Improving Sparse Autoencoder with Dynamic Attention

Dongsheng Wang, Jinsen Zhang, Dawei Su, Hui Huang*
College of Computer Science and Software Engineering, Shenzhen University, China
{dongshengwang,2400101100,2023110003}@szu.edu.cn, hhzhiyan@gmail.com

Abstract

Recently, sparse autoencoders (SAEs) have emerged as a promising technique for interpreting activations in foundation models by disentangling features into a sparse set of concepts. However, identifying the optimal level of sparsity for each neuron remains challenging in practice: excessive sparsity might lead to poor reconstruction, whereas insufficient sparsity harms interpretability. While existing activation functions such as ReLU and TopK provide certain sparsity guarantees, they typically require additional sparsity regularization or cherry-picked hyperparameters. We show in this paper that adaptive sparse attention mechanisms using sparsemax can bridge this trade-off, due to their ability to determine the number of concepts in a data-dependent manner. Specifically, we first explore a new class of SAEs based on the cross-attention architecture with the latent features as queries and the learnable dictionary as the key and value matrices. To encourage sparse pattern learning, we employ a sparsemax-based attention strategy that automatically infers a sparse set of concepts according to the complexity of each neuron, resulting in a more flexible and efficient activation function. Through comprehensive evaluation and visualization, we show that our approach successfully achieves lower reconstruction loss while producing high-quality concepts. Moreover, the sparsity level automatically determined by our approach can serve as tuning guidance to improve existing SAEs. The code is available https://github.com/qyj-bkjx/Sparsemax-SAE.

1. Introduction

The impressive gains in reasoning and accuracy of recent large-scale machine learning models like CLIP [6, 55] and GPT [1, 54] have generally come at the cost of a loss of transparency into their functioning. Typically, neurons in these models are polysemantic, and they respond to seemingly unrelated inputs simultaneously. This can be explained as superposition [17], where models learn more independent features than they have neurons by viewing each feature as a linear combination of neurons. Fortunately, sparse autoencoders (SAEs) [19, 27, 28, 32, 45] have emerged as a promising technique for addressing this fundamental challenge by learning an overcomplete yet sparse representation of neural activations, effectively disentangling these superimposed features into more interpretable concepts [20, 60, 63, 66].

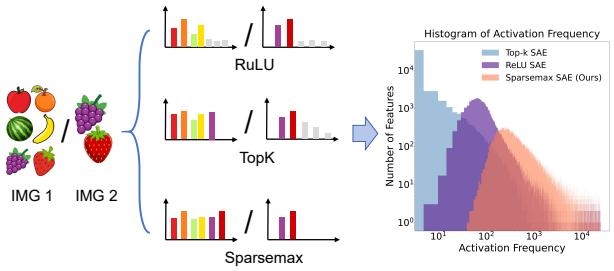

Figure 1. Comparisons of our proposed Sparsemax SAE with previous SAEs (left) and histogram of activation frequency of the learned concepts on the ImageNet dataset (right). ReLU-based SAEs often suffer from the feature shrinkage issue, while TopK-based models tend to produce dead concepts as they only keep K largest concepts. In contrast, our Sparsemax-based SAE dynamically select the number of concepts based on feature complexity, thereby discovering more concepts.

Despite its successes in reversing the effects of superposition, determining the optimal sparse level for the latent features remains an open problem. For example, assigning too many concepts to each feature may compromise interpretability, while insufficient concepts can degrade reconstruction—both scenarios lead to suboptimal concept learning. Early SAEs [28, 34] adopt the ReLU as the activation function and combine it with $\mathcal{L}_1$ regularization to balance sparsity and reconstruction. However, the $\mathcal{L}_1$ penalty often leads to feature shrinkage, where all activations tend toward zero (Seen as Fig. 1). GatedReLU [57] suggests decomposing ReLU into direction selection and magnitude estimation functions via the gate mechanism. JumpReLU [58] finds that zeroing out activations below a positive threshold is a better option. However, these models require additional regularizations to prompt the sparsity, and the balancing coefficient needs to be carefully selected to achieve satisfactory

*Corresponding author.

performance [27].

On the other hand, recent attempts propose to limit the number of concepts explicitly. For example, TopK SAEs [20] employ K-sparse autoencoder [37] to directly choose the K largest concepts and zero the rest. This approach eliminates the need for an explicit sparsity penalty but imposes a rigid constraint on the number of active concepts per sample. BatchTopK SAEs [5] relax the top-K constraint to a batch-level constraint, enabling the SAEs to represent each sample within a batch with a variable number of concepts, resulting in more flexible and efficient utilization of the concept dictionary. Unfortunately, both TopK and BatchTopK view K as a hyperparameter, and how to set K properly remains an unsolved problem.

In this paper, we aim to improve SAEs with adaptive sparse attention mechanisms under the cross-attention framework. Moving beyond the traditional single-layer MLP-style encode-decode structure, we here explore a new class of SAEs based on the transformer architecture due to its successes in various tasks [49, 53, 65, 68]. Specifically, we first view the to-be-learned dictionary as a set of concept vectors, which will be used as the key and value matrices via the corresponding projections. For each latent feature in the neural networks, we view it as the query and apply the cross-attention operation to obtain the reconstructed feature. Intuitively, the calculation of attention weights can be viewed as the encoding stage of SAEs, whose output measures the relevance score between the query and concepts. Notably, this transformer-based SAE connects the encoding and decoding stages by sharing the same concept vectors, rather than viewing them as two independent MLPs, achieving coherent, high-quality concept learning.

With the designed transformer-based architecture, it is natural to replace the softmax with any well-studied sparse attention operations [8, 14, 24, 50]. Here, we adopt sparsemax [39] as our solution. On one hand, sparsemax is differentiable everywhere, and can be easily applied with gradient-based optimization. On the other hand, sparsemax has the ability to assign exactly zero probability to some of its outputs, sharing the same motivation as SAEs. Most importantly, unlike previous works that require hyperparameter K to constrain the number of activations, sparsemax dynamically estimates a threshold function for each sample according to its complexity. This enables the model to output the most relevant concepts while truncating others to zero. As shown in Fig 1, TopK-based SAEs sometimes fail to correctly assign the number of concepts due to the incorrect K. In contrast, our sparsemax successfully assigns six concepts to complex images and two concepts to simple images. Our sparsemax can be viewed as a more precise version of BatchTopK, where we set K at the sample level rather than the batch level, thereby achieving greater flexibility and accuracy in sparse estimation.

To sum up, our contributions are as follows:

- We formalize a novel transformer-based SAEs under the cross-attention framework, which bridges the gap between the encoder and decoder of SAEs by sharing the same concept vectors, resulting in more coherent concept learning.
- A novel sparsemax function is developed to replace the original softmax function in the cross-attention operation. Sparsemax can determine the number of activations dynamically for each sample, without any regularization or hard TopK truncation.
- We provide extensive validation across image and text tasks, demonstrating that our approach not only captures coherent concepts but also achieves superior reconstruction results.

2. Related Work

Mechanistic interpretability with SAEs. Mechanistic interpretability aims to uncover and explain the black box characteristics, enabling models to understand input data and generate reasonable responses [56]. Recently, Sparse Autoencoder (SAEs) have been applied to language models due to their inherent ability to generate interpretable latent concepts [28, 59]. Building upon the standard architecture with ReLU activation [4], a series of studies have developed numerous improvements to the original design. For example, Gated SAE [57], Switch SAE [43] aim to design complex encoders to ensure sparse outputs; Focusing on the feature shrinkage issue [4], JumpReLU [58] introduces threshold parameters for each concept to truncate the concepts with small scores. Topk-based SAEs [5, 20] directly keep K concepts with large scores and zero out the others; In terms of the sparsity regularization, P-annealing [30] and mutual feature regularization (MFR) [38] are designed as alternatives to L_0 and L_1 loss.

Inspired by the successful application of SAE in LLMs, PatchSAE and its variants explore to train SAEs on top of the CLIP and DiNOv2 [34, 46, 52, 62], showing great potential in interpreting visual concepts. Additionally, there has also been interest in steering the generation process of diffusion models via SAEs [11, 20]. More recently, several studies have increasingly applied SAEs to multimodal LLMs [47, 71], and show that SAE can learn shared concepts across the vision and text modalities. However these model primarily extend existing SAEs (such as ReLU and TopK) to the vision domain, without designing new SAE architectures.

Sparse Attention Mechanisms. Traditional attention mechanisms commonly employ the softmax transformation to convert scores into probability distributions [65]. However, softmax produces dense distributions, assigning non-zero attention weights to all elements, which limits interpretability and efficiency [33]. SlidingWindow is a commonly used strategy that allows the query to compute at-

tention only within a fixed windlow [2, 7, 69]. However, these models rely on pre-defined sparse patterns, limiting their potential for application in SAEs. Top-K attention [25] shares similar ideas with TopK SAEs and selects a set of keys with the K largest scores. SeerAttention [21] separates queries and keys into spatial blocks and performs blockwise selection for sparsity. α-entmax attention [51] provides natural, input-dependent sparsity patterns with an exact and differentiable transformation, attracting increasing attention in recent research. For example, Adaptively sparse transformers [10] employ α-entmax attention where attention heads can learn α dynamically. SparseFinder [64] aims to address the efficiency issues of α-entmax by predicting a prior. Sparse Flash Attention [12, 23] combines the efficiency of GPU-optimized algorithms with the sparsity benefits of α-entmax, showing great runtime and memory efficiency. Sparsemax [40] can be viewed as a special case of α-entmax with $\alpha = 2$. It maps inputs onto the probability simplex while allowing exact zeros in the output. Moreover, sparsemax yields piecewise-linear activations and a well-defined Jacobian, enabling efficient gradient computation. It has been successfully applied in multi-label classification and attention-based networks, producing more selective and interpretable attention maps without sacrificing differentiability [29, 40, 42].

In this paper, we aim to introduce a new transformer-based SAEs based on the sparsemax attention. This approach enables SAE to be trained solely with the reconstruction loss, eliminating the need for additional penalty regularization or hyperparameter tuning. The proposed SAEs can also be easily applied to both visual and textual domains.

3. Methodology

In this section, we first briefly review SAEs, then introduce our proposed model in detail.

3.1. SAE and its variants

To disentangle the polysemantic activations (or features) $\boldsymbol{x} \in \mathbb{R}^d$ into a set of monosemantic and interpretable concepts $\mathcal{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_M\} \in \mathbb{R}^{d\times M}$, where $M \gg d$ represents the concept space dimension, SAEs typically represent $\boldsymbol{x}$ as a sparse linear combination of these concepts under the encoder-decoder framework:

$$\begin{aligned} \boldsymbol{z} &= \sigma(\mathbf{W}_{\text{enc}}(\boldsymbol{x} - \boldsymbol{b}_{\text{enc}})), \\ \hat{\boldsymbol{x}} &= \mathbf{W}_{\text{dec}}\boldsymbol{z} + \boldsymbol{b}_{\text{dec}}, \end{aligned} \tag{1}$$

where $\mathbf{W}_{\text{enc}} \in \mathbb{R}^{M\times d}$ and $\mathbf{W}_{\text{dec}} \in \mathbb{R}^{d\times M}$ denote the weight matrices of the single-layer encoder and decoder, respectively. $\boldsymbol{b}_{\text{enc}}, \boldsymbol{b}_{\text{dec}} \in \mathbb{R}^d$ are two bias terms. The columns of $\mathbf{W}_{\text{dec}}$ are the to-be-learned concepts $\mathcal{C}$, and the reconstruction $\hat{x}$ is obtained by weighting these concepts via $\boldsymbol{z}$. To prompt the sparse combination, various structures of σ have been developed in previous studies:

ReLU-based σ. Early SAEs often employ the ReLU activation function to generate sparse weights due to its simplicity of implementation. To address the feature shrinkage issue in ReLU, where activations in $\boldsymbol{z}$ tend toward zero, GatedReLU and JumpRelu are proposed. Although effective in learning sparse $\boldsymbol{z}$, these models require additional sparse regularizations, for example:

$$\mathcal{L} = ||\boldsymbol{x} - \hat{x}||_2^2 + \lambda S(\boldsymbol{z}), \tag{2}$$

where S denotes the function penalizing non-sparse decompositions, *e.g.*, $\mathcal{L}_1$ in ReLU and GatedReLU and $\mathcal{L}_0$ in JumpReLU. λ sets the trade-off between sparsity and reconstruction, requiring careful tuning to achieve a balance between the two.

TopK-based σ. Gao et al. [20] suggests that the TopK is another option to learn sparse $\boldsymbol{z}$, where only the K largest concepts are kept for each sample, with all others set to zero. Due to its explicit sparsity selection, TopK SAEs can be trained via the reconstruction loss. Built upon TopK, BatchTopK is further developed to replace the sample-level TopK operation with a batch-level BatchTopK function, where the top $n \times K$ activations across the entire batch of n samples are selected, while all others are set to zero. Compared to ReLU-based SAEs, TopK-based SAEs empirically show a better balance between the sparsity and reconstruction. However, how to choose the optimal K in these models remains an open problem.

In this paper, we aim to further relax the constraints of BatchTopK based on the sparsemax function, which dynamically determines the number of activations according to the feature's complexity, rather than setting a fixed K.

3.2. Sparsemax SAE

As illustrated in Fig. 2, we improve SAEs in two aspects: 1) a transformer-based architecture is designed to connect the encoder and decoder by sharing the same concept vectors; and 2) within the transformer-based framework, each sample has the ability to determine its sparse level dynamically via sparsemax attention. Below, let us introduce each of them in detail.

Transformer-based SAE. Inspired by the great successes of transformer-based structures in various fields [49, 53, 65, 67], we aim to explore a novel SAE with the cross-attention mechanism. Mathematically , we rewrite Eq. 1 as:

$$\begin{aligned} \mathbf{Q} = \boldsymbol{x}^T\mathbf{W}_Q, \quad \mathbf{K} = \mathcal{C}^T\mathbf{W}_K, \quad \mathbf{V} = \mathcal{C}^T\mathbf{W}_V, \\ \hat{x} = \sigma(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}, \end{aligned} \tag{3}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d\times d}$ denotes the query, key, and value projections, respectively. σ is the sparsemax function, which will be introduced later. Intuitively, Eq. 3 views the

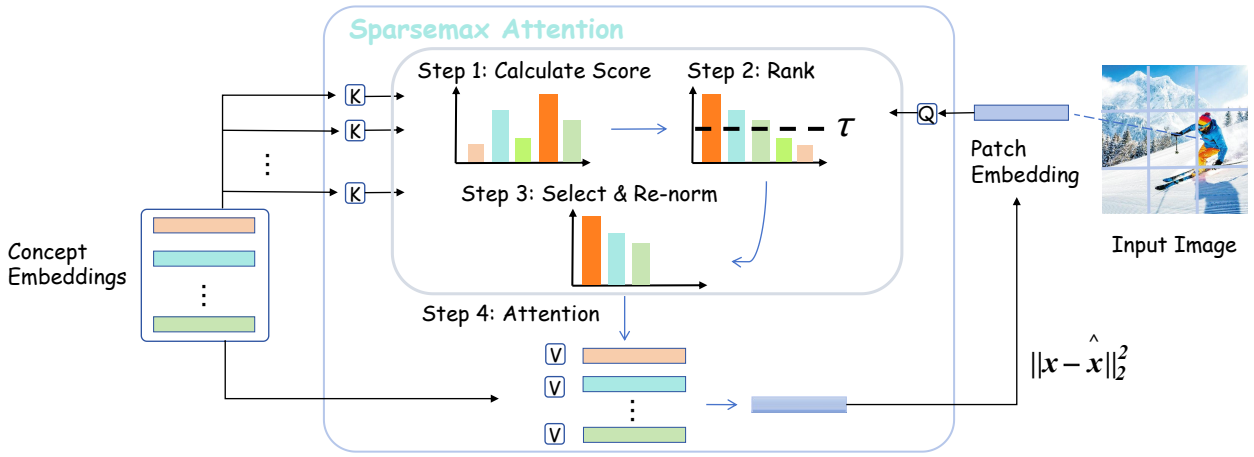

Figure 2. Framework of our Saprsemax SAE, which reconstructs the input feature under the transformer architectures, and the sparsemax attention is employed to dynamically determine the number of concepts by estimating the threshold τ.

input feature as a query and reconstructs it using a set of concepts via the cross-attention framework. Compared to MLP-based SAEs in Eq. 1 that directly output $\boldsymbol{z}$ via $\mathbf{W}_{enc}$, our approach models the activation weights as the similarity score of the query and concepts explicitly, thereby enabling more precise weight estimation. For example, a higher $\boldsymbol{z}$ denotes a closer distance between the query and concepts in the embedding space. More importantly, unlike Eq. 1 views $\mathbf{W}_{\text{enc}}$ and $\mathbf{W}_{\text{dec}}$ as two independent learnable projections, both the key and value in Eq. 3 originate from the same concept $\mathcal{C}$. This reinforces the synergy between the concept vector $\mathbf{V}$ and its weights during the weighting (decoding) stage, showing stronger reconstruction capabilities.

Sparsemax Attention. One of the core ideas of SAEs is to assign a sparse set of concepts for each latent feature, while the widely-used softmax function in transformers often outputs dense activations [65]. To this end, we introduce the sparsemax attention into our transformer-based SAE, as it is capable of producing exactly zero value to low-scoring concepts. Let $\boldsymbol{z} = \mathbf{Q}\mathbf{K}^T \in \mathbb{R}^M$ denote the similarity score between the query and M concepts, sparsemax aims to project $\boldsymbol{z}$ onto the probability simplex with Euclidean distance:

$$\text{sparsemax}(\boldsymbol{z}) = \arg\min_{p \in \Delta^{M-1}} \|p - z\|^2, \tag{4}$$

where $\Delta^{M-1} := \left\{ p \in \mathbb{R}^M \,\middle|\, p_i \geq 0 \;, \sum_{i=1}^M p_i = 1 \right\}$ is the $(M-1)$-dimensional simplex. Sparsemax aims to find the point inside the simplex that is nearest to $\boldsymbol{z}$. Therefore, for small coordinates of $\boldsymbol{z}$, the closest point in the simplex will force them to zero, in which case $\text{sparsemax}(\boldsymbol{z})$ becomes sparse. Fortunately, Eq. 4 can be solved with linear-time algorithms [16, 41].

Proposition 1. *The closed-form solution of Eq. 4 is:*

$$\text{sparsemax}(\boldsymbol{z})_m = \max(\boldsymbol{z}_m - \tau, 0), \tag{5}$$

where τ is a threshold calculated so the result sums to 1: $\sum_{j \in S}(\boldsymbol{z}_j - \tau) = 1$ for every selected $\boldsymbol{z}_j$, e.g., $S = \{j : \boldsymbol{z}_j > \tau\}$. Furthermore, the support set S (and hence τ) can be computed efficiently via sorting: if we sort z in descending order as $z_{(1)} \geq \cdots \geq z_{(M)}$, define

$$k = \max\left\{ r \in \{1, \ldots, M\} \,\middle|\, z_{(r)} + \frac{1 - \sum_{i=1}^r z_{(i)}}{r} > 0 \right\}, \tag{6}$$

then,

$$\tau = \frac{\sum_{i=1}^k z_{(i)} - 1}{k}. \tag{7}$$

Proof. We provide a detailed derivation in the appendix. □

Unlike TopK-based algorithms that truncate $\boldsymbol{z}$ by setting a hard threshold, Prop. 1 suggests a dynamical τ by measuring the content complexity of the input feature. For example, if the query feature consists of multiple concepts, the $\boldsymbol{z}$ tends to contain many comparable values, resulting in a large set S. Conversely, if the query feature represents pure concepts, the resulting S becomes very small. We summarize the Sparsemax algorithm in Alg. 1.

4. Experiments

In this section, we first outline the experimental setup, followed by the evaluation of Sparsemax SAEs. We conduct experiments in both visual and textual domains and compare our approach with recent advances in terms of image classification and reconstruction tasks. Finally, we visualize the learned concepts at both the image and patch levels, revealing clear and interpretable visual patterns behind these concepts.

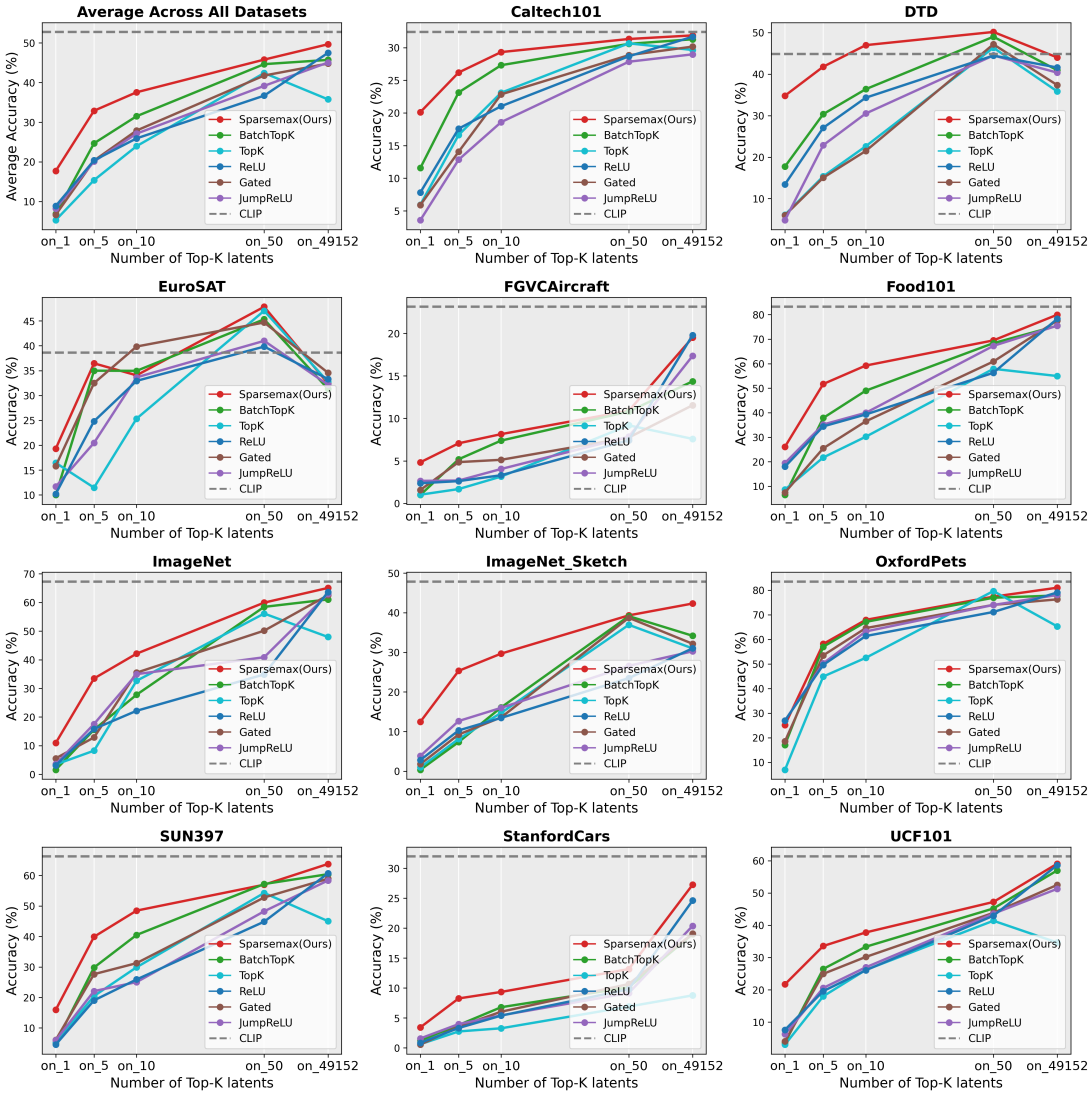

Figure 3. Comparisions of zero-shot image classfication using top-n concepts on 11 datasets. All results are calculated as the mean value of three runs with different random seeds. CLIP denotes the results using the original image features in CLIP, and on_49152 denotes all concepts are used.

Table 1. NMSE scores with various dictionary sizes M on the OpenWeb and WikiText-103 test datasets.

Method	**OpenWeb**				**WikiText-103**			
	M=3072	M=6144	M=12288	M=24576	M=3072	M=6144	M=12288	M=24576
RELU	0.064	0.064	0.064	0.059	0.064	0.064	0.064	0.064
JumpRELU	0.051	0.050	0.050	0.051	0.058	0.056	0.055	0.567
Gated	0.078	0.092	0.129	0.489	0.088	0.106	0.196	0.527
TopK	0.014	0.059	0.010	0.055	0.024	0.063	0.018	0.061
BatchTopK	0.014	0.061	0.060	0.060	0.024	0.064	0.018	0.062
Saprsemax SAE (Ours)	**0.005**	**0.038**	**0.004**	**0.039**	**0.008**	**0.046**	**0.007**	**0.045**

4.1. Experimental Setup

Datasets. For the visual modality, we use the CLIP model with an image encoder of ViT-B/16. This results in a CLS and 14×14 image patch tokens as inputs. Following Patch-SAE [34], we extract the ViT output from the residual stream of the second-last attention layer. Therefore, the training

Table 2. CE degradation with various dictionary sizes M on the OpenWeb and WikiText-103 test datasets.

Method	OpenWeb				WikiText-103			
	M=3072	M=6144	M=12288	M=24576	M=3072	M=6144	M=12288	M=24576
Relu	-4.709	-4.656	-4.614	-1.562	-4.709	-4.656	-4.614	-4.615
JumpRELU	-0.586	-1.300	-0.928	-1.151	-3.272	-3.060	-2.980	-3.260
Gated	-1.738	-1.422	-2.679	-1.309	-6.637	-5.791	-5.453	-5.557
TopK	0.209	-1.778	0.306	-1.261	-0.898	-4.587	-0.569	-4.251
BatchTopK	0.196	-1.742	0.302	-1.740	-0.867	-4.659	-0.566	-4.356
Saprsemax SAE (Ours)	**0.031**	**-1.516**	**0.012**	**-0.395**	**-0.106**	**-2.234**	**-0.113**	**-2.079**

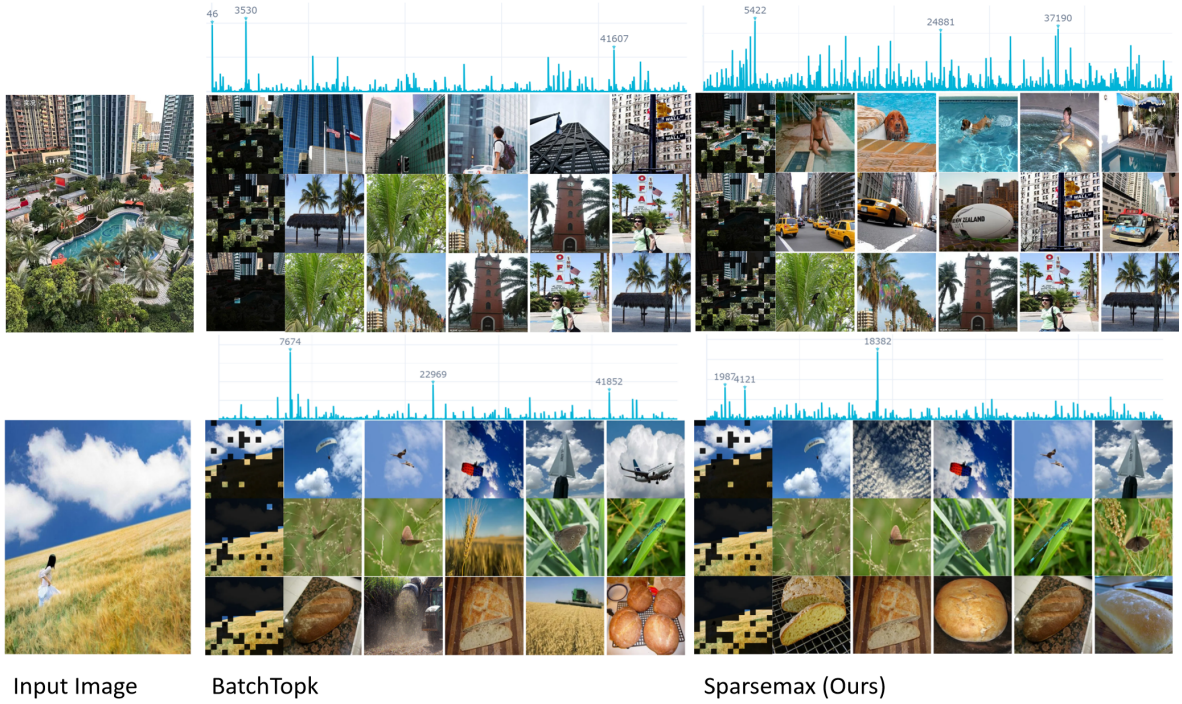

Figure 4. Visualization of top three concepts given the reference image. For each concept, we provide its masking map within the input image and top five reference image from the ImageNet dataset. Compared to BatchSAE, our Sparsemax SAE learns clearer and more interpretable concepts.

Algorithm 1 Sparsemax Attention

1: **Input:** $\boldsymbol{z}$
2: Sort $\boldsymbol{z}$ as $z_{(1)} \geq \cdots \geq z_{(M)}$
3: Find $k(\boldsymbol{z}) := \max \left\{ r \in [M] \,\middle|\, z_{(r)} + \frac{1-\sum_{i=1}^{r} z_{(i)}}{r} > 0 \right\}$, where $[M] := \{1, \ldots, M\}$
4: Define $\tau(\boldsymbol{z}) = \frac{\sum_{i=1}^{k} z_{(i)} - 1}{k\boldsymbol{z}}$.
5: **Output:** $\mathbf{p}$ such that $p_i = max\{0, z_i - \tau(\mathbf{z})\}$

visual data has a size of $N_{img} \times N_{patch} \times d$, with N_{img} and N_{patch} denoting the number of images and patches per image, respectively. We train all SAEs on the ImageNet dataset [13], and evaluate the performance on 11 classification datasets: ImageNet [13] and Caltech101 [18] for generic object classification, OxfordPets [48], StanfordCars [31], Flowers102 [44], Food101 [3] and FGVCAircraft [36] for fine-grained image recognition, EuroSAT [26] for satellite image classification, UCF101 [61] for action classification, DTD [9] for texture classification, and SUN397 [70] for scene recognition. For the textual modality, we choose GPT-2 Small [54] as our pre-trained model, and extract the hidden embeddings from the residual stream of the 8-th transformer layer. We train all SAEs on the training set of the OpenWebText dataset [22], which was processed into sequences of a maximum of 128 tokens for input into the language models. We report the reconstruction results on the test sets of the OpenWebText and WikiText-103 datasets.

Baselines We compare our Sparsemax SAE against a range of state-of-the art baselines, including **1) ReLU-based SAEs**: ReLUSAE [28], a pioneering method that explains LLMs using SAE; PatchSAE [34] views the image patches as tokens and extracts interpretable concepts at both the image and patch levels; GateSAE [57] designs two encoders to model the position and coefficients of the concepts simultaneously; JumpReLU [58] introduces a learnable threshold into ReLU to alleviate the feature shrinkage issue. And **2) TopK-based SAEs**: TopKSAE [63] directly keeps the K largest concepts and zeroes out the others; BatchTopK [5] relaxes TopKSAE by introducing the batch-level operation, where samples within a batch size share $K \times N_{batch}$ activations.

Unlike above models that require additional regularization loss or hyperparameter tuning, Our sparsemax SAE dynamically determines the optimal sparsity level based on feature complexity, demonstrating greater flexibility and interpretability.

Implementation Details. Following prior research [34], for the CLIP model, we set the number of concepts to $M = 49,152$, as a value 64 times the latent dimension of the ViT model. The batch size is 32, and the training continued until a total of $2,621,440$ patches were feed. For the GPT-2 Small model, we conduct experiments with $M = 3072, 6144, 12288$ and 24576 to test the performance with different dictionary sizes. The batch size is 128 and training continued until a total of 1×10^9tokens were processed. All models were trained using the Adam optimizer with a learning rate of 3×10^{-4}, $\beta_1 = 0.9$ and $\beta_2 = 0.99$. For all baselines, we load the suggested hyperparameters according to their papers ($K = 32$ for TopK-based SAEs and the sparsity weighting set to $1e^{-3}$ for ReLU-based SAEs).

4.2. Results Analysis

4.2.1. Zero-Shot Image Classification

To measure the quality of the learned concepts of SAEs and explore whether the activated concepts per class capture the core class-level concepts, we conduct zero-shot image classification tasks by replacing the intermediate embeddings of ViT in CLIP with the SAE reconstruction embeddings using only the top $n = 1, 5, 10, 50$ concepts. The final prediction is calculated by the cosine similarity between textual prototypes and steered image features. To specify the subset concepts per class, we follow PatchSAE [34] and first collect SAE latent activations from the training set of each dataset, and then select the largest n concepts according to their activation frequency. These concepts are then act as masks to control the ViT's reconstruction. Intuitively, the selected top-n concepts capture key information of that class, and higher classification performance denotes a higher quality of the learned concepts.

Fig. 3 reports the classification results of our Sparsemax SAE and baselines on 11 datasets. We also report the results with original ViT embeddings (CLIP) and results with all concepts (on_49152). From the results, we have the following interesting finds: **1)** Overall, our Sparsemax SAE achieves the best average performance across 11 datasets on all top-n settings (top-left subfigure). Particularly at extremely small n values (i.e., $n = 1, 5, 10$), our method significantly outperformed the second-best model. these results demonstrate that our sparse attention-based SAE is much more effective than ReLU and TopK-based SAEs. Our approach successfully assigns the most relevant concepts to the latent features, improving the representation learning of the dictionary. **2)** SAE-based models outperform CLIP on the EuroSAT and DTD datasets $n = 10, 50$ settings, whose images differ significantly from the pre-trained natural images. This demonstrates that the learned concepts are also generalizable and can be useful in zero-shot scenarios due to their monotonicity. **3)** Intuitively, more concepts should yield better reconstruction quality and thus higher prediction scores. However,we observe performance declining from on_50 to on_49152 across multiple datasets. we attribute this to the denoising capability of SAEs, where the top-n concepts capture the clear visual patterns that aid in identifying object labels.

4.2.2. Reconstruction Results on Text

Following previous works [5], we evaluate the reconstruction performance of our approach in terms of normalized mean squared error (NMSE) and cross-entropy (CE) degradation on the OpenWeb and WikiText-103 corpora, and report the results in Table. 1 and 2, respectively. Similarly, we replace the intermediate features in GPT-2 Small with the reconstruction output of SAEs and measure the difference with the original outputs. Experiments demonstrate that across all datasets and dictionary sizes, our proposed Sparsemax SAE not only achieves significantly lower NMSE scores than other methods but also exhibits reduced CE degradation. This proves that Sparsemax SAE not only decouples polysemantic features into interpretable concepts but also reconstructs input data with lower information loss. The dynamic sparse attention mechanism enables the model to leverage more conceptual representations for complex features, fully demonstrating its effectiveness.

4.2.3. Ablation Study

In this section, we aim to ablate the impacts of our proposed modules: transformer-based architecture and sparsemax attention. Specifically, we mainly consider two variants: MLP-based SAEs with the sparsemax function and transformer-based SAEs with the ReLU function. For the latter, we employ the L_1 regularization to promote sparsity. Table. 12 reports the image classification results on the ImageNet dataset. From the ablations, we find that both the introduced modules improve the performance of our base-

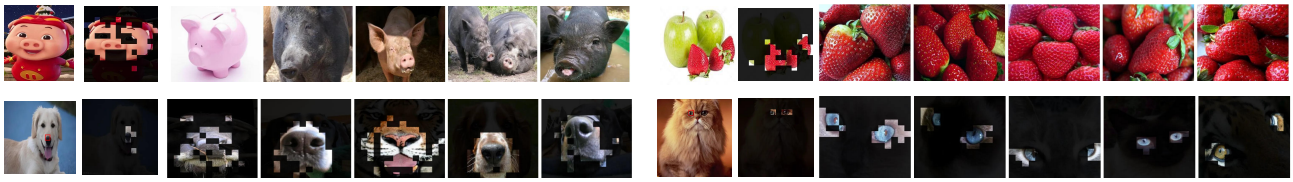

Figure 5. Visualizations of the top one concept. For the query image, we interprete its most relevant concept from the image-level (top row) and patch-level (bottom row), respectively.

Table 3. Ablation results on the ImageNet dataset.

	on_1	on_5	on_10	on_50	on_49152
ReLU SAE	3.12	15.83	22.17	34.87	63.67
Transformer + ReLU	3.86	16.85	24.08	36.33	63.94
MLP + Sparsemax	7.91	29.87	39.73	55.32	64.74
Sparsemax SAE (Ours)	10.93	33.47	42.13	59.95	65.09

Table 4. Zero-shot classification results of different K on the Food101 dataset.

	on_1	on_5	on_10	on_50	on_49152
TopK ($K = 32$)	8.64	21.74	30.21	57.89	54.96
BatchTopK ($K = 32$)	6.46	37.86	48.99	68.37	75.44
TopK ($K = 24$)	0.99	24.42	42.88	66.80	47.67
BatchTopK ($K = 24$)	7.36	44.53	49.52	69.40	75.60
Sparsemax SAE (Ours)	26.11	51.71	59.23	69.49	79.95

line. The transformer framework connects the encoder and decoder by sharing the same concept vectors, resulting in more coherent concepts. The sparsemax function enables the model to dynamically determine the number of activations according to the feature's complexity, improving the trade-off between the sparsity and reconstruction.

4.2.4. Further Analysis

Our sparsemax SAE is trained solely with reconstruction loss, with sparsity determined by the dataset. This enables optimal K selection for TopK-based SAE.Specifically, we first collect all activations of our sparsemax SAE on the ImageNet dataset, and calculate the average number of activated concepts per sample $K^* = 24$. We then re-train TopK-based SAE with the new K^*. Table. 4 reports the results of $K = 24, 32$. We find that the calculated K using our approach is a better choice for both TopK and BatchTopK SAEs in most cases. This improvement suggests that our sparsemax attention is able to estimate the level of sparsity.

4.2.5. Visualization Results

In addition to the above quantitative analysis, we also provide visualizations of the learned concepts. First, we aim to evaluate the top three concepts activated for each given image. Specifically, we use the `CLS` token as the global representation of the images, and obtain the top three concepts with the largest activation scores. Fig. 4 shows the comparison results of BatchTopK and our Sparsemax SAE. For each model, we also visualize the activation scores of all concepts. For each selected concept, we visualize its corresponding patches of the input image and the top five reference images from the ImageNet dataset, which provide an easy tool to understand the meaning of the concepts. From the results, we find that our approach successfully disentangles the core concepts from the input image[35],and the top three concepts show clear and specific visual patterns. For example, Sparsemax SAE successfully extracts concepts of the swimming pool, building, and tree from the input image. Compared to BatchTopK, we find that both models are able to extract the relevant concepts. However, BatchTopK sometimes produces redundant or unclear patterns. For example, the second and third concepts of the first image learned from BatchTopK share similar reference images, and the third concept of the second image contains both the wheat and bread patterns.

We also provide the visualization of the top one concept at the image level (top row of Fig. 5) and patch level (bottom row of Fig. 5). We find that the top one concept successfully extracts the key information of the given image. For the patch-level visualization, our approach accurately localizes the dog's nose and cat's eyes in the reference images, which demonstrates the high-quality of the learned concepts. This suggests that the learned concepts exhibit internal semantic structure that could be further modeled with graph-based discovery methods [15].

5. Conclusion

In this paper, we present a novel transformer-based SAE with sparsemax attention mechanism, where the input feature acts as the query and the concepts are modeled as the key and value matrices. The sparsemax function is then employed to steer the sparsity between the query and key attention. The transformer structure connects the encoder and decoder of SAEs by sharing the same concept vectors, while the sparsemax function enables the model to dynamically determine the sparsity level based on the feature's complexity. This synergy enhances the concept learning and reconstruction performance of traditional SAEs. Extensive experiments across image classification, text reconstruction, and visualization validate the effectiveness of our model. We hope our sparsemax SAE will offer novel insights for secure artificial intelligence and interpretability research.

Acknowledgments

This work was supported in part by National Key R&D Program of China (2024YFB3908500, 2024YFB3908502), NSFC (62506237, 62576215), ICFCRT (W2441020), Guangdong Basic and Applied Basic Research Foundation (2023B1515120026), Shenzhen Science and Technology Program (KQTD20210811090044003, KJZD20240903100022028, RCJC20200714114435012), and Scientific Development Funds from Shenzhen University.

References

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

[2] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 3

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer, 2014. 6

[4] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, 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

[5] Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*. 2, 7

[6] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2818–2829, 2023. 1

[7] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 3

[8] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with performers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 2

[9] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 6

[10] Gonçalo M Correia, Vlad Niculae, and André FT Martins. Adaptively sparse transformers. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 2174–2184. Association for Computational Linguistics, 2019. 3

[11] Bartosz Cywiński and Kamil Deja. Saeuron: Interpretable concept unlearning in diffusion models with sparse autoencoders. *arXiv preprint arXiv:2501.18052*, 2025. 2

[12] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022. 3

[13] 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. 6

[14] Zhibin Duan, Dongsheng Wang, Bo Chen, Chaojie Wang, Wenchao Chen, Yewen Li, Jie Ren, and Mingyuan Zhou. Sawtooth factorial topic embeddings guided gamma belief network. In *International Conference on Machine Learning*, pages 2903–2913. PMLR, 2021. 2

[15] Zhibin Duan, Yishi Xu, Bo Chen, Dongsheng Wang, Chaojie Wang, and Mingyuan Zhou. Topicnet: Semantic graph-guided topic discovery. In *Advances in Neural Information Processing Systems*, 2021. 8

[16] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008. 4

[17] 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. *Transformer Circuits Thread*, 2022. 1

[18] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 6

[19] 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 *International Conference on Machine Learning*, pages 16543–16572. PMLR, 2025. 1

[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. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. 1, 2, 3

[21] Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Peiyuan Zhou, Jiaxing Qi, Junjie Lai, Hayden Kwok-Hay So, Ting Cao, Fan Yang, et al. Seerattention: Learning intrinsic sparse attention in your llms. *arXiv preprint arXiv:2410.13276*, 2024. 3

[22] Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus, 2019. 6

[23] Nuno Gonçalves, Marcos V Treviso, and Andre Martins. Adasplash: Adaptive sparse flash attention. In *Forty-second International Conference on Machine Learning*. 3

[24] Nuno Gonçalves, Marcos V Treviso, and Andre Martins. Adasplash: Adaptive sparse flash attention. In *Forty-second International Conference on Machine Learning*, 2025. 2

[25] Ankit Gupta, Guy Dar, Shaya Goodman, David Ciprut, and Jonathan Berant. Memory-efficient transformers via top-k attention. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 39–52, 2021. 3

[26] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 6

[27] Sai Sumedh R. Hindupur, Ekdeep Singh Lubana, Thomas Fel, and Demba E. Ba. Projecting assumptions: The duality between sparse autoencoders and concept geometry. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 1, 2

[28] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 1, 2, 7

[29] Tao Jin and Jiaming Liu. A text classification method by integrating mobile inverted residual bottleneck convolution networks and capsule networks with adaptive feature channels. *Scientific Reports*, 15(1):855, 2025. 3

[30] Adam Karvonen, Benjamin Wright, Can Rager, Rico Angell, Jannik Brinkmann, Logan Smith, Claudio Mayrink Verdun, David Bau, and Samuel Marks. Measuring progress in dictionary learning for language model interpretability with board game models, 2024. page 16. 2

[31] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 6

[32] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19, 2006. 1

[33] Miaoge Li, Dongsheng Wang, Xinyang Liu, Zequn Zeng, Ruiying Lu, Bo Chen, and Mingyuan Zhou. Patchct: Aligning patch set and label set with conditional transport for multi-label image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15348–15358, 2023. 2

[34] Hyesu Lim, Jinho Choi, Jaegul Choo, and Steffen Schneider. Sparse autoencoders reveal selective remapping of visual concepts during adaptation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. 1, 2, 5, 7

[35] Xinyang Liu, Dongsheng Wang, Miaoge Li, Bowei Fang, Yishi Xu, Zhibin Duan, Bo Chen, and Mingyuan Zhou. Patch-prompt aligned bayesian prompt tuning for vision-language models. In *Uncertainty in Artificial Intelligence*, 2024. 8

[36] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 6

[37] Alireza Makhzani and Brendan Frey. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013. 2

[38] Luke Marks, Alasdair Paren, David Krueger, and Fazl Barez. Enhancing neural network interpretability with feature-aligned sparse autoencoders. *arXiv preprint arXiv:2411.01220*, 2024. 2

[39] Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR, 2016. 2

[40] André F. T. Martins and Ramón Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1614–1623. PMLR, 2016. 3

[41] Christian Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of n. *Journal of Optimization Theory and Applications*, 50(1):195–200, 1986. 4

[42] Timur Mudarisov, Mikhail Burtsev, Tatiana Petrova, et al. Limitations of normalization in attention. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 3

[43] Anish Mudide, Joshua Engels, Eric J Michaud, Max Tegmark, and Christian Schroeder de Witt. Efficient dictionary learning with switch sparse autoencoders, 2024. 273233368. 2

[44] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 6

[45] 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. 1

[46] 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, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024. 2

[47] Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. *arXiv preprint arXiv:2504.02821*, 2025. 2

[48] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 6

[49] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 2, 3

[50] Ben Peters, Vlad Niculae, and André F. T. Martins. Sparse sequence-to-sequence models. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1504–1519. Association for Computational Linguistics, 2019. 2

[51] Ben Peters, Vlad Niculae, and André F. T. Martins. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1504–1519. ACL, 2019. 3

[52] Yao Qiang, Chengyin Li, Prashant Khanduri, and Dongxiao Zhu. Interpretability-aware vision transformer. *arXiv preprint arXiv:2309.08035*, 2023. 2

[53] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 2, 3

[54] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1, 6

[55] 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

[56] Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*, 2024. 2

[57] Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*, 2024. 1, 2, 7

[58] Senthooran 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, 7

[59] Lee Sharkey, Dan Braun, and Beren Millidge. Taking features out of superposition with sparse autoencoders. 2022. 2023. 2

[60] Wei Shi, Sihang Li, Tao Liang, Mingyang Wan, Guojun Ma, Xiang Wang, and Xiangnan He. Route sparse autoencoder to interpret large language models. *arXiv preprint arXiv:2503.08200*, 2025. 1

[61] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6

[62] Samuel Stevens, Wei-Lun Chao, Tanya Berger-Wolf, and Yu Su. Sparse autoencoders for scientifically rigorous interpretation of vision models. *arXiv preprint arXiv:2502.06755*, 2025. 2

[63] 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. 1, 7

[64] Marcos Treviso, António Góis, Patrick Fernandes, Erick Fonseca, and André FT Martins. Predicting attention sparsity in transformers. In *Proceedings of the Sixth Workshop on Structured Prediction for NLP*, pages 67–81, 2022. 3

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3, 4

[66] Dongsheng Wang, Dandan Guo, He Zhao, Huangjie Zheng, Korawat Tanwisuth, Bo Chen, and Mingyuan Zhou. Representing mixtures of word embeddings with mixtures of topic embeddings. In *International Conference on Learning Representations*, 2022. 1

[67] Dongsheng Wang, Miaoge Li, Xinyang Liu, MingSheng Xu, Bo Chen, and Hanwang Zhang. Tuning multi-mode token-level prompt alignment across modalities. *Advances in Neural Information Processing Systems*, 36:52792–52810, 2023. 3

[68] Dongsheng Wang, Jiequan Cui, Miaoge Li, Wang Lin, Bo Chen, and Hanwang Zhang. Instruction tuning-free visual token complement for multimodal llms. In *European Conference on Computer Vision*, pages 446–462. Springer, 2024. 2

[69] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023. 3

[70] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 6

[71] Kaichen Zhang, Yifei Shen, Bo Li, and Ziwei Liu. Large multi-modal models can interpret features in large multi-modal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3650–3661, 2025. 2