

Interpretable and Steerable Concept Bottleneck Sparse Autoencoders

Akshay Kulkarni¹ Tsui-Wei Weng¹ Vivek Narayanaswamy²
Shusen Liu² Wesam A. Sakla² Kowshik Thopalli²

¹University of California, San Diego ²Lawrence Livermore National Laboratory

{a2kulkarni, lweng}@ucsd.edu {narayanaswam1, liu42, sakla1, thopalli1}@llnl.gov

Abstract

Sparse autoencoders (SAEs) promise a unified approach for mechanistic interpretability, concept discovery, and model steering in LLMs and LVLMs. However, realizing this potential requires learned features to be both interpretable and steerable. To that end, we introduce two new computationally inexpensive interpretability and steerability metrics for a systematic analysis of LVLM SAEs. This uncovers two observations; (i) a majority of SAE neurons exhibit either low interpretability or low steerability or both, rendering them ineffective for downstream use; and (ii) user-desired concepts are often absent in the SAE, thus limiting their practical utility. To address these limitations, we propose *Concept Bottleneck Sparse Autoencoders (CB-SAE)*¹—a novel post-hoc framework that prunes low-utility neurons and augments the latent space with a lightweight concept bottleneck aligned to a user-defined concept set. The resulting CB-SAE improves interpretability by +32.1% and steerability by +14.5% across LVLMs and image generation tasks.

1. Introduction

Sparse autoencoders (SAEs) [5, 17, 37] have emerged as a foundational tool for mechanistic interpretability, mapping dense polysemantic activations into sparse monosemantic latents in large language models (LLMs) [23], vision models [12, 42, 43, 46], and large vision-language models (LVLMs) [33]. However, realizing this promise requires SAE features to be semantically meaningful and causally effective, *i.e.* *interpretable* and *steerable* respectively.

Recent work on SAEs [1, 48] in the context of LLMs shows that interpretability does not guarantee steering effectiveness, *i.e.* features that activate strongly for a human-understandable concept may fail to control it when intervened upon [1, 16, 17, 49]. Although this trade-off has been observed in language models, its presence and implications in vision encoders and LVLMs remain largely unexplored. To investigate this in the LVLM setting, we conducted an

¹Code: github.com/Trustworthy-ML-Lab/CB-SAE

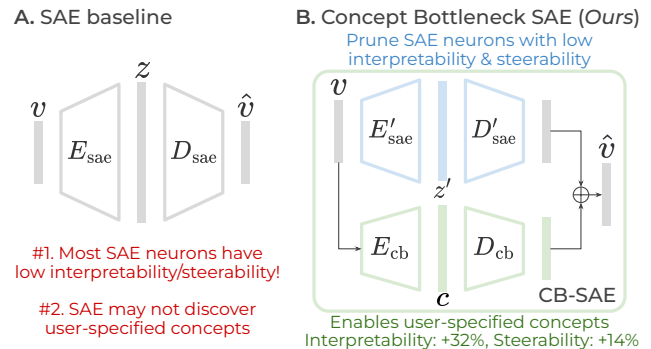


Figure 1. **A.** We find that the majority of SAE neurons in vision models have low interpretability or steerability, with no guarantee of discovering user-specified concepts. **B.** Our CB-SAE addresses both limitations by pruning SAE neurons with low interpretability and steerability, and replacing them with a user-specified concept bottleneck that improves both interpretability and steerability.

empirical study of SAEs trained on activations from the CLIP [36] vision encoder. We introduced metrics to quantify both interpretability and steerability at the neuron level and analyze their alignment across the SAE’s latent space. Our findings reveal that only about 19% of SAE neurons exhibit both high interpretability and steerability. Moreover, despite the SAE’s large dictionary size (65,536 neurons), it fails to represent 27–45% of concepts drawn from established ImageNet-derived benchmarks [44], even when trained on the corresponding data. This highlights the inability of unsupervised SAEs to cover user-specific concepts reliably.

These findings surface two key limitations that constrain the practical utility of SAEs: (i) the inability to ensure comprehensive coverage of semantically meaningful concepts, and (ii) the lack of mechanisms for explicitly encoding user-defined concepts into the latent space to support better steerability. As a result, practitioners are left to work with the latent features the SAE happens to discover and searching post hoc for relevant activations, with no guarantee of alignment with task-specific requirements. This motivates the need for a unified framework that supports both unsupervised discovery and user-guided specification.

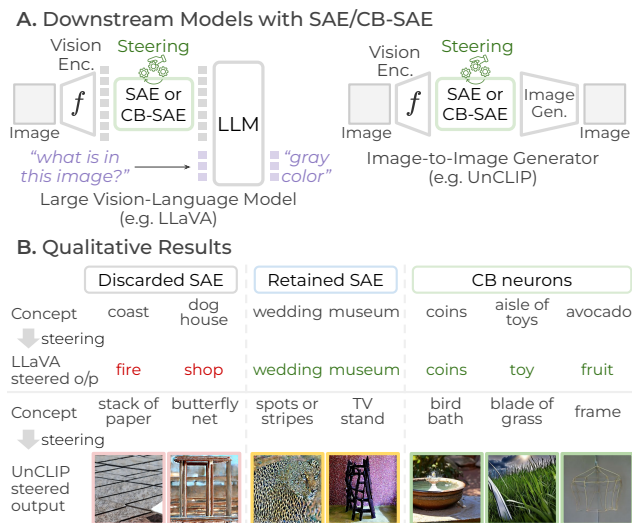


Figure 2. **A.** Our CB-SAE and baseline SAE can steer multiple downstream models like large vision-language models (LLaVA [25]) or image generative models (UnCLIP [40]). **B.** Examples of steering LLaVA and UnCLIP when using unit vector steering (zeroing out all SAE/CB-SAE neurons except the selected concept).

A counterpart to SAEs are Concept Bottleneck Models (CBMs) [19, 21, 31, 41, 45, 54], with concept learning from a supervised perspective. CBMs explicitly train a model to predict a fixed set of human-interpretable concepts by introducing a bottleneck layer that mediates the final prediction. This enables guaranteed concept coverage, but limits CBMs to predefined concepts, preventing discovery of novel features unlike SAEs. These complementary strengths highlight the need for a unified framework that combines CBMs’ controllability with SAEs’ discovery capabilities.

Motivated by these observations, we propose Concept Bottleneck Sparse Autoencoders (CB-SAE) – a unified framework that combines the unsupervised discovery capabilities of SAEs with the controllability of concept bottlenecks. We begin by pruning SAE features that lack interpretability and steerability, and then augment the resulting latent space with a lightweight CB autoencoder [21], trained to align with a user-specified concept set (Fig. 1B). The model is optimized using tailored loss functions that preserve reconstruction fidelity, interpretability, and steerability. As a result, our CB-SAE produces latent features that are both semantically meaningful and causally effective.

We evaluate CB-SAE on two challenging downstream tasks: controlled text generation via vision–language models (LLaVA-1.5-7B [24], LLaVA-MORE [9]) and controlled image synthesis using UnCLIP [40]. CB-SAE consistently outperforms standard SAEs, with average gains of +32.1% in interpretability and +14.5% in steerability across all models and metrics. This improved performance is also shown qualitatively in Fig. 2B, where the retained SAE and CB neurons consistently outperform discarded SAE neurons w.r.t.

steerability. To our knowledge, CB-SAE is the first framework to unify sparse autoencoders with concept bottleneck models, enabling robust interpretation and control of vision representations across modalities and architectures.

2. Related Work

Sparse Autoencoders. SAEs aim to discover interpretable features in neural networks by learning overcomplete decompositions of activations [27]. Recent work [6, 14] showed SAEs can decompose LLM representations into monosemantic features. Various architectural innovations improved SAEs, like Batch-Top- k sparsity [7], JumpReLU [37], and Matryoshka SAEs [8] with multi-level feature hierarchies [10]. Large-scale efforts trained LLM SAEs across multiple layers and models [13, 23], with systematic benchmarks [17]. However, we uncover two key limitations of SAEs: they do not guarantee the discovery of user-desired concepts, and many SAE neurons exhibit low interpretability or utility in downstream steering [1, 12, 16, 17, 49, 51].

Concept Bottleneck Models. CBMs [19, 54] provide a framework for building interpretable models by constraining predictions through a human-understandable concept layer, enabling both interpretation and steering. This approach has been extended to label-free settings [31], enhanced with vision-language guidance [41, 50, 53], applied to image generative models [15, 21] as well as LLMs [45]. DN-CBM [38] used an SAE encoder as a concept bottleneck layer, and trained a linear layer to classify using the SAE concepts. However, their approach is limited to image classification and susceptible to SAE limitations highlighted in our work. A concurrent work, AlignSAE [52], independently devised a similar approach to introduce supervised concepts in SAEs. They attempt to disentangle the supervised concepts from the unsupervised SAE neurons with an orthogonality loss, while our approach explicitly prunes the low utility SAE neurons and only introduces the supervised concepts absent from the retained SAE neurons. Further, AlignSAE focuses on text-based LLM SAEs while we focus on vision SAEs for multimodal LLMs and image-to-image generative models. Our work bridges SAEs and CBMs into our novel CB-SAE, combining the expressiveness of overcomplete feature decomposition with user-specified concepts, steerability, and interpretability of concept-guided learning.

SAEs for Vision and Vision-Language Models. Recent work showed that SAEs can learn interpretable, monosemantic features in vision models [43] as well as vision-language models [33, 55]. SpLiCE [3] interprets CLIP with sparse concept embeddings using supervised reconstruction and a sparsity constraint, but does not support unsupervised concept discovery unlike SAE/CB-SAE. Another line of work [29, 34, 47] investigated how visual information maps to language via SAEs for cross-modal interpretability [26, 28]. However, prior works neither address the challenges of ensur-

ing discovered features are both interpretable and steerable, nor guarantee the discovery of user-specified concepts. Our CB-SAE addresses both limitations through post-hoc pruning and concept bottleneck training.

3. Background

SAE preliminaries. Let $v = f_l(x) \in \mathbb{R}^d$ denote the dense activations from layer l of a deep pre-trained vision model (e.g., CLIP image encoder [36]) f for an input image $x \in \mathbb{X}$. Here d denotes the activation dimension and \mathbb{X} corresponds to the space of images. SAEs decomposes the polysemantic activations v into sparse, overcomplete latent representations $z \in \mathbb{R}^\omega$ ($\frac{\omega}{d} \gg 1$) with the aim of associating every unit in z to distinct, interpretable concepts. Here $\frac{\omega}{d}$ corresponds to the expansion factor of the SAE [13]. Formally, an SAE is parameterized by a linear encoder $E_{\text{sae}} \in \mathbb{R}^{\omega \times d}$, a linear decoder $D_{\text{sae}} \in \mathbb{R}^{d \times \omega}$, a shared bias term $b \in \mathbb{R}^d$, and a non-linear activation function $\sigma_{\text{sae}} : \mathbb{R}^\omega \rightarrow \mathbb{R}^\omega$:

$$z = \sigma_{\text{sae}}(E_{\text{sae}}(v - b)) \quad (1)$$

$$\hat{v} = D_{\text{sae}}z + b \quad (2)$$

The SAE training objective is given by $\mathcal{L}_r = \|v - \hat{v}\|_2^2 + \lambda \|z\|_1$, where $\lambda \geq 0$ balances reconstruction fidelity and sparsity, where \hat{v} represents the SAE reconstruction. In addition to standard ℓ_1 regularization, sparsity can be enforced directly via the activation function $\sigma_{\text{sae}}(\cdot)$, such as top- k [13], batch top- k [7], or ReLU with a learnable threshold [23, 37].

Measuring SAE interpretability. After training, SAEs are typically evaluated using reconstruction fidelity or sparsity [17, 26]. However, these metrics do not directly quantify interpretability which is namely the extent to which individual SAE neurons correspond to human-understandable concepts. While existing work [18, 33] relies on manual inspection of top- k activating inputs or autointerpretability scores [4, 35] that depend on external language models or measuring the monosemanticity [33], they are often subjective, computationally expensive, and difficult to scale. To this end, we leverage a popular neuron interpretability tool CLIP-Dissect [30], which utilizes a user-specified concept set \mathcal{C} and a pretrained vision-language model to assign each neuron j of an SAE to a human-interpretable text concept c_j . Our interpretability score is the similarity score computed in CLIP-Dissect. This approach is computationally inexpensive, scalable, and is used for the first time with SAEs. Neuron interpretability scores have also been useful for downstream tasks like adversarial defense [20]. Please refer to Appendix for further details on CLIP-Dissect.

Measuring SAE steerability. Beyond interpretability, SAEs have been shown to enable controllable manipulation of model behavior across language [1], large-scale vision [43], and large vision-language models [26, 33] such

as LLaVA [25]. Steerability refers to the ability to influence model outputs through targeted modifications of SAE neuron activations, thereby inducing semantically consistent changes [1]. It is typically quantified by measuring the alignment between the steered output and the concept label of the intervened neuron. Since, in our study, the base model f is a vision-only encoder, we employ a downstream generative model to evaluate the effect of SAE latent interventions.

Following [33], we adopt LLaVA [25], which maps an image-text pair (x, t) to a text output o . In LLaVA, the vision encoder f (e.g. CLIP [36]) produces visual tokens $\{v_i\}_{i=1}^N$, which are projected by an adapter into the LLM’s word embedding space, where they are combined with prompt tokens to generate o . Similar to [33], to probe steerability, we use a white image with the prompt “*What is shown in this image? Use exactly one word!*”. For a given target neuron $j \in [\omega]$ of the SAE, we overwrite its activation across all tokens (from this white image) with a fixed value α , reconstruct the modified latents $\{\tilde{v}_i\}$ through the SAE decoder, and feed them into LLaVA to produce the steered output \tilde{o}_j .

We then compute the cosine similarity between \tilde{o}_j and the neuron’s CLIP-Dissect-assigned concept c_j in a sentence-transformer embedding space [39]. Higher similarity indicates greater steerability, as the neuron reliably drives the output toward its associated concept. Unlike [33], which compared steered outputs to top-activating images in CLIP’s image-text space, our method compares \tilde{o}_j with concepts identified by CLIP-Dissect, which produces these descriptions by aggregating activations across all images thus yielding a more robust, semantically grounded steerability metric. Note that while steering (image,text)-to-text LLaVA [25] is one way to compute a steerability metric, a similar metric can be computed using an image-to-image generator like UnCLIP [40] (see Appendix for analysis with UnCLIP).

4. Interpretability vs Steerability in SAEs

In this section, we empirically analyze SAEs from two complementary perspectives: (i) their capacity to capture interpretable concepts and (ii) their ability to steer model outputs and quantify their trade-offs. While these two measures *i.e.*, interpretability and steerability are related, they capture distinct yet complementary aspects of SAE behavior [48]. In practice, interpretable neurons may not be steerable if their activations are weakly causal or entangled, while steerable neurons may encode abstract features misaligned with user objectives [1, 48]. The insights from this analysis on large vision-language models motivate our hybrid framework, introduced in Sec. 5, which integrates SAEs with principles from concept bottleneck models [19, 31, 41].

Expt. 1: Are all SAE neurons interpretable & steerable?

Setup. We train a Matryoshka Batch Top- k SAE [7, 55] with $\bar{\omega} = 65536$ and an expansion factor of 64 on layer $l = 22$ of the CLIP-ViT-L/14-336 vision encoder [36], following

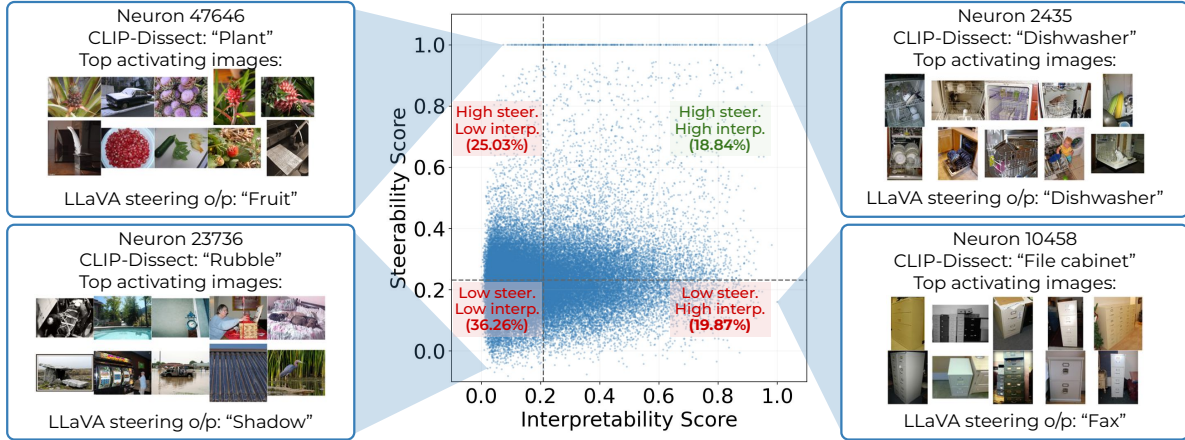


Figure 3. We analyze the interpretability and steerability of 65,536 neurons of an SAE trained for a CLIP image encoder. We also visualize the CLIP-Dissect assigned concept, top-activating images, and LLaVA steering outputs for some characteristic neurons. The dashed lines indicate the average scores along each axis, and we observe that most SAE neurons have either low interpretability, low steerability, or both.

the setup of [33], using the ImageNet-1K dataset [11] (see Appendix for more details and results with other layers and models). As CLIP-Dissect requires a predefined concept set \mathcal{C} , we employ the Broden dataset [2], ($|\mathcal{C}|=1197$) which provides both low-level attributes and object-level visual concepts. We then compute interpretability and steerability scores for each SAE neuron as described in Sec. 3.

Observations. Fig. 3 illustrates the trade-off between interpretability and steerability scores, with dashed horizontal and vertical lines marking their respective mean values to delineate four distinct neuron groups. For each group, the figure shows the top-10 activating images, CLIP-Dissect-assigned concepts (interpretability), and LLaVA-steered outputs (steerability) for a representative neuron, highlighting characteristic behaviors. The distribution of neurons across these groups is as follows:

- *Low interpretability, low steerability* (36.26%, 23,763): These neurons are largely inactive and contribute minimally to either semantic meaning or controllable behavior.
- *High interpretability, low steerability* (19.87%, 13,022): These neurons capture clear, human-understandable concepts but have limited influence on model outputs.
- *Low interpretability, high steerability* (25.03%, 16,403): These neurons effectively steer outputs but correspond to abstract or composite features that lack semantic clarity.
- *High interpretability, high steerability* (18.84%, 12,348): This is the most desirable group of neurons that are both interpretable and causally effective.

These indicate that a vast majority of SAE neurons are not directly useful for downstream tasks such as explanation or control, reinforcing the need for hybrid approaches that jointly enhance interpretability and steerability.

Expt. 2: Can SAEs represent all user-specified concepts?

A key question is whether the SAE can represent all concepts from a given concept set, thereby supporting both human

interpretability and controllable model behavior. Although the SAE contains 65,536 neurons, far exceeding the size of standard concept sets, its ability to represent concepts varies considerably with the diversity and complexity of the set.

Observations. Using CLIP-Dissect, we evaluate the coverage of unique concepts across multiple concept sets:

- Broden [2]: 1,153/1,197 (96.3%)
- VLG-CBM [41]: 3,445/4,729 (72.8%)
- DECIDER [44]: 4,333/7,827 (55.3%)
- 3k common English words [31]: 1,857/3,000 (61.9%)
- 20k common English words [31]: 5,596/20,000 (28.0%)

The SAE performs well on the smaller and well-defined Broden concept set, capturing 96.3% of its visual concepts. However, coverage drops sharply for larger or linguistically diverse sets, with only 28–73% of concepts represented on average. Notably, despite being trained on ImageNet, the SAE fails to capture 27–45% of ImageNet-related concepts from the VLG-CBM and DECIDER sets. These results indicate that, while SAEs effectively capture simple, low-level concepts, their latent spaces struggle to generalize to broader, user-specified concept sets—limiting their utility for downstream interpretability and nuanced steerability.

Our analysis reveals two key requirements: (i) expanding the SAE latent space to capture a broader range of semantically distinct concepts while remaining effective for downstream tasks such as steering, and (ii) enabling explicit user specification of concepts within the SAE. Simply pruning neurons with low interpretability and steerability degrades reconstruction fidelity. To address this and enable user-specified concepts, we propose to train a *concept bottleneck autoencoder* [19, 21] alongside the retained SAE. This hybrid framework combines (supervised) concept alignment with (unsupervised) discovery, restoring reconstruction fidelity while enhancing concept coverage and steerability.

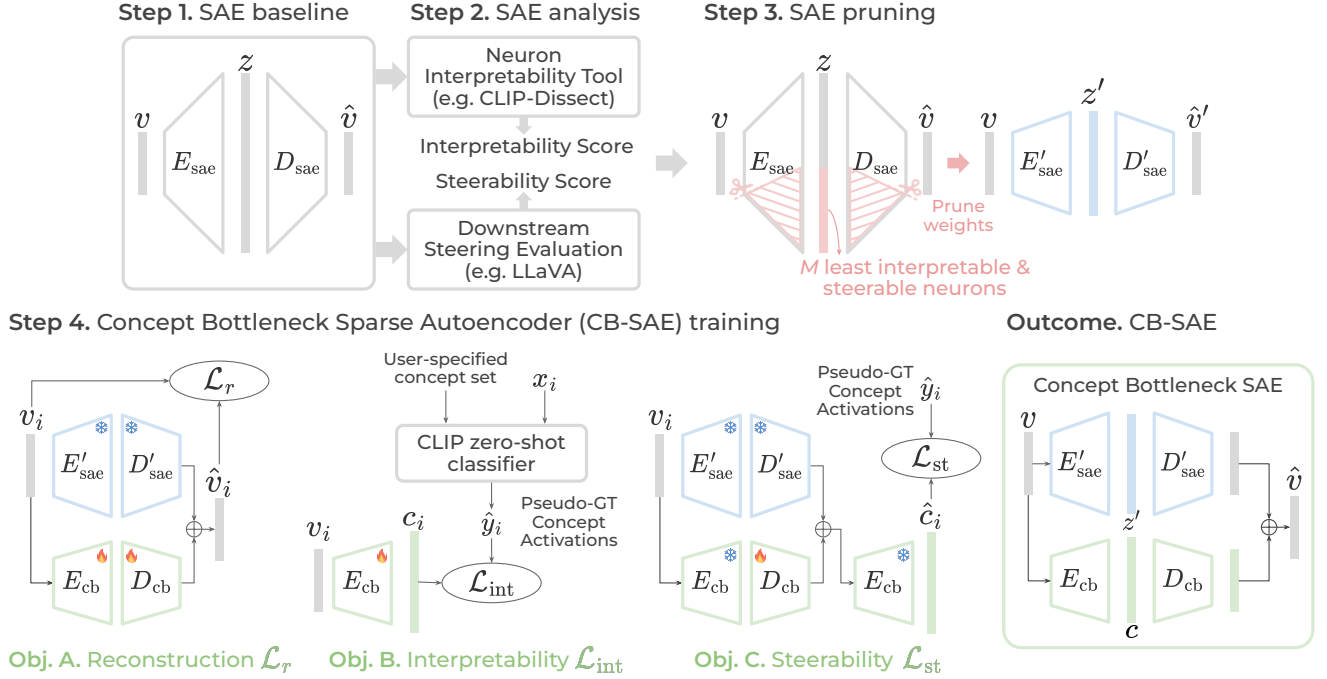


Figure 4. **Pipeline for building CB-SAE.** **Step 1.** A baseline SAE is trained and **Step 2.** evaluated with CLIP-Dissect and downstream steering to obtain interpretability and steerability scores per SAE neuron. **Step 3.** The M least interpretable and steerable neurons are pruned by deleting the corresponding SAE weights. **Step 4.** We train the CB-SAE with frozen, pruned SAE weights with three objectives: **A.** recover the reconstruction ability lost by pruning using \mathcal{L}_r , **B.** incorporate the user-specified concept set with \mathcal{L}_{int} , and **C.** promote steerability with a cyclic reconstruction loss \mathcal{L}_{st} . **Outcome.** CB-SAE uses both retained SAE and CB-AE for inference.

5. Our Approach: CB-SAE

We propose a novel concept bottleneck sparse autoencoder (CB-SAE) based on our analysis to address two limitations of sparse autoencoders namely low interpretability/steerability and the lack of support for user-specified concepts.

5.1. Pruning SAE neurons

Step 1 (Fig. 4) As discussed in Sec. 3, we begin with training an SAE on layer l activations from the vision model f .

Step 2 (Fig. 4) Following our analysis experiments, we compute the interpretability and steerability scores for each sparse neuron in the trained SAE denoted by $I \in [0, 1]^\omega$ and $S \in [0, 1]^\omega$ respectively.

Step 3 (Fig. 4) We prune the SAE weights $E_{\text{sae}}, D_{\text{sae}}$ to remove the M least interpretable and steerable SAE neurons as they are unsuitable for downstream applications. Concretely, the set of M SAE neurons to be pruned is $\mathcal{P} = \{m \mid I_m + S_m < \tau, m \in [\omega]\}$ where τ is the threshold that determines $|\mathcal{P}| = M$ and $[\omega] = \{1, 2, \dots, \omega\}$. In practice, we simply sort the $I + S$ in descending order and select the bottom- M neurons that constitute \mathcal{P} .

SAE consists of $E_{\text{sae}} \in \mathbb{R}^{\omega \times d}$ and $D_{\text{sae}} \in \mathbb{R}^{d \times \omega}$. We can then prune the selected set of neurons \mathcal{P} by deleting the corresponding rows and columns in E_{sae} and D_{sae} respectively.

In other words, the retained SAE weights E'_{sae} and D'_{sae} have all rows and columns other than those in \mathcal{P} respectively:

$$E'_{\text{sae}} = E_{\text{sae}}[[\omega] \setminus \mathcal{P}, :] \quad (3)$$

$$D'_{\text{sae}} = D_{\text{sae}}[:, [\omega] \setminus \mathcal{P}] \quad (4)$$

Here, \setminus denotes the set minus operator. Note, the shared bias term $b \in \mathbb{R}^d$ does not change as it is independent of the number of SAE neurons ω . The retained SAE consists of the retained encoder $E'_{\text{sae}} \in \mathbb{R}^{(\omega-M) \times d}$, retained decoder $D'_{\text{sae}} \in \mathbb{R}^{d \times (\omega-M)}$, and the bias $b \in \mathbb{R}^d$. Using Eq. (1), (2) with $E'_{\text{sae}}, D'_{\text{sae}}$, the retained SAE latent changes to $z' \in \mathbb{R}^{(\omega-M)}$ and the reconstructed latent is $\hat{v}' \in \mathbb{R}^d$ (Fig. 4, Step 3). While the reconstructed \hat{v}' has the same dimensions as v , the average reconstruction loss $\mathbb{E}_v[v - \hat{v}']$ will be higher than without pruning $\mathbb{E}_v[v - \hat{v}]$, due to loss of information to effectively reconstruct the activations. As discussed earlier, to recover this lost reconstruction ability and to incorporate user-specified concepts, we introduce a concept bottleneck.

5.2. Training CB-SAE

Step 4 (Fig. 4) We introduce a concept bottleneck autoencoder [21] alongside the retained SAE (Fig. 1B). Our CB-SAE consists of the retained SAE $E'_{\text{sae}}, D'_{\text{sae}}$, a lin-

ear concept encoder $E_{\text{cb}} \in \mathbb{R}^{|\mathcal{C}| \times d}$, a linear concept decoder $D_{\text{cb}} \in \mathbb{R}^{d \times |\mathcal{C}|}$, and a non-linear activation function $\sigma_{\text{cb}} : \mathbb{R}^{|\mathcal{C}|} \rightarrow \mathbb{R}^{|\mathcal{C}|}$ similar to the SAE, where \mathcal{C} is a pre-defined concept set. For an input $v \in \mathbb{R}^d$, the CB-SAE reconstructs $\hat{v}' \in \mathbb{R}^d$ as,

$$z' = \sigma_{\text{sae}}(E'_{\text{sae}}(v - b)) \quad (5)$$

$$c = E_{\text{cb}}(v - b) \quad (6)$$

$$\hat{v}' = D'_{\text{sae}}z' + b + D_{\text{cb}}\sigma_{\text{cb}}(c) \quad (7)$$

We use a top- k function as σ_{cb} with $k \ll |\mathcal{C}|$ to ensure that sparsity constraints are similar to the original SAE. The bias term $b \in \mathbb{R}^d$ is shared with the retained SAE.

Concept Set Selection. Based on our motivation to support user-specified concepts, the concept set \mathcal{C} can be specified by the user as a list of text-based concepts, similar to prior work [31, 41]. However, as shown in Sec. 4, the SAE can already represent some concepts well and including them in the concept set \mathcal{C} would be redundant. Hence, we only use concepts absent from the retained SAE in the CB-SAE concept set. Let $\mathcal{C}_{\text{user}}$ be the user-specified concept set, $\mathcal{C}_{\text{rsae}} \subset \mathcal{C}_{\text{user}}$ be the concepts present in the retained SAE (found using CLIP-Dissect before pruning the SAE, Fig. 4 Step 2). Then our CB-SAE concept set is given by $\mathcal{C} = \mathcal{C}_{\text{user}} \setminus \mathcal{C}_{\text{rsae}}$.

Training Objectives. We propose three training objectives to guide the CB-SAE. First, the concept bottleneck should recover the reconstruction fidelity lost by pruning the SAE neurons. Second, the CB neurons should be interpretable w.r.t. the concept set \mathcal{C} . And third, the CB neurons should be steerable w.r.t. the concept set \mathcal{C} . The neurons of the retained SAE already meet the reconstruction, interpretability, and steerability objectives for their discovered concepts $\mathcal{C}_{\text{rsae}}$, so we keep the retained SAE weights frozen.

Objective A: Reconstruction \mathcal{L}_r (Fig. 4, Step 4A). Similar to the SAE, we optimize the mean-squared error \mathcal{L}_r between the input latent $v \in \mathbb{R}^d$ and the CB-SAE reconstruction \hat{v}' ,

$$\min_{E_{\text{cb}}, D_{\text{cb}}} [\mathcal{L}_r(v, \hat{v}')] \quad (8)$$

Instead of an ℓ_1 regularizer for sparsity, we use a top- k activation function σ_{cb} as mentioned earlier.

Objective B: Interpretability \mathcal{L}_{int} (Fig. 4, Step 4B). To ensure that each CB neuron in c activates for its corresponding concept in the concept set \mathcal{C} , we use a CLIP zero-shot classifier [36] with \mathcal{C} as the classes and obtain pseudo-ground-truth concept activations similar to prior work in CBMs [21, 31]. This enables concept-label-free training of the CB-SAE, *i.e.* does not require explicit concept labels and also supports any arbitrary user-specified concept sets.

The zero-shot classifier $\mathcal{M} : \mathbb{X} \times \mathbb{T}^{|\mathcal{C}|} \rightarrow \mathbb{R}^{|\mathcal{C}|}$ (\mathbb{T} refers to text space) takes in an image $x \in \mathbb{X}$, list of concept names \mathcal{C} , and predicts concept logits $\hat{y} = \mathcal{M}(x, \mathcal{C}) \in \mathbb{R}^{|\mathcal{C}|}$. We use

a cosine-cubed similarity loss \mathcal{L}_{int} following [31] between the concept encoder’s predictions $c = E_{\text{cb}}(v)$ and \hat{y} ,

$$\min_{E_{\text{cb}}} [\mathcal{L}_{\text{int}}(c, \hat{y})] \quad (9)$$

Here, $v = f_l(x)$ is the vision encoder f output at layer l for the same image x used with \mathcal{M} . Further, note that we use the sparsity constraint of a top- k activation function σ_{cb} only for the decoder in Eq. (7). This is because the concept encoder E_{cb} should be able to interpret all concepts in an image, but the concept decoder D_c can only use the top- k concepts for reconstruction. This is further ensured by updating only E_{cb} with the interpretability objective \mathcal{L}_{int} . We defer more details of the cosine-cubed similarity loss [31] to the Appendix.

Objective C: Steerability \mathcal{L}_{st} (Fig. 4, Step 4C). Prior work on CBMs for generative models [21] leveraged the downstream image generation task to design explicit steerability objectives. In contrast, we propose a simple task-agnostic cyclic reconstruction objective for steerability. With this, we show in Sec. 6 that the same CB-SAE can steer two different downstream tasks: image-to-image generation and image-text-to-text generation.

Concretely, as shown in Fig. 4, Step 4C, we pass the reconstructed latent \hat{v}' back through the concept encoder E_{cb} to produce cyclically reconstructed concepts $\hat{c} = E_{\text{cb}}(\hat{v}' - b)$. Then, we use the same pseudo-ground-truth concept activations \hat{y} computed for Objective B and optimize the same loss as Objective B with \hat{c} (denoted by \mathcal{L}_{st} for clarity),

$$\min_{D_{\text{cb}}} [\mathcal{L}_{\text{st}}(\hat{c}, \hat{y})] \quad (10)$$

Only the concept decoder D_{cb} is updated with the steerability loss. This is because only the decoder D_{cb} is responsible to appropriately modify the latent \hat{v}' when a concept in c is modified for steering. On the other hand, the concept encoder E_{cb} should focus only on interpreting the input v and updating it with \mathcal{L}_{st} could hurt interpretability.

Instead of loss weighting hyperparameters, we train by alternately minimizing the objectives via separate Adam optimizers which adaptively scale weight updates [22].

6. Experiments

We extensively evaluate our proposed CB-SAE w.r.t. interpretability and steerability on two downstream tasks, (image,text)-to-text generation and image-to-image generation. We also performed detailed ablation and sensitivity analysis experiments to validate our design choices.

6.1. Setup

Baseline SAE and CB-SAE. We follow Pach et al. [33] and train a Matryoshka Batch Top- k SAE [55] with expansion factor $\frac{\omega}{4} = 64$ as the baseline SAE on the ImageNet-1k [11]

Table 1. **Interpretability and Steerability Evaluation with LLaVA and UnCLIP.** All four metrics are in 0-1 range (higher is better), CD indicates CLIP-Dissect score and MS indicates monosemanticity score.

Downstream Task	Steered Model	Method	Interpretability		Steerability	
			CD	MS	Unit-Vec	White Image
Image + Text → Text Generation	LLaVA-1.5-7B [25] (CLIP-ViT-L + Vicuna-7B)	SAE [33]	0.154	0.517	0.198	0.203
		CB-SAE (<i>Ours</i>)	0.244	0.556	0.261	0.250
	LLaVA-MORE [9] (DINOv2-L + Gemma2-9B)	SAE [33]	0.194	0.553	0.179	0.177
		CB-SAE (<i>Ours</i>)	0.291	0.598	0.192	0.189
Image → Image Generation	UnCLIP [40] (CLIP-ViT-L + SD-2.1)	SAE [33]	0.058	0.540	0.642	0.654
		CB-SAE (<i>Ours</i>)	0.092	0.594	0.659	0.664

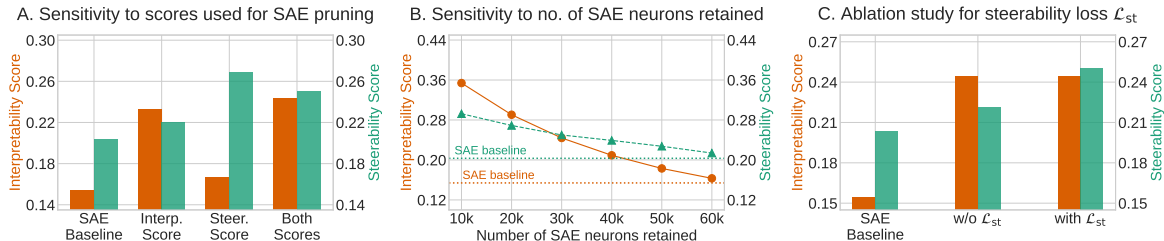


Figure 5. **A.** Sensitivity of CB-SAE to the choice of scores used for SAE pruning. **B.** Sensitivity of CB-SAE to the number of SAE neurons retained. **C.** Ablation study for our proposed steerability objective \mathcal{L}_{st} from Eq. (10).

Table 2. Evaluating interpretability and steerability of discarded SAE neurons, retained SAE neurons, and CB neurons separately.

Set of Neurons	Interpretability	Steerability	
	CLIP-Dissect	Unit-Vec	White Image
All SAE neurons	0.154	0.198	0.203
Discarded SAE neurons	0.084	0.144	0.162
Retained SAE neurons	0.238	0.263	0.252
CB neurons	0.323	0.231	0.219
All CB-SAE neurons	0.244	0.261	0.250

dataset. Our CB-SAE is also trained on the same intermediate activations as the baseline SAE for a fair comparison. We retain $\omega - M = 30k$ neurons in the SAE pruning and use a top- k function as σ_{cb} with $k = 5$ in our CB-SAE. We use the VLG-CBM ImageNet concept set [31, 41] for the CB neurons. In our training, we use a CLIP-ViT-B/16 [36] model for obtaining the pseudo-ground-truth concept activations.

Evaluation Metrics. To evaluate interpretability, we use the CLIP-Dissect interpretability score introduced in Sec. 3 and the monosemanticity score from [33] using the ImageNet validation set. To ensure a fair evaluation, we use a stronger CLIP-ViT-L/14 model (w.r.t. smaller ViT-B/16 used for training CB-SAE). To evaluate steerability, we use our proposed steerability score (Sec. 3). Concretely, we evaluate the steerability of each CB/SAE neuron in two ways:

- Unit Vector: The selected neuron is activated to a high value $\alpha = 50$ (as in [33]) & all other neurons are set to 0.
- White Image: The selected neuron is activated to a high

value $\alpha = 50$ and all other neurons have the values predicted when using an empty white image (following [33]) as input, instead of 0 like in unit vector steering.

The interpretability and steerability scores of individual neurons are averaged to obtain the overall scores. For experiments where the steered output is text, we compare the similarity between the steered text and the CLIP-Dissect assigned concept for the selected neuron in a sentence transformer embedding space (as in Sec. 3). For experiments where steered output is an image, we compute the average similarity between the steered image and top-16 highly activating images for the selected neuron in the DINOv2 [32] embedding space. This is because the diffusion model being steered (UnCLIP [40]) may rarely return partially or completely noisy images after steering, which cannot be properly evaluated with an image-text similarity score (e.g. CLIP) that expects clean images. All metrics are normalized in 0-1 range and higher values indicate better performance.

6.2. Quantitative Comparison

In Table 1, we compare our CB-SAE with the baseline SAE [33] across several downstream models as well as tasks. For two different variants of the (image,text)-to-text LLaVA [9, 25] model, our CB-SAE demonstrates consistent gains over the SAE baseline across both interpretability (avg. +33.0% for CLIP+Vicuna and avg. +29.0% for DINOv2+Gemma) and steerability metrics (avg. +27.5% for CLIP+Vicuna and avg. +14.0% for DINOv2+Gemma). Similarly, for the image-to-image UnCLIP [40] generative model,

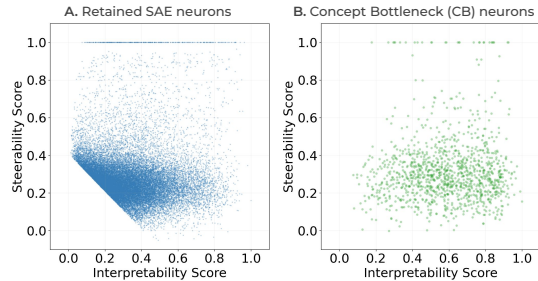


Figure 6. Visualizing the interpretability and steerability of retained SAE neurons and CB neurons, similar to Fig. 3.

our CB-SAE outperforms the SAE baseline (avg. **+34.3%** interpretability and avg. **+2.1%** steerability). To the best of our knowledge, we are the first to show that an SAE (and CB-SAE) trained with the same method can be used to steer different downstream tasks.

6.3. Analysis of our CB-SAE

Effect of CB neurons. In Table 2, we separately evaluate the interpretability and steerability of discarded and retained SAE neurons, as well as CB neurons. We observe CB neurons have significantly higher interpretability than SAE neurons. Whereas, CB neurons’ steerability is worse than retained SAE neurons but significantly better than discarded SAE neurons as well as all SAE neurons (discarded+retained). Intuitively, the retained SAE neurons contain many highly steerable neurons because steerability (and interpretability) were used to prune SAE neurons.

Sensitivity to scores used for SAE pruning. In Fig. 5A, we compare the CB-SAE performance while varying the choice of metrics for SAE pruning: either interpretability score or steerability score or both. We find that prioritizing either score leads to some loss in performance on the other, while using both scores gives balanced performance. This is beneficial as users can choose the score or even design a new score for pruning based on their target downstream usecase.

Sensitivity to no. of SAE neurons retained. In Fig. 5B, we evaluate CB-SAE models trained with varying number of SAE neurons retained after pruning using both interpretability and steerability scores. We find that lower number of SAE neurons retained leads to higher interpretability and steerability scores. This is because pruning keeps a smaller subset of SAE neurons with higher scores. However, further reducing the number of SAE neurons would hurt performance as reconstruction becomes more difficult.

Ablation study for steerability loss \mathcal{L}_{st} . In Fig. 5C, we analyze the impact of our proposed steerability loss \mathcal{L}_{st} on the CB-SAE. We observe similar interpretability with and without \mathcal{L}_{st} , which is significantly better than the SAE baseline. And using \mathcal{L}_{st} improves steerability by 2.9%, validating its usefulness in CB-SAE training.

Visualizing retained SAE and CB neurons. We visualize the distribution of CB neurons and retained SAE neurons

		Discarded SAE		Retained SAE		CB neurons		
Unit Vector	Neuron	#35447	#54185	#2610	#3008	#30506	#31437	#31018
	Concept	chocolate syrup	race	bathroom	bread	face	rubber	mansion
	LLaVA steered o/p	cookie	truth	bathroom	bread	face	rubber	mansion
White Image	Neuron	#40393	#26259	#1013	#3459	#31068	#30624	#30192
	Concept	stone circle	floral design	office	lake	mess	glass surface	building
	LLaVA steered o/p	circle	flower	office	lake	mess	glass	building
Unit Vector	Neuron	#58036	#58239	#4048	#27760	#31714	#30714	#31209
	Concept	colorful exterior	prey animal	sky	toiletries	stamps	heel	paper clip
	LLaVA steered o/p	interior	fish	sky	strawberry	stamp	foot	pole
Unit Vector	Neuron	#10542	#44896	#6660	#8278	#29561	#29318	#30333
	Concept	made of pores	fish bowl	kalahari desert	leather/fabric	cross	blunt edge	windows
	UnCLIP steered output							
Unit Vector	Neuron	#53739	#51681	#56619	#10722	#30216	#30893	#29712
	Concept	slot for mail	short tufted tail	round metal dial	speedo meter	staircase	super hero	fire
	UnCLIP steered output							

Figure 7. Qualitative examples of steering UnCLIP and LLaVA. Green indicates successful steering, yellow indicates partial success, and red indicates failure cases. See Appendix for more results.

w.r.t. interpretability and steerability scores in Fig. 6. Compared to Fig. 3, the retained SAE neurons do not include low interpretability/steerability neurons as per our SAE pruning (Fig. 4, Step 2). Whereas the CB neurons have higher interpretability and similar steerability scores to the retained SAE neurons. Hence, there is potential to further improve steerability by designing better or more task-specific losses.

Qualitative examples of steering. We report qualitative examples of steering LLaVA and UnCLIP with discarded SAE neurons, retained SAE neurons, and CB neurons in Fig. 7. We find discarded SAE neurons to be worse in steering than retained SAE neurons, which is expected as pruning uses the steerability score. Specifically, in the image-to-image generator UnCLIP [40], we find steering the CB neurons produce higher quality images compared to the SAE neurons which often produce noisy images, likely due to the explicit concept supervision in CB-SAE. We also highlight some **failure cases** and **partially correct** steering for all neurons.

7. Conclusion

In this work, we made the first attempt to unify two complementary paradigms - SAEs for unsupervised concept discovery and CBM for interpretable concept control - into a single unified framework, CB-SAE. Motivated by insights derived from our comprehensive analysis of SAEs in LVLMs, we first pruned the low-utility neurons in the SAE. We then introduced a light-weight CB module trained alongside the frozen, retained SAE using three principled objectives. Across vision-language (LLaVA) and image generation (UnCLIP) tasks, CB-SAE consistently improves interpretability and steerability, while enabling user-specified concept control.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344, Lawrence Livermore National Security, LLC. and was supported by the LLNL-LDRD Program under Project No. 25-SI-001 and DOE ECRP 51917/SCW1885. LLNL-CONF-2013863. A. Kulkarni and T-W. Weng are also partially supported by National Science Foundation under Grant No. 2313105, 2430539, Hellman Fellowship, ARL Award, and Intel Rising Star Faculty Award.

References

- [1] Dana Arad, Aaron Mueller, and Yonatan Belinkov. SAEs are good for steering – if you select the right features. In *EMNLP*, 2025. 1, 2, 3
- [2] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, 2017. 4
- [3] Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P Calmon, and Himabindu Lakkaraju. Interpreting CLIP with sparse linear concept embeddings (SpLiCE). In *NeurIPS*, 2024. 2
- [4] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models, 2023. 3
- [5] Joseph Bloom, Curt Tigges, Anthony Duong, and David Chanin. SAELens: An open-source library for training and analyzing sparse autoencoders, 2024. 1
- [6] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermy, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. 2
- [7] Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK sparse autoencoders. In *NeurIPS*, 2024. 2, 3
- [8] Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. In *ICML*, 2025. 2
- [9] Federico Cocchi, Nicholas Moratelli, Davide Caffagni, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. LLaVA-MORE: A Comparative Study of LLMs and Visual Backbones for Enhanced Visual Instruction Tuning. In *ICCV*, 2025. 2, 7
- [10] Valérie Costa, Thomas Fel, Ekdeep Singh Lubana, Bahareh Tolooshams, and Demba E. Ba. From flat to hierarchical: Extracting sparse representations with matching pursuit. In *NeurIPS*, 2025. 2
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 4, 6
- [12] Thomas Fel, Ekdeep Singh Lubana, Jacob S. Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, Binxu Wang, Martin Wattenberg, Demba E. Ba, and Talia Konkle. Archetypal SAE: Adaptive and stable dictionary learning for concept extraction in large vision models. In *ICML*, 2025. 1, 2
- [13] Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *ICLR*, 2025. 2, 3
- [14] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *ICLR*, 2024. 2
- [15] Aya Abdelsalam Ismail, Julius Adebayo, Hector Corrada Bravo, Stephen Ra, and Kyunghyun Cho. Concept bottleneck generative models. In *ICLR*, 2024. 2
- [16] Subhash Kantamneni, Joshua Engels, Senthooan Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? a case study in sparse probing. In *ICML*, 2025. 1, 2
- [17] Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBenchmark: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *ICML*, 2025. 1, 2, 3
- [18] Dahye Kim, Xavier Thomas, and Deepti Ghadiyaram. Revelio: Interpreting and leveraging semantic information in diffusion models. In *ICCV*, 2025. 3
- [19] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *ICML*, 2020. 2, 3, 4
- [20] Akshay Kulkarni and Tsui-Wei Weng. Interpretability-guided test-time adversarial defense. In *ECCV*, 2024. 3
- [21] Akshay Kulkarni, Ge Yan, Chung-En Sun, Tuomas Oikarinen, and Tsui-Wei Weng. Interpretable generative models through post-hoc concept bottlenecks. In *CVPR*, 2025. 2, 4, 5, 6
- [22] Jogendra Nath Kundu, Maharshi Gor, and R Venkatesh Babu. BiHMP-GAN: Bidirectional 3D human motion prediction GAN. In *AAAI*, 2019. 6
- [23] Tom Lieberum, Senthooan Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma Scope: Open sparse autoencoders everywhere all at once on Gemma 2. In *BlackboxNLP Workshop*, 2024. 1, 2, 3
- [24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *NeurIPS*, 2023. 2
- [25] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 2, 3, 7
- [26] Hantao Lou, Changye Li, Jiaming Ji, and Yaodong Yang. SAE-V: Interpreting multimodal models for enhanced alignment. In *ICML*, 2025. 2, 3
- [27] Alireza Makhzani and Brendan Frey. k-Sparse autoencoders. In *ICLR*, 2014. 2

- [28] Ali Nasiri-Sarvi, Hassan Rivaz, and Mahdi S Hosseini. SPARC: Concept-aligned sparse autoencoders for cross-model and cross-modal interpretability. *TMLR*, 2026. 2
- [29] Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. Towards interpreting visual information processing in vision-language models. In *ICLR*, 2025. 2
- [30] Tuomas Oikarinen and Tsui-Wei Weng. CLIP-Dissect: Automatic description of neuron representations in deep vision networks. In *ICLR*, 2023. 3
- [31] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *ICLR*, 2023. 2, 3, 4, 6, 7
- [32] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 7
- [33] Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. In *NeurIPS*, 2025. 1, 2, 3, 4, 6, 7
- [34] Isabel Papadimitriou, Huangyuan Su, Thomas Fel, Sham Kakade, and Stephanie Gil. Interpreting the linear structure of vision-language model embedding spaces. In *COLM*, 2025. 2
- [35] Gonçalo Santos Paulo, Alex Troy Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In *ICML*, 2025. 3
- [36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 3, 6, 7
- [37] Senthoooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with JumpReLU sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024. 1, 2, 3
- [38] Sukrut Rao, Sweta Mahajan, Moritz Böhle, and Bernt Schiele. Discover-then-Name: Task-agnostic concept bottlenecks via automated concept discovery. In *ECCV*, 2024. 2
- [39] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019. 3
- [40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3, 7, 8
- [41] Divyansh Srivastava, Ge Yan, and Lily Weng. VLG-CBM: Training concept bottleneck models with vision-language guidance. In *NeurIPS*, 2024. 2, 3, 4, 6, 7
- [42] Samuel Stevens, Jacob Beattie, Tanya Berger-Wolf, and Yu Su. Towards open-ended visual scientific discovery with sparse autoencoders. *arXiv preprint arXiv:2511.17735*, 2025. 1
- [43] Samuel Stevens, Wei-Lun Chao, Tanya Berger-Wolf, and Yu Su. Interpretable and testable vision features via sparse autoencoders. *arXiv preprint arXiv:2502.06755*, 2025. 1, 2, 3
- [44] Rakshith Subramanyam, Kowshik Thopalli, Vivek Narayanaswamy, and Jayaraman J Thiagarajan. Decider: Leveraging foundation model priors for improved model failure detection and explanation. In *ECCV*, 2024. 1, 4
- [45] Chung-En Sun, Tuomas Oikarinen, Berk Ustun, and Tsui-Wei Weng. Concept bottleneck large language models. In *ICLR*, 2025. 2
- [46] Harrish Thasarathan, Julian Forsyth, Thomas Fel, Matthew Kowal, and Konstantinos G Derpanis. Universal sparse autoencoders: Interpretable cross-model concept alignment. In *ICML*, 2025. 1
- [47] Constantin Venhoff, Ashkan Khakzar, Sonia Joseph, Philip Torr, and Neel Nanda. How Visual Representations Map to Language Feature Space in Multimodal LLMs. In *CVPRW*, 2025. 2
- [48] Xu Wang, Yan Hu, Benyou Wang, and Difan Zou. Does higher interpretability imply better utility? a pairwise analysis on sparse autoencoders. In *NeurIPS*, 2025. 1, 3
- [49] Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. AxBench: Steering LLMs? Even simple baselines outperform sparse autoencoders. In *ICML*, 2025. 1, 2
- [50] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition. In *ICCV*, 2023. 2
- [51] Xinyuan Yan, Shusen Liu, Kowshik Thopalli, and Bei Wang Phillips. Visual exploration of feature relationships in sparse autoencoders with curated concepts. In *NeurIPS*, 2025. 2
- [52] Minglai Yang, Xinyu Guo, Mihai Surdeanu, and Liangming Pan. AlignSAE: Concept-aligned sparse autoencoders. *arXiv preprint arXiv:2512.02004*, 2025. 2
- [53] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *CVPR*, 2023. 2
- [54] Mert Yuksekogul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR*, 2023. 2
- [55] Vladimir Zaiğrajew, Hubert Baniecki, and Przemysław Biecek. Interpreting CLIP with hierarchical sparse autoencoders. In *ICML*, 2025. 2, 3, 6