Evaluating Sparse Autoencoders for Monosemantic Representation

Moghis Fereidouni, Muhammad Umair Haider, Peizhong Ju, A.B. Siddique
University of Kentucky
{moghis.fereidouni, muhammadumairhaider, peizhong.ju, ab.siddique}@uky.edu

Abstract

A key barrier to interpreting large language models is polysemanticity, where neurons activate for multiple unrelated concepts. Sparse autoencoders (SAEs) have been proposed to mitigate this issue by transforming dense activations into sparse, more interpretable features. While prior work suggests that SAEs promote monosemanticity, no quantitative comparison has examined how concept activation distributions differ between SAEs and their base models. This paper provides the first systematic evaluation of SAEs against base models through activation distribution lens. We introduce a fine-grained concept separability score based on the Jensen–Shannon distance, which captures how distinctly a neuron's activation distributions vary across concepts. Using two large language models (Gemma-2-2B and DeepSeek-R1) and multiple SAE variants across five datasets (including word-level and sentence-level), we show that SAEs reduce polysemanticity and achieve higher concept separability. To assess practical utility, we evaluate concept-level interventions using two strategies: full neuron masking and partial suppression. We find that, compared to base models, SAEs enable more precise concept-level control when using partial suppression. Building on this, we propose Attenuation via Posterior Probabilities (APP), a new intervention method that uses concept-conditioned activation distributions for targeted suppression. APP achieves the smallest perplexity increase while remaining highly effective at concept removal [1].

1 Introduction

Large language models (LLMs) have achieved remarkable performance across a wide range of natural language tasks, often matching or surpassing human-level performance (Luo et al., 2025; Achiam et al., 2023; Touvron et al., 2023; Guo et al., 2025; Eslamian and Cheng, 2025). Nonetheless, understanding how these models internally represent and manipulate concepts remains a major challenge. A key obstacle is polysemanticity; the phenomenon where individual neurons respond to multiple, semantically distinct concepts rather than encoding single, interpretable features (Janiak et al., 2023; Olah et al., 2017; Nguyen et al., 2016). This entanglement complicates the interpretation and analysis of model behavior, posing a significant barrier to building transparent and controllable AI systems (Sharkey et al., 2025; Marshall and Kirchner, 2024; Bereska and Gavves, 2024).

Dictionary learning via sparse autoencoders (SAEs) (Huben et al., 2024; Gao et al., 2025) has recently emerged as a promising approach to mitigating polysemanticity in neural representations. SAEs aim to transform dense activations of a desired component of the base LLM into sparse features by enforcing sparsity and encouraging each neuron to specialize in distinct, concept-specific features (Huben et al., 2024; Rajamanoharan et al., 2024a,b; Gao et al., 2025). The goal is to produce monosemantic representations, where individual neurons respond to single, well-defined concepts (Huben et al., 2024; Rajamanoharan et al., 2024a,b; Gao et al., 2025). The underlying hypothesis is intuitive; if we can force the model to use fewer neurons simultaneously, each active neuron should correspond to a more distinct and interpretable concept. Empirical studies have shown that SAEs can uncover interpretable features across domains such as vision and language, facilitating improved interpretability (Shu et al., 2025; Huben et al., 2024; Pach et al., 2025).

Most existing evaluations of SAE interpretability are qualitative, relying on case studies or anecdotal neuron visualizations that provide limited systematic insight (Kissane et al., 2024; Li et al., 2025). The remaining quantitative efforts mainly examine whether neurons are active for given

[1] Source code available at https://github.com/MultifacetedNLP/SAEMonosemanticity.

Findings of the Association for Computational Linguistics: EACL 2026, pages 5969–5984
March 24-29, 2026 ©2026 Association for Computational Linguistics

concepts, rather than capturing the full distributional structure of concept activations across neurons (Minegishi et al., 2025; Karvonen et al., 2025).

In this work, we conduct a systematic investigation into the effectiveness of SAEs in promoting monosemanticity in the internal representations of LLMs through the *distributional lens*. Particularly, we conduct comprehensive evaluations using two large language models (Gemma-2-2B (Rivière et al., 2024) and DeepSeek-R1 (Guo et al., 2025)) and various SAEs of different widths and sparsity levels on five benchmark datasets, including both word-level (POS tagging and NER) and sentence-level (e.g., AG News) tasks. We begin by quantifying polysemanticity using overlap statistics, measuring the fraction of salient neurons that respond to multiple, semantically distinct concepts. While SAEs exhibit lower polysemanticity than their base models, this overlap-based analysis remains coarse-grained; it treats all neurons that respond to multiple concepts as equally entangled, without considering how their activations vary across those concepts. In practice, a neuron may activate for several concepts, yet do so with clearly distinct activation distributions, suggesting behavior that may still be considered monosemantic. That is, monosemanticity is not solely about binary activation overlap, but rather about the separability of a neuron's activation distributions across concepts.

We formalize this view by introducing a new concept separability score, based on the Jensen–Shannon distance (Lin, 1991). This fine-grained, distribution-aware metric quantifies how well a neuron's activations separate across different concepts by measuring the distance between their activation distributions. Using this score, we find that SAEs exhibit higher concept separability than their dense counterparts.

To further evaluate the practical utility of monosemantic representations of SAEs, we examine their effectiveness in enabling concept-level model interventions. Specifically, we assess how precisely concept-related behavior can be suppressed in SAEs compared to base models. We evaluate two intervention strategies; full neuron masking, which suppresses all activations of salient neurons associated with a target concept, and partial suppression, which intervenes in activations selectively based on their distributional association with the concept. Across SAEs and the base model, partial suppression outperforms full masking in most cases, achieving more effective suppression of the target concept while better preserving unrelated model behavior. Furthermore, our results show that SAEs support more precise and effective concept removal than their dense counterparts, especially when applying partial suppression methods. These findings reinforce the idea that concept separability, when defined in terms of activation distributions rather than binary activation overlap, offers better model control and interpretability.

Additionally, motivated by the varying separability of concept activations across neurons, we introduce Attenuation via Posterior Probabilities (APP), a new intervention method that leverages concept-conditioned activation distributions to selectively suppress target concepts with minimal side effects. Specifically, APP computes the posterior probability that a given activation corresponds to a target concept and attenuates it accordingly. Among all methods evaluated, APP achieves the smallest degradation in language modeling quality (lowest perplexity increase) while remaining highly competitive with other baselines in targeted concept removal across both SAEs and base models.

In summary, this work makes the following contributions:

- We present the first quantitative analysis of monosemantic representations in SAEs relative to their base LLM through distributional lens.
- We introduce a concept separability score, based on the Jensen-Shannon distance, a fine-grained, distribution-aware metric that captures how well neuron activations separate for different concepts.
- We propose a new intervention method, which is the least invasive concept erasure technique and highly competitive with existing methods in removing the targeted concept.

2 Preliminaries

Neuron. A neuron refers to a component of a hidden state vector in a transformer layer. Given a hidden state $h^l \in \mathbb{R}^d$ at layer l, the j-th neuron is denoted by h^l_j.

Concept. A concept $c_i \in C$ is a semantic category assigned to each input (or its components), where $C = c_1, \ldots, c_k$. For instance, sentence-level types (e.g., declarative, interrogative) or word-level tags (e.g., noun, verb) can serve as concepts. In this work, we focus both on *sentence-level* and *word-level* concepts.

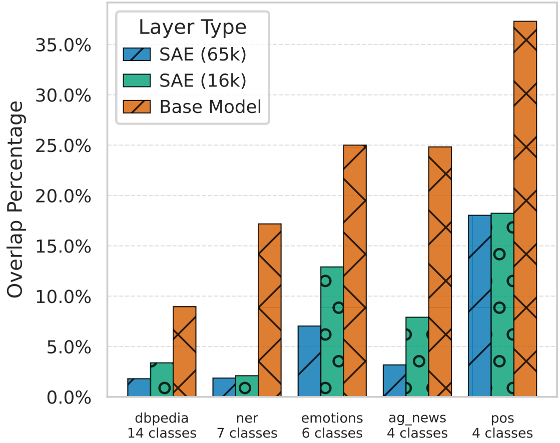

Figure 1: SAEs reduce neuron overlap compared to the base model, indicating lower polysemanticity. Higher-capacity SAEs (65k) further reduce overlap, enabling better separation of distinct concepts.

Datasets and Models. We use five datasets: Part-of-Speech Tagging (POS) (Pasini et al., 2021), Named Entity Recognition (NER) (Ding et al., 2021), AG News (Zhang et al., 2015), Emotions (Saravia et al., 2018), and DBpedia (Zhang et al., 2015). Our analysis is conducted using two large language models: (1) the Gemma-2-2B model (Rivière et al., 2024) along with its corresponding JumpReLU Sparse Autoencoders (SAEs) from GemmaScope (Lieberum et al., 2024), and (2) the DeepSeek-R1 model (Guo et al., 2025) with its associated JumpReLU SAE from LlamaScope (He et al., 2024).

3 Analyzing Polysemanticity in SAEs

A neuron is considered polysemantic when it responds to multiple, distinct concepts rather than a single, well-defined one. Several studies have demonstrated that such polysemantic behavior is common in neural networks (Elhage et al., 2022; Bau et al., 2017; Scherlis et al., 2022; Lecomte et al., 2024; Marshall and Kirchner, 2024; Olah et al., 2017; Nguyen et al., 2016). This polysemanticity reduces the interpretability of models, motivating the development of Sparse Autoencoders (SAEs) (Huben et al., 2024; Rajamanoharan et al., 2024a,b; Gao et al., 2025). SAEs are designed to encourage sparsity in neural activations, aiming to align each neuron with a specific, distinct concept and thereby promote monosemanticity and interpretability. In the following, we evaluate SAEs to better understand their effectiveness in improving monosemanticity.

3.1 Salient Neuron Overlap

As a first step, we quantify polysemanticity by measuring the overlap percentage of salient neurons (i.e., neurons with high mean activation) across concepts. Specifically, for each concept, we identify the top 80 salient neurons by mean activation. Then, the overlap percentage is computed as the intersection-over-union of these top-k sets across concepts (see Appendix H for the corresponding top-p analysis). This metric captures the extent to which neurons are shared across concepts, reflecting shared saliency and polysemanticity. The Figure 1 compares this shared saliency for the base model and two Sparse Autoencoders (SAEs) with different latent dimensions (16k and 65k), while maintaining comparable sparsity levels (116 vs. 93 active neurons). In the DBpedia dataset, for instance, we observe that nearly 9% of the top-activated (salient) neurons in the base model are shared across all 14 concepts. Moreover, the Figure 1 confirms that SAEs exhibit reduced conceptual overlap, suggesting less polysemanticity; however, polysemantic neurons are still present. For example, in the POS dataset, the percentage of shared salient neurons drops from around 38% in the base model to approximately 18% in the SAE with 16k dimensions. This is still a relatively high percentage, indicating that over 18% of the most salient neurons are shared across all four concepts. Moreover, another notable observation is that the 65k-dimensional SAEs exhibit lower polysemanticity than their 16k-dimensional counterparts across all datasets. This reinforces the idea that larger SAEs have greater capacity to allocate distinct neurons to specific concepts, thereby enhancing interpretability. For additional analysis evaluating all active SAE neurons, not just the top 80, see Appendix E.

3.2 SAEs Activation Distributions

So far, we have analyzed polysemanticity by quantifying neuron overlap based on activation frequency, specifically, how often a neuron is active or salient across different concepts. While these approaches offer useful aggregate insights, they treat neuron activation as a binary or averaged signal, overlooking the distributional characteristics of how neurons respond to each concept. In other words, a neuron might be shared across multiple concepts, but the manner in which it activates for each could vary significantly, ranging from broad, overlapping responses to distinct, well-separated patterns. To

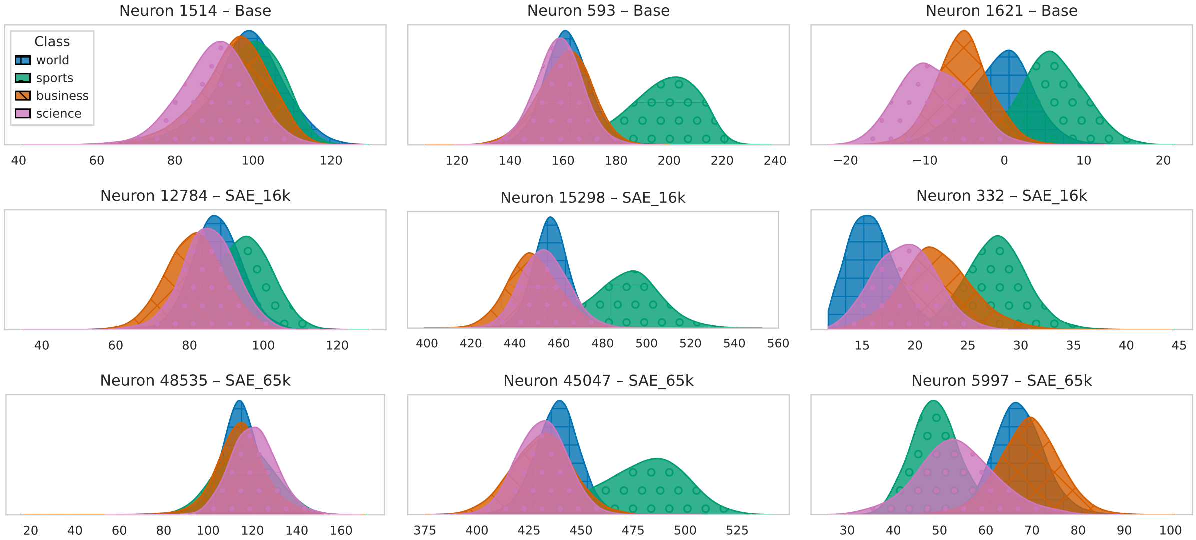

Figure 2: Across base model and SAEs (SAE-16k, SAE-65k), neurons exhibit varying degrees of separability in their activations. Some have completely overlapping activations across concepts, others show partial or clear separation. This variability underscores the importance of using distribution-aware metrics when assessing monosemanticity.

better capture this nuance, Figure 2 displays the full activation distributions of selected neurons across concept classes in the AG News dataset, comparing the base model with two SAE variants of differing capacities. We include an analogous visualization for the NER dataset in Appendix B.

These plots reveal two key patterns: (1) consistent with prior observations (Haider et al., 2025), neuron activations in both the base model and SAEs tend to follow approximately Gaussian distributions; and (2) while some neurons are shared across multiple concepts, their activation distributions can range from highly overlapping to clearly separable. This underscores a fundamental limitation of mean-based or binary overlap measurements, which can overlook meaningful distinctions in activation behavior. To more accurately measure polysemanticity, it is therefore necessary to employ a metric that captures the full shape of activation distributions across concepts. To this end, we introduce a new concept separability score, based on the Jensen-Shannon distance, which quantifies the degree of separation between concept-specific activation distributions.

	POS	AG News	Emotions	DBpedia	NER
Base	0.343	0.366	0.322	0.405	0.308
SAE (width: 16k, $\ell_0 : 116$)	0.539	0.650	0.431	0.621	0.581
SAE (width: 65k, $\ell_0 : 93$)	**0.600**	**0.709**	**0.446**	**0.680**	**0.621**

Table 1: Separability Score S across five datasets for the base model and Sparse Autoencoders (SAEs) with varying widths and sparsity levels (ℓ_0).

Concept Separability Score. To quantify how separable a neuron's activation distributions are across k concepts, we first define the probability density function as:

$$f_{h_j^l|c_i}(x) = p\big(h_j^l = x \mid c_i\big), \quad i = 1, \ldots, k,$$

with this density function at hand, we define mixture as $M^{(l,j)}(x) = \frac{1}{k}\sum_{i=1}^{k} f_{h_j^l|c_i}(x)$, and then compute the generalized Jensen–Shannon divergence (Lin, 1991) as

$$\mathrm{JSD}\big(f_{h_j^l|c_1}, \ldots, f_{h_j^l|c_k}\big) = H\big(M^{(l,j)}\big) - \tfrac{1}{k}\sum_{i=1}^{k} H\big(f_{h_j^l|c_i}\big).$$

Next, we take the square root of the JSD and normalize it to obtain a proper distance metric bounded within $[0, 1]$:

$$D_{\mathrm{JS}}\big(f_{h_j^l|c_1}, \ldots, f_{h_j^l|c_k}\big) = \frac{\sqrt{\mathrm{JSD}\big(f_{h_j^l|c_1}, \ldots, f_{h_j^l|c_k}\big)}}{\sqrt{\log_2 k}},$$

Moreover, we assign $D_{\mathrm{JS}} = 1$ whenever a neuron's activations are all attributed to a single concept. Finally, the layer-level separability score is

$$S^l = \frac{1}{d}\sum_{j=1}^{d} D_{\mathrm{JS}}\big(f_{h_j^l|c_1}, \ldots, f_{h_j^l|c_k}\big),$$

where d is the number of neurons in layer l.

Table 1 reports the concept separability score $S \in [0, 1]$ for Gemma, where higher values indicate more distinct activation distributions across concepts, indicating higher monosemanticity (Full JS scores for all settings and both LLMs are in Appendix G). Across all five datasets, SAEs substantially improve separability over the base model. For instance, on DBpedia (14 classes), the score increases from 0.405 in the base model to 0.621 and 0.680 for the 16K- and 65K-dimensional SAEs, respectively, an increase of over 50%. Furthermore, higher-capacity SAEs consistently yield greater separability, supporting the view that more expressive latent spaces better disentangle concepts.

Building on our finding that sparse autoencoders yield more separable concept representations, we next assess whether this improved separability enables more precise concept erasure interventions.

4 Concept Erasure

Concept erasure encompasses interventions that aim to remove a specific concept from a model's internal representation, ideally without affecting other concepts (Dalvi et al., 2019a,b; Dai et al., 2022; Morcos et al., 2018). Formally, let M be a trained model that maps an input x to a concept label $M(x) = c$. An ideal erasure yields a modified model M'_{ideal} satisfying:

$$M'_{\text{ideal}}(x) = \begin{cases} \neq M(x), & \text{if } M(x) = c, \\ = M(x), & \text{if } M(x) \neq c. \end{cases}$$

That is, the model should unlearn the target concept c while preserving its behavior on all other concepts.

As shown in Figure 2, some neurons exhibit considerable overlap in their activation distributions across concepts, while others show separability. This pattern appears in both the base model and SAEs, suggesting that concept erasure techniques should not treat all activation values in one neuron identically. Instead, we propose a more targeted approach that considers where an activation falls within the distribution. Specifically, values in regions uniquely tied to a concept (i.e., those regions of distribution that are clearly separable from other concepts) should be suppressed more strongly, while regions shared across concepts should be dampened more conservatively to preserve other concepts. To achieve this, we introduce Attenuation via Posterior Probabilities (APP), which modulates suppression based on distributional separability.

4.1 Attenuation via Posterior Probabilities (APP)

Given all neurons h^l_j of layer l, with individual activations x^l_j for $j = 1, \dots, d$, and a target concept $c_i \in C$, our goal is to selectively suppress the activation x^l_j that is attributable to c_i, while preserving contributions from other concepts. We begin by computing the posterior probability that a given activation x^l_j arose from concept c_i, under the assumption that all concepts are a priori equally likely:

$$\begin{aligned} \pi_{j,i}(x^l_j) = p(c_i \mid h^l_j = x^l_j) &= \frac{p(h^l_j = x^l_j \mid c_i)\, p(c_i)}{\sum_{m=1}^{k} p(h^l_j = x^l_j \mid c_m)\, p(c_m)}, \\ &= \frac{p(h^l_j = x^l_j \mid c_i)}{\sum_{m=1}^{k} p(h^l_j = x^l_j \mid c_m)} \equiv \frac{f_{h^l_j \mid c_i}(x^l_j)}{\sum_{m=1}^{k} f_{h^l_j \mid c_m}(x^l_j)}. \end{aligned}$$

By definition, $\sum_{i=1}^{k} \pi_{j,i}(x^l_j) = 1$.

To avoid unreliable posterior estimates from low-density regions, we limit our attention to the central region of the target concept's activation distribution, where density estimates are more reliable. Let $\mu_{j,i}$ and $\sigma_{j,i}$ denote the mean and standard deviation of neuron h^l_j under concept c_i, and define the valid damping window as:

$$W_{j,i} = \left[\mu_{j,i} - 2.5\,\sigma_{j,i},\ \mu_{j,i} + 2.5\,\sigma_{j,i}\right].$$

The damping factor $\alpha_{j,i}(x)$ is then defined as:

$$\alpha_{j,i}(x) = \begin{cases} 1 - \pi_{j,i}(x), & x \in W_{j,i}, \\ 1, & \text{otherwise.} \end{cases}$$

With $\alpha \approx 0$ when x is very typical of the target concept c_i. Finally, we apply this factor to dampen the activation:

$$\tilde{x}^l_j = \alpha_{j,i}(x^l_j)\, x^l_j = \begin{cases} \left[1 - \pi_{j,i}(x^l_j)\right] x^l_j, & |x^l_j - \mu_{j,i}| \leq 2.5\,\sigma_{j,i}, \\ x^l_j, & \text{otherwise.} \end{cases}$$

This formulation enables precise, concept-aware suppression while leaving unrelated or uncertain activations unchanged.

4.2 Baseline Methods

To comprehensively evaluate concept erasure effectiveness, we compare APP (which is a partial suppression method) against three other partial methods and one full-masking baseline.

AURA (Suau et al., 2024): Ranks neurons by AUROC, selects those with $\mathrm{AUROC} > 0.5$, and dampens their output based on AUROC.

Range Masking (Haider et al., 2025): The activations of concept-relevant neurons (highly activated) are suppressed when they fall within their typical range ($\mu \pm 2.5\sigma$).

Adaptive Dampening (Haider et al., 2025): The activations of concept-relevant neurons (highly activated) are dampened in proportion to their distance from the concept mean.

Full Masking (Dalvi et al., 2019a; Dai et al., 2022; Antverg and Belinkov, 2022): concept-relevant neurons (highly activated) are fully zeroed out to eliminate the target concept.

4.3 Metrics

We evaluate the causal effect of our interventions using three metrics: task accuracy, confidence, and perplexity.

Accuracy and confidence are measured both before and after intervention, for the target concept c and all auxiliary concepts $c' \neq c$. The goal is to assess how much the intervention selectively affects the target concept while minimizing disruption to others.

Let D_{Acc} denote the drop in accuracy for the target concept, and D'_{Acc} the average drop in accuracy across auxiliary concepts. Similarly, let D_{Conf} and D'_{Conf} be the drops in confidence score for the target and auxiliary concepts, respectively, where we use the model's predictive probability as a proxy for confidence score.

Using these, we compute two scores:

$$\Delta_{\text{Acc}} = D_{\text{Acc}} - D'_{\text{Acc}}, \quad \Delta_{\text{Conf}} = D_{\text{Conf}} - D'_{\text{Conf}}.$$

Higher values of Δ_{Acc} and Δ_{Conf} indicate more precise interventions, strongly affecting the target concept while preserving performance on others. In the main text, we report only Δ_{Acc} and Δ_{Conf}; the full metric breakdowns are included in Appendix C.

Lastly, to capture the overall impact on the model's generative ability, we measure the increase in perplexity:

$$\mathrm{DPPL} = \mathrm{PPL}_{\text{post}} - \mathrm{PPL}_{\text{base}}.$$

Comprehensive implementation details are presented in Appendix D, including our histogram-based KDE for concept-conditioned densities.

4.4 Results and Analysis

Comparison of Intervention Effectiveness: SAEs vs. Base Model. As it can be seen in Table 2, across nearly all settings, we find that partial intervention methods (particularly APP and AURA) consistently achieve higher Δ_{Acc} and Δ_{Conf} when applied to SAE representations compared to the base model. Specifically, for APP, SAE-based interventions outperformed the base model in 38 out of 40 comparisons. For AURA, SAE-based interventions were more effective in all 40 cases. In contrast, full masking shows less benefit from SAE representations; 19 out of 40 interventions resulted in better outcomes than when applied to the base model. This discrepancy suggests that coarse suppression methods fail to capitalize on the increased concept separability offered by SAEs. These findings reinforce that SAE representations are more disentangled and that fine-grained, distribution-aware methods are better equipped to exploit this structure for effective concept removal. Furthermore, focusing specifically on the APP intervention (which is a partial intervention), we observe that within the SAE family, increasing capacity consistently enhances intervention quality. In particular, Δ_{Conf} consistently increases as we scale from 16k to 65k latent dimensions, reflecting improved confidence suppression for the target concept. Δ_{Acc} also improves across most datasets, with only minor exceptions, further underscoring the role of latent dimensionality in enabling more precise and effective concept removal.

Partial Interventions Vs Full Interventions. Full masking ranks as the worst-performing method in 36 out of 40 Δ-metrics on the SAEs and it also produces the largest perplexity increase in all DPPL evaluations across SAEs and the base model. This underscores that distribution-aware partial methods (e.g., APP, AURA), which leverage activation distributions, are far more effective for targeted concept removal than the coarse, distribution-agnostic full-masking approach.

APP is least disruptive and highly competitive on concept erasure. APP consistently achieves the smallest DPPL across all 25 experiments, making it the least disruptive method. Adaptive follows as the second-best approach, attaining the second-smallest DPPL in 19 out of 25 cases. For concept

Table 2: Concept Erasure Results by Method and Model Type across Datasets (Gemma-2-2B). Bolded values indicate the best performance, and underlined values denote the second-best. Results are grouped by intervention method (e.g., APP, AURA, Adaptive) and model type (Base vs. SAE variants) across five benchmark datasets. For each SAE variant, the exact SAE width and sparsity level (ℓ_0) are explicitly specified.

Type	**Method**	**POS**			**AG News**			**Emotions**			**DBpedia**			**NER**		
		Δ_{Acc} ↑	Δ_{Conf} ↑	DPPL ↓	Δ_{Acc} ↑	Δ_{Conf} ↑	DPPL ↓	Δ_{Acc} ↑	Δ_{Conf} ↑	DPPL ↓	Δ_{Acc} ↑	Δ_{Conf} ↑	DPPL ↓	Δ_{Acc} ↑	Δ_{Conf} ↑	DPPL ↓
Base	APP	**0.224**	**0.210**	**0.571**	**0.347**	**0.266**	**0.394**	**0.051**	0.095	**0.151**	0.113	**0.104**	**0.076**	0.079	0.129	**0.201**
	Aura	0.063	0.091	1.195	$\underline{0.112}$	0.092	2.130	-0.019	0.004	2.273	0.092	0.064	2.792	0.033	0.044	1.515
	Range	0.151	0.1006	2.258	0.081	$\underline{0.098}$	1.026	0.036	$\underline{0.104}$	0.784	**0.168**	**0.109**	0.700	**0.135**	**0.209**	1.978
	Adaptive	$\underline{0.157}$	$\underline{0.129}$	$\underline{1.158}$	0.050	0.089	$\underline{0.545}$	0.032	0.090	$\underline{0.428}$	$\underline{0.120}$	0.099	$\underline{0.387}$	$\underline{0.095}$	0.173	$\underline{1.052}$
	Full	0.115	0.069	22.873	0.096	0.080	16.914	$\underline{0.038}$	**0.113**	17.438	**0.169**	$\underline{0.102}$	10.922	**0.134**	$\underline{0.176}$	23.598
SAE width: 65k $\ell_0 : 93$	APP	$\underline{0.302}$	$\underline{0.270}$	**0.342**	**0.601**	**0.442**	**0.230**	$\underline{0.231}$	$\underline{0.279}$	**0.075**	**0.399**	**0.219**	**0.114**	$\underline{0.507}$	0.520	**0.161**
	Aura	**0.406**	**0.312**	$\underline{0.360}$	0.577	$\underline{0.367}$	0.592	**0.336**	**0.311**	0.286	$\underline{0.357}$	**0.213**	0.490	**0.519**	0.514	0.359
	Range	0.052	0.055	0.741	0.587	0.125	0.542	0.146	0.024	0.413	0.252	0.056	0.446	0.497	$\underline{0.549}$	0.433
	Adaptive	0.095	0.105	0.533	$\underline{0.592}$	0.215	$\underline{0.358}$	0.182	0.084	$\underline{0.226}$	0.222	$\underline{0.083}$	$\underline{0.279}$	0.494	**0.557**	$\underline{0.308}$
	Full	0	0.0003	1.712	**0.602**	0.006	4.763	0.120	0.002	5.462	0.275	0.018	3.892	0.382	0.391	3.046
SAE width: 16k $\ell_0 : 116$	APP	$\underline{0.216}$	**0.191**	**0.468**	$\underline{0.582}$	**0.369**	**0.569**	$\underline{0.265}$	$\underline{0.274}$	**0.166**	**0.312**	**0.152**	**0.140**	$\underline{0.377}$	$\underline{0.397}$	**0.531**
	Aura	**0.250**	**0.192**	$\underline{0.577}$	**0.596**	$\underline{0.286}$	1.262	**0.339**	**0.280**	0.493	$\underline{0.257}$	$\underline{0.140}$	0.836	**0.389**	**0.415**	0.741
	Range	0.061	0.046	0.925	0.505	0.103	0.879	0.099	0.009	0.418	0.208	0.053	0.446	0.298	0.305	0.915
	Adaptive	0.097	$\underline{0.085}$	0.655	0.508	0.163	$\underline{0.676}$	0.128	0.047	$\underline{0.267}$	0.177	0.071	$\underline{0.288}$	0.329	0.347	$\underline{0.721}$
	Full	0	0.0005	3.738	0.462	0.001	5.882	0.048	0.0005	6.454	0.187	0.0007	5.639	0.262	0.256	3.872
SAE width: 65k $\ell_0 : 197$	APP	$\underline{0.350}$	**0.322**	**0.398**	**0.615**	**0.458**	**0.433**	$\underline{0.247}$	$\underline{0.285}$	**0.406**	**0.362**	**0.235**	**0.083**	$\underline{0.530}$	0.654	**0.346**
	Aura	**0.384**	$\underline{0.317}$	$\underline{0.576}$	**0.611**	$\underline{0.397}$	0.940	**0.311**	**0.327**	0.584	$\underline{0.297}$	**0.235**	0.596	**0.567**	**0.681**	$\underline{0.510}$
	Range	0.046	0.059	0.902	$\underline{0.346}$	0.060	0.748	0.135	0.024	0.729	0.292	0.106	0.335	0.503	$\underline{0.662}$	0.613
	Adaptive	0.110	0.130	0.634	0.345	0.130	$\underline{0.554}$	0.170	0.110	$\underline{0.574}$	0.257	$\underline{0.147}$	$\underline{0.205}$	0.503	0.656	0.815
	Full	0	0.00001	5.599	0.344	0.0007	7.422	0.110	0.002	7.613	0.216	0.019	6.203	0.292	0.384	3.625
SAE width: 16k $\ell_0 : 285$	APP	$\underline{0.382}$	$\underline{0.306}$	**0.634**	**0.628**	**0.356**	**0.882**	$\underline{0.212}$	$\underline{0.228}$	**0.492**	**0.274**	**0.147**	**0.167**	**0.423**	**0.482**	**0.370**
	Aura	**0.433**	**0.359**	$\underline{0.778}$	$\underline{0.549}$	$\underline{0.346}$	1.671	**0.286**	**0.300**	$\underline{0.788}$	0.158	$\underline{0.111}$	0.975	**0.428**	$\underline{0.469}$	0.718
	Range	0.051	0.047	1.356	0.456	0.069	1.852	0.111	0.012	1.395	$\underline{0.181}$	0.057	0.958	0.338	0.392	0.924
	Adaptive	0.179	0.148	0.975	0.441	0.149	$\underline{1.312}$	0.121	0.072	0.948	0.145	0.083	$\underline{0.582}$	$\underline{0.376}$	0.435	$\underline{0.669}$
	Full	0	0.001	5.880	0.213	0.0004	16.284	0.062	0.001	8.117	0.153	0.023	9.510	0.167	0.191	8.661

removal, APP demonstrates strong and consistent performance; it most frequently achieves the best Δ_{Conf} (14/25 cases), while ranking first or second in 21 out of 25 settings. Regarding Δ_{Acc}, AURA leads with 14 wins, though APP follows closely and ranks in the top two positions for 23 out of 25 settings. Overall, APP effectively removes target concepts while preserving predictive fluency better than all other baselines.

Comparative Analysis of APP and AURA. The superior effectiveness of AURA and APP on both Δ_{Acc} and Δ_{Conf} stems from the fact that they explicitly model not only the target-concept distribution but also the distributions of all auxiliary concepts. By calibrating their interventions to maximize disruption of c while minimizing collateral effects on $c' \neq c$, both methods achieve higher Δ values than approaches that consider only the target distribution. Between these two, APP pulls ahead of AURA in terms of perplexity (DPPL) because it leverages fine-grained, activation-specific damping rather than a single, per-neuron factor. AURA mutes an "expert" neuron uniformly, regardless of whether a particular activation is highly characteristic of the target concept, whereas APP computes $\pi_{j,i}(x)$ for each activation and suppresses only the portions of the distribution uniquely associated with c_i. This activation-aware attenuation not only removes the targeted concepts effectively but also best preserves the model's overall fluency, as evidenced by the smallest DPPL.

Cross-Model Validation (DeepSeek). To verify that our observations are not specific to Gemma-2, we replicate experiments on DeepSeek-R1 (Appendix A). The results show the same trends, SAEs enable more selective concept removal than the base model. Moreover, APP achieves the smallest perplexity degradation (DPPL) across all datasets except one, while remaining highly competitive on the concept-removal metrics (Δ_{Acc}, Δ_{Conf}).

4.5 Relation Between Erasure Methods and JS Distance

Our distribution-aware separability score S (see Subsection 3.2) serves as a natural predictor for the precision of concept erasure methods. Intuitively, the more separable a neuron's activation distributions are, the easier it should be to suppress only the target concept while preserving unrelated behavior. We empirically validate this hypothesis in Figure 3, which plots our JS separability score S (x-axis) against the change in accuracy difference ΔAcc (y-axis), aggregating data from both the Gemma and DeepSeek experiments (Both SAE and base models). A strong positive Pearson correlation ($r = 0.771$, p < 0.001) emerges for the average per-

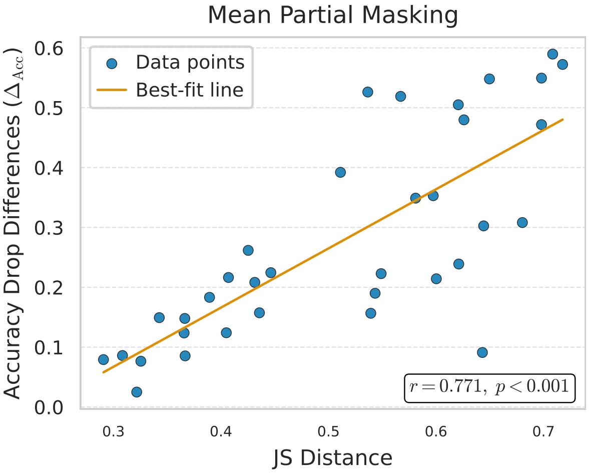

Figure 3: Separability Score vs. Erasure Ability (Partial)

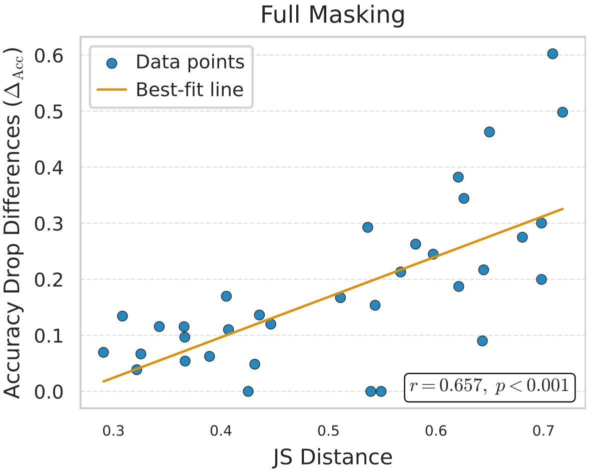

Figure 4: Separability Score vs. Erasure Ability (Full)

formance of partial erasure methods (APP, AURA, Range, and Adaptive), confirming that higher separability reliably predicts more selective accuracy drops on the target concept. Detailed per-method correlations are provided in Appendix F. For comparison, we conducted the same analysis for the full masking approach (Figure 4). While a positive correlation also appears ($r = 0.657$, p < 0.001), it is notably weaker than that observed for partial masking. This gap reinforces a key insight that full masking cannot exploit the fine-grained separability of activation distributions, whereas distribution-aware partial methods (e.g., APP, AURA) do.

5 Related Works

5.1 Sparse Autoencoders for Feature Discovery

Sparse Autoencoders (SAEs) have emerged as a powerful method for learning interpretable, monosemantic features from neural network activations (Huben et al., 2024). Recent advances have focused on improving reconstruction quality and scaling through architectural and training strategy innovations such as JumpReLU activations, BatchTopK sparsity, gated, and end-to-end training frameworks (Rajamanoharan et al., 2024a,b; Gao et al., 2025; Bussmann et al., 2024; Braun et al., 2024). Empirical analyses have validated SAEs' ability to discover meaningful structures across different domains, from vision-language models to algorithmic patterns like temporal difference learning in LLMs (Sun et al., 2025b; Pach et al., 2025; Demircan et al., 2025). Moreover, evaluation studies have highlighted both their utility for interpretability tasks and remaining challenges with polysemantic representations (Kantamneni et al., 2025; Minegishi et al., 2025; Karvonen et al., 2025). *However, none of these empirical analyses evaluated the separability of activation distributions in SAEs as a measure of polysemanticity.*

5.2 SAE-Based Model Control

The interpretable features learned by SAEs enable precise control over language model behavior. Several works have demonstrated effective steering by carefully selecting and manipulating SAE features, with approaches ranging from supervised methods for identifying relevant dimensions to frameworks using hypernetworks (Arad et al., 2025; He et al., 2025a,b; Bayat et al., 2025; Sun et al., 2025a; Minegishi et al., 2025). *However, prior SAE-based control methods did not utilize posterior probabilities, limiting their precision.*

5.3 Base Model Control and Causal Analysis

Complementing SAE-based approaches, researchers have developed techniques for direct activation control and causal analysis in the base language models. General frameworks for transporting activations facilitate intervention across model architectures (Rodriguez et al., 2025), while causal tracing methods enable precise localization and editing of specific knowledge or biases (Vig et al., 2020; Meng et al., 2022, 2023). These approaches offer foundational tools for probing and manipulating model behavior at the activation level. *However, again, they do not leverage posterior probabilities for better intervention and have been applied exclusively to base models, not to representations learned by SAEs.*

6 Conclusion

This work presents the first quantitative analysis of monosemanticity in SAEs compared to their dense base models through distributional lens. To better characterize monosemanticity, we introduce an activation distribution-aware concept separability score based on the Jensen–Shannon distance, which captures fine-grained distinctions in neuron activations across concepts. We also demonstrate that SAEs support more precise concept-level interventions than base models, particularly when using partial suppression. Building on this, we propose a new method, Attenuation via Posterior Probabilities, which achieves effective concept removal with least possible side effects.

7 Limitations

To make density estimation computationally feasible at scale, APP replaces standard kernel density estimation (KDE) with a histogram-based approximation. While this approach substantially improves efficiency, it also introduces certain limitations. As the number of histogram bins increases, the accuracy of the estimated activation distributions improves, but so does the computational cost. Consequently, achieving the best possible performance of APP, in terms of precise density estimation and separability, requires significantly higher computational resources and runtime. Future work could explore alternative KDE methods to better balance accuracy and efficiency.

Acknowledgment

This work was supported in part by the National Science Foundation (NSF) under grant IIS-2401685.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Omer Antverg and Yonatan Belinkov. 2022. On the pitfalls of analyzing individual neurons in language models. In *International Conference on Learning Representations*.

Dana Arad, Aaron Mueller, and Yonatan Belinkov. 2025. Saes are good for steering–if you select the right features. *arXiv preprint arXiv:2505.20063*.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*.

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. 2025. Steering large language model activations in sparse spaces. *arXiv preprint arXiv:2503.00177*.

Leonard Bereska and Stratis Gavves. 2024. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*. Survey Certification, Expert Certification.

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. 2024. Identifying functionally important features with end-to-end sparse dictionary learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Bart Bussmann, Patrick Leask, and Neel Nanda. 2024. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. 2019a. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Fahim Dalvi, Avery Nortonsmith, Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, and James Glass. 2019b. Neurox: A toolkit for analyzing individual neurons in neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9851–9852.

Can Demircan, Tankred Saanum, Akshay Kumar Jagadish, Marcel Binz, and Eric Schulz. 2025. Sparse autoencoders reveal temporal difference learning in large language models. In *The Thirteenth International Conference on Learning Representations*.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online. Association for Computational Linguistics.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/toy_model/index.html.

Ali Eslamian and Qiang Cheng. 2025. Tabnsa: Native sparse attention for efficient tabular data learning. *arXiv preprint arXiv:2503.09850*.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2025. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Muhammad Umair Haider, Hammad Rizwan, Hassan Sajjad, Peizhong Ju, and AB Siddique. 2025. Neurons speak in ranges: Breaking free from discrete neuronal attribution. *arXiv preprint arXiv:2502.06809*.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, and 1 others. 2024. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *arXiv preprint arXiv:2410.20526*.

Zirui He, Mingyu Jin, Bo Shen, Ali Payani, Yongfeng Zhang, and Mengnan Du. 2025a. Sae-ssv: Supervised steering in sparse representation spaces for reliable control of language models. *arXiv preprint arXiv:2505.16188*.

Zirui He, Haiyan Zhao, Yiran Qiao, Fan Yang, Ali Payani, Jing Ma, and Mengnan Du. 2025b. Saif: A sparse autoencoder framework for interpreting and steering instruction following of language models. *arXiv preprint arXiv:2502.11356*.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Jett Janiak, Chris Mathwin, and Stefan Heimersheim. 2023. Polysemantic attention head in a 4-layer transformer. https://www.lesswrong.com/posts/nuJFTS5iiJKT5G5yh/polysemantic-attention-head-in-a-4-layer-transformer. LessWrong Blog.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. 2025. Are sparse autoencoders useful? a case study in sparse probing. In *Forty-second International Conference on Machine Learning*.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, and 1 others. 2025. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *arXiv preprint arXiv:2503.09532*.

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

Victor Lecomte, Kushal Thaman, Rylan Schaeffer, Naomi Bashkansky, Trevor Chow, and Sanmi Koyejo. 2024. What causes polysemanticity? an alternative origin story of mixed selectivity from incidental causes. In *ICLR 2024 Workshop on Representational Alignment*.

Aaron J Li, Suraj Srinivas, Usha Bhalla, and Himabindu Lakkaraju. 2025. Interpretability illusions with sparse autoencoders: Evaluating robustness of concept representations. *arXiv preprint arXiv:2505.16004*.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 278–300, Miami, Florida, US. Association for Computational Linguistics.

J. Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Xiaoliang Luo, Akilles Rechardt, Guangzhi Sun, Kevin K Nejad, Felipe Yáñez, Bati Yilmaz, Kangjoo Lee, Alexandra O Cohen, Valentina Borghesani, Anton Pashkov, and 1 others. 2025. Large language models surpass human experts in predicting neuroscience results. *Nature human behaviour*, 9(2):305–315.

Simon C Marshall and Jan H Kirchner. 2024. Understanding polysemanticity in neural networks through coding theory. *arXiv preprint arXiv:2401.17975*.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Gouki Minegishi, Hiroki Furuta, Yusuke Iwasawa, and Yutaka Matsuo. 2025. Rethinking evaluation of sparse autoencoders through the representation of polysemous words. In *The Thirteenth International Conference on Learning Representations*.

Ari S. Morcos, David G.T. Barrett, Neil C. Rabinowitz, and Matthew Botvinick. 2018. On the importance of single directions for generalization. In *International Conference on Learning Representations*.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2016. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 2017. Feature visualization. *Distill*. Https://distill.pub/2017/feature-visualization.

Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. 2025. Sparse autoencoders learn monosemantic features in vision-language models. *arXiv preprint arXiv:2504.02821*.

Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Xl-wsd: An extra-large and cross-lingual evaluation framework for word sense disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13648–13656.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. 2024a. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. 2024b. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, and 80 others. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, marco cuturi, and Xavier Suau. 2025. Controlling language and diffusion models by transporting activations. In *The Thirteenth International Conference on Learning Representations*.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Adam Scherlis, Kshitij Sachan, Adam S Jermyn, Joe Benton, and Buck Shlegeris. 2022. Polysemanticity and capacity in neural networks. *arXiv preprint arXiv:2210.01892*.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, and 1 others. 2025. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*.

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

Xavier Suau, Pieter Delobelle, Katherine Metcalf, Armand Joulin, Nicholas Apostoloff, Luca Zappella, and Pau Rodríguez. 2024. Whispering experts: neural interventions for toxicity mitigation in language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Jiuding Sun, Sidharth Baskaran, Zhengxuan Wu, Michael Sklar, Christopher Potts, and Atticus Geiger. 2025a. Hypersteer: Activation steering at scale with hypernetworks. *arXiv preprint arXiv:2506.03292*.

Xiaoqing Sun, Alessandro Stolfo, Joshua Engels, Ben Wu, Senthooran Rajamanoharan, Mrinmaya Sachan, and Max Tegmark. 2025b. Dense sae latents are features, not bugs. *arXiv preprint arXiv:2506.15679*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

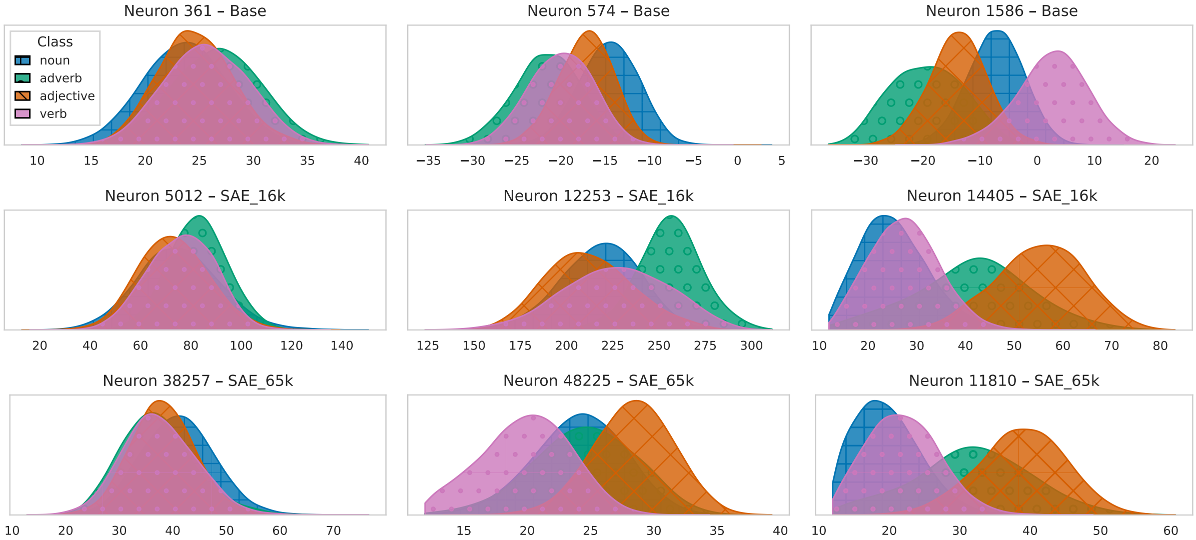

Figure 5: Across base model and SAEs (SAE-16k, SAE-65k), neurons exhibit varying degrees of separability in their activations. Some have completely overlapping activations across concepts, others show partial or clear separation. This variability underscores the importance of using distribution-aware metrics when assessing neuron monosemanticity.

Table 3: Concept Erasure Results (Deepseek)

Type	Method	POS			AG News			Emotions			DBpedia			NER		
		$\Delta_{\text{Acc}}\uparrow$	$\Delta_{\text{Conf}}\uparrow$	DPPL $\downarrow$	$\Delta_{\text{Acc}}\uparrow$	$\Delta_{\text{Conf}}\uparrow$	DPPL $\downarrow$	$\Delta_{\text{Acc}}\uparrow$	$\Delta_{\text{Conf}}\uparrow$	DPPL $\downarrow$	$\Delta_{\text{Acc}}\uparrow$	$\Delta_{\text{Conf}}\uparrow$	DPPL $\downarrow$	$\Delta_{\text{Acc}}\uparrow$	$\Delta_{\text{Conf}}\uparrow$	DPPL $\downarrow$
Base	APP	**0.185**	0.210	0.269	**0.184**	**0.384**	**0.182**	**0.100**	**0.191**	**0.031**	0.155	0.251	**0.133**	0.089	0.143	**0.043**
	Aura	0.049	0.069	**0.253**	0.015	0.071	0.250	0.067	0.146	0.205	**0.211**	0.281	0.607	0.038	0.071	0.284
	Range	0.057	**0.223**	0.722	0.174	0.217	1.100	0.078	0.151	0.427	0.149	**0.325**	1.191	0.089	**0.256**	1.315
	Adaptive	0.050	0.139	0.354	0.122	0.276	0.491	0.072	0.160	0.249	0.115	0.244	0.752	**0.090**	0.198	0.583
	Full	0.054	0.181	2.710	0.115	0.115	5.665	0.070	0.135	1.403	0.136	0.289	4.710	0.067	0.173	6.895
SAE	APP	0.307	0.327	7.077	0.487	0.809	**4.244**	0.092	0.473	**1.320**	**0.651**	0.658	**1.515**	0.567	0.877	**3.393**
	Aura	**0.414**	0.412	**2.461**	**0.514**	0.774	4.450	0.089	0.410	1.439	**0.652**	**0.774**	6.279	**0.589**	**0.883**	4.206
	Range	0.353	0.422	15.374	0.453	**0.846**	8.247	0.090	0.658	1.926	0.498	0.646	2.715	0.540	0.830	11.409
	Adaptive	0.338	**0.431**	12.211	0.433	0.831	6.934	**0.093**	0.652	1.538	0.488	0.631	2.287	0.501	0.789	9.057
	Full	0.245	0.307	23.146	0.200	0.478	30.988	0.090	**0.744**	3.085	0.498	0.676	6.877	0.300	0.489	29.032

A Deepseek Concept Erasure Experiments

On DeepSeek, our method APP is highly competitive on the concept-removal metrics. Across all 20 comparisons of Δ_{Acc} and Δ_{Conf} (5 datasets $\times$ 2 metrics $\times$ Base/SAE), APP ranks as the best or second-best method in 13 cases. By contrast, the next strongest method, Range, achieves a top-two ranking in 12 out of 20 comparisons. This demonstrates that APP performs on par with or better than the existing approaches. Importantly, APP is also the least disruptive method, achieving the lowest increase in perplexity (DPPL) in 8 out of 10 DeepSeek experiments (all except POS for Base and POS for SAE). In one of these two exceptions, APP remains highly competitive with a perplexity increase of 0.269 vs. 0.253 for AURA. Overall, APP combines strong concept-removal performance with the smallest degradation in language modeling quality on DeepSeek.

Additionally, as shown in Table 3, applying interventions in the SAE representation yields larger Δ_{Acc} and Δ_{Conf} than in the Base representation across methods: for APP, Δ_{Conf} increases on all five datasets and Δ_{Acc} increases on 4 out of 5 (slightly lower on *Emotions*), indicating that SAEs promote greater concept separability and thereby enable more effective concept removal.

B NER Neurons Activation

As shown in Figure 5, the NER dataset exhibits patterns similar to AG News. Certain neurons in both the base model and SAEs (e.g., leftmost plots) show considerable overlap in their activation distributions across the four classes, indicating limited class discrimination. In contrast, neurons in the middle and right columns reveal more separable activation patterns.

Table 4: Concept Erasure Detailed metrics D_{Acc}, D'_{Acc}, D_{Conf}, and D'_{Conf} for Gemma-2-2b.

Type	Method	POS				AG News				Emotions				DBpedia				NER			
		$D_{\text{Acc}} \uparrow$	$D'_{\text{Acc}} \downarrow$	$D_{\text{Conf}} \uparrow$	$D'_{\text{Conf}} \downarrow$	$D_{\text{Acc}} \uparrow$	$D'_{\text{Acc}} \downarrow$	$D_{\text{Conf}} \uparrow$	$D'_{\text{Conf}} \downarrow$	$D_{\text{Acc}} \uparrow$	$D'_{\text{Acc}} \downarrow$	$D_{\text{Conf}} \uparrow$	$D'_{\text{Conf}} \downarrow$	$D_{\text{Acc}} \uparrow$	$D'_{\text{Acc}} \downarrow$	$D_{\text{Conf}} \uparrow$	$D'_{\text{Conf}} \downarrow$	$D_{\text{Acc}} \uparrow$	$D'_{\text{Acc}} \downarrow$	$D_{\text{Conf}} \uparrow$	$D'_{\text{Conf}} \downarrow$
Base	APP	0.276	0.051	0.209	-0.002	0.356	0.008	0.254	-0.013	0.066	0.015	0.090	-0.005	0.126	0.013	0.101	-0.003	0.092	0.012	0.124	-0.006
	Aura	0.128	0.065	0.131	0.039	0.151	0.039	0.176	0.084	0.027	0.047	0.033	0.029	0.170	0.077	0.095	0.031	0.069	0.036	0.093	0.048
	Range	0.699	0.547	0.546	0.446	0.276	0.194	0.328	0.229	0.125	0.089	0.210	0.105	0.357	0.189	0.195	0.085	0.259	0.124	0.362	0.152
	Adaptive	0.567	0.409	0.446	0.317	0.145	0.094	0.220	0.131	0.096	0.064	0.148	0.058	0.239	0.118	0.136	0.037	0.176	0.081	0.237	0.063
	Full	0.710	0.594	0.553	0.483	0.284	0.187	0.328	0.248	0.126	0.087	0.212	0.099	0.364	0.194	0.196	0.093	0.259	0.125	0.358	0.182
SAE width: 65k ℓ_0 : 93	APP	0.581	0.278	0.386	0.116	0.624	0.022	0.412	-0.031	0.268	0.036	0.278	-0.001	0.411	0.012	0.216	-0.004	0.541	0.033	0.548	0.028
	Aura	0.615	0.209	0.388	0.075	0.593	0.015	0.411	0.044	0.374	0.037	0.318	0.007	0.413	0.055	0.216	0.003	0.554	0.034	0.560	0.045
	Range	0.676	0.624	0.494	0.438	0.672	0.085	0.434	0.309	0.414	0.268	0.381	0.357	0.412	0.159	0.219	0.163	0.552	0.054	0.557	0.007
	Adaptive	0.675	0.579	0.492	0.386	0.669	0.077	0.433	0.217	0.413	0.230	0.378	0.294	0.411	0.189	0.218	0.134	0.541	0.047	0.545	-0.012
	Full	0.676	0.676	0.495	0.495	0.674	0.071	0.434	0.428	0.415	0.295	0.382	0.380	0.413	0.138	0.220	0.201	0.553	0.171	0.559	0.167
SAE width: 16k ℓ_0 : 116	APP	0.519	0.303	0.323	0.131	0.619	0.037	0.343	-0.026	0.290	0.025	0.273	-0.001	0.332	0.020	0.150	-0.003	0.409	0.031	0.433	0.036
	Aura	0.397	0.147	0.254	0.061	0.623	0.027	0.343	0.057	0.375	0.036	0.306	0.026	0.333	0.076	0.151	0.011	0.414	0.025	0.441	0.026
	Range	0.617	0.555	0.414	0.368	0.624	0.118	0.348	0.244	0.396	0.297	0.342	0.333	0.334	0.126	0.153	0.100	0.413	0.114	0.442	0.136
	Adaptive	0.616	0.519	0.412	0.327	0.623	0.115	0.348	0.184	0.396	0.268	0.342	0.295	0.334	0.157	0.153	0.082	0.413	0.083	0.438	0.091
	Full	0.617	0.617	0.414	0.414	0.625	0.162	0.348	0.346	0.396	0.348	0.342	0.342	0.336	0.148	0.153	0.152	0.414	0.151	0.442	0.186
SAE width: 65k ℓ_0 : 197	APP	0.572	0.222	0.435	0.113	0.639	0.024	0.420	-0.038	0.273	0.026	0.275	-0.010	0.391	0.028	0.230	-0.005	0.556	0.025	0.667	0.013
	Aura	0.560	0.175	0.405	0.088	0.654	0.043	0.425	0.027	0.349	0.037	0.324	-0.003	0.392	0.095	0.234	-0.002	0.593	0.025	0.700	0.019
	Range	0.583	0.537	0.481	0.421	0.655	0.309	0.434	0.374	0.403	0.267	0.389	0.364	0.391	0.099	0.236	0.130	0.587	0.084	0.694	0.032
	Adaptive	0.583	0.472	0.480	0.350	0.655	0.310	0.434	0.304	0.401	0.230	0.384	0.273	0.391	0.134	0.235	0.088	0.569	0.066	0.671	0.014
	Full	0.583	0.583	0.481	0.481	0.656	0.311	0.435	0.434	0.403	0.293	0.389	0.386	0.392	0.175	0.236	0.216	0.591	0.298	0.697	0.313
SAE width: 16k ℓ_0 : 285	APP	0.526	0.144	0.414	0.107	0.669	0.041	0.353	-0.004	0.252	0.039	0.241	0.013	0.322	0.047	0.149	0.001	0.443	0.019	0.510	0.028
	Aura	0.505	0.071	0.390	0.030	0.674	0.125	0.350	0.003	0.321	0.034	0.283	-0.017	0.323	0.165	0.152	0.040	0.449	0.021	0.515	0.046
	Range	0.679	0.628	0.560	0.513	0.673	0.216	0.361	0.291	0.369	0.257	0.346	0.334	0.322	0.140	0.153	0.095	0.449	0.110	0.522	0.129
	Adaptive	0.679	0.499	0.558	0.410	0.674	0.232	0.360	0.211	0.369	0.247	0.345	0.273	0.322	0.177	0.153	0.069	0.448	0.071	0.519	0.083
	Full	0.680	0.680	0.560	0.559	0.676	0.462	0.361	0.360	0.369	0.306	0.346	0.345	0.323	0.170	0.153	0.129	0.449	0.282	0.523	0.332

Table 5: Concept Erasure Detailed metrics D_{Acc}, D'_{Acc}, D_{Conf}, and D'_{Conf} for DeepSeek-R1.

Type	Method	POS				AG News				Emotions				DBpedia				NER			
		$D_{\text{Acc}} \uparrow$	$D'_{\text{Acc}} \downarrow$	$D_{\text{Conf}} \uparrow$	$D'_{\text{Conf}} \downarrow$	$D_{\text{Acc}} \uparrow$	$D'_{\text{Acc}} \downarrow$	$D_{\text{Conf}} \uparrow$	$D'_{\text{Conf}} \downarrow$	$D_{\text{Acc}} \uparrow$	$D'_{\text{Acc}} \downarrow$	$D_{\text{Conf}} \uparrow$	$D'_{\text{Conf}} \downarrow$	$D_{\text{Acc}} \uparrow$	$D'_{\text{Acc}} \downarrow$	$D_{\text{Conf}} \uparrow$	$D'_{\text{Conf}} \downarrow$	$D_{\text{Acc}} \uparrow$	$D'_{\text{Acc}} \downarrow$	$D_{\text{Conf}} \uparrow$	$D'_{\text{Conf}} \downarrow$
Base	APP	0.198	0.013	0.199	-0.011	0.193	0.009	0.363	-0.021	0.106	0.007	0.162	-0.029	0.160	0.005	0.242	-0.008	0.095	0.007	0.133	-0.010
	Aura	0.081	0.032	0.103	0.034	0.029	0.014	0.125	0.054	0.099	0.032	0.179	0.033	0.241	0.030	0.348	0.067	0.051	0.013	0.074	0.003
	Range	0.067	0.011	0.288	0.065	0.303	0.130	0.687	0.469	0.148	0.070	0.315	0.164	0.181	0.032	0.347	0.021	0.144	0.054	0.379	0.123
	Adaptive	0.060	0.010	0.140	0.001	0.191	0.069	0.519	0.243	0.121	0.050	0.237	0.077	0.136	0.021	0.228	-0.016	0.121	0.031	0.236	0.039
	Full	0.062	0.008	0.291	0.109	0.287	0.171	0.691	0.575	0.147	0.078	0.318	0.183	0.183	0.047	0.353	0.064	0.127	0.060	0.371	0.199
SAE	APP	0.325	0.018	0.338	0.011	0.507	0.020	0.674	-0.134	0.098	0.006	0.339	-0.134	0.660	0.008	0.573	-0.085	0.581	0.014	0.838	-0.039
	Aura	0.424	0.010	0.455	0.043	0.522	0.008	0.688	-0.086	0.098	0.009	0.339	-0.071	0.664	0.012	0.576	-0.198	0.604	0.015	0.869	-0.014
	Range	0.420	0.067	0.479	0.057	0.497	0.044	0.644	-0.202	0.098	0.008	0.329	-0.329	0.515	0.017	0.347	-0.299	0.596	0.055	0.855	0.025
	Adaptive	0.375	0.036	0.423	-0.008	0.465	0.032	0.607	-0.224	0.098	0.005	0.328	-0.324	0.499	0.011	0.337	-0.294	0.546	0.045	0.793	0.004
	Full	0.422	0.177	0.481	0.174	0.501	0.301	0.652	0.174	0.098	0.008	0.329	-0.415	0.520	0.022	0.350	-0.326	0.603	0.303	0.867	0.378

C Details of Experiments

The detailed metrics (D_{Acc}, D'_{Acc}, D_{Conf}, and D'_{Conf}) before subtraction for Gemma-2-2b are reported in Table 4, and those for DeepSeek-R1 are presented in Table 5.

D Implementation Details

Hyperparameters. Range Masking, Adaptive Dampening, and Full Masking each rely on a saliency threshold hyperparameter that determines the top-$p\%$ of neurons considered most relevant to a given concept. For Gemma, we set $p = 0.3$ for POS, AG News and NER, $p = 0.2$ for Emotions and DBpedia. For DeepSeek, we use $p = 0.3$, 0.4, 0.1, 0.2, and 0.4 for POS, AG News, Emotions, DBpedia, and NER, respectively. In contrast, AURA (Suau et al., 2024) and our method, APP, do not require this hyperparameter. Both apply interventions across all neurons in the selected layer, avoiding the need to tune or justify a saliency cutoff. However, because SAE neurons typically activate on only a small subset of inputs for any given concept, we introduce an activation-frequency threshold τ to ensure reliability. Specifically, for each SAE neuron h_j^l, and each concept c_i, let $\mathcal{X}_{c_i}$ be the set of corresponding input samples. We define the firing frequency of h_j^l on c_i as

$$f_{j,i} = \frac{\left|\{\, x \in \mathcal{X}_{c_i} : h_j^l(x) > 0 \}\right|}{|\mathcal{X}_{c_i}|}.$$

We exclude neuron h_j^l from all concept-erasure methods if $f_{j,i} < \tau$, as sparse activations preclude meaningful intervention. In all experiments, we set $\tau = 0.1$.

Histogram-based KDE. To estimate the densities $f_{h_j^l|c_i}(x)$ required by APP, we use a kernel density estimation (KDE) implemented via a histogram-based approximation that preserves accuracy while greatly improving efficiency. During training, each neuron–concept activation distribution is discretized into B uniform-width bins (we use $B = 2048$) and we store the bin centers, counts, bandwidths, and normalization constants. At inference, a query activation x is evaluated only against these B centers rather than all N training activations. Let F be the number of neurons and Q the number of query points; this reduces complexity from $\mathcal{O}(FNQ)$ time and $\mathcal{O}(FN)$ memory for naïve KDE to $\mathcal{O}(FBQ)$ and $\mathcal{O}(FB)$, respectively, yielding an approximate N/B-fold improvement in both inference speed and memory usage ($B \ll N$).

Intervention Location and Scope. All interventions are applied at a consistent computation point (immediately before the language-model head) to ensure comparability across models and methods. Specifically, interventions are introduced after the MLP and residual addition in the final transformer block. For SAEs, they are applied directly after the SAE activation nonlinearity. To isolate causal effects on prediction, interventions are restricted to the token corresponding to the model's output label. This design allows precise measurement of each concept-erasure method's influence on the final decision while avoiding confounding effects from earlier tokens.

Dataset Statistics. We report the dataset sizes and splits used in all experiments to ensure transparency and reproducibility. The AG News dataset contains 120,000 training and 7,600 test examples. The DBpedia dataset includes 100,000 training and 25,000 test examples, both randomly sampled using a fixed seed to ensure reproducibility. For Emotions, we use 16,000 training and 2,000 test examples. The POS Tagging and NER datasets each consist of 100,000 training and 10,000 test examples, also sampled with a fixed random seed.

Model Specifications. As described in the main text, we use the Gemma-2 model with 2 billion parameters (Hugging Face name: `google/gemma-2-2b`). For DeepSeek, we employ the DeepSeek-R1 model, specifically the distilled variant based on Llama-8B with 8 billion parameters (Hugging Face name: `deepseek-ai/DeepSeek-R1-Distill-Llama-8B`).

Computation Details. All experiments were conducted using the university's high-performance computing (HPC) cluster managed via the Slurm workload manager. We used NVIDIA A100 (40 GB) and V100 (32 GB) GPUs for activation collection and intervention process. Each job was allocated 1 GPU, 1–10 CPU cores, and 50 GB of RAM.

Result Reliability. All quantitative results reported in the concept-erasure tables correspond to a single representative run. To verify the stability of our findings, we reran selected datasets (e.g., AG News) multiple times and observed identical outcomes and consistent relative rankings across methods. For instance, the top-performing method in the initial run remained the best in repeated trials.

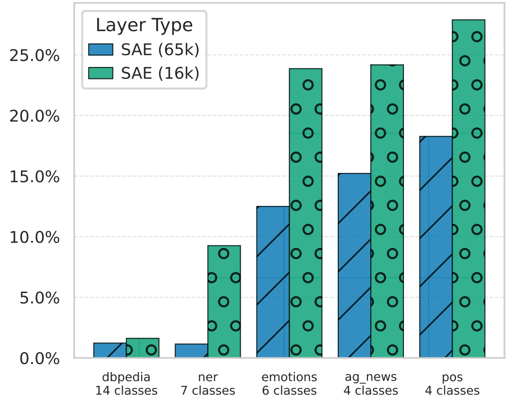

Figure 6: 65k SAEs show lower overlap than 16k, further supporting that greater capacity enables more distinct neuron-to-concept mappings.

These consistent results indicate that the reported values are reliable and reproducible.

Package Implementation and Parameter Settings. Our implementation leverages several standard Python and deep learning libraries. The Hugging Face Transformers package was used to load and run language models (AutoTokenizer, AutoModelForCausalLM). The transformer_lens and sae_lens libraries were employed for model inspection and sparse autoencoder integration. The scikit-learn library was used for evaluation metrics such as ROC-AUC, while pandas and numpy supported data processing. Default configurations were used for all libraries, with reproducibility ensured by fixing random.seed(42).

E All SAE Neurons Analysis

While our salient neuron analysis provides valuable insight into the most strongly responding neurons, it considers only a narrow slice of the activation space (specifically, the top 80 neurons per concept). This limited scope may miss neurons that, although not highly ranked by mean activation, are still consistently active across multiple concepts and contribute to polysemanticity. To address this limitation, we broaden our analysis to include all neurons that exhibit non-zero activation for any concept, offering a more comprehensive view of concept overlap beyond the most salient neurons. This extended analysis is conducted only for SAEs, since in the dense base model, all neurons are active across all inputs, rendering such overlap statistics uninformative. The results are shown in Figure 6, which reports the intersection-over-union of active neurons across concepts. By capturing both highly active and more subtly engaged neu-

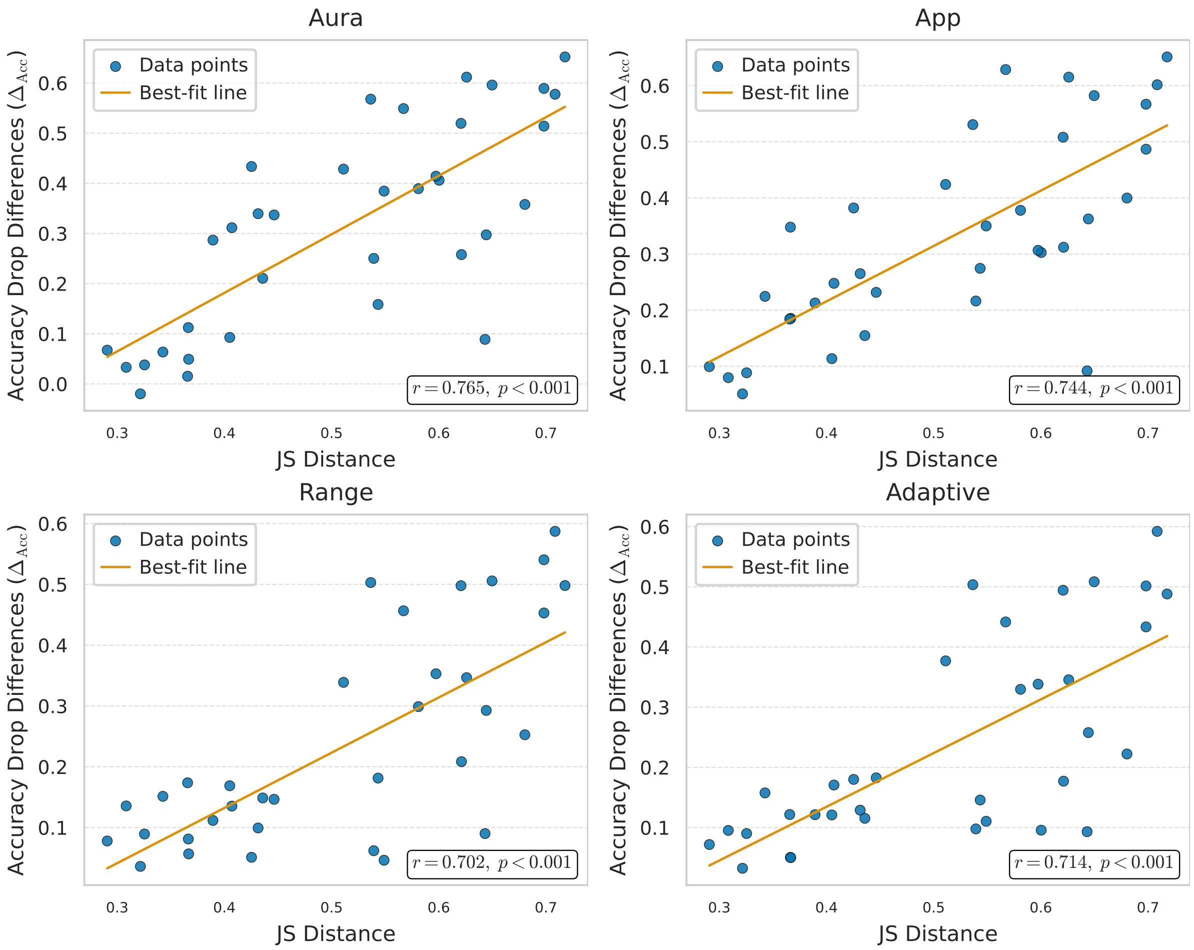

Figure 7: JS distance vs. erasure ability for all partial erasure methods.

rons, this analysis reveals a complete picture of polysemantic behavior. Consistent with our earlier findings shown in Figure 1, we observe that higher-capacity SAEs (e.g., 65k dimensions) exhibit lower neuron overlap than their lower-capacity counterparts. Together, Figures 1 and 6 demonstrate that while SAEs significantly reduce polysemanticity, they do not eliminate it entirely. Polysemantic neurons remain present, albeit to a lesser extent.

F Concept Erasure and JS Distance

As illustrated in Figure 7, all partial concept erasure methods exhibit a strong Pearson correlation with our JS separability score metric. Specifically, each method achieves a correlation coefficient greater than 0.7 with statistically significant p-values ($p < 0.01$), indicating a robust relationship between concept separability and erasure precision. These results suggest that partial erasure techniques can effectively leverage the separability inherent in activation representations to suppress the targeted concept while minimizing interference with unrelated ones. Among the four partial erasure approaches evaluated, AURA and APP demonstrate the highest correlations (0.765 and 0.744, respectively), highlighting their superior ability to exploit distributional distinctions between concept activations. We attribute this performance advantage to the fact that both AURA and APP explicitly model the distributions of both target and auxiliary concepts as discussed in Subsection 4.4 (Comparative Analysis of APP and AURA).

G JS Separability: DeepSeek vs. Gemma

Table 6 presents the JS separability scores computed for both the Gemma-2-2B and DeepSeek-R1 models across all datasets. As indicated in the table, every SAE variant consistently achieves higher separability scores than its respective base model. This consistent improvement confirms that the incorporation of SAEs enhances the distinction between concept activation distributions, leading to more interpretable internal representations. Moreover, when comparing SAEs with the same capac-

	POS	AG News	Emotions	DBpedia	NER
Gemma					
Base	0.343	0.366	0.322	0.405	0.308
SAE (width: 16k, ℓ_0 : 116)	0.539	0.650	0.431	0.621	0.581
SAE (width: 16k, ℓ_0 : 285)	0.425	0.567	0.389	0.543	0.511
SAE (width: 65k, ℓ_0 : 93)	**0.600**	**0.709**	**0.446**	**0.680**	**0.621**
SAE (width: 65k, ℓ_0 : 197)	0.549	0.626	0.407	0.644	0.537
DeepSeek					
Base	0.367	0.366	0.291	0.436	0.325
SAE	**0.597**	**0.698**	**0.643**	**0.718**	**0.698**

Table 6: JS separability score comparison across datasets for Gemma and DeepSeek. Bold values indicate the best score per dataset.

ity, those with higher sparsity (corresponding to lower ℓ_0 values) exhibit greater separability. These results confirms that sparsity plays a critical role in improving the distinctness of concept distributions.

H Salient Neuron Overlap with Top-P Selection.

In addition to the top-k analysis reported in the main text, we repeat the salient neuron overlap analysis using a top-p criterion, varying $p \in \{0.6, 0.7, 0.8, 0.9\}$. The results are reported in Table 7. Across all tested values of p, we observe the same qualitative trends as in Figure 1; SAEs consistently exhibit lower neuron overlap than the base model, indicating reduced polysemanticity. Moreover, higher-capacity (65k) SAEs demonstrate lower overlap than their 16k counterparts across all datasets. The consistency of these results across different top-p thresholds shows that our conclusions are robust to the choice of saliency selection strategy.

Dataset	Type	top-p = 0.6	top-p = 0.7	top-p = 0.8	top-p = 0.9
DBpedia	SAE (65k)	0.91%	0.99%	0.98%	1.25%
	SAE (16k)	1.05%	1.19%	1.22%	1.44%
	Base	11.57%	14.71%	16.88%	20.97%
NER	SAE (65k)	0.60%	0.75%	0.86%	1.03%
	SAE (16k)	4.46%	4.96%	6.25%	8.05%
	Base	24.20%	25.79%	29.83%	33.90%
Emotions	SAE (65k)	7.21%	8.02%	9.41%	11.42%
	SAE (16k)	14.75%	17.01%	19.79%	22.29%
	Base	37.17%	37.83%	40.42%	45.05%
AG News	SAE (65k)	8.08%	9.88%	11.75%	13.92%
	SAE (16k)	14.11%	16.26%	19.66%	22.92%
	Base	31.27%	33.93%	36.21%	40.34%
POS	SAE (65k)	12.97%	14.37%	15.97%	17.42%
	SAE (16k)	19.43%	22.17%	24.14%	26.17%
	Base	42.28%	46.71%	49.68%	51.99%

Table 7: Salient neuron overlap between models across datasets, where salient neurons are selected using top-p.