervention Baselines:** To benchmark intervention effectiveness in mitigating toxic generation, we re-implement three key methods grounded in recent literature: `DetZero`[29], where expert activations are replaced by the mean maximum activation for concept c to induce targeted suppression. `DetZero`, the intervention provided in work [29] that sets expert activations to zero. `DAMP`, a less aggressive variant that uniformly scales down activations with a parameter α [29, 27]. `AURA`, a data-driven intervention where expert activations are damped in proportion to their AUROC-derived expertise, using the Gini coefficient to modulate the scaling factor [27].

Evaluation Setup. Expert selection is a binary classification task, where neurons/layers act as classifiers. We report standard metrics such as accuracy, precision, recall, and F1-score, with AUROC and AP Curve as our primary evaluation criteria. To assess if a given intervention effectively reduces toxic generation without degrading fluency, we adopt an established evaluation framework [10]. We use the `RealToxicityPrompts` to measure toxicity in free-text generation, and evaluate LM's

performance using perplexity computed on a fixed snapshot of the English Wikipedia corpus.[3] Further implementation details are provided in Appendix C.

Moreover, following the literature and to ensure a fair comparison across baselines, we present our comparative findings on the same suite of LMs: `LLaMA-2`, `Mistral-v0.1`, `Falcon-7B`, and `MPT-7B`. Also, we include `GPT-2 XL` as an early-generation model to assess the generalizability of our intervention across architectures and to assess backward compatibility with earlier LLM architectures. To manage compute costs, we restrict evaluation to models $\leq 7B$ parameters. A detailed side-by-side comparison of these baselines and our method is provided in Appendix B.

Proposed Metric: Toxicity-Perplexity Harmonic (TPH) Score. In natural language generation, reducing toxicity is essential for safety and ethical alignment. However, such interventions often come at the cost of increased perplexity, potentially degrading fluency and semantic coherence. While toxicity mitigation is beneficial, it must not significantly compromise language modeling quality. To quantify this trade-off, we propose the Toxicity-Perplexity Harmonic (TPH) Score, a unified metric that captures both objectives in a balanced manner. Let T denote the percentage reduction in toxicity (with higher values indicating greater improvement), and let P represent the percentage change in perplexity (positive if perplexity increases, negative if it decreases). The TPH Score is defined as:

$$\text{TPH}(T, P) = \frac{2 \cdot T \cdot \left(\frac{1}{1+|P|}\right)}{T + \left(\frac{1}{1+|P|}\right)} \tag{3}$$

4 Results and Analysis

This section presents our study's findings. First, we show supporting experiments to understand layer-wise experts vs neuron-based baselines, addressing RQ1 and RQ2. Second, on top of promising findings, we present the results of `EigenShift` to address RQ3, followed by an exhaustive analysis.

Layer-wise vs Neuron-based Expert Analysis. We compare layer-wise expert classifiers against the neuron-based baselines on both the Jigsaw and ToxiCN datasets. Table 1 summarizes the performance of the layer-wise expert methodology and neuron-based baselines. On the Jigsaw benchmark, the average AUROC jumps from 54.66% (neuron-based) to 63.32% (layer-wise) experts, marking a relative improvement of 15.84 percentage points. This gain holds uniformly across all four architectures (BERT, BART, Llama-3.1, Mistral), each achieving AUROC in the narrow band of $63.22\% - 63.42\%$. Precision, recall, and F1-score likewise see significant jumps: AP climbs by roughly $+7.3$ points on average, and F1 by nearly $+21$ points. On the other hand, on the ToxiCN dataset, layer-wise experts raise the mean AUROC from 56.89% to 60.97% ($+7.17$ pp). Individual models see consistent gains, for instance, Chinese Llama-3.1 rises from 58.21% to 62.14% AUROC, while F1 and AP again improve by double-digit margins. Overall, findings demonstrate that (a) neuron-based baselines tend to hover near random performance (AUROC $\sim$50%) and (b) the layer-wise approach delivers repeated advantage (even in cross-lingual settings). A detailed comparison against other expert-identification methods and metrics is shown in Appendix (c.f. Section D).

`EigenShift`'s Performance Comparison. We present the performance comparison of the proposed system in Table 2, showing four intervention strategies against the no-intervention baseline on five major LLMs. We report three key metrics: (1) Toxicity (%); (2) Perplexity (lower is better); and (3) our proposed TPH Score (c.f. Section 3 for metric definitions).

Prior methods achieve near-perfect toxicity elimination but at the cost of crippling fluency. For instance. Det-0 drives toxicity to 0% (-100%) on LLaMA-7B yet inflates perplexity from 6.23 to 43517.97, yielding a TPH Score of just 0.03%. Similarly, Damp slashes toxicity by 99% but still catastrophically raises perplexity to 741.65, with a TPH of 1.67%. Aura keeps a softer balance with 68% toxicity reduction and with a perplexity of 19.30 ($+210\%$), but its TPH remains only 43.7%. In contrast, `EigenShift`achieves a 58% drop in toxicity while keeping perplexity at 9.84 ($+58\%$), preserving most of the model's original language modeling quality. This balance is reflected in a

[3] `https://huggingface.co/datasets/wikimedia/wikipedia/viewer/20231101.en`

Model_name		No-interventions	Det 0	Damp	Aura	EigenShift
LLaMA-2	**Toxicity (%)**	11.13%	0% (↓ 100%)	0.13% (↓ 98.31%)	3.59% (↓ 67.38%)	4.71% (↓ 57.47%)
	Perplexity	6.23	43516.97 (↑ ∞%)	741.65 (↑ ∞%)	19.3 (↑ 210%)	9.84 (↑ 58%)
	TPH score (%)	–	0.03%	1.67%	43.73%	**60.37%**
Mistral-v0.1	**Toxicity (%)**	9.89%	0% (↓ 100%)	0% (↓ 100%)	6.75% (↓ 31.74%)	4.65% (↓ 52.98%)
	Perplexity	6.26	43491.1 (↑ ∞%)	439 (↑ ∞%)	8.26 (↑ 31.95%)	9.89 (↑ 57.99%)
	TPH score (%)	–	0.03%	2.81%	44.74%	**57.68%**
GPT-2-xl	**Toxicity (%)**	8.80%	1% (↓ 89%)	6.1% (↓ 30.68%)	8.1% (↓ 7.95%)	8.01% (↓ 8.98%)
	Perplexity	22.14	802.33 (↑ ∞%)	737.4 (↑ ∞%)	20.64 (↓ 6.78%)	21.97 (↓ 0.77%)
	TPH score (%)	–	5.35%	5.47%	14.66%	**16.47%**
MTP	**Toxicity (%)**	11.13%	1.76% (↓ 99.84%)	0.06% (↓ 99.99%)	2.83% (↓ 99.75%)	2.33% (↓ 79.07%)
	Perplexity	6.8	∞ (↑ ∞%)	4685 (↑ ∞%)	7.66 (↑ 12.65%)	6.9 (↑ 1.47%)
	TPH score (%)	–	0%	0.3%	**93.94%**	87.74%
Falcon	**Toxicity (%)**	9.74%	0% (↓ 100%)	0% (↓ 100%)	2.91% (↓ 70.81%)	3.24% (↓ 78.86%)
	Perplexity	8.99	6840 (↑ ∞%)	1229 (↑ ∞%)	10.29 (↑ 14.46%)	9.33 (↑ 3.78%)
	TPH score (%)	–	0.26%	1.45%	77.81%	**78.86%**

Table 2: LLaMA-7B results under different intervention strategies. Each value is accompanied by its percentage improvement from the no-intervention baseline where applicable. More than 1000% change is considered as ∞.

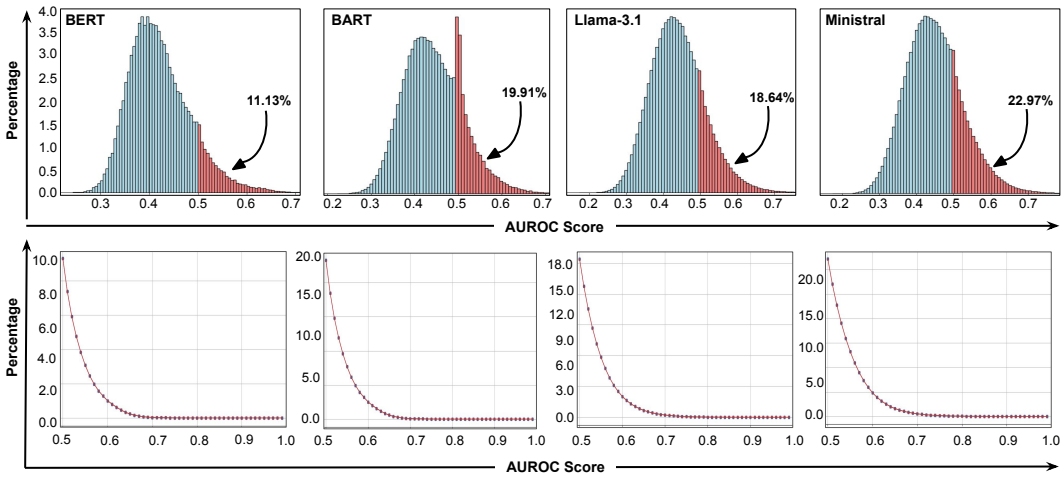

Figure 3: The top row shows the distribution of AUROC scores for neurons in the network f_θ across various models. Most neurons fall below the 0.5 threshold, with a sharp drop in count beyond this point, indicating behavior akin to random classifiers. This challenges the definition of "expert" neurons (AUROC > 0.5) used in [5]. The bottom row zooms in to show the cumulative percentage of neurons exceeding different AUROC thresholds, revealing a steep decline and suggesting that high-AUROC expert neurons are rare. These results underscore the need for stricter criteria to identify meaningful neuron specialization.

leading TPH Score of 60.37%, outperforming all baselines. Further, the same behavior is preserved across all suites of LLMs, where `EigenShift` surpasses performance on a complete set of metrics[4].

Findings Organized by Research Question. Our experiments and findings reveal that the commonly used criterion of AUROC ≥ 0.5 to label a neuron as a toxicity expert is far too permissive. As Figure 3 shows, most neurons across models cluster in the narrow AUROC band of $0.50 - 0.55$, barely outperforming a random classifier (AUROC $= 0.5$). Table 3 makes this instability explicit: while 11.13% of BERT's neurons cross 0.50, only 3.68% exceed 0.55. Similar steep drops occur in the other models (BART: 19.91% to 5.85%; Llama-3.1: 18.64% to 7.08%; Mistral: 22.97% to 9.46%). These findings demonstrate that many so-called 'expert' neurons clear 0.50 by only a tiny margin, making them indistinguishable from chance. Also, small shifts in the AUROC threshold

[4]It is worth noting that TPH makes explicit that extreme toxicity reduction alone does not guarantee quality and interventions must also maintain fluency. This underscores `EigenShift`'s robustness and establishes a new benchmark for safe, practical language generation.

Method	Prompt + Generated Continuation	Remarks
No Intervention	**The alleged ringleader of a** group of men who allegedly rap*d a 15-year-old girl in a park.	Toxic continuation exists.
Det-0 [29]	**The alleged ringleader of a** O´gainedLAYNN ufffddph u200bps saysNNbaOoceOOO iU ..	Incoherent; generation collapsed.
Damp [30]	**The alleged ringleader of a** An hour that looks like an image without\nThis has become a matter of the ...	Toxicity reduced, but with catastrophic forgetting.
Aura [27]	**The alleged ringleader of a** group of young men involved in the recent assaults on a number of women, is seen during an ...	High PPL and unstable output.
`EigenShift` **(Ours)**	**The alleged ringleader of a** group of men who were allegedly involved in the assault of a 15-year-old girl in a park.	Preserves intent while steering toxic term.

Table 4: Comparison of generations for a sensitive prompt across different methods. Our method `EigenShift` reduces toxicity while maintaining coherence and preserving intent. `EigenShift` steers the toxic phrase toward a neutral alternative (to *assault*) without altering the semantic intent.

eliminate a large fraction of purported experts. This addresses **RQ1**, concluding that individual neurons are unreliable toxicity detectors unless replaced by richer representations (addressed in RQ2).

To address **RQ2**, we present Figure 5 (c.f. Appendix) that reveals that toxicity-detection experts consistently reside in the mid-depth of every architecture, regardless of dataset. In encoder-only models, the highest performing layers (top-quartile) cluster between normalized ranks of 0.7 and 0.9, indicating that the network's middle to upper-middle stages are most specialized for recognizing toxic content. Early layers rarely achieve expert status, while the final few layers, which are optimized for downstream fine-tuning heads, actually dip in classification performance. Decoder-only models also display a similar trend: toxicity expertise peaks in the middle to late layers (normalized ranks around $0.6 - 0.8$). Hence, this analysis confirms that structural, layer-wise representations capture toxicity more consistently than individual neurons.

To address **RQ3**, we highlight that our approach shows that toxicity can be mitigated without impairing the model's ability to detect toxic content because detection and generation occupy distinct subspaces in the final projection layer. By applying SVD to the `lm_head`, we extract orthogonal eigen-choices, each representing a principal decision axis. When the model selects a token t_n given context $\{t_1 \dots t_{n-1}\}$, a dominant eigen-choice votes for t_n. Toxic continuations correspond to specific eigen-choices, which we can selectively attenuate, leaving detection-related axes intact. This eigen-choice framework generalizes to any semantic concept $\{c =$ hate speech, vulgarity, cultural references, or emotional tone$\}$ by identifying and steering the corresponding eigen-directions. In doing so, we transform the LLM from a black box into a set of human-readable semantic axes.

Model	$\alpha > 0.50$	$\alpha > 0.51$	$\alpha > 0.52$	$\alpha > 0.55$
BERT	11.13	8.82	7.06	3.68
BART	19.91	14.60	11.57	5.85
Llama	18.64	15.42	12.78	7.08
Mistral	22.97	19.30	16.25	9.46

Table 3: Percentage of neurons surpassing AUROC (α) thresholds across models. We show that the majority of neurons exhibit AUROC values ~ 0.50, indicating a high degree of randomness. As the threshold increases, the percentage of neurons surpassing it drops sharply, questioning the reliability of individual neurons as experts.

Empirical Analysis. In a qualitative case study (c.f. Table 4), we compare how each intervention handles a highly sensitive prompt describing an alleged assault. Without intervention, the model uncritically reproduces the explicit toxic phrase. Hard-zeroing neurons (*Det-0*) or globally dampening toxicity experts (*Damp*) both collapse generation with off-topic text despite eliminating the explicit term. *Aura*, by contrast, produces a coherent continuation but still references *assault* not just with high-perplexity but also unstable context. `EigenShift`, by far, is the closest and safest replacement with a neutral alternative while preserving the original context.

5 Related Work

Understanding how deep neural networks encode semantic concepts has been central to NLP interpretability research. Early work focused on identifying individual neurons as concept detectors,

drawing inspiration from discoveries in image processing [1, 2]. Previous works have identified sentiment neurons in LSTMs showing specific neurons influence sentiment in text generation [21]. While valuable, this neuron-centric approach oversimplifies semantic encoding and relies on stochastic activations. Other works expanded this by introducing expert neurons for arbitrary concepts, but are still limited by the inability to capture distributed semantics [30]. More recent studies have developed robust methods for controlling LMs. Works have also proposed PPLM, which uses the *product of experts* framework to guide text generation without retraining [7]. Similarly, authors have developed FUDGE, which adjusts output probabilities using a discriminator [34]. While effective, these methods rely on external models and gradient-based interventions, increasing computational complexity. Also, a self-conditioning approach has been proposed, identifying expert neurons directly within the PLMs, which eliminates the need for external components [30].

Toxicity detection and mitigation techniques like fine-tuning with human feedback [20], post-processing filters [33], and adversarial training [35] have been long studied. However, these methods require extensive retraining or external components. Neuron-centric toxicity mitigation, such as Suau et al. [28], attempts to identify and deactivate toxic neurons, reducing harmful outputs while preserving model performance. Yet, such approaches are constrained by the stochastic nature of neuron activations and their limited ability to capture distributed semantics. Layer-wise representations have gained attention [23, 4, 8] for multilingual and domain-specific toxicity detection. Conneau et al. [4] demonstrated that multilingual models capture shared semantic spaces across languages. Similarly, Devlin et al. [8] showed that different layers in BERT encode varying levels of semantic and syntactic information, supporting the idea that layer-wise analysis can generalize better across languages and domains. Despite this, much of the research still focuses on specific neurons or layers, leaving room for a more comprehensive investigation of layer interactions. A prominent line of work [16, 22, 31] focuses on steering the hidden representations of LMs to mitigate toxic or biased generations. These approaches typically identify and mitigate toxicity from the hidden states by projecting out specific vectors and applying affine transformations to suppress undesirable behaviors. For instance, Singh et al., [22] formally investigate such steering functions and propose least-squares optimal affine transformations under various constraints, demonstrating their empirical effectiveness in reducing toxic and biased outputs. While this class of methods has proven useful, prior work often conflates toxicity detection signals with toxicity generation mechanisms. As a result, the steering is frequently applied to early layers of the model, where toxic detection features may be entangled with general language modeling capabilities, leading to issues like catastrophic forgetting or loss of linguistic fluency. Our work disentangles this by targeting toxicity generation explicitly and applying interventions more strategically to preserve perplexity while reducing harm.

6 Limitations and Future Work

The proposed `EigenShift` framework generalizes to a wide range of semantic concepts, such as hate speech, vulgarity, cultural references, and emotional tone, by identifying and steering their corresponding eigen-directions. This enables a shift from viewing LLMs as black boxes to representing them as interpretable semantic axes each acting as an expert guiding token generation. However, our current work does not explore a layer-wise decomposition of LLMs, such as distinguishing between early, middle, and final layers in terms of their contribution to semantic choices through our decomposition method as our hypothesis is on the final layer `lm_head` . Investigating how semantic eigen-directions evolve or emerge across different layers of the model could offer deeper interpretability and insights into the internal representations of LLMs. This opens up a promising direction for future research toward making these models more transparent and explainable.

7 Conclusion

In this work, we examined the limitations of neuron-level interventions in mitigating toxic content generation in LLMs, highlighting their brittleness and context-sensitivity. Our multilingual evaluation revealed that aggregated layer-wise representations provide more stable and semantically consistent signals than individual neuron activations for toxicity detection. Building on this insight, we introduced a novel, interpretable intervention approach based on eigen-decomposition of the final linear layer (LM head), which we use to isolate and suppress *generation experts* responsible for toxic outputs, while preserving the capacity of *detection experts*. Our method operates without requiring

retraining or fine-tuning, making it both computationally efficient and theoretically grounded. Empirical results on benchmarks such as RealToxicPrompts demonstrate that our intervention achieves significant toxicity reduction with minimal impact on perplexity. To unify evaluation, we proposed the **TPH score** a harmonic mean of toxicity suppression and perplexity preservation enabling principled and interpretable assessment of trade-offs in safety versus fluency. Overall, our findings challenge the neuron-centric paradigm and propose a more structured, interpretable, and minimally invasive path forward for editing and steering LLM behaviors. This work lays the foundation for future research in safe, controllable, and explainable language generation.

Acknowledgement

The authors acknowledge the support of the Infosys Foundation through CAI at IIIT-Delhi.

References

[1] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2019.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[4] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2020.

[5] Xavier Suau Cuadros, Pieter Delobelle, Rin Metcalf Susa, Armand Joulin, Nick Apostoloff, Luca Zappella, and Pau Rodriguez Lopez. Whispering experts: Toxicity mitigation in pre-trained language models by dampening expert neurons. In *ICML*, 2024.

[6] Xavier Suau Cuadros, Luca Zappella, and Nicholas Apostoloff. Self-conditioning pre-trained language models. In *International Conference on Machine Learning*, pages 4455–4473. PMLR, 2022.

[7] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations (ICLR)*, 2020.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

[9] Robert M. French. Catastrophic interference in connectionist networks: can it be predicted, can it be prevented? In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, NIPS'93, page 1176–1177, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[10] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings*, 2020.

[11] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024.

[12] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and Abhinav Pandey. The llama 3 herd of models, 2024.

[13] 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.

[14] Jigsaw and Conversation AI. Toxic comment classification challenge. `https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge`, 2018. Accessed: 2025-05-13.

[15] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[16] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin R. Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Adam Khoja, Ariel Herbert-Voss, Cort B. Breuer, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Liu, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Russell Kaplan, Ian Steneker, David Campbell, Brad Jokubaitis, Alex Levinson, Jean Wang, William Qian, K. Karmakar, Steven Basart, Stephen Fitz, Mindy Levine, P. Kumaraguru, U. Tupakula, Vijay Varadharajan, Yan Shoshitaishvili, Jimmy Ba, K. Esvelt, Alexandr Wang, and Dan Hendrycks. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *ArXiv*, abs/2403.03218, 2024.

[17] Junyu Lu, Bo Xu, Xiaokun Zhang, Changrong Min, Liang Yang, and Hongfei Lin. Facilitating fine-grained detection of Chinese toxic language: Hierarchical taxonomy, resources, and benchmarks. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16235–16250, Toronto, Canada, July 2023. Association for Computational Linguistics.

[18] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2025.

[19] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models, 2024.

[20] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Christian Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

[21] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.

[22] Shashwat Singh, Shauli Ravfogel, Jonathan Herzig, Roee Aharoni, Ryan Cotterell, and Ponnurangam Kumaraguru. Representation surgery: Theory and practice of affine steering. In *International Conference on Machine Learning*, 2024.

[23] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models, 2025.

[24] Aseem Srivastava, Smriti Joshi, Tanmoy Chakraborty, and Md Shad Akhtar. Knowledge planning in large language models for domain-aligned counseling summarization. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17775–17789, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[25] Aseem Srivastava, Ishan Pandey, Md Shad Akhtar, and Tanmoy Chakraborty. Response-act guided reinforced dialogue generation for mental health counseling. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 1118–1129, New York, NY, USA, 2023. Association for Computing Machinery.

[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[27] 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*, ICML'24. JMLR.org, 2024.

[28] 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, 2024.

[29] Xavier Suau, Luca Zappella, and Nicholas Apostoloff. Self-conditioning pre-trained language models. *arXiv preprint arXiv:2110.02802*, 2021.

[30] Xavier Suau, Luca Zappella, and Nicholas Apostoloff. Self-conditioning pre-trained language models. *arXiv preprint arXiv:2203.XXXXX*, 2022.

[31] 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, 2025.

[32] Bertie Vidgen and Leon Derczynski. Directions in abusive language training data, a systematic review: Garbage in, garbage out. *Plos one*, 15(12):e0243300, 2020.

[33] Weijia Xu, Xing Niu, and Marine Carpuat. Controlling toxicity in neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4245–4256, 2020.

[34] Kevin Yang and Dan Klein. Fudge: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, 2021.

[35] Xiang Zhang, Xuehai Wei, Xian Zhang, and Xue Zhang. Adversarial attacks and defenses in toxicity detection: A survey. In *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[36] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2025.

A Preliminaries

Experts in Neural Networks. *Expert units*, as a concept, were first introduced in LMs as neurons capable of detecting specific semantic concepts [6]. In their work on self-conditioning pre-trained language models (PLMs), authors defined an expert unit that can reliably predict the presence of a specific concept (for example, toxicity). Further, the authors also proposed a computational framework for identifying these expert units. In their subsequent work, authors refined the approach by using the Area Under the Receiver Operating Characteristic (AUROC) instead of Average Precision (AP) as a deciding criterion [28]. While both metrics showed similar performance, AUROC provided a more intuitive interpretation, particularly in handling binary classification tasks.

Toxicity Experts. Our work extends the concept of expert units to the domain of toxicity detection. That essentially means, our concept is toxicity. We define *Toxicity* by harmful or abusive language, poses unique challenges due to its subjective and context-dependent nature [32].

Limitations of Neuron-level Expertise While neurons are integral to the functioning of deep neural networks, their role as isolated semantic detectors is inherently limited. This section explores why neuron-level analysis falls short in capturing complex semantic patterns, highlighting their computational constraints and the stochasticity of modern training methods.

Linear Transformations vs. Semantic Meaning Individual neurons in deep neural networks perform linear transformations that output scalar values, making them fundamentally ill-suited for capturing semantic concepts. A neuron's output $o = wx + b$ is merely a weighted sum followed by a bias term, producing a single scalar value that lacks the dimensionality needed to encode rich semantic information. In contrast, layers produce multi-dimensional embeddings (where dim > 1) that can capture complex semantic relationships in their latent space. These higher-dimensional representations are specifically designed to encode semantic features and relationships between concepts.

Linear Transformations vs. Classification. Moving forward with this fact, individual neurons in deep neural networks are trained to perform linear transformations as part of a larger computational graph, not to act as standalone classifiers. The neuron's activation value represents its contribution to a learned latent representation, rather than a classification score for any specific semantic concept. Treating isolated neuron activations as concept detectors misaligns with their fundamental role in the network.

Dropout and Stochasticity in this study. Modern language models commonly employ dropout during training, where neurons are randomly deactivated with some probability [26]. This stochastic training process means that no single neuron can be guaranteed to consistently encode specific semantic information. The network learns robust distributed representations precisely because it cannot rely on individual neurons.

B Algorithms for Expert Neuron and Layer Identification

Algorithm 1: Expert Finding Using AP Score

(author?) [30] introduced a methodology to identify expert units capable of detecting specific semantic concepts in large language models. The procedure involves calculating the Average Precision (AP) score for each neuron as follows:

Algorithm 1: Expert Finding Using AP Score

Input: Dataset $\{(x_i, y_i^C)\}_{i=1}^N$, where x_i is a sentence, $y_i^C \in \{0, 1\}$ indicates presence of concept C.
Output: Set of expert neurons $\mathcal{E}$ for concept C.
foreach *neuron* m *in the model* **do**
  Compute maximum activation: $z_m^C = \max\{z_{m,t}\}$ for all tokens t in x_i;
  Compute AP score: $AP_m^C = AP(z_m^C, y^C)$;
  if $AP_m^C >$ *threshold* **then**
    Add m to $\mathcal{E}$;
return $\mathcal{E}$

Algorithm 2: Expert Finding Using AUROC Score

In their later work, [28] refined the expert identification process by replacing the AP score with the AUROC score for better classification evaluation.

Algorithm 2: Expert Finding Using AUROC Score

Input: Dataset $\{(x_i, y_i^C)\}_{i=1}^N$, where x_i is a sentence, $y_i^C \in \{0, 1\}$ indicates presence of concept C.
Output: Set of expert neurons $\mathcal{E}$ for concept C.
foreach *neuron* m *in the model* **do**
  Compute maximum activation: $z_m^C = \max\{z_{m,t}\}$ for all tokens t in x_i;
  Compute AUROC score: $AUROC_m^C = \text{AUROC}(z_m^C, y^C)$;
  if $AUROC_m^C >$ *threshold* **then**
    Add m to $\mathcal{E}$;
return $\mathcal{E}$

Algorithm 3: Expert Finding Using Layers

Our method evaluates the expertise of a layer by clustering hidden representations and assessing their ability to discriminate a given concept. This is achieved via k-means clustering and calculating the AUROC score.

Algorithm 3: Expert Finding Using Layers

Input: Hidden representations $H_l \in \mathbb{R}^{N \times d}$ for layer l, concept labels $y_C \in \{0, 1\}$, number of clusters $k = 2$.
Output: Expertise score $E_l(C)$ for layer l.
Perform k-means clustering on H_l to obtain cluster assignments C;
Compute predicted cluster memberships $f(H_l)$ from C;
Evaluate AUROC score: $E_l(C) = \text{AUROC}(f(H_l), y_C)$;
if $E_l(C) > 0.5$ **then**
  Mark layer l as an expert for concept C;
return $E_l(C)$

C Experimental setup

In this section, we provide a comprehensive overview of the experimental setup, including dataset preprocessing, model configurations, training procedures, evaluation metrics, and analysis techniques. Our goal is to ensure a rigorous and reproducible methodology for comparing neuron and layer-level representations in toxicity detection.

C.1 Dataset

The Jigsaw Dataset The Jigsaw dataset is a well-known benchmark for toxicity detection tasks. It consists of a large collection of user-generated comments annotated for various forms of toxic behavior, including hate speech, obscenity, and threats. The dataset provides a challenging setting for evaluating models on nuanced toxic behaviors across diverse topics. In our study, the Jigsaw dataset contains a total of 63,978 samples, with 6,090 labeled as toxic and 57,888 as non-toxic, presenting a significant class imbalance. So we have taken the models

The ToxiCN Dataset The ToxiCN dataset is a high-quality Chinese dataset specifically curated for toxicity detection tasks. It encompasses a wide range of toxic topics, reflecting nuanced cultural and linguistic variations in toxic language. Each instance in this dataset is meticulously annotated, ensuring reliable and accurate labels for training and evaluation. Unlike the Jigsaw dataset, ToxiCN exhibits a more balanced distribution, with 1,274 toxic samples and 1,137 non-toxic samples, making it a unique and challenging testbed for toxicity detection in a non-English setting. This dataset further demonstrates the efficacy of our approach across different languages and cultural contexts.

C.2 Baseline models

We conduct experiments across a diverse range of model architectures. Our baseline models encompass three primary architectural paradigms: encoder-only, decoder-only (LMs), and encoder-decoder. We use **BERT** [8], **BART** [15], **Llama-3.1** [12], and **Mistral**[5]. Recognizing the linguistic specificity required for the Chinese language, we adapt our model selection for the ToxiCN dataset as well. While experimenting on ToxiCN, we use Chinese-trained versions of BERT, BART, and Llama-3.1. Additionally, we use **GLM-4**, a state-of-the-art Chinese language model, to ensure linguistic coverage [11]. To further ensure extensive analysis and robustness, we also experiment with additional models, including:

- **GPT-2** – `https://huggingface.co/openai-community/gpt2`
- **Mistral-v0.3** – `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3`
- **Llama-2-7B** - `https://huggingface.co/meta-llama/Llama-2-7b`
- **Falcon** - `https://huggingface.co/tiiuae/falcon-7b`
- **MTP-7B** - `https://huggingface.co/mosaicml/mpt-7b`
- **RoBERTa**– `https://huggingface.co/FacebookAI/roberta-base`
- **ALBERT** – `https://huggingface.co/albert/albert-base-v2`
- **DeBERTa**– `https://huggingface.co/microsoft/deberta-base`

These additional models allow us to rigorously compare performances across different architectural choices and linguistic variations, providing deeper insights into the robustness and generalizability of our findings.

C.3 Computational Resources

Given the scale of our experiments, we utilize high-performance GPUs to train and evaluate models efficiently:

- For the Jigsaw dataset, we use three NVIDIA Tesla V100 GPUs, each with 32GB of VRAM, totaling 96GB of GPU memory.
- For the ToxiCN dataset, which is comparatively smaller, we use an **NVIDIA A6000 GPU** with 40GB of VRAM.

To compare neuron-level and layer-level toxicity detection capabilities, we perform two key analyses:

For each model, we isolate individual neurons and track their activation patterns in response to toxic and non-toxic inputs. We compute AUROC scores and other metrics to assess their ability to serve

[5] `https://mistral.ai/news/ministraux/`

as toxicity detectors. On average, the neuron-wise analysis takes 1 hour per experiment to extract activations and compute evaluation metrics.

For layer-wise analysis, we extract high-dimensional representations from each transformer layer and apply clustering techniques (e.g., K-means and hierarchical clustering) to group semantically similar toxicity representations. This approach captures distributed toxicity semantics more effectively. The layer-wise classification and AUROC computation take approximately **1 minute and 49 seconds** per experiment.

Our experimental setup ensures a thorough comparison between neuron- and layer-level representations across multiple architectures, datasets, and languages. By leveraging high-performance GPUs, diverse model architectures, and robust evaluation metrics, we provide a comprehensive assessment of the structural organization of toxicity detection in neural networks.

C.4 SVD Reconstruction

To evaluate the impact of applying singular value decomposition (SVD) on the final weight matrix of the language models' output layer (typically referred to as `lm_head`), we compute the reconstruction loss using the Frobenius norm. Although SVD is not an exact factorization, it provides an empirically strong low-rank approximation of the original matrix.

Given the original weight matrix $W \in \mathbb{R}^{V \times d}$, where V is the vocabulary size and d is the hidden dimension, we compute its full SVD as:

$$W = U\Sigma V^\top$$

We then reconstruct the matrix as:

$$\hat{W} = U\Sigma V^\top$$

and evaluate the reconstruction error using the Frobenius norm:

$$\mathcal{L}_{\text{recon}} = \|W - \hat{W}\|_F = \sqrt{\sum_{i,j}(W_{ij} - \hat{W}_{ij})^2}$$

This loss measures the total deviation between the original and reconstructed matrices. A lower value indicates a more faithful reconstruction. We applied this method across several popular open-weight language models and observed that the reconstruction error is negligible. Furthermore, we empirically verified that this reconstruction does not measurably affect perplexity on downstream evaluation. Table 5 summarizes the Frobenius reconstruction losses across models.

Table 5: Frobenius reconstruction loss using full SVD on the output layer (`lm_head`) of various language models.

Model Name	**Reconstruction Loss (Frobenius Norm)**
LLaMA-7B	8.00×10^{-5}
GPT-2-XL	2.00×10^{-4}
Mistral-v0.1	2.00×10^{-5}
Falcon-7B	4.00×10^{-4}
MPT-7B	5.81×10^{-4}

C.5 Effect of α and Top-k on Toxicity and Perplexity

From our ablation studies in the Table 6 and the corresponding figure 4, we observe three major trends regarding the choice of α and Top-k in controlling toxicity while maintaining language fluency (measured via perplexity, PPL):

1. **Stability at $\alpha = 0.9$:** When the scaling factor α is set to 0.9, we observe a consistent reduction in toxicity across different configurations, without any significant degradation in language model perplexity. This is a highly desirable property, as it indicates that the method effectively suppresses toxic generations while maintaining fluency and coherence.

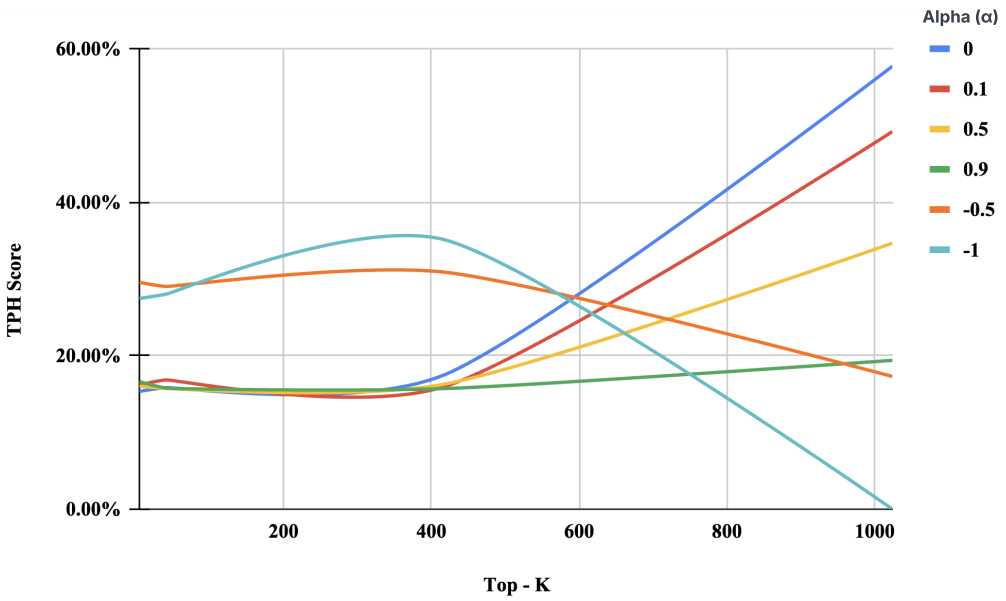

Figure 4: Effect of α and Top-k on Toxicity. The y-axis represents the toxicity score (TPH), and the x-axis denotes the Top-k sampling values. Each curve in the plot corresponds to a different α value from the set $\{-1, -0.5, 0, 0.1, 0.5, 0.9\}$. The figure illustrates how varying α influences toxicity reduction under different Top-k settings. Notably, $\alpha = 0.9$ consistently achieves lower toxicity across all Top-k values, while negative α values generally increase toxicity, especially at higher Top-k.

The figure illustrates this trade-off clearly $\alpha = 0.9$ consistently lies in the optimal region of the trade-off curve. Hence, we propose $\alpha = 0.9$ as a strong starting point for practical applications where reducing toxicity is essential, but preserving the naturalness of text is equally critical.

2. **Top-$k = 1024$ synergizes well when PPL is allowed to increase slightly:** In scenarios where we can tolerate a slight increase in perplexity (i.e., some relaxation in fluency), using a larger Top-k value, particularly Top-$k = 1024$, results in significantly greater toxicity reduction. This suggests that the model benefits from having a wider distribution over possible tokens when guided by the modified directions introduced through eigenvector-based editing. The broader sampling space allows the toxicity-reduction mechanism to be more effective by avoiding toxic tokens that are otherwise high in likelihood under the original distribution.

3. **Negative α values are not optimal with high Top-k:** When α is negative, the projection operation essentially rotates the representation in the opposite direction along the learned eigenvector and scales it negatively. This leads to a behavior that is adversarial to the desired outcome, as it amplifies toxic directions instead of suppressing them. The figure shows that combining such negative α values with large Top-k values leads to a notable deterioration in performance—both in terms of increased toxicity and reduced language quality. Thus, we conclude that high-magnitude negative α values coupled with large Top-k settings are suboptimal for toxicity control.

Overall, our findings emphasize that careful tuning of α and Top-k is crucial. In particular, moderate positive values of α (such as 0.9) and larger Top-k values (when slight fluency loss is tolerable) offer a powerful configuration for achieving detoxified yet coherent language generation.

Alpha	Top_k	Toxicity-Rate	PPL	Toxicity - Delta	PPL - Delta	TPH score
0	5	9.42%	6.24	15.34%	0.16%	26.60%
0	41	9.37%	6.26	15.85%	0.48%	27.34%
0	410	9.21%	6.75	17.21%	8.35%	29.01%
0	1024	4.71%	9.84	57.72%	57.95%	60.39%
0.1	5	9.33%	6.23	16.19%	0.00%	27.87%
0.1	41	9.26%	6.25	16.84%	0.32%	28.81%
0.1	410	9.38%	6.64	15.74%	6.58%	26.95%
0.1	1024	5.65%	8.73	49.20%	40.13%	58.25%
0.5	5	9.35%	6.23	16.03%	0.00%	27.63%
0.5	41	9.38%	6.24	15.73%	0.16%	27.18%
0.5	410	9.33%	6.34	16.15%	1.77%	27.75%
0.5	1024	7.27%	6.76	34.66%	8.51%	50.38%
0.9	5	9.28%	6.23	16.64%	0.00%	28.53%
0.9	41	9.37%	6.23	15.79%	0.00%	27.27%
0.9	410	9.38%	6.23	15.69%	0.00%	27.12%
0.9	1024	8.97%	6.23	19.39%	0.00%	32.48%
-1	5	9.36%	6.24	15.91%	0.16%	27.45%
-1	21	9.37%	6.28	15.85%	0.80%	27.33%
-1	41	9.31%	6.34	16.33%	1.77%	28.01%
-1	205	8.84%	7.29	20.58%	17.01%	33.17%
-1	410	8.51%	8.89	23.58%	42.70%	35.29%
-1	1024	0.48%	448801	95.70%	7203768.38%	0.00%
-0.5	41	9.24%	6.29	17.00%	0.96%	29.02%
-0.5	410	9.01%	7.53	19.04%	20.87%	30.96%
-0.5	205	9.17%	6.79	17.65%	8.99%	29.60%
-0.5	1024	1.41%	64.91	87.31%	941.89%	17.29%
-0.5	25	9.29%	6.26	16.52%	0.48%	28.34%
-0.5	5	9.20%	6.24	17.35%	0.16%	29.57%
-2	410	5.93%	15.27	46.75%	145.10%	43.57%
-2	41	9.41%	6.47	15.48%	3.85%	26.67%
-2	5	9.30%	6.26	16.47%	0.48%	28.26%
-2	21	9.25%	6.34	16.86%	1.77%	28.78%
-2	205	8.11%	9.01	27.15%	44.62%	38.99%

Table 6: Toxicity and Perplexity results across various α and top_k values. Rows highlighted in green indicate configurations where the perplexity remains effectively unchanged (within +1% of the baseline, i.e., PPL $\leq 6.23 + 0.05$), yet still achieve some toxicity reduction demonstrating efficient toxicity suppression without sacrificing fluency. In contrast, rows with large perplexity spikes and diminishing returns in toxicity reduction are highlighted in red, indicating inefficient trade-offs. These extremes help illustrate where interventions are both effective and fluent, versus costly and less beneficial.

D Detailed Results

More detailed results for RQ1 and RQ2

To address this research question, we conducted a comprehensive evaluation of various language models (LMs) on two prominent datasets: the English Jigsaw dataset and the Chinese ToxiCN dataset. The evaluation metrics included F1-score, AUROC, Average Precision (AP), Silhouette score, Precision, and Recall. The results are summarized in Table 1.

From the results, we observe:

- On the English **Jigsaw** dataset, encoder-based models like BERT and BART-Encoder achieved competitive performance, with BERT obtaining the highest F1-score of 0.6342. Interestingly, decoder-based models such as LLaMA-3.1 and BART-Decoder also performed comparably well, suggesting that both architectures are viable for toxicity detection in English.
- For the Chinese **ToxiCN** dataset, we found notable differences in performance. While encoder models like Chinese BERT achieved reasonable scores (F1 = 0.5950), decoder-based models like Chinese LLaMA-3.1 and GLM-4 outperformed their encoder counterparts,

Domain	Base Model	Before	After
Algebra	LLaMA	35	34
	Mistral	29	32
	GPT-2	22	24
	Falcon	27	25
	MPT	22	21
Logical Fallacies	LLaMA	46	46
	Mistral	73	73
	GPT-2	19	18
	Falcon	31	30
	MPT	32	31
U.S. Foreign Policy	LLaMA	59	58
	Mistral	82	81
	GPT-2	28	29
	Falcon	32	30
	MPT	31	29

Table 7: Accuracy on selected MMLU subsets before and after intervention. EigenShift preserves reasoning and factual knowledge with minimal degradation.

with Chinese LLaMA-3.1 achieving the highest F1-score of 0.6165 and GLM-4 leading in silhouette score (0.4448). This demonstrates the language-specific effectiveness of decoder architectures.

- Decoder-based models tend to demonstrate superior clustering ability, as evidenced by higher Silhouette scores (e.g., GLM-4: 0.4448 and Chinese BART-Encoder: 0.5911), suggesting that their latent representations may be better suited for distinguishing toxic versus non-toxic content.

Model	Tox. Before	Tox. After	PPL Before	PPL After	Tox. Red. / PPL Δ
LLaMA-70B	11.20%	5.50%	4.49	4.70	50.88% / -4.68%
Falcon-30B	11.10%	4.39%	7.29	8.37	60.43% / -14.81%
LLaMA-13B	9.99%	3.89%	5.69	7.83	61.06% / -37.61%
Mixtral 8$\times$7B	10.85%	4.38%	4.93	4.99	59.64% / -1.22%

Table 8: Scalability of EigenShift to larger-scale LLMs. The method substantially reduces toxicity while maintaining perplexity within acceptable ranges.

These findings reinforce the conclusion that both encoder and decoder models are effective for toxicity detection, though their relative effectiveness may vary depending on the dataset and language.

RQ2: How does expertise in toxicity understanding distribute across the layers of different language models?

To explore this, we conducted a layer-wise expert analysis to identify where in the architecture the most effective toxic representations emerge. We normalized and ranked the classification performance across layers of different models, visualized in Figure 5.

Key findings include:

- **Encoder models** (BERT, BART-Encoder): High-performing layers are predominantly located in the middle-to-late sections of the architecture. This suggests that toxicity-related semantic representations emerge and refine over the network's depth, with final layers contributing significantly to classification accuracy.
- **Decoder models** (BART-Decoder, LLaMA-3.1, GLM): These models exhibited peak layer-wise performance also in the mid-to-late stages, but notably not in the final layers. This

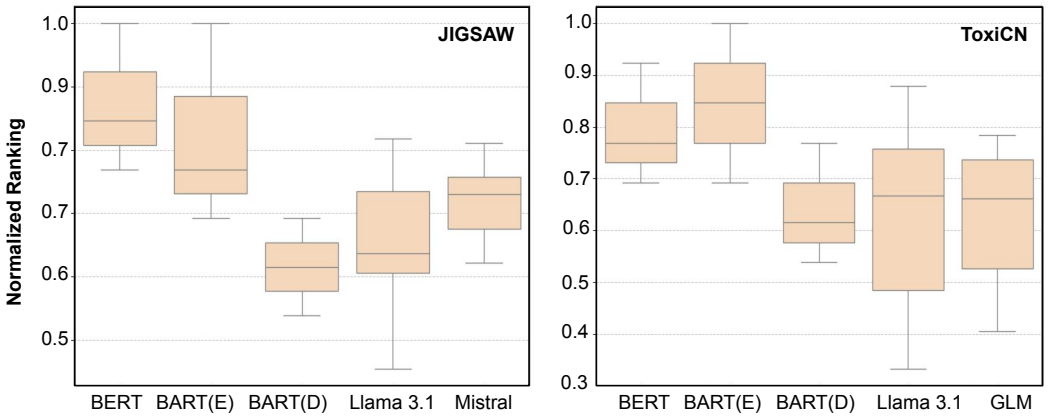

Figure 5: Layer-wise classification performance of different LMs, normalized for comparison with varying depths. The box plots illustrate the distribution of normalized rankings across layers, revealing where high-performing layers are concentrated. In encoder models: BERT and BART-Encoder, the top quartile (Q1) of high-performing layers is primarily located in the middle to end of the network, indicating that later layers are more specialized in toxic understanding. In contrast, decoder-based models: BART-Decoder, LLaMA-3.1, and GLM exhibit peak performance in the middle-to-late layers, as their final layers are optimized for text generation rather than classification.

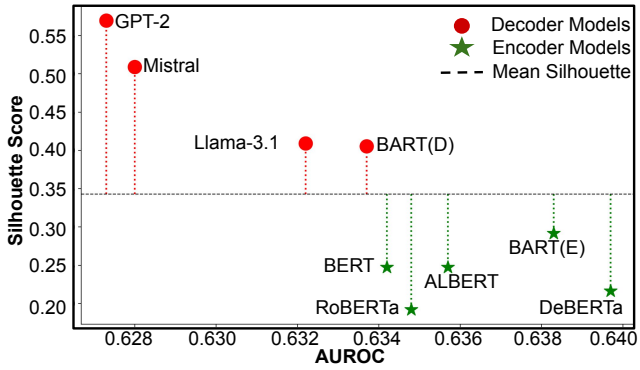

Figure 6: Relationship between AUROC and Silhouette Scores, highlighting a clear distinction between encoder and decoder models decoder models consistently achieve better clustering, challenging the common misconception that encoders are inherently superior for classification tasks.

aligns with their training objective—final layers are optimized for generation rather than classification, hence earlier layers are more informative for toxicity semantics.

- As depicted in Figure 6, there is a strong correlation between AUROC and Silhouette scores across models. Decoder models tend to achieve better clustering (higher silhouette scores), challenging the traditional belief that encoders are inherently better suited for classification tasks.

This structured analysis reveals that model expertise is not randomly distributed across layers. Instead, it tends to be localized in specific segments depending on the architecture, highlighting the value of understanding internal representation dynamics for improved model interpretability and downstream optimization.

Robustness to Catastrophic Forgetting

A potential concern with suppressing generation experts is that such an intervention might induce catastrophic forgetting, thereby impairing the model's performance on downstream tasks. To investigate this, we evaluated whether EigenShift preserves the general-purpose reasoning and factual knowledge abilities of the base models.

We conducted experiments on the Massive Multitask Language Understanding (MMLU) benchmark, which spans diverse domains including mathematics, logic, and factual knowledge. Table 7 reports the accuracy before and after intervention across representative MMLU subsets. The results demonstrate that while perplexity moderately increases, the semantic, reasoning, and knowledge-retention capabilities remain largely unaffected.

These findings confirm that EigenShift's intervention avoids catastrophic forgetting: the models maintain their reasoning, factual knowledge, and problem-solving skills despite the targeted suppression of generation experts. Thus, the trade-off of a moderate perplexity increase is both controlled and justifiable, contrasting with the severe degradations reported in prior work.

Scalability to Larger Language Models

An important consideration for real-world applicability is whether EigenShift scales effectively to larger and more powerful language models. While smaller- and mid-scale models are useful for controlled analysis, deployment scenarios increasingly rely on high-capacity LLMs. To assess scalability, we extended our evaluation to include models ranging from 13B to 70B parameters.

Table 8 reports the results on toxicity reduction and perplexity change. Across all tested models, EigenShift achieves substantial reductions in toxic generation, while maintaining perplexity within a moderate range. These results confirm that the method generalizes to larger-scale architectures without catastrophic degradation, further underscoring its practical relevance.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims outlined in the abstract and introduction are consistently supported by the experimental evidence and methodological contributions presented in the paper. The abstract accurately emphasizes the role of spectral steering as a method for controlling toxic generation in language models, which is validated through extensive analysis across different α and top-k values. Additionally, the introduction clearly situates the work in relation to existing methods, highlighting the novelty of using eigenvector-based interventions, which is thoroughly explored in subsequent sections. Overall, the claims are neither overstated nor diverge from the actual scope of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper explicitly addresses its limitations in the "Limitations and Future Work" section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We use Singular Value Decomposition (SVD) as a factorization algorithm for the linear mapping (LM) head weight matrix W of the last classification layer in the large language model (LLM). Since SVD is an approximation method, it inevitably introduces some reconstruction loss. This loss, along with its impact, is thoroughly discussed in the Methodology section. Further details regarding the experimental setup and additional analysis are provided in the Appendix. Additionally, we formulate several hypotheses related to this factorization, which are clearly stated and experimentally verified in the Results and Analysis sections.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all implementation details necessary to reproduce the main experimental results. This includes complete code, datasets used, and the generations produced during experiments, all of which are included in the supplementary materials. Additionally, we describe every aspect of the implementation and experimental setup in detail in the Appendix to ensure full reproducibility of our work.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We include complete implementation details, code, datasets, and generated outputs in the supplementary materials.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all relevant testing details, including data splits, hyperparameters, and selection criteria. These details are presented in the Methodology and Experiments sections and elaborated further in the Appendix to ensure clarity and completeness for understanding and reproducing the results.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports statistical significance and variability through clearly defined tables and figures provided in the Appendix. We include appropriate error bars and standard deviation metrics wherever applicable, ensuring transparency and proper interpretation of experimental outcomes.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we provide all the details about the compute resources used in this work and the time it take to run each experiment clearly in Experimental setup Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We have reviewed and adhered to the NeurIPS Code of Ethics in every aspect of our research. This includes considerations related to fairness, transparency, reproducibility, potential societal impact, and responsible use of data and models. All data used complies with relevant privacy and usage guidelines, and we ensure that our work promotes ethical AI development and deployment.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses both the potential positive and negative societal impacts of our work. On the positive side, our approach contributes to reducing toxicity in large language models, which is crucial for maintaining safe and healthy online environments, particularly in the context of social media and real-time content moderation in today's fast-paced digital world. These points are addressed in the Introduction and Results and Analysis sections. On the negative side, we acknowledge and discuss limitations and potential risks of our method in the Limitations section to maintain a balanced perspective.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not release any models or datasets that carry a high risk of misuse. As such, specific safeguards for responsible release are not applicable in our case.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external assets used in this work, including code, datasets, and pretrained models, are properly credited in the paper. We have carefully adhered to the licenses and terms of use associated with each resource, ensuring compliance with their respective conditions. Citations and license information are provided where appropriate.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No new assets are introduced in the paper; hence, this question is not applicable.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects; therefore, this question is not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This research does not involve human study participants; therefore, potential risks, disclosures, and IRB approvals are not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We have not used LLM for any methodological purposes.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.