Steering CLIP's vision transformer with sparse autoencoders

Sonia Joseph[1,2] Praneet Suresh[1,2] Ethan Goldfarb[3]
Lorenz Hufe[4] Yossi Gandelsman[5] Robert Graham[2]
Danilo Bzdok[1,2] Wojciech Samek[4] Blake Aaron Richards[1,2]

[1]Mila [2]McGill University [3]Independent Researcher
[4]Fraunhofer HHI [5]UC Berkeley

sonia.joseph@mila.quebec
(Corresponding author)

Abstract

While vision models are highly capable, their internal mechanisms remain poorly understood– a challenge which sparse autoencoders (SAEs) have helped address in language, but which remains underexplored in vision. We address this gap by training SAEs on CLIP's vision transformer and uncover key differences between vision and language processing, including distinct sparsity patterns for SAEs trained across layers and token types. We then provide the first systematic analysis on the steerability of CLIP's vision transformer by introducing metrics to quantify how precisely SAE features can be steered to affect the model's output. We find that 10-15% of neurons and features are steerable, with SAEs providing thousands more steerable features than the base model. Through targeted suppression of SAE features, we then demonstrate improved performance on three vision disentanglement tasks (CelebA, Waterbirds, and typographic attacks), finding optimal disentanglement in middle model layers, and achieving state-of-the-art performance on defense against typographic attacks.

1. Introduction

Vision transformers have become fundamental to modern computer vision and have achieved remarkable performance across diverse tasks [7, 13, 31, 40]. However, despite their widespread adoption and success, we lack a deep understanding of how these models process and represent visual information internally. Although recent progress has been made in understanding the features of language models using decomposition techniques like sparse autoencoders (SAEs) [6, 9], similar interpretability advances in vision transformers remain limited.

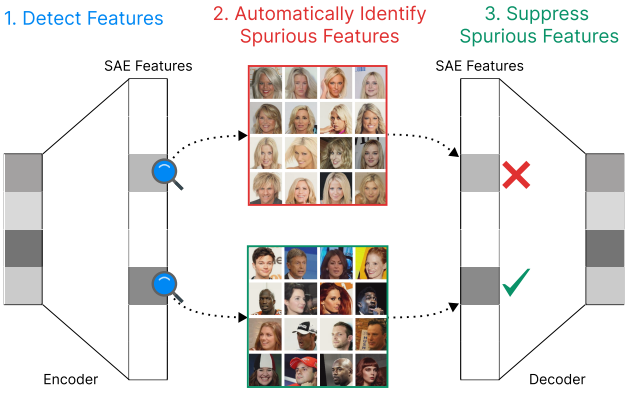


Figure 1. Our method improves performance on vision disentanglement tasks by detecting and suppressing features with CLIP SAEs. For CelebA, we suppress blondeness to improve gender classification. For details, see Section 7.1.

Our work addresses this gap by applying an SAE-based analysis to CLIP's vision transformer [40], revealing fundamental differences between vision and language processing mechanisms. This analysis is particularly important as vision transformers increasingly serve as building blocks for larger multimodal systems [15, 30, 31], where understanding their internal representations becomes crucial for safety and reliability.

We make several key contributions. First, we train SAEs on CLIP's vision encoder. We observe properties about the sparsity of the resulting SAEs, as measured by the L0-norm, or the number of SAE features that activate for a given input token. The L0 values of SAEs trained on the spatial tokens are higher at the center of the image. Spatial tokens have much higher L0s than the CLS token and SAEs trained on a language model, suggesting fundamental differences in the sparsity of vision and language distributions.

Continuing to characterize our SAEs, we introduce a steerability metric S, which measures how precisely SAE features can be manipulated to influence model outputs. Our analysis shows that approximately 10-15% features are steerable (as determined by a threshold on S). While SAEs and the base model have similar proportions of steerable components, SAEs' higher dimensionality yields a much larger absolute number of steerable elements. The S measurement provides a concrete framework for assessing the downstream utility of our SAEs.

Finally, we demonstrate the practical utility of our SAEs through improving the performance on three challenging vision disentanglement tasks: CelebA attribute separation, Waterbirds background suppression, and defense against typographic attacks. By selectively suppressing specific SAE features, we achieve improved performance on these tasks, with the most effective steering occurring in the middle layers of the model. For defense against typographic attacks, our method beats state-of-the-art performance.

Our findings not only advance our theoretical understanding of CLIP vision transformers but also provide practical tools for improving model behavior. We release our trained SAE models and code to facilitate further research in vision transformer interpretability. Our work establishes a foundation for more systematic approaches to understanding and controlling these increasingly important models.

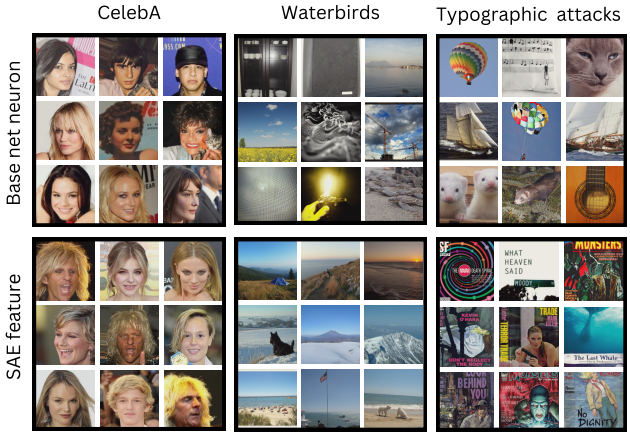


Figure 2. The top activating images show that base net features are polysemantic, while SAE features capture task-relevant attributes: blondeness (CelebA), land/water backgrounds (Waterbirds), and typographic images (typographic attacks). Feature selection details are in Section 7.1 and more examples are in Appendix 12, Figure 10.

2. Background

2.1. CLIP-ViT preliminaries

2.1.1. Contrastive Pre-training

CLIP [40] is trained via a contrastive loss to produce image representations from weak text supervision. The model includes an image encoder M_{image} and a text encoder M_{text} that map images and text descriptions to a shared latent space $\mathbb{R}^d$. The two encoders are trained jointly to maximize the cosine similarity between the output representations $M_{\text{image}}(I)$ and $M_{\text{text}}(t)$ for matching input text-image pairs (t, I):

$$\text{sim}(I,t) = \frac{\langle M_{\text{image}}(I), M_{\text{text}}(t)\rangle}{\|M_{\text{image}}(I)\|_2 \|M_{\text{text}}(t)\|_2}. \tag{1}$$

2.1.2. Zero-shot Classification with CLIP

Given a set of classes, the name of each class c_i (e.g., the class "tabby cat") is mapped to a fixed template(c_i) (e.g., "A photo of a {class}"), and encoded via the text encoder $M_{\text{text}}(\text{template}(c_i))$. The classification prediction for a given image I is the class c_i whose text representation is most similar to the image representation:

$$\arg\max_{c_i} \text{sim}(I, \text{template}(c_i)).$$

2.1.3. CLIP-ViT Architecture

The CLIP-ViT image encoder consists of a Vision Transformer followed by a linear projection[1]. Denoting the projection matrix by $P \in \mathbb{R}^{d \times d'}$:

$$M_{\text{image}}(I) = P(\text{ViT}(I)). \tag{2}$$

The input I to ViT is first split into K non-overlapping image patches that are encoded into K d'-dimensional image tokens. An additional learned token, called the class (CLS) token, is included and used later as the output token. Tokens are processed simultaneously by applying L alternating residual layers of multi-head self-attention (MSA) and MLP blocks.

2.2. Feature disentanglement with sparse autoencoders

One major challenge in neural network interpretability lies in the often non-interpretable and polysemantic nature of individual neurons [16]. Motivated by the sparse feature hypothesis Olshausen and Field [36], recent advancements in sparse dictionary learning Bricken et al. [6], Cunningham et al. [9] have shown that sparse autoencoders (SAEs) can effectively identify interpretable monosemantic directions in the latent space. For an input activation $x \in \mathbb{R}^{d_{\text{model}}}$

[1]The Vision Transformer (ViT) is applied to the input image $I \in \mathbb{R}^{H \times W \times 3}$ to obtain a d'-dimensional representation ViT(I)

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

from these components, the SAE computes the following decomposition:

$$\hat{x} + \epsilon(x) = \sum_{j=1}^{d_{\text{SAE}}} f_j(x) n_j + b + \epsilon(x). \tag{3}$$

This formulation decomposes the input into a reconstruction $\hat{x}$ through a sparse combination of feature vectors $n_j \in \mathbb{R}^{d_{\text{model}}}$, where each n_j is normalized to unit length. The feature activations $f_j(x) \in \mathbb{R}$ serve as sparse coefficients, while $b \in \mathbb{R}^{d_{\text{model}}}$ represents a bias term. The model is optimized using a combination of L_2 reconstruction loss and L_1 regularization to enforce sparsity in the activations.

3. Related Work

3.1. CLIP Interpretability

Several works have explored interpreting CLIP's behavior and representations. Studies have investigated CLIP's biases [45], attention layers [18, 25], and neurons [19, 21]. Recent works [4, 47] introduce task-agnostic concept discovery, demonstrating that CLIP's internal representations naturally align with human-interpretable concepts. Our work builds upon these findings by using SAEs not only to understand CLIP's representations at a finer degree of resolution, but also to actively steer CLIP's behavior through identified interpretable features.

3.2. Concept Bottleneck Models and Interpretability

Recent efforts to make deep neural networks more interpretable have led to various approaches, with Concept Bottleneck Models (CBMs) emerging as a promising direction. Traditional CBMs [26] require manually specified concepts and labeled attribute datasets, limiting their scalability. Recent work leverages large language models (LLMs) and vision-language models to overcome this limitation [8, 35, 38]. However, these approaches are subject to the biases of LLMs and still rely on pre-selecting concepts based on downstream tasks, which may not align with the model's learned representations. In contrast, our work uses SAEs, which extract unsupervised features from the model's internal representations.

3.3. Sparse Autoencoders for Model Interpretability

Sparse autoencoders (SAEs) have recently gained attention as tools for mechanistic interpretability [6, 9]. These works demonstrate that SAEs can effectively decompose neural networks into interpretable features, particularly in language models.

Recent studies have expanded the application of SAEs to vision tasks. Abdulaal et al. [1], Fry [17] and Daujotas [10, 11] explored how SAEs can extract interpretable features from CLIP's vision encoder, and Daujotas showed their potential in modifying image generation with diffusion models. Meanwhile, Rao et al. [41] leveraged CLIP embeddings to label SAE-derived concepts, enabling task-agnostic concept bottleneck models. Gorton [22] applied SAEs to InceptionV1, uncovering missing curve detectors. Additionally, Lim et al. [28] introduced PatchSAE for spatially localized attributions for fine-grained concept extraction.

Our paper builds upon past insights, uncovering novel vision-specific sparsity patterns in the SAEs' feature space, performing the first systematic and quantitative analysis on the steerability of CLIP's features, and applying CLIP SAEs to practical disentanglement tasks, achieving state-of-the-art performance on defense against typographic attacks.

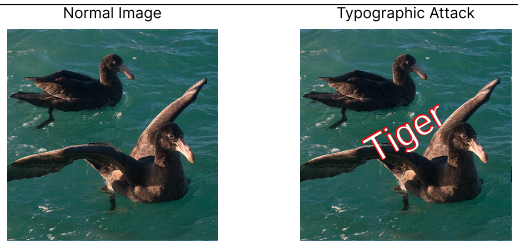


Figure 3. Typographic Attack on CLIP: On the left, an ImageNet-100 sample. On the right, the same image with 'tiger' written on it. As demonstrated by Goh et al. [21], this simple text overlay can mislead CLIP's zero-shot classification towards the attacker's intended label.

3.4. Typographic Attacks in CLIP Models

Typographic attacks [21] are a specific attack vector targeting vision-language models like CLIP, arising from their multimodal pretraining. These attacks involve inserting human-readable text into an input image to manipulate the model's prediction, as illustrated in Figure 3.

To improve robustness against such attacks, Materzyńska et al. [33] proposed a learned transformation applied to CLIP's output. PAINT [24] fine-tunes on synthetically generated images of typographic attacks and uses weight interpolation of the fine-tuned and the original model, to gain a more robust model. Azuma and Matsui [3] introduced Defense-Prefixes (DPs) which extend the idea of "class-prefix learning" [27, 29, 42, 44] towards defending against typographic attacks by inserting a learned token before the class name, leading to improved robustness without changing the model's parameters.

4. Training CLIP Vision SAEs

We train two variants of SAEs on activations from the residual stream of each layer of the CLIP-ViT-B-32 model. The Vanilla SAEs use the ReLU activation function with

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

the sparsity induced by L1 regularization, while the Top-K SAEs [20] use the Top-K activation function with a fixed sparsity as defined by the hyperparameter k. Both sets of SAEs are trained in general and task-specific settings. The general purpose SAEs are trained on ImageNet-1K to populate the dictionary with as many features as possible from the ViT's latent space, whereas the task-specific SAEs are trained on CelebA and Waterbirds datasets to populate the dictionary with features that are task-relevant. Full training and evaluation details of our SAEs are in Appendix 10.

5. Properties of CLIP SAEs

5.1. Comparing Spatial and CLS Token Behaviors

The L0 metric, which measures the number of activated features per patch and serves as a key indicator of SAE sparsity, reveals distinct activation patterns in CLIP SAEs. Spatial tokens maintain high activation counts (300-700), with central patches showing a higher L0, suggesting greater information density in these regions (Figure 4 a). The spatial bias in L0 persists up through the final layer, showing that some spatial information is retained (Appendix 9, Figure 8). The CLS token follows a distinct pattern from the spatial tokens, transitioning from sparse to rich representations around layers 6-7 (Figure 4 b, c). The L0 of the spatial tokens has a high degree of variance, reflecting high norm tokens (Figure 4).

5.2. Comparing CLIP and Language Model Sparsity

When contrasting CLIP with GPT-2, we find striking differences in the L0 values of the SAEs trained on their representations (Figure 4c). CLIP's spatial tokens maintain substantially higher L0 values, showing 3-14x higher activation counts than GPT-2 tokens (which average L0 = 20-50). Interestingly, CLIP's CLS token exhibits sparsity patterns more similar to those of language models in early layers before diverging.

These patterns suggest fundamental differences in information processing strategies. While language models maintain relatively consistent sparsity throughout their layers, CLIP demonstrates a dual behavior: spatial tokens preserve rich local features while the CLS token captures the compression, which aligns with observations in [2].

6. CLIP SAE Feature Steering

Building on our analysis of CLIP SAEs, we explore their potential for model control through feature activation steering—manipulating SAE features to influence model outputs predictably [46]. This investigation raises several key questions: how to measure the impact of feature manipulations on outputs, how SAE steering compares to direct model steering, how steering effectiveness varies across layers, and what proportion of features can be effectively steered.

To address these questions, we introduce a metric called *steerability*, which quantitatively characterizes an SAE feature's steering performance. This metric measures output probability distribution changes in response to changing the activation of a given feature, enabling systematic comparisons between SAE features, across SAE layers, and between SAEs and their base networks. Below, we define steerability metrics and use them to analyze properties of our CLIP SAEs.

6.1. Steerability Metrics

We conduct our analysis in the context of zero-shot CLIP classification. Following [41], we use CLIP's text encoder to encode a vocabulary $\mathcal{V} = v_1, v_2, \ldots, v_m$ containing thousands of potential feature names into a library of text embeddings $\mathcal{T} = t_1, t_2, \ldots t_m$.

For a given image $i \in I$, we first obtain its embedding i_{emb} through CLIP's vision encoder. Computing the dot product between i_{emb} and $\mathcal{T}$ followed by a softmax operation yields CLIP's probability distribution $P_i \in \mathbb{R}^{|\mathcal{V}|}$ over the vocabulary $\mathcal{V}$. To measure steering effects, we then select a feature f and replace its feature activation across all patches with a steering strength s during the forward pass. Let $\tilde{P}_i \in \mathbb{R}^{|\mathcal{V}|}$ CLIP's steered probability distribution over the vocabulary $\mathcal{V}$.

We now define feature-level metrics that help us to compare the steering capabilities of different features, and furthermore propose metrics to help us to compare SAE feature-level and neuron-level representations with regard to steerability.

6.1.1. Feature Level Steerability Metrics

The average probability difference of a feature f across images $\mathcal{I}$ is given by:

$$\Delta P_f = \left| \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} (\tilde{P}_i - P_i) \right|, \tag{4}$$

where $\tilde{P}_i$ and P_i are the predicted probabilities after and before feature activation, respectively.

Furthermore, we quantify f's steerability by:

$$S_f = \left| \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} (\tilde{P}_i - P_i)^2 \right|. \tag{5}$$

$\mathcal{S}_f$ is designed to quantify how a feature distributes its weight across concept vocabulary elements. For a feature with probability mass uniformly distributed among its top n_c concepts with value $\frac{1}{n_c}$, the metric has the property: $\sum_{i=1}^{n_c} \left(\frac{1}{n_c}\right)^2 = n_c \cdot \left(\frac{1}{n_c}\right)^2 = \frac{1}{n_c}$. This formulation ensures that polysemantic features with approximately uniform distributions will have the desirable property that a feature with

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

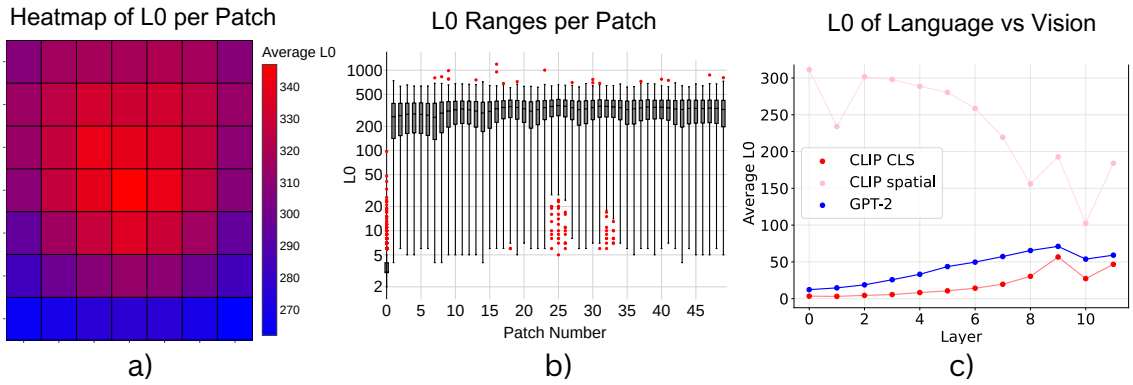


Figure 4. A visualization of the L0 values for an x64 vanilla SAE trained on all patches of CLIP-B-32 for Layer 0. a) A heatmap of average L0 per patch, overlaid on the original image grid, shows that there is a bias toward the center. The center bias remains constant for all layers (see Appendix 9). b) A box plot of L0s per patch reflects high-norm spatial tokens and a low-norm CLS token. c) A comparison between the L0s SAEs trained on the residual stream of GPT-2 and CLIP.

$n_c - 1$ top concepts always scores higher than one with n_c top concepts.

6.1.2. Layer Level Steerability Metrics

The layer-level metrics provide a more comprehensive view of steerability across the features within a layer. First, we define the average steerability of a layer, which reflects the overall steerability of all features in that layer:

$$\mathcal{S} = \frac{1}{|F|} \sum_{f \in F} S_f, \tag{6}$$

where F denotes the set of all features in the layer and S_f is the steerability of feature f. This metric aggregates the individual steerabilities of all features to give a sense of how well the layer as a whole can be steered.

To identify how many features within a layer are steerable, we define the number of steerable features as the proportion of features whose steerability exceeds a given threshold γ:

$$\#\mathcal{S} = \frac{1}{|F|} \sum_{f \in F} \mathbf{1}[S_f > \gamma], \tag{7}$$

where $\mathbf{1}[\cdot]$ is the indicator function. This metric indicates how many features exhibit significant steerability, characterizing how steering a given layer impacts the model's output.

Finally, we quantify concept coverage as the number of features whose steerability exceeds a higher threshold β, corresponding to a meaningful increase in concept probability:

$$\#\mathcal{C} = \sum_{f \in F} \mathbf{1}[S_f > \beta]. \tag{8}$$

The $\#C$ metric only indicates the number of vocabulary concepts a feature adheres to, without capturing the semantic relationships between promoted concepts. Thus, a feature may exhibit high specificity while remaining polysemantic (e.g., equally promoting semantically distant concepts like "carrot" and "organization"). We elaborate on this limitation in Appendix 11.2 and leave further exploration to future work.

6.2. Empirical Results on CLIP SAE Steerability

Our analysis reveals three key findings about steerable features in CLIP shown in Figures 5, 6, and 7. First, we discover features that can strongly influence CLIP's output distribution when steered, including "perfectly" steerable features that direct all probability mass towards one concept when steered to its asymptote. Second, using our layer-level metrics, we quantify how common these steerable features are. In CLIP's deeper layers, typically 10-15% of features are steerable, where $\#\mathcal{S} > |F| \cdot 0.10$ (Eq. 7). Finally, we empirically show that steering at the feature-level provides much better control over concept space than neuron-level steering, achieving more than 10 times the concept coverage $\#\mathcal{C}$ (Eq. 8) compared to the base model.

6.2.1. Some Properties of Steerable Features

When we amplify features to their maximum strength[2], their effect on probability distribution varies. Some features, like the 'Dragon' feature shown in Figure 5, strongly concentrate probability mass toward a single concept—in this case, the dragon logit—or a small cluster of related concepts. Other features, however, have a more diffuse effect: some appear to have little to no impact on the probability distribution, while others spread probability mass across a broad range of concepts, sometimes affecting hundreds or even thousands of them.

[2]We determine a value of 150 to be sufficient, as larger values produced no additional changes. This threshold may vary by SAE and base model.

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

This second pattern could have several explanations: these features might not meaningfully affect CLIP's output, they might need to work in concert with other features (suggesting our steering pushes activations too out-of-distribution), or our vocabulary might lack terms that accurately capture the feature's true semantic direction in CLIP's concept space (Appendix 11, Figure 9) [3].

6.2.2. The Frequency of Steerable Features

Using Eq. 5, we analyze the prevalence of steerable features in an SAE. Figure 6 presents our findings for a Vanilla SAE trained on layer 11 of CLIP's residual stream. Due to computational constraints, we analyze a subset of 12,000 features. Using a steerability threshold of $\gamma = 0.10$ we find 1,322 steerable features among 12,000 total features. While these constitute a minority, they still form a substantial subset of controllable features, including approximately 100 that are "perfectly" steerable. This metric provides a valuable tool for practitioners to optimize SAE training, increasing the number of steerable features and identifying suboptimal training outcomes.

6.2.3. Feature Steering vs. Neuron Steering

Our comparison of feature-level and neuron-level steering reveals a significant advantage for feature-level. While individual steerable neurons can match the steering strength of features, they are far rarer and provide access to a much smaller concept space (Figure 7). In a sample of 25% of features and neurons from layer 11, neuron steering accessed only 1% of concepts (about 50 concepts from our 5000-word vocabulary) in $\mathcal{V}$ (Eq. 8), while feature steering accessed approximately 11% (about 550 concepts).

7. Improving on Disentanglement Tasks with CLIP SAE Feature Steering

In this section, we apply our CLIP SAEs to three distinct downstream tasks designed to assess their disentanglement capabilities and demonstrate the universality of the learned features.

Our first objective is to evaluate the effectiveness of SAEs in suppressing spurious correlations, drawing inspiration from frameworks such as Pahde et al. [37] and Dreyer et al. [14]. To achieve this, we adopt experimental settings similar to those used in Nam et al. [34], Idrissi et al. [23], and Pham et al. [39]:

1. **Waterbirds** [43]: is an artificially generated dataset which combines bird photographs in the Caltech-UCSD Birds dataset[48] with background images from the Places dataset[49]. The goal is to classify the binary target attribute $Y =$ waterbird and $\overline{Y} =$ landbird given the spurious correlations with the background

[3] We choose $\#\mathcal{V} = 5,000$ to balance coverage and compute; a larger vocabulary could mitigate this limitation.

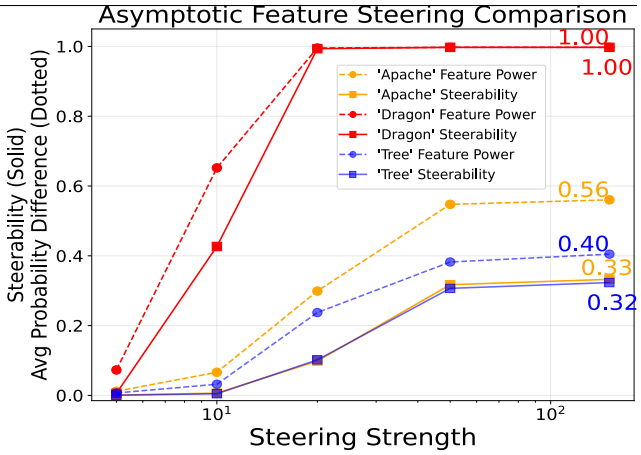


Figure 5. Asymptotic Feature Steerability Plot showing ΔP_f (dotted) and $\mathcal{S}_f$ (solid) versus steering strength. The "dragon" feature achieves perfect steering to a single concept, while "tree" and "apache" have similar $\mathcal{S}_f$ but different ΔP_f - 'tree' steers precisely to tree concepts, whereas "apache" disperses across helicopter-related concepts (e.g., "aircraft", "aviation", "rescue").

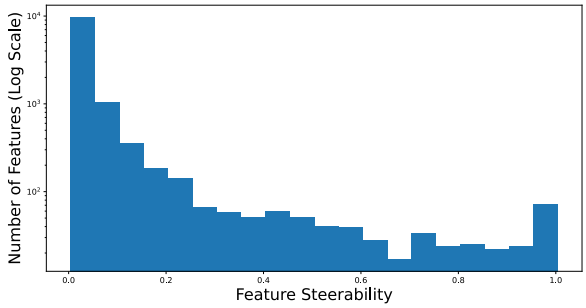


Figure 6. Log-Scale SAE Feature Steerability Histogram. Steerability scores at $s = 150.0$ for a Vanilla SAE trained on the residual stream of Layer 11. This is a sample of 12,000 features (representing roughly 25% of a total 49,152), of which 1,322 are steerable ($\mathcal{S}_f > 0.10$).

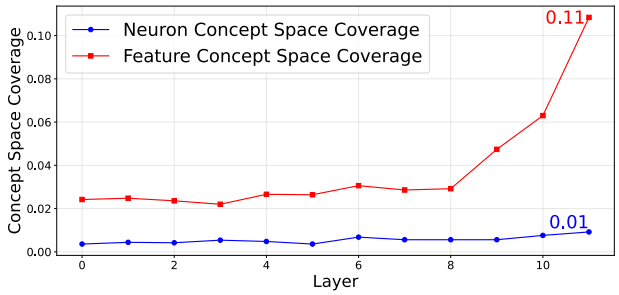


Figure 7. Concept Space Coverage, Feature vs Neurons, by Layer (Residual Stream). Notably, at layer 11, feature steering allows access to more than 10x the span of concept space that neuron steering does.

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

	CelebA				Waterbirds			
	Original Model		SAE		Original Model		SAE	
	Overall	Worst	Overall	Worst	Overall	Worst	Overall	Worst
Baseline	92.78	77.78	-	-	68.81	22.43	-	-
Layer 0	92.78	78.89	**93.19**	**79.44**	68.81	22.43	**69.42**	22.43
Layer 1	92.78	77.78	**92.92**	**78.89**	68.81	22.43	68.81	22.43
Layer 2	92.78	77.78	**93.47**	**79.44**	68.81	22.43	**71.95**	***24.61**
Layer 3	92.78	77.78	**93.06**	**78.89**	68.81	22.43	**69.54**	**24.14**
Layer 4	92.78	77.78	**93.19**	**80.00**	68.81	22.43	**68.93**	**23.21**
Layer 5	92.78	77.78	**93.19**	**78.89**	68.81	22.43	**69.05**	**22.74**
Layer 6	92.78	77.78	**93.19**	**80.00**	68.81	22.43	68.81	22.43
Layer 7	92.64	77.78	**93.61**	***81.11**	68.81	22.43	68.81	22.43
Layer 8	92.64	77.78	**93.47**	***81.11**	68.81	22.43	68.81	22.43
Layer 9	92.78	77.78	**93.06**	**79.44**	68.81	22.43	**70.19**	**22.74**
Layer 10	92.78	77.78	**93.47**	**80.00**	68.81	22.43	68.81	22.43
Layer 11	92.50	77.22	**93.33**	**80.56**	68.81	22.43	68.81	22.43

Table 1. Performance comparison showing overall and worst-group accuracy across different layers and tasks after targeted zero-ablations on the original model neurons (control), and targeted SAEs zero-ablations of the strict feature set $F^l_{\tau^*}$ (our method) (See Section 7.1). Ablations using SAE feature outperform ablations on the base model, due to SAE's finer level of granularity in representing the spurious feature. We pick the top-performing SAE for each layer and task. Full results for all SAE types, random controls, and the relaxed condition are in Appendix 13.

landscape A = water background and $\overline{A}$ = land background

2. **CelebA** [32]: consists of the face pictures of celebrities, featuring annotations on a variety of features. We use the binary gender label[4] as the target attribute Y = male and $\overline{Y}$ = female and utilize the feature A = blond and $\overline{A}$ = not blond as a spurious correlation.

Secondly, we aim to assess the utility of our SAEs in safety-critical applications by evaluating their effectiveness in enhancing the robustness of the CLIP model against typographic attacks. To this end, we follow the evaluation setup of Azuma and Matsui [3], using ImageNet-100[5], a subset of the ImageNet dataset [12], for tuning our method. We then benchmark our approach on RTA-100 [3], PAINT [24], and the dataset published by Materzyńska et al. [33].

7.1. Suppressing Spurious Correlations

To suppress the spurious correlations, we identify the SAE features that are most closely aligned to the spurious correlation A (Figure 1). To achieve this, we split the train dataset D into two subsets, D_A and $D_{\overline{A}}$, based on whether the spurious correlation is present in the datapoint. Then, for each layer l, select the SAE features F^l whose average activation on D_A is at least τ higher than their average activation on $D_{\overline{A}}$. Formally,

$$F^l = \left\{ j \mid \mathbb{E}_{x \sim D_A}[f^l_j(x)] > \mathbb{E}_{x \sim D_{\overline{A}}}[f^l_j(x)] + \tau \right\}, \quad (9)$$

where $f^l_j(\cdot)$ is the activation of the jth SAE feature of the lth layer, and $\tau \in \mathbb{R}$ is a scalar threshold.

We evaluate the CLIP model on the validation set, applying zero ablation to the set of SAE features F^l independently for each layer.

To systematically determine an appropriate threshold, we perform a grid search over $\tau \in [10^{-6}, 1.0]$ for each layer l. For each threshold value, we obtain a corresponding feature set, denoted as F^l_τ. These feature sets are evaluated to assess their impact using two metrics: $F^l_{\tau^*}$, which maximizes both overall accuracy and worst-group accuracy; and $F^l_{\tau'}$, which allows for a controlled performance drop ($\leq 4\%$ drop in accuracy) in non-target groups while prioritizing improvements in the worst-performing group.

These thresholds $F^l_{\tau^*}$ and $F^l_{\tau'}$ are chosen to balance overall model performance with fairness considerations. The 4% threshold is chosen to ensure that gains for the worst-performing group are achieved without excessive degradation in other groups.

7.1.1. Evaluation

We evaluate the CLIP model on the held out test set, while ablating the SAE features sets $F^l_{\tau^*}$ and $F^l_{\tau'}$. Additionally, we apply the same technique to CLIP's original feature space to verify that the SAE enhances disentanglement compared to the unmodified CLIP representation. To further validate our approach, we perform random ablations in

[4] While we acknowledge that gender is not binary, we follow this benchmark as it is standard in the literature.

[5] https://www.kaggle.com/datasets/ambityga/imagenet100

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

both the SAE and original CLIP feature spaces, ensuring that our method accurately identifies the relevant features.

7.1.2. Results

Targeted feature suppression with SAEs consistently improves disentanglement for every layer of the model for CelebA, and for many layers for Waterbirds (Table 1). Based on maximally activating images for $F^l_{\tau^*}$, the base model neurons are highly polysemantic, while SAE features show the task-relevant attribute (Figure 2).

The optimal layer for disentanglement is Layers 7 and 8 of CelebA, with worst group accuracy improving from 77.78% to 81.11%, and Layer 2 for Waterbirds, with worst group accuracy improving from 22.43% to 24.61% (Table 1). Under the relaxed condition, the best-performing accuracy goes up to 86.67% at Layer 9 for CelebA (Appendix 13, Table 6) and to 36.6% at Layer 5 for Waterbirds (Appendix 13, Table 8). We hypothesize that more diffuse features (like land and water backgrounds) may be optimally disentangled in earlier layers, before they become entangled with other features during the forward pass. More localized features (like blonde hair) may be better disentangled in later layers.

7.2. Suppressing Typographic Attacks

Defense	RTA 100	PAINT	Materzyńska et al. [33]
No Defense	0.66	0.63	0.54
DP	**0.73**	0.66	0.83
SAE (Ours)	0.72	**0.67**	0.79
DP + Ours	**0.73**	**0.67**	**0.88**

Table 2. Performance comparison of our method, Defense-Prefix (SoA), and their combination across three standard benchmarks for typographic attacks. See more in Section 7.2

.

To obtain a set of SAE features that encode typographic knowledge, we follow a similar approach to Section 7.1. For clarity, we focus our study of typographic attacks on CLIP's last residual layer. Throughout the remainder of this section, we omit the layer superscript (e.g., ^{l}) for readability.

We define two datasets: ImageNet-100 serves as $D_{\overline{T}}$, while D_T is constructed by applying a synthetically generated typographic attack to each image in ImageNet-100. Accordingly, the SAE feature set F is obtained following Equation 9.

Next, we construct a simple extension to F to improve the recall of typographic features. Specifically, for each feature direction n_j, we check whether it has a cosine similarity higher than λ with any n_m for $m \in M = \{m \mid f_m \in F\}$. If so, it is added to an enhanced set of features F^*, thereby capturing additional relevant features that may not have been initially selected.

7.2.1. Evaluation

Similar to Section 7.1, we perform a sweep over the λ and τ values on the validation set. The test results on the three benchmarks—RTA-100 [3], PAINT [24], and the dataset published by Materzyńska et al. [33]—are reported in Table 2.

Additionally, we report the results of Azuma and Matsui [3], who train a Defense-Prefix (DP) to improve CLIP's robustness against typographic attacks, as it represents the current state-of-the-art (SOTA). We also evaluate the combination of our method with DP. This integration is straightforward, as DP modifies only the input to CLIP's text transformer, while our approach exclusively affects the vision transformer.

7.2.2. Results

We conduct our tests with $\lambda = 0.2$ and $\tau = 1$, which result in F^* containing 495 out of 49,152 possible SAE features ($\sim 1\%$). The drop in ImageNet top-1 accuracy was 0.7 percentage points. Our method outperforms the current state-of-the-art on the PAINT dataset and remains competitive with DP on the [3] and Materzyńska et al. [33] datasets. When combining our method with the SOTA, we achieve top performance across all benchmarks (Table 2).

8. Conclusion

In this work, we train sparse autoencoders (SAEs) on CLIP's vision transformer, revealing sparsity differences between vision and language processing. We find that 10-15% of features in deeper layers are steerable, with CLIP SAE features providing 10x better concept coverage than base neurons. Through these steerable features, we demonstrate improved disentanglement capabilities and achieved state-of-the-art performance on typographic attack defense.

References

[1] Ahmed Abdulaal, Hugo Fry, Nina Montaña-Brown, Ayodeji Ijishakin, Jack Gao, Stephanie Hyland, Daniel C Alexander, and Daniel C Castro. An x-ray is worth 15 features: Sparse autoencoders for interpretable radiology report generation. *arXiv preprint arXiv:2410.03334*, 2024. 3

[2] Reduan Achtibat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. Attnlrp: attention-aware layer-wise relevance propagation for transformers. *arXiv preprint arXiv:2402.05602*, 2024. 4

[3] Hiroki Azuma and Yusuke Matsui. Defense-prefix for preventing typographic attacks on clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3644–3653, 2023. 3, 7, 8

[4] Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio Calmon, and Himabindu Lakkaraju. Interpreting CLIP with sparse linear concept embeddings (spliCE). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3

[5] Joseph Bloom. Open-source sparse autoencoders for all residual stream, 2024. Accessed: 2025-01-30. 2

[6] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023. 1, 2, 3

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1

[8] Aditya Chattopadhyay, Ryan Pilgrim, and Rene Vidal. Information maximization perspective of orthogonal matching pursuit with applications to explainable AI. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[9] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023. 1, 2, 3

[10] Gytis Daujotas. Case study: Interpreting, manipulating, and controlling clip with sparse autoencoders, 2024. 3

[11] Gytis Daujotas. Interpreting and steering features in images, 2024. 3

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 1

[14] Maximilian Dreyer, Frederik Pahde, Christopher J Anders, Wojciech Samek, and Sebastian Lapuschkin. From hope to safety: Unlearning biases of deep models by enforcing the right reasons in latent space. *arXiv preprint arXiv:2308.09437*, 2023. 6

[15] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1

[16] 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. Toy models of superposition, 2022. 2

[17] Hugo Fry. Towards multimodal interpretability: Learning sparse interpretable features in vision transformers, 2024. 3

[18] Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting CLIP's image representation via text-based decomposition. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[19] Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting the second-order effects of neurons in CLIP. In *The Thirteenth International Conference on Learning Representations*, 2025. 3

[20] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders, 2024. 4

[21] Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021. https://distill.pub/2021/multimodal-neurons. 3

[22] Liv Gorton. The missing curve detectors of inceptionv1: Applying sparse autoencoders to inceptionv1 early vision. In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024. 3

[23] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pages 336–351. PMLR, 2022. 6

[24] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. *Advances in Neural Information Processing Systems*, 35:29262–29277, 2022. 3, 7, 8

[25] Sonia Joseph and Neel Nanda. Laying the foundations for vision and multimodal mechanistic interpretability & open problems. *AI Alignment Forum*, 2024. 3

[26] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020. 3

[27] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023. 3

[28] Hyesu Lim, Jinho Choi, Jaegul Choo, and Steffen Schneider. Sparse autoencoders reveal selective remapping of visual concepts during adaptation, 2024. 3

[29] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 3

[30] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 1

[31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1

[32] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018. 7

[33] Joanna Materzyńska, Antonio Torralba, and David Bau. Disentangling visual and written concepts in clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16410–16419, 2022. 3, 7, 8

[34] Junhyun Nam, Jaehyung Kim, Jaeho Lee, and Jinwoo Shin. Spread spurious attribute: Improving worst-group accuracy with spurious attribute estimation. *arXiv preprint arXiv:2204.02070*, 2022. 6

[35] Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[36] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997. 2

[37] Frederik Pahde, Maximilian Dreyer, Leander Weber, Moritz Weckbecker, Christopher J Anders, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. Navigating neural space: Revisiting concept activation vectors to overcome directional divergence. *arXiv preprint arXiv:2202.03482*, 2022. 6

[38] Konstantinos P. Panousis, Dino Ienco, and Diego Marcos. Sparse linear concept discovery models, 2023. 3

[39] Alan Pham, Eunice Chan, Vikranth Srivatsa, Dhruba Ghosh, Yaoqing Yang, Yaodong Yu, Ruiqi Zhong, Joseph E Gonzalez, and Jacob Steinhardt. The effect of model size on worst-group generalization. *arXiv preprint arXiv:2112.04094*, 2021. 6

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2

[41] Sukrut Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery, 2024. 3, 4

[42] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 3

[43] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. 6

[44] Idan Schwartz, Vésteinn Snæbjarnarson, Hila Chefer, Serge Belongie, Lior Wolf, and Sagie Benaim. Discriminative class tokens for text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22725–22735, 2023. 3

[45] Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, Philip Mansfield, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. Large language models encode clinical knowledge, 2022. 3

[46] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. 4

[47] Johanna Vielhaben, Dilyara Bareeva, Jim Berend, Wojciech Samek, and Nils Strodthoff. Beyond scalars: Concept-based alignment analysis in vision transformers. *arXiv preprint arXiv:2412.06639*, 2024. 3

[48] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 6

[49] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 6

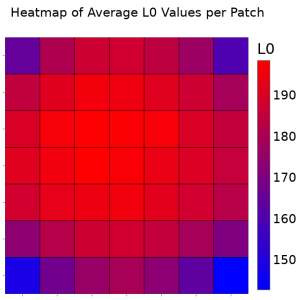
Steering CLIP's vision transformer with sparse autoencoders

Supplementary Material

9. L0 comparison details

The L0 values for language SAEs were collected from open source sparse autoencoders trained on GPT2-s residual stream [5].

Figure 8. Heatmap of average L0 per layer shows patch sparsity still retains its spatial bias even in Layer 11.



10. SAE training details and statistics

10.1. Training procedure

Architecture and Optimization. The SAEs were configured with an expansion factor of 64, mapping the base model's 768-dimensional activation space to a dictionary of 49,152 features. For initialization, we set the encoder weights to be the transpose of the decoder weights. The SAEs were trained using the Adam optimizer, sweeping the initial learning rate from 1e-5 to 1e-1, and employed a cosine annealing learning rate schedule with a 200-step warmup. Training used a batch size of 4096 samples.

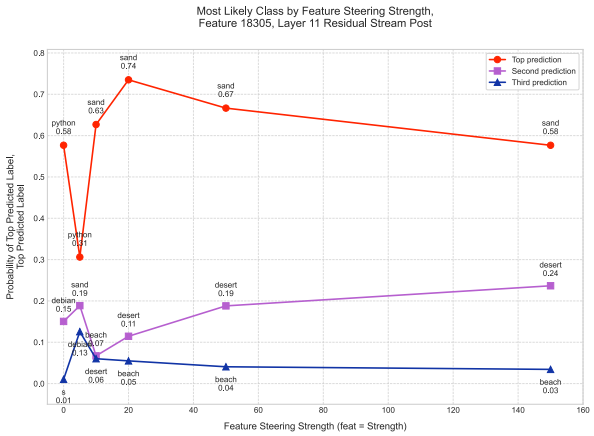
Sparsity and Losses. Two SAE variants were trained to minimize the MSE reconstruction loss while enforcing sparsity through different mechanisms: (1) Top-K SAEs, which enforced fixed sparsity levels ($k \in \{64, 128, 256\}$) via masking, and (2) Vanilla SAEs, which used ℓ_1 regularization with the coefficient swept over $[10^{-12}, 1]$, to induce variable sparsity. To prevent dead features that rarely activate, ghost grads auxiliary loss was used.

Training Data. General-purpose SAEs were trained on ImageNet1k for 1 epoch. For task-specific SAEs, we used CelebA and an augmented Waterbirds dataset, both trained for 2 epochs. The Waterbirds dataset augmentations included rotations ($\pm 10°$, $\pm 20°$), 5–15% edge cropping, contrast enhancements (factors of 1.1–1.3), and horizontal flips.

11. Steering Metric

11.1. More steering curves

Figure 9. Feature Concept Space Throughout Steering. This is a plot of the top three logits for each steering strength run. This visualizes the causal link between steering strength and CLIP prediction, and indicates whether a feature is semantically coherent or polysemantic.



In this case, we believe this feature's 'true concept' to not be present in the chosen vocabulary due to lack of asymptotic convergence towards a single concept. However, this feature exhibits strong semantic coherence, as the top 10 most highly promoted concepts for this feature are: 'sand', 'desert', 'beach', 'ground', 'rings', 'floor', 'ring', 'soil', 'mouse', and 'shell'

11.2. Limitations of the steerability metric

To address the limitation of the steerability metric not capturing the relationship between semantic concepts, we propose measuring the weighted distance between promoted concepts and the mean vocabulary vector:

$$\mu_V = \frac{1}{|V|} \sum_{v \in V} v \tag{10}$$

For a feature f, we compute:

$$D_f = \sum_{v \in V} P(v|f) \|v - \mu_V\|_2 \tag{11}$$

where $P(v|f)$ is the probability assigned to vocabulary element v by feature f. This metric shows positive correlation with qualitatively observed monosemantic/steerable features.

The metric is theoretically justified when vocabulary elements are uniformly distributed on the unit sphere (implying $\mu_V \approx 0$), as semantically related concepts cluster to-

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

gether while unrelated concepts are approximately orthogonal. We leave the rest to future work.

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

12. Maximally Activating Images

Figure 10. The top activating images show that base net features are polysemantic, while SAE features capture task-relevant attributes: blondeness (CelebA), land/water backgrounds (Waterbirds), and typographic images (typographic attacks).

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

12.1. SAE Evaluations

12.1.1. Vanilla SAEs (all patches)

L1 Coef.	Exp. Var.	L0	Layer	Sublayer	Avg Img L0	Avg CLS L0	Cos Sim	Recon Cos Sim	CE	Recon CE	Zero Abl CE	CE Rec
1e-4	0.892	57.31	0	mlp_out	2862.89	3.39	0.954	0.978	3.412	3.501	4.339	90.35
8e-5	0.910	84.88	0	mlp_out	4094.50	5.13	0.962	0.982	3.415	3.491	4.342	91.77
5e-5	0.943	160.48	0	mlp_out	7718.12	8.24	0.974	0.988	3.411	3.467	4.336	93.93
1e-5	0.987	598.23	0	mlp_out	29296.53	36.43	0.994	0.998	3.414	3.430	4.341	98.37
1e-4	0.817	249.88	1	mlp_out	11700.90	3.69	0.910	0.897	3.414	4.521	16.305	91.42
8e-5	0.851	329.70	1	mlp_out	15910.30	5.28	0.928	0.921	3.417	4.252	16.309	93.52
5e-5	0.907	592.62	1	mlp_out	28769.50	6.85	0.955	0.953	3.413	3.895	16.308	96.27
1e-5	0.984	1478.89	1	mlp_out	72397.80	88.58	0.992	0.994	3.415	3.477	16.309	99.52
8e-5	0.850	390.00	2	mlp_out	19407.09	13.54	0.936	0.984	3.413	3.504	6.179	96.71
1e-4	0.817	294.43	2	mlp_out	14676.46	12.23	0.922	0.979	3.411	3.541	6.176	95.31
5e-5	0.903	690.89	2	mlp_out	33472.50	21.24	0.958	0.991	3.412	3.460	6.180	98.28
1e-5	0.981	1939.23	2	mlp_out	88634.21	364.95	0.992	0.999	3.413	3.421	6.183	99.71
1e-4	0.777	417.00	3	mlp_out	20666.06	7.29	0.905	0.988	3.415	3.497	4.581	92.93
8e-5	0.823	572.56	3	mlp_out	27773.92	8.99	0.924	0.991	3.413	3.479	4.582	94.37
5e-5	0.886	972.85	3	mlp_out	48126.45	16.30	0.952	0.995	3.413	3.451	4.582	96.82
1e-5	0.982	1977.63	3	mlp_out	93453.44	628.76	0.993	0.999	3.415	3.423	4.583	99.36
1e-4	0.771	406.16	4	mlp_out	20074.38	16.24	0.903	0.990	3.409	3.481	4.789	94.84
8e-5	0.816	560.38	4	mlp_out	27463.11	20.32	0.923	0.992	3.411	3.465	4.788	96.09
5e-5	0.883	984.73	4	mlp_out	48444.25	35.74	0.951	0.995	3.409	3.440	4.786	97.73
1e-5	0.982	1975.18	4	mlp_out	94364.62	1113.69	0.993	0.999	3.412	3.416	4.789	99.72
1e-4	0.768	368.31	5	mlp_out	18387.25	31.38	0.903	0.990	3.409	3.470	5.142	96.46
8e-5	0.810	506.74	5	mlp_out	25240.54	39.93	0.921	0.992	3.412	3.461	5.144	97.22
5e-5	0.882	930.02	5	mlp_out	46145.45	93.39	0.951	0.995	3.414	3.438	5.145	98.58
1e-5	0.983	1840.60	5	mlp_out	87745.34	1284.78	0.994	0.999	3.407	3.410	5.139	99.85

Table 3. CLIP-ViT-B-32 vanilla sparse autoencoder performance metrics for all patches.

12.1.2. Top-K SAEs (all patches)

Exp. Var.	L0	Layer	Sublayer	Cos Sim	Recon Cos Sim	CE	Recon CE	Zero Abl CE	CE Rec
0.83	64	1	resid_post	0.888	1.0	6.762	6.762	6.908	100.0
0.68	64	2	resid_post	0.779	1.0	6.762	6.762	6.908	100.0
0.70	64	3	resid_post	0.790	1.0	6.762	6.762	6.908	100.0
0.80	64	4	resid_post	0.858	1.0	6.762	6.762	6.908	100.0
0.69	64	5	resid_post	0.791	1.0	6.762	6.762	6.908	100.0
0.78	64	6	resid_post	0.863	1.0	6.762	6.762	6.908	100.0
0.79	64	7	resid_post	0.881	1.0	6.762	6.762	6.908	100.0
0.81	64	8	resid_post	0.897	1.0	6.762	6.762	6.908	100.0
0.83	64	9	resid_post	0.906	1.0	6.762	6.762	6.908	100.0
0.82	64	10	resid_post	0.893	1.0	6.762	6.762	6.908	100.0
0.78	64	11	resid_post	0.883	1.0	6.762	6.762	6.908	100.0

Table 4. CLIP-ViT-B-32 Top-K sparse autoencoder performance metrics for all patches.

13. Disentanglement Task Full Results

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

Table 5. Layer-wise Accuracies (%) for CelebA, under the strict condition

	Vanilla		Top K=64		Top K=128		CelebA K=64		CelebA K=128	
Layer	Overall	Worst	Overall	Worst	Overall	Worst	Overall	Worst	Overall	Worst
SAE Feature Ablation										
L0	92.78	**78.89**	92.78	77.78	**93.19**	**79.44**	92.78	77.78	92.78	77.78
L1	**92.92**	77.78	92.78	77.78	**92.92**	**78.89**	92.78	77.78	92.78	77.78
L2	92.78	77.78	**93.47**	**79.44**	92.64	**78.33**	92.78	77.78	-	-
L3	92.64	**78.33**	92.78	77.78	**93.06**	**78.89**	92.78	77.78	92.78	77.78
L4	**92.92**	**78.33**	92.78	77.78	**93.19**	**80.0**	92.78	77.78	92.78	77.78
L5	**93.06**	**78.89**	**92.92**	**78.33**	**93.19**	**78.89**	**93.19**	**78.89**	92.64	**78.33**
L6	**93.19**	**79.44**	**93.19**	**79.44**	**93.06**	**79.44**	**93.06**	**79.44**	**93.19**	**80.0**
L7*	**93.06**	**80.56**	**93.47**	**80.56**	**93.33**	***81.11**	**93.33**	**80.56**	**93.61**	***81.11**
L8*	**93.61**	**80.56**	**93.47**	***81.11**	**93.33**	**80.56**	**93.19**	**80.0**	**92.92**	**78.33**
L9	**92.92**	**78.89**	**92.92**	**78.33**	**92.92**	**78.33**	**93.06**	**79.44**	-	-
L10	**93.47**	**80.0**	**92.92**	**78.89**	92.78	77.78	**92.92**	**79.44**	92.78	77.78
L11	**93.33**	**80.56**	92.78	77.78	**92.92**	**78.33**	**92.92**	**78.89**	**93.19**	**78.89**
Random SAE Feature Ablation										
L0	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78
L1	92.64	77.78	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78
L2	92.64	77.78	92.78	77.78	92.78	77.78	92.78	77.78	-	-
L3	92.64	77.78	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78
L4	92.64	77.78	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78
L5	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78
L6	92.78	77.78	92.78	77.78	92.64	77.78	92.64	77.78	92.78	77.78
L7	92.78	77.78	92.78	77.78	92.78	77.78	92.64	77.78	92.78	77.78
L8	92.64	77.78	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78
L9	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78	-	-
L10	92.64	77.78	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78
L11	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78
Base Network Neuron Ablation										
L0	92.78	**78.89**	-	-	-	-	-	-	-	-
L1	92.78	77.78	-	-	-	-	-	-	-	-
L2	92.78	77.78	-	-	-	-	-	-	-	-
L3	92.78	77.78	-	-	-	-	-	-	-	-
L4	92.78	77.78	-	-	-	-	-	-	-	-
L5	92.78	77.78	-	-	-	-	-	-	-	-
L6	92.78	77.78	-	-	-	-	-	-	-	-
L7	92.64	77.78	-	-	-	-	-	-	-	-
L8	92.64	77.78	-	-	-	-	-	-	-	-
L9	92.78	77.78	-	-	-	-	-	-	-	-
L10	92.78	77.78	-	-	-	-	-	-	-	-
L11	92.5	77.22	-	-	-	-	-	-	-	-

Note: Results on the CelebA disentanglement task with targeted ablation of SAE features on the residual stream. The baseline values are 92.78% for overall accuracy and 77.78% for worst group accuracy. Bolded values show an improvement over baseline. The best-performing layers (by worst group accuracy) and their corresponding SAEs are marked with an asterisk. The base network ablation does not depend on SAE type, so the value is the same across all SAE types.

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

Table 6. Layer-wise Accuracies (%) for CelebA, under the relaxed condition

	Vanilla		Top K=64		Top K=128		CelebA K=64		CelebA K=128	
Layer	Overall	Worst	Overall	Worst	Overall	Worst	Overall	Worst	Overall	Worst
SAE Feature Ablation										
L0	92.78	**78.89**	92.08	76.11	**93.61**	**81.67**	**92.92**	**78.33**	**93.06**	**78.33**
L1	**93.06**	**80.56**	**93.33**	**79.44**	**93.89**	**80.0**	**93.33**	**80.56**	92.78	77.78
L2	**93.89**	**82.78**	**92.92**	**81.11**	**93.47**	**79.44**	**94.17**	**82.22**	-	-
L3	**94.03**	**83.33**	**93.19**	**81.11**	**93.06**	**78.89**	**93.75**	**80.56**	**93.47**	**82.22**
L4	**93.33**	**79.44**	**93.06**	**78.33**	**93.19**	**80.0**	**93.19**	**80.56**	**93.47**	**79.44**
L5	**93.61**	**80.56**	**93.47**	**80.0**	**93.06**	**79.44**	**93.19**	**78.89**	92.64	**78.33**
L6	**93.47**	**81.11**	92.5	**81.11**	**93.06**	**79.44**	92.36	77.78	**93.19**	**80.0**
L7	**93.06**	**80.56**	**93.75**	**84.44**	**93.47**	**81.67**	**94.03**	**84.44**	**94.03**	**84.44**
L8	**94.03**	**82.78**	**93.47**	**81.11**	**94.03**	**85.0**	**93.89**	**83.89**	**94.17**	**85.56**
L9*	**94.72**	***86.67**	**93.06**	**79.44**	**93.47**	**81.11**	**93.89**	**86.11**	-	-
L10	**93.47**	**80.0**	**92.92**	**78.89**	92.78	77.78	**93.19**	**79.44**	**94.17**	**83.89**
L11	**94.31**	**83.33**	92.78	77.78	92.36	**78.33**	**92.92**	**78.89**	**93.19**	**79.44**
Random SAE Feature Ablation										
L0	92.78	77.78	92.78	77.78	**92.92**	**78.33**	92.78	77.78	92.64	77.78
L1	92.64	77.78	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78
L2	92.64	77.78	92.78	77.78	92.78	77.78	92.5	**78.33**	-	-
L3	92.5	77.22	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78
L4	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78
L5	92.64	77.78	92.64	77.78	92.64	77.78	92.78	77.78	92.78	77.78
L6	92.78	**78.89**	**92.92**	77.78	92.64	77.78	92.78	77.78	92.78	77.78
L7	92.78	77.78	92.64	77.78	92.5	77.22	92.64	77.78	92.64	77.78
L8	92.78	**78.33**	92.78	77.78	92.64	77.22	92.64	77.78	**92.92**	**78.33**
L9	92.64	77.78	92.78	77.78	92.64	77.78	**92.92**	**78.89**	-	-
L10	92.64	77.78	92.78	77.78	92.78	77.78	92.78	77.78	92.64	77.78
L11	92.78	77.78	92.78	77.78	92.78	77.78	92.78	77.78	92.5	77.22
Base Network Neuron Ablation										
L0	**93.19**	**80.0**	-	-	-	-	-	-	-	-
L1	**93.47**	**80.56**	-	-	-	-	-	-	-	-
L2	**93.47**	**79.44**	-	-	-	-	-	-	-	-
L3	92.78	77.78	-	-	-	-	-	-	-	-
L4	**92.92**	77.78	-	-	-	-	-	-	-	-
L5	**92.92**	77.78	-	-	-	-	-	-	-	-
L6	92.78	77.78	-	-	-	-	-	-	-	-
L7	**93.75**	**80.0**	-	-	-	-	-	-	-	-
L8	92.64	77.78	-	-	-	-	-	-	-	-
L9	92.78	77.78	-	-	-	-	-	-	-	-
L10	92.78	77.78	-	-	-	-	-	-	-	-
L11	**93.33**	**79.44**	-	-	-	-	-	-	-	-

Note: The baseline values are 92.78% for overall accuracy and 77.78% for worst group accuracy. Base network ablation is not dependent on the SAE type, so the value is the same across all SAE types.

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

Table 7. Layer-wise Accuracies (%) for Waterbirds, under the strict condition

	Vanilla		Top K=64		Waterbirds K=64	
Layer	Overall	Worst	Overall	Worst	Overall	Worst
SAE Feature Ablation						
L0	68.81	22.43	**69.42**	20.25	68.81	22.43
L1	68.81	22.43	68.81	22.43	68.81	22.43
L2*	**68.99**	***24.61**	**71.95**	21.65	68.81	22.43
L3	**69.54**	**24.14**	68.81	22.43	68.81	22.43
L4	**68.93**	**23.21**	68.81	22.43	68.81	22.43
L5	**69.05**	**22.74**	68.81	22.43	68.81	22.43
L6	68.81	22.43	68.81	22.43	68.81	22.43
L7	68.81	22.43	68.81	22.43	68.81	22.43
L8	68.81	22.43	68.81	22.43	68.81	22.43
L9	**70.19**	**22.74**	68.81	22.43	68.81	22.43
L10	68.81	22.43	68.81	22.43	68.81	22.43
L11	68.81	22.43	68.81	22.43	68.81	22.43
Random SAE Feature Ablation						
L0	68.81	22.43	68.81	22.43	68.81	22.43
L1	68.81	22.43	68.81	22.43	68.81	22.43
L2	68.81	22.43	64.2	22.27	68.81	22.43
L3	**68.85**	22.27	68.81	22.43	68.81	22.43
L4	68.81	22.43	68.81	22.43	68.81	22.43
L5	68.81	22.43	68.81	22.43	68.81	22.43
L6	68.81	22.43	68.81	22.43	68.81	22.43
L7	68.81	22.43	68.81	22.43	68.81	22.43
L8	68.81	22.43	68.81	22.43	68.81	22.43
L9	68.81	22.43	68.81	22.43	68.81	22.43
L10	68.81	22.43	68.81	22.43	68.81	22.43
L11	68.81	22.43	68.81	22.43	68.81	22.43
Base Network Neuron Ablation						
L0	68.81	22.43	-	-	-	-
L1	68.81	22.43	-	-	-	-
L2	68.81	22.43	-	-	-	-
L3	68.81	22.43	-	-	-	-
L4	68.81	22.43	-	-	-	-
L5	68.81	22.43	-	-	-	-
L6	68.81	22.43	-	-	-	-
L7	68.81	22.43	-	-	-	-
L8	68.81	22.43	-	-	-	-
L9	68.81	22.43	-	-	-	-
L10	68.81	22.43	-	-	-	-
L11	68.81	22.43	-	-	-	-

Note: The baseline values are 68.81% for overall accuracy and 22.43% for worst group accuracy.

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)

Table 8. Layer-wise Accuracies (%) for Waterbirds, under the relaxed condition

	Vanilla		Top K=64		Waterbirds K=64	
Layer	Overall	Worst	Overall	Worst	Overall	Worst
SAE Feature Ablation						
L0	68.81	22.43	**69.42**	20.25	68.81	22.43
L1	68.81	22.43	**69.68**	**30.84**	**70.26**	19.16
L2	**68.99**	**24.61**	**71.95**	21.65	**71.94**	**23.68**
L3	**68.86**	**33.96**	68.81	22.43	**70.25**	19.31
L4	**70.19**	**27.73**	**68.83**	**27.26**	68.81	22.43
L5*	**70.07**	***36.6**	**70.76**	**25.55**	68.81	22.43
L6	**70.04**	**33.18**	**69.0**	**24.14**	**71.37**	**30.22**
L7	**70.62**	**28.66**	68.81	22.43	68.81	22.43
L8	**71.3**	**24.92**	68.81	22.43	68.81	22.43
L9	**70.19**	**22.74**	68.81	22.43	68.66	22.43
L10	68.73	22.27	68.81	22.43	68.81	22.43
L11	**69.12**	**23.83**	**69.68**	**22.59**	**70.25**	21.18
Random SAE Feature Ablation						
L0	68.81	22.43	68.81	22.43	68.81	22.43
L1	68.81	22.43	68.81	22.43	68.81	22.43
L2	68.81	22.43	64.2	22.27	68.81	22.43
L3	**68.83**	22.43	68.81	22.43	68.78	22.43
L4	68.8	22.12	68.81	22.43	68.81	22.43
L5	68.62	22.43	**68.85**	22.43	68.81	22.43
L6	68.59	22.27	68.81	22.43	**68.83**	**22.59**
L7	**68.92**	22.43	68.81	22.43	68.81	22.43
L8	68.76	22.43	68.81	22.43	68.81	22.43
L9	68.81	22.43	68.81	22.43	68.81	22.43
L10	68.81	22.43	68.81	22.43	68.81	22.43
L11	**68.83**	**22.59**	68.81	22.43	68.8	22.43
Base Network Neuron Ablation						
L0	68.81	22.43	-	-	-	-
L1	68.81	22.43	-	-	-	-
L2	68.81	22.43	-	-	-	-
L3	68.81	22.43	-	-	-	-
L4	68.81	22.43	-	-	-	-
L5	68.81	22.43	-	-	-	-
L6	68.81	22.43	-	-	-	-
L7	68.81	22.43	-	-	-	-
L8	68.81	22.43	-	-	-	-
L9	68.81	22.43	-	-	-	-
L10	68.81	22.43	-	-	-	-
L11	68.81	22.43	-	-	-	-

Note: The baseline values are 68.81% for overall accuracy and 22.43% for worst group accuracy.

Accepted to the CVPR 2025 Workshop on Mechanistic Interpretability for Vision (MIV)