ABSTOPK: RETHINKING SPARSE AUTOENCODERS FOR BIDIRECTIONAL FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse autoencoders (SAEs) are a cornerstone of interpretability for large language models (LLMs), aiming to decompose hidden states into meaningful semantic features. While several SAE variants have been proposed, there remains no principled framework to derive SAEs from the original dictionary learning formulation. In this work, we introduce such a framework by unrolling the proximal gradient method for sparse coding. We show that a single-step update naturally recovers common SAE variants, including ReLU, JumpReLU, and TopK. Through this lens, we reveal a fundamental limitation of existing SAEs: their sparsity-inducing regularizers enforce non-negativity, preventing a single feature from representing bidirectional concepts (e.g., male vs. female). This structural constraint fragments semantic axes into separate, redundant features, limiting representational completeness. To address this issue, we propose AbsTopK SAE, a new variant derived from the ℓ_0 sparsity constraint that applies hard thresholding over the largest-magnitude activations. By preserving both positive and negative activations, AbsTopK uncovers richer, bidirectional conceptual representations. Comprehensive experiments across four LLMs and seven probing and steering tasks show that AbsTopK improves reconstruction fidelity, enhances interpretability, and enables single features to encode contrasting concepts. Remarkably, AbsTopK matches or even surpasses the Difference-in-Mean method—a supervised approach that requires labeled data for each concept and has been shown in prior work to outperform SAEs.

1 INTRODUCTION

The pursuit of interpretability has become a central objective in modern machine learning, as it is essential for the assurance, debugging, and fine-grained control of large language models (LLMs) (Marks et al., 2025; Park et al., 2023; Luo et al., 2024; Arora et al., 2018). Within this domain, sparse dictionary learning methods (Poggio & Serre, 2006; Fel et al., 2023), and specifically sparse autoencoders (SAEs), have re-emerged as a prominent methodology for systematically enumerating the latent concepts a model may employ in its predictions (Hindupur et al., 2025; Bussmann et al., 2024; Rajamanoharan et al., 2025; Gao et al., 2025).

An SAE decomposes a model's hidden representations into an overcomplete basis of latent features (Elhage et al., 2022; Thasarathan et al., 2025), which ideally correspond to abstract, data-driven concepts whose linear superposition reconstructs the original activation vector (Higgins et al., 2017; Fel, 2025). Empirical evidence indicates that SAE latents capture semantically coherent features across diverse domains. In LLMs, these features exhibit selectivity for specific entities (e.g., *Golden Gate Bridge*), linguistic behaviors (e.g., sycophantic phrasing), and symbolic systems (e.g., Hebrew script) (Templeton et al., 2024; Csordás et al., 2024; Durmus et al., 2024). Similarly, in vision models, they respond to distinct objects (e.g., barbers, dog shadows) and complex scene properties (e.g., foreground-background separation, facial detection in crowds) (Fel, 2024; Thasarathan et al., 2025). In protein models, they have been shown to correlate with functional elements such as binding sites and structural motifs (Garcia & Ansuini, 2025; Adams et al., 2025).

The discovery of such interpretable, semantically grounded features suggests a natural avenue for steering models: by amplifying, suppressing, or combining specific latents, one can intervene to modulate downstream behavior, which is a principal motivation for research into SAEs (Gao et al.,

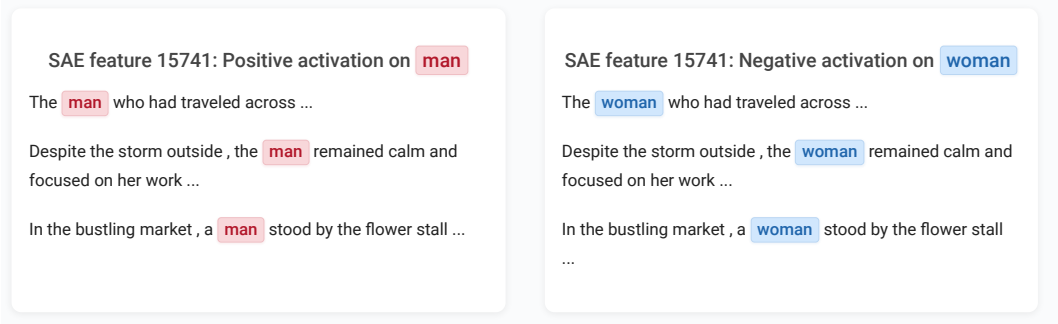

Figure 1: **AbsTopK enables single latent features to encode opposing concepts by leveraging both positive and negative activations.** To test this, we generated controlled sentence pairs with only one differing token (*man* vs. *woman*). The shown feature activates positively for *man* and negatively for *woman*, demonstrating bidirectional encoding. Unlike conventional SAEs, which are restricted by a non-negativity constraint, AbsTopK more compactly captures opposing semantics within a single dimension, yielding richer and more coherent representations.

2025; Bricken et al., 2023; Kantamneni et al., 2025). This control is predicated on the assumption that the concepts identified by SAEs faithfully correspond to the features underlying a model's predictions (Arditi et al., 2024; Uppaal et al., 2024; Engels et al., 2025). While recent studies suggest that simpler techniques such as Difference-in-Means (DiM) can outperform SAEs on practical steering benchmarks and tasks (Arditi et al., 2024; Wu et al., 2025), they are limited to extracting a single vector for a pre-specified, labeled concept, which restricts their flexibility compared to SAEs, especially in exploratory or interpretability contexts. However, these findings introduce uncertainty regarding the degree to which SAEs recover a model's internal features. The comparatively simpler baselines that can rival or outperform SAEs on downstream control tasks suggest that the features extracted by SAEs only partially align with the model's internal decision-making processes, thereby raising concerns about their fidelity as faithful representations.

We posit that one source of this misalignment lies in a structural limitation of conventional SAEs: their systematic neglect of negative activations, despite evidence that many meaningful directions in representation space are inherently bidirectional (Mao et al., 2022). The *linear representation hypothesis* (Mikolov et al., 2013) suggests that a model's internal states can be approximated as linear combinations of semantic vectors, where conceptual transformations correspond to both positive and negative displacements along these vector axes (Arora et al., 2018; Uppaal et al., 2024; Luo et al., 2024). Classic word analogies, such as the vector operation $\boldsymbol{v}_{\text{king}} - \boldsymbol{v}_{\text{man}} + \boldsymbol{v}_{\text{woman}} \approx \boldsymbol{v}_{\text{queen}}$ (Pennington et al., 2014), illustrate how semantic differences are encoded as generalizable vector offsets. Nevertheless, by enforcing non-negativity or retaining only the topk signed activations (Bussmann et al., 2024; Gao et al., 2025), conventional SAEs either fragment such contrastive concepts into separate, unidirectional bases (e.g., "male" and "female") or discard one direction of the semantic axis entirely. This practice not only compromises the representational faithfulness of the learned features but also diminishes their utility for controlled interventions, for which bidirectional traversal of a semantic axis is often essential.

To address this limitation at a foundational level, we analyze the SAE framework from a proximal operator perspective. This reveals that the neglect of negative activations is an inherent property of common nonlinearities, which can be interpreted as the proximal map for a specific regularizer. Motivated by this insight, we use the hard-thresholding as the proximal operator and propose AbsTopK, a principled modification that selects features based on activation magnitude. By preserving both positive and negative activations, the AbsTopK method enables a single feature to represent opposing concepts, as illustrated in Figure 1. This mechanism directly counteracts the fragmentation of contrastive ideas into separate, unrelated features.

Contributions. Our contributions are threefold:

- **A Unified Framework for Designing SAEs.** We introduce a principled framework for designing SAEs by unrolling the proximal gradient method for sparse coding with sparsity-inducing regular-

izers. A single-step update naturally induces common SAE variants, including ReLU, JumpReLU, and TopK (Templeton et al., 2024; Gao et al., 2025; Rajamanoharan et al., 2025). This framework provides a rigorous tool for analyzing their implicit regularizers and identifying shared limitations.

- **Absolute TopK (AbsTopK) for Learning Bidirectional Features** Building on this framework, we propose a new SAE variant derived from the vanilla sparsity constraint (ℓ_0 norm) without a non-negative constraint, which results in a hard-thresholding operator that selects the largest-magnitude activations. By preserving both positive and negative activations, AbsTopK SAE uncovers richer, bidirectional conceptual representations.
- **Comprehensive Empirical Validation.** We conducted a comprehensive empirical evaluation across four LLMs, comparing the proposed AbsTopK SAE with TopK and JumpReLU SAEs on a suite of seven probing and steering tasks, along with three unsupervised metrics. The results demonstrate that AbsTopK outperforms TopK and JumpReLU SAEs, producing representations with higher fidelity and interpretability. Additionally, a case study illustrates that AbsTopK can encode a bidirectional semantic axis within a single latent feature, effectively capturing contrasting concepts. Notably, AbsTopK achieves performance comparable to—or even exceeding—the Difference-in-Mean method, which relies on labeled data and has been shown in prior work to outperform SAEs.

2 From Proximal Interpretations of SAEs to AbsTopK

2.1 Preliminaries

We denote vectors by lowercase bold letters (e.g., $\boldsymbol{x}$) and matrices by uppercase bold letters (e.g., $\boldsymbol{X}$). With an input sequence of N tokens, $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, where each $\boldsymbol{x}_j$ denotes the embedding of the j-th token, the LLM can be viewed as a function $f : \mathbb{R}^{d\times N} \to \mathbb{R}^{V\times N}$, where V is the vocabulary size and $f(\boldsymbol{X})$ gives the output logits for all tokens in the sequence. For our purposes, we abstract away the internal details of f and instead study the representations in the hidden layers. Consider interventions at layer ℓ in the residual stream. Supposing that that the model comprises L layers, then f can be decomposed as

$$f(\boldsymbol{X}) \;=\; \phi_{\ell+1:L}\big(\phi_{1:\ell}(\boldsymbol{X})\big), \tag{1}$$

where $\phi_{1:\ell}(\boldsymbol{X})$ denotes the representation after the first ℓ layers and $\phi_{\ell+1:L}$ represents the remaining computation from layer ℓ to L. We denote by $\boldsymbol{x}_j^{(\ell)}$ the embedding of the residual stream at the ℓ-th layer corresponding to the j-th token of the input sequence $\boldsymbol{X}$. In the following presentation, when the context is clear, we omit the superscript (ℓ) and the subscript (token index j) for notational simplicity, and denote the hidden embedding of a token in a given layer by $\boldsymbol{x}$.

The linear representation hypothesis (Park et al., 2023) assumes that the hidden representation $\boldsymbol{x}$ can be expressed as a linear superposition of latent concepts:

$$\boldsymbol{x} \;=\; \sum_{p=1}^{P} \alpha_p \boldsymbol{h}_p \;+\; \text{context (noise)}, \tag{2}$$

where $\{\boldsymbol{h}_p\}_{p=1}^{P}$ are referred to as concept directions or feature vectors, such as gender or sentiment, and α_i are activation coefficients. Since a particular token—although it encodes information from previous tokens in the context—typically contains only a small subset of concepts or features, its representation is expected to be *sparse*; that is, most of the coefficients α_p are zero, resulting in a sparse linear representation, often simply referred to as *sparse representation*.

To find these concept directions or feature vectors, supervised approaches such as the Difference-in-Mean (DiM) method construct labeled datasets for each target attribute. While effective for isolating specific concepts, these methods are inherently limited to predefined features and do not scale to the large number of latent dimensions present in LLM representations. In contrast, dictionary learning provides an unsupervised and scalable alternative: it can simultaneously recover a more complete dictionary that approximates the underlying concept directions, uncovering a richer and more comprehensive set of latent features than DiM, which is typically restricted to a single concept vector. Consequently, while DiM may achieve stronger control on a specific concept (Wu et al., 2025), dictionary-learning methods have gained popularity due to their ability to uncover a richer, more comprehensive set of latent features.

2.2 Dictionary Learning and the Proximal Perspective on Sparse Autoencoders

In a nutshell, dictionary learning (Olshausen & Field, 1996) seeks to construct a dictionary $\boldsymbol{D}$ consisting of basis vectors $\{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_P\}$, which are also called as *atoms*, such that it can (approximately) provide sparse linear combination for all token embeddings $\boldsymbol{x}$ from the same layer. Since the total number of concept vectors P' is unknown, P is typically set to a relatively large value to ensure that as many concepts as possible can be learned. This typically requires solving a training problem of form (Mairal et al., 2011)

$$\min_{\boldsymbol{D}\in\mathbb{R}^{d\times P},\boldsymbol{b}\in\mathbb{R}^d} \mathbb{E}_{\boldsymbol{x}}\left[\min_{\boldsymbol{z}\in\mathbb{R}^s} \underbrace{\frac{1}{2}\|\boldsymbol{x}-(\boldsymbol{D}\boldsymbol{z}+\boldsymbol{b})\|_2^2}_{g(\boldsymbol{z})} + \lambda R(\boldsymbol{z})\right], \tag{3}$$

where $R(\boldsymbol{z})$ is a sparsity-inducing regularizer, $\lambda > 0$ controls the trade-off between reconstruction fidelity and sparsity, $\boldsymbol{b}$ is an additional bias vector. In classical dictionary learning, the data is often preprocessed to have zero global mean, so the bias term is not used. Alternatively, the bias term can be incorporated into the dictionary as $\boldsymbol{D}\boldsymbol{z}+\boldsymbol{b} = [\boldsymbol{D} \quad \boldsymbol{b}]\begin{bmatrix}\boldsymbol{z}\\ 1\end{bmatrix}$. In this work, however, we explicitly include $\boldsymbol{b}$ to align with the structure of commonly used SAEs which will be described later.

The main challenge in solving the problem (3) lies in jointly estimating both the dictionary $(\boldsymbol{D}, \boldsymbol{b})$ and the sparse coefficients $\boldsymbol{z}$. When one of these variables is fixed, optimizing over the other becomes relatively easier,[1] though still nontrivial in practice. In particular, given a dictionary $\boldsymbol{D}$ and bias $\boldsymbol{b}$, the problem reduces to finding a sparse approximation of $\boldsymbol{x}$, a step commonly referred to as *sparse coding*. An efficient method for solving this problem is the proximal gradient method (Parikh et al., 2014; Silva & Rodriguez, 2020), which is especially suitable when the regularizer $R(\boldsymbol{z})$ is non-differentiable, such as the ℓ_1 norm used in Lasso (Tibshirani, 1996) or ℓ_0 norm that directly enforce sparsity (Foucart, 2011; Bao et al., 2014; Rajamanoharan et al., 2025).

Proximal gradient methods induce encoders For a function $r : \mathbb{R}^d \to \mathbb{R}$, its proximal operator is defined by (Parikh et al., 2014)

$$\mathrm{prox}_r(\boldsymbol{u}) = \arg\min_{\boldsymbol{v}\in\mathbb{R}^d} \frac{1}{2}\|\boldsymbol{v}-\boldsymbol{u}\|^2 + r(\boldsymbol{v}).$$

Now starting from an initialization $\boldsymbol{z}^{(0)}$, the proximal gradient method for optimizing $\boldsymbol{z}$ in (3) performs iterative updates of the form

$$\boldsymbol{z}^{(t+1)} = \mathrm{prox}_{\mu\lambda R}\left(\boldsymbol{z}^{(t)} - \mu\nabla g(\boldsymbol{z}^{(t)})\right) = \mathrm{prox}_{\mu\lambda R}\left(\boldsymbol{z}^{(t)} - \mu\boldsymbol{D}^\top(\boldsymbol{D}\boldsymbol{z}+\boldsymbol{b}-\boldsymbol{x})\right), \tag{4}$$

where $\mu > 0$ is the step size. This perspective naturally leads to *unrolled networks* (Gregor & LeCun, 2010; Chen et al., 2022), where each proximal gradient step can be interpreted as a layer in a neural network that iteratively refines the latent code $\boldsymbol{z}$ while enforcing sparsity (Daubechies et al., 2004). In particular, with $\boldsymbol{z}^{(0)} = 0$ and $\mu = 1$, the first update becomes

$$\boldsymbol{z}^{(1)} = \mathrm{prox}_{\lambda R}\left(\boldsymbol{D}^\top\boldsymbol{x} - \boldsymbol{D}^\top\boldsymbol{b}\right). \tag{5}$$

Inspired by prior work on unrolled networks (Gregor & LeCun, 2010; Chen et al., 2022), we relax the fixed parameters $\boldsymbol{D}, \boldsymbol{b}$ in the above update by learnable counterparts: replacing $\boldsymbol{D}$ with a trainable weight matrix $\boldsymbol{W}$, and $-\boldsymbol{D}^\top\boldsymbol{b}$ with a learnable bias vector $\boldsymbol{b}_\mathrm{e}$, in order to further accelerate convergence. Then the update (5) becomes

$$\boldsymbol{z}^{(1)} = \mathrm{prox}_{\lambda R}\left(\boldsymbol{W}^\top\boldsymbol{x} + \boldsymbol{b}_\mathrm{e}\right), \tag{6}$$

which resembles an encoder. The following result shows that certain regularizers give rise to proximal operators commonly used in SAEs.

[1] This observation has motivated alternating minimization methods such as MOD (Cai et al., 2016) and K-SVD (Aharon et al., 2006).

Lemma 1. *Denote by* $\mathrm{ReLU}_\lambda, \mathrm{JumpReLU}_\theta, \mathrm{TopK}_k$ *as the following operators:*

$$(\mathrm{ReLU}_\lambda(\boldsymbol{u}))_i = \max\{u_i - \lambda, 0\}, \quad (\mathrm{JumpReLU}_\theta(\boldsymbol{u}))_i = \begin{cases} 0, & u_i < \theta, \\ u_i, & u_i \geq \theta, \end{cases},$$
$$(\mathrm{TopK}_k(\boldsymbol{u}))_i = \begin{cases} \max\{u_i, 0\}, & i \in \mathcal{T}_k(\boldsymbol{u}), \\ 0, & i \notin \mathcal{T}_k(\boldsymbol{u}), \end{cases} \tag{7}$$

where $\mathcal{T}_k(\boldsymbol{u})$ *denotes the set of indices corresponding to the* k *largest entries*[2] *of* $\boldsymbol{u}$*. Here* λ*,* θ *and* k *are hyper-parameters subject to design choices. They can be induced by the following choices of sparse regularizers:*

- *Case I:* $R(\boldsymbol{z}) = \|\boldsymbol{z}\|_1 + \iota_{\{\boldsymbol{z} \geq 0\}}(\boldsymbol{z})$*, then* $\mathrm{prox}_{\lambda R} = \mathrm{ReLU}_\lambda$*;*
- *Case II:* $R(\boldsymbol{z}) = \|\boldsymbol{z}\|_0 + \iota_{\{\boldsymbol{z} \geq 0\}}(\boldsymbol{z})$*, then* $\mathrm{prox}_{\lambda R} = \mathrm{JumpReLU}_{\sqrt{2\lambda}}$*;*
- *Case III:* $R(\boldsymbol{z}) = \iota_{\{\|\boldsymbol{z}\|_0 \leq k, \boldsymbol{z} \geq 0\}}(\boldsymbol{z})$*, then* $\mathrm{prox}_{\lambda R}(\boldsymbol{u}) = \mathrm{TopK}_k(\boldsymbol{u})$*.*

Here ι_A *is the indicator function of set* A*, i.e.,* $\iota_A(\boldsymbol{z}) = 0$ *if* $\boldsymbol{z} \in \mathbb{A}$ *and* $\iota_A(\boldsymbol{z}) = +\infty$ *if* $\boldsymbol{z} \notin \mathbb{A}$*, and* $\boldsymbol{z} \geq 0$ *means* $z_i \geq 0$ *for all* i*.*

A detailed proof is provided in the Appendix C. Note that ReLU_λ reduces to the standard ReLU when $\lambda \to 0$. The operators $\mathrm{ReLU}\,\lambda$ and $\mathrm{JumpReLU}_\theta$ are commonly referred to as soft thresholding and hard thresholding (except restricted to the nonnegative orthant), respectively, in signal and image processing, where they are used to enforce sparsity (Foucart, 2011; Acuña et al., 2020). The TopK operator in (7) follows the original formulation in Gao et al. (2025), which includes an additional ReLU to ensure nonnegative activations. Nevertheless, if $\boldsymbol{u}$ has at least k nonnegative entries—which is typically the case since k is much smaller than the ambient dimension s—then the ReLU inside TopK is redundant, and the operator simply retains the largest k entries while setting the rest to zero. This phenomenon is also observed in Gao et al. (2025), where the training curves were found to be indistinguishable. In a nutshell, Lemma 1 establishes that several prevalent nonlinearities in SAEs, including ReLU, JumpReLU, and TopK, are precisely the proximal operators of sparse-enforcing regularizers.

One-step proximal gradient method leads to Sparse Autoencoders. With Lemma 1, applying a one-step proximal gradient method to the sparse coding problem naturally leads to SAEs. Specifically, (6) defines a mapping from an input representation $\boldsymbol{x}$ to a sparse code $\boldsymbol{z}$, which is then decoded to reconstruct the original representation, formally given by

$$\text{encoder: } \boldsymbol{z} = \mathrm{prox}_{\lambda R}\left(\mathbf{W}^\top \boldsymbol{x} + \boldsymbol{b}_{\mathrm{e}}\right), \quad \text{decoder: } \widehat{\boldsymbol{x}} = \boldsymbol{D}\boldsymbol{z} + \boldsymbol{b}. \tag{8}$$

Choosing different regularizers R as in Lemma 1 yields different variants of SAEs, including the vanilla version with ReLU (Cunningham et al., 2023), a version with JumpReLU (Rajamanoharan et al., 2025), and one with TopK (Gao et al., 2025). For simplicity, we refer to these as ReLU SAE, JumpReLU SAE, and TopK SAE, respectively. This observation situates diverse SAE architectures within a unified proximal framework, where each activation function is interpreted as the proximal map for a specific regularizer R. Consequently, design choices for SAEs correspond directly to the selection of an implicit sparsity-inducing penalty, which in turn provides a principled basis for comparing and extending these models. For instance, our analysis in Lemma 1 shows that ReLU SAE corresponds to the ℓ_1 norm regularizer (a convex relaxation of sparsity) with weight $\lambda \to 0$, whereas JumpReLU and TopK correspond directly to the sparsity-inducing ℓ_0 norm regularizers with a non-vanishing λ, thereby enforcing stronger sparsity. This provides a principled explanation for the improved performance of JumpReLU and TopK over ReLU observed in (Rajamanoharan et al., 2025; Gao et al., 2025).

Substituting (6) into (3) yields the training objective for SAEs (Cunningham et al., 2023; Rajamanoharan et al., 2025; Gao et al., 2025):

$$\min_{\substack{\boldsymbol{D}, \mathbf{W} \in \mathbb{R}^{d \times P} \\ \boldsymbol{b} \in \mathbb{R}^d, \boldsymbol{b}_{\mathrm{e}} \in \mathbb{R}^P}} \mathbb{E}_{\boldsymbol{x}} \left[\frac{1}{2} \left\| \boldsymbol{x} - (\boldsymbol{D}\boldsymbol{z} + \boldsymbol{b}) \right\|_2^2 + \lambda R(\boldsymbol{z}), \text{ where } \boldsymbol{z} = \mathrm{prox}_{\lambda R}\left(\mathbf{W}^\top \boldsymbol{x} + \boldsymbol{b}_{\mathrm{e}}\right) \right]. \tag{9}$$

[2] In case k largest components are not uniquely defined, one can choose among them—for example, by selecting the components with the smallest indices—to ensure exactly k entries are kept.

In practice, the two instances of λ in (9) may be decoupled to provide additional flexibility for hyper-parameter tuning.

The use of a parameterized encoder is a key design choice that circumvents the challenging non-convex optimization in the original dictionary learning formulation (3), which requires simultaneous optimization over the sparse codes $\boldsymbol{z}$ and the dictionary parameters $\boldsymbol{D}$ and $\boldsymbol{b}$. By decoupling this joint optimization, SAEs yield a more tractable training procedure. The encoder arises as a single proximal gradient step, augmented with uncoupled, learnable parameters for the dictionary and bias to reduce the approximation error relative to exact sparse coding. Consequently, the training problem (9) can be efficiently solved via stochastic gradient descent, and SAEs can be implemented efficiently at inference time, making them attractive for interpretability research.

This perspective also provides a principled foundation for developing SAE variants with improved performance. For example, by incurring additional computational cost, one may extend (6) to multi-step variants, yielding multi-layer encoders (Tolooshams & Ba, 2022) that produce more accurate sparse codes and potentially capture finer-grained structure in the representation space. We leave this direction to future work. In the next subsection, we turn to SAEs induced by alternative sparsity regularizers.

2.3 Beyond non-negativity: Sparse Autoencoders with AbsTopK

The proximal perspective developed above suggests that design choices for SAEs can be interpreted as the selection of the sparsity-inducing penalty. While this view explains their sparsity-inducing effect, it also reveals a fundamental limitation of current SAEs in Equation (7): they prompt sparsity but also enforce non-negativity, discarding half of the representation space. As many semantic axes are naturally bidirectional (e.g., *male v.s. female*, *positive v.s. negative sentiment*), restricting sparse codes to be nonnegative fragments these concepts into two separate directions or collapses one side entirely.

Fragmentation of Conventional SAE To formalize this, consider a single semantic concept direction $\boldsymbol{h}$ in (2) represented by a dictionary atom $\boldsymbol{d} \in \mathbb{R}^d$. An ideal sparse code would represent concepts along this axis as $\alpha \boldsymbol{d}$, where the sign of the scalar α encodes directionality. However, under the non-negativity constraint $\boldsymbol{z} \geq 0$, this is impossible. Instead, a standard SAE must allocate two distinct dictionary atoms, $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$, oriented in opposite directions, with nonnegative activations $z_i \geq 0$ and $z_j \geq 0$ respectively. Each atom is activated only for one direction, leading to a fragmented representation that arises directly from the non-negativity constraint. This fragmentation is a direct consequence of the non-negativity constraint.

Removing non-negativeness as a remedy. To address this issue, we propose using a sparse regularizer without the non-negativity constraint. Different variants of sparse regularizers can be considered, with representative examples discussed in Lemma 1. In this work, we adopt the ℓ_0 norm due to its simplicity and its direct connection to sparsity. Specifically, in the dictionary learning formulation (3), we use the regularizer $R(\boldsymbol{z}) = \iota_{\{\|\boldsymbol{z}\|_0 \leq k\}}$ which removes the non-negativity constraint present in the TopK-inducing regularizer. The corresponding proximal operator is

$$\operatorname{prox}_{\lambda R}(\boldsymbol{u}) = \arg\min_{\boldsymbol{z} \in \mathbb{R}^d} \tfrac{1}{2}\|\boldsymbol{u} - \boldsymbol{z}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{z}\|_0 \leq k, \tag{10}$$

whose closed-form solution is further given by

$$\left(\operatorname{prox}_R(\boldsymbol{u})\right)_i = \left(\operatorname{AbsTopK}_k(\boldsymbol{u})\right)_i = \begin{cases} u_i, & i \in \mathcal{H}_k(\boldsymbol{u}), \\ 0, & i \notin \mathcal{H}_k(\boldsymbol{u}), \end{cases} \tag{11}$$

where $\mathcal{H}_k$ denotes the indices of the k largest (in modulus) components[3]. In words, this operator preserves the k largest-magnitude components of a vector and sets all others to zero. In the compressive sensing literature, it is referred to as the *hard thresholding operator* (Foucart, 2011). Here, we refer to it as Absolute TopK (AbsTopK) to distinguish it from the TopK operator commonly used in SAE.

[3]Similarly, if the k largest components are not uniquely defined, one can, for instance, select those with the smallest indices to ensure exactly k entries are retained.

This principle of hard thresholding can also be applied to JumpReLU, introducing a threshold on both positive and negative activations. This achieves a similar effect by eliminating small-magnitude features and enforcing sparsity. However, to isolate and directly test our core hypothesis, the value of representing concepts along a bipolar axis, this work focuses on AbsTopK, as it provides the most direct implementation of a global k-sparsity constraint. We remain JumpReLU variants for future investigation.

AbsTopK SAE. Following the derivation in the previous section, we integrate the AbsTopK nonlinearity operator into the framework (9) to obtain a new SAE architecture, which we term AbsTopK SAE:

$$\boldsymbol{z} = \mathrm{AbsTopK}(\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}_\mathrm{e}),\ \widehat{\boldsymbol{x}} = \boldsymbol{D}\boldsymbol{z} + \boldsymbol{b}. \tag{12}$$

The overall training problem becomes

$$\min_{\substack{\boldsymbol{D},\boldsymbol{W}\in\mathbb{R}^{d\times P}\\ \boldsymbol{b}\in\mathbb{R}^d,\boldsymbol{b}_\mathrm{e}\in\mathbb{R}^P}} \mathbb{E}_{\boldsymbol{x}}\left[\frac{1}{2}\left\|\boldsymbol{x}-(\boldsymbol{D}\boldsymbol{z}+\boldsymbol{b})\right\|_2^2,\ \text{where } \boldsymbol{z}=\mathrm{AbsTopK}(\boldsymbol{W}^\top\boldsymbol{x}+\boldsymbol{b}_\mathrm{e})\right]. \tag{13}$$

By design, AbsTopK preserves both positive and negative activations, enabling a single feature to capture contrastive concepts along a unified semantic axis. This simple modification circumvents the fragmentation induced by non-negativity constraints, and yields features that more faithfully reflect the bidirectional structure of semantic representations.

3 Experiments: Empirical Validation of SAE Behavior

To empirically validate our theoretical claims and demonstrate the practical advantages of the AbsTopK operator, we perform a suite of experiments which involve training JumpReLU, TopK, and AbsTopK SAEs on `monology/pile-uncopyrighted` (Gao et al., 2020) across the GPT2-SMALL, Pythia-70M, Gemma2-2B, and Qwen3-4B models (Radford et al., 2019; Biderman et al., 2023; Team, 2024; Yang et al., 2025). To compare the different SAEs, we evaluate their performance along several dimensions: (i) reconstruction quality on base datasets, (ii) effectiveness on a range of steering tasks, and (iii) impact on general capabilities of the models. For further experimental details and extended results, we refer the reader to Appendix B.

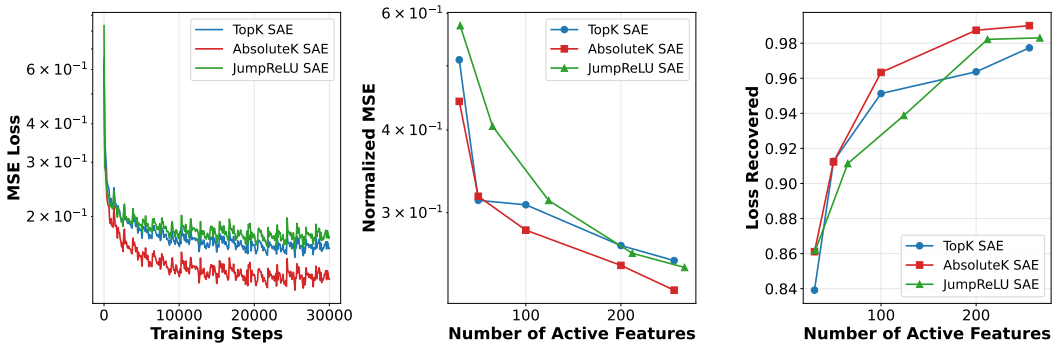

Figure 2: **Performance comparison of JumpReLU, TopK, and AbsTopK SAEs on Qwen3 4B Layer 20**, showing **(a)** MSE Training Loss, **(b)** Normalized MSE, and **(c)** Loss Recovered. Additional results across models and layers are provided in Appendix D.

3.1 Unsupervised Metrics

This section presents a comparative evaluation of SAE architectures, utilizing a suite of complementary metrics engineered to assess distinct facets of model performance. The investigation encompasses three primary analyses: (a) an examination of the training mean squared error (MSE) to evaluate optimization stability and convergence rates; (b) the measurement of normalized reconstruction error as a function of feature sparsity to ascertain representational fidelity; and (c) a relative cross-entropy loss recovered score to determine the preservation of language modeling performance. For Topk and AbsTopK, sparsity is explicitly controlled by directly specifying the number of active

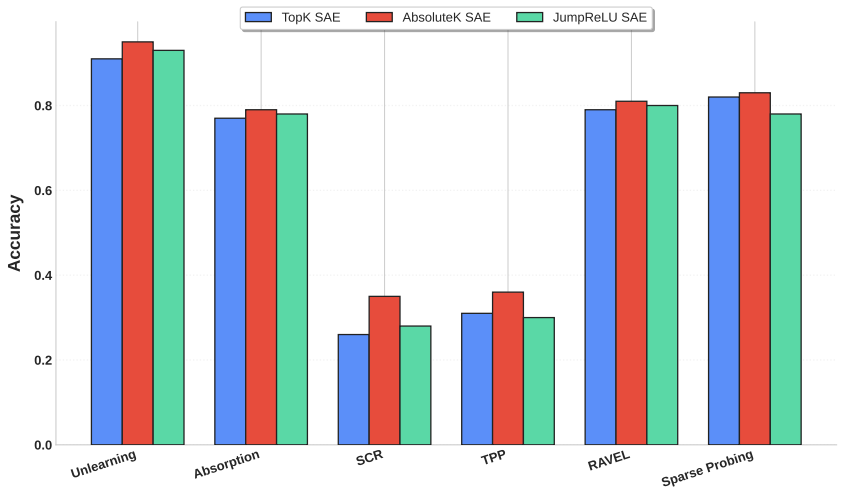

Figure 3: **Performance comparison of SAE variants (TopK, AbsTopK, and JumpReLU) across tasks on Qwen3-4B Layer 18.** For all tasks, higher scores indicate better performance; the Unlearning and Absorption scores have been transformed as $1-$original score to maintain this consistency. For more details, see Appendix E.

features k; in contrast, for JumpReLU, sparsity is varied by manually adjusting the threshold parameter θ, thereby simulating different sparsity levels.

The normalized reconstruction error in (b) is defined as $\text{nMSE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2 / \|\boldsymbol{x}\|_2^2$ (Gao et al., 2025), thereby controlling for scale differences across representations. The Loss Recovered score in (c) measures how well SAE reconstructions preserve predictive performance (Karvonen et al., 2025), defined as $(H^* - H_0)/(H_{\text{orig}} - H_0)$, where H_{orig} is the cross-entropy of the original model, H^* that after substitution, and H_0 under zero-ablation, with values closer to one indicating better preservation.

The empirical results of this evaluation delineate a distinct performance hierarchy among the three architectures under consideration. The AbsTopK architecture demonstrates the most favorable performance characteristics. Specifically, the AbsTopK model consistently yields lower reconstruction errors across most tested sparsity levels and manifests minimal cross-entropy degradation. This demonstrates its superior capacity for preserving the integrity of the language model's performance under conditions of representational compression.

The observed disparities in performance can be attributed to the relative expressiveness of the constraints inherent to each architectural formulation. The TopK and JumpReLU architectures both impose a non-negativity constraint on activations. This imposition results in a conical decomposition of the feature space, which has the effect of fragmenting concepts that are inherently bidirectional in nature. In contrast, the AbsTopK architecture accommodates both positive and negative activation values, a flexibility that facilitates a more faithful linear decomposition of the model's latent states. This capacity for bidirectionality furnishes a more compact and interpretable representational basis, as it permits a single feature to encode oppositional concepts through its algebraic sign. As will be further substantiated in subsequent qualitative analyses, this fundamental property enables the AbsTopK model to acquire dictionary atoms that exhibit a closer alignment with the underlying conceptual structures embedded within the model's representations.

3.2 Results on Probe and Steering Tasks

To assess the utility of learned SAE features for model control, a comprehensive benchmarking evaluation was conducted across a diverse suite of steering and probing tasks. These tasks were specifically designed to probe various dimensions of feature quality, from basic concept representation to the capacity for precise interventional control. A detailed methodological overview for each metric is provided in the Appendix E.

Table 1: **Performance comparison on MMLU (↑) and HarmBench (↑) across steering methods.** The best result among all methods for each metric is highlighted in **bold**. For details of the steering methods, see Appendix F.

Model	Layer	Metric	Original	JumpReLU SAE	TopK SAE	AbsTopK SAE	DiM
Qwen3-4B	18	MMLU	77.3	75.0	75.2	**75.9**	75.8
		HarmBench	17.0	79.1	78.2	**81.3**	80.6
	20	MMLU	77.3	75.7	75.0	**76.4**	**76.4**
		HarmBench	17.0	78.5	77.0	79.0	**80.0**
Gemma2-2B	12	MMLU	52.2	48.8	49.1	**51.3**	51.0
		HarmBench	19.0	69.5	69.8	70.2	**70.8**
	16	MMLU	52.2	48.2	48.5	**51.0**	50.8
		HarmBench	19.0	69.8	70.2	71.7	**72.0**

The empirical results, as shown in Table 3, demonstrate the superiority of the AbsTopK methodology. Across the entire suite of evaluated tasks, AbsTopK SAE outperforms both the TopK SAE and JumpReLU SAE baselines. This performance advantage is especially conspicuous in bidirectional steering metrics, such as SCR, which directly quantify the reliability of interventions. In these critical evaluations, AbsTopK shows marked improvements over the alternatives.

We posit that this consistent outperformance is directly attributable to the core mechanism of the AbsTopK methodology: the retention of both positive and negative feature activations. Unlike TopK approaches, which enforce a hard sparsity constraint that discards all but the most prominent positive activations, AbsTopK preserves a richer, more complete semantic representation. This retention is critical for interventions that require nuanced and bidirectional control. By encoding not only the presence of a concept but also its negation or semantic opposition, AbsTopK features provide a more robust and granular basis for manipulation.

3.3 Empirical Results on Steering vs. Utility

Model steering confronts a fundamental tradeoff: enhancing specific behaviors often degrades general capabilities. It has often been assumed in prior literature that DiM interventions are more effective for specific concept manipulation than SAEs despite their reliance on labeled data and limitation to extracting only a single concept vector (Arditi et al., 2024; Wu et al., 2025; Zhu et al., 2025). To systematically evaluate this trade-off, we conducted an empirical study measuring general capability preservation via the MMLU benchmark (Hendrycks et al., 2021) and safety alignment using HarmBench (Mazeika et al., 2024). For this evaluation, we focus on Qwen and Gemma models, as smaller models, Pythia-70M and GPT-2 Small, only have very low score on MMLU benchmark.

As shown in Table 1, the empirical results indicate that conventional SAE steering methods successfully improve safety metrics but at a detriment to general performance. In contrast, the proposed AbsTopK methodology achieves a more optimal balance between these competing objectives. It facilitates substantial enhancements in safety alignment on HarmBench while simultaneously mitigating the degradation of MMLU scores. Compared to DiM, AbsTopK is competitive on safety, sometimes slightly lower, but consistently retains more general ability. This pattern highlights that carefully designed SAE steering can rival and, in some cases, surpass intervention strategies that rely on labeled data.

4 Conclusion

This work identifies the non-negativity constraint in SAEs as a core cause of semantic feature fragmentation. In response, we introduce the AbsTopK operator, which replaces this constraint with direct k-sparsity enforced via an ℓ_0 proximal operator. This modification enables single features to capture bipolar semantics, and our empirical results confirm that AbsTopK yields reconstructions of superior compactness and fidelity. Our work pioneers a shift towards bipolar sparse representations and suggests future research into more efficient, neurally-plausible approximations of the ℓ_0 operator for large-scale models.

Reproducibility Statement

We provide all implementation details, including hyperparameters and algorithms, in Appendix B. Our code is also submitted as part of the supplementary material to ensure full reproducibility.

Ethics Statement

This work investigates sparse autoencoders in the context of large language models, including their potential to improve model steerability and harmlessness. While such techniques may contribute to safer and more interpretable systems, they could also be misused to influence model behavior in unintended ways. We therefore encourage responsible application of these methods and acknowledge the broader ethical implications associated with controllability in LLMs.

Use of LLMs

We used LLMs to assist in the preparation of this paper, mainly for polishing the presentation, clarifying the wording of technical content, and checking grammar. All research ideas, experimental designs, implementations, and analyses were developed by the authors.

References

Sebastian Acuña, Ida S Opstad, Fred Godtliebsen, Balpreet Singh Ahluwalia, and Krishna Agarwal. Soft thresholding schemes for multiple signal classification algorithm. *Optics Express*, 28(23): 34434–34449, 2020.

Etowah Adams, Liam Bai, Minji Lee, Yiyang Yu, and Mohammed AlQuraishi. From mechanistic interpretability to mechanistic biology: Training, evaluating, and interpreting sparse autoencoders on protein language models. *bioRxiv*, 2025. doi: 10.1101/2025.02.06.636901. URL https://www.biorxiv.org/content/early/2025/02/08/2025.02.06.636901.

M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. doi: 10.1109/TSP.2006.881199.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.

Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. In *Conference on learning theory*, pp. 113–149. PMLR, 2015.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018. doi: 10.1162/tacl_a_00034. URL https://aclanthology.org/Q18-1034/.

Kola Ayonrinde, Michael T Pearce, and Lee Sharkey. Interpretability as compression: Reconsidering sae explanations of neural activations with mdl-saes. *arXiv preprint arXiv:2410.11179*, 2024.

Chenglong Bao, Hui Ji, Yuhui Quan, and Zuowei Shen. L0 norm based dictionary learning by proximal methods with global convergence. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3858–3865, 2014. doi: 10.1109/CVPR.2014.493.

Chenglong Bao, Hui Ji, Yuhui Quan, and Zuowei Shen. Dictionary learning for sparse coding: Algorithms and convergence analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1356–1369, 2016. doi: 10.1109/TPAMI.2015.2487966.

Boaz Barak, Jonathan A Kelner, and David Steurer. Dictionary learning and tensor decomposition via the sum-of-squares method. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 143–151, 2015.

Heinz H. Bauschke and Patrick L. Combettes. Convex analysis and monotone operator theory in hilbert spaces. In *CMS Books in Mathematics*, 2011. URL https://api.semanticscholar.org/CorpusID:117838286.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Usha Bhalla, Suraj Srinivas, Asma Ghandeharioun, and Himabindu Lakkaraju. Towards unifying interpretability and control: Evaluation via intervention, 2025. URL https://openreview.net/forum?id=uOrfve3prk.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

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. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. URL https://openreview.net/forum?id=d4dpOCqybL.

Shuting Cai, Shaojia Weng, Binling Luo, Daolin Hu, Simin Yu, and Shuqiong Xu. A dictionary-learning algorithm based on method of optimal directions and approximate k-svd. In *2016 35th Chinese Control Conference (CCC)*, pp. 6957–6961, 2016. doi: 10.1109/ChiCC.2016.7554453.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, and Joseph Isaac Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders, 2025. URL https://openreview.net/forum?id=LC2KxRwC3n.

Niladri S. Chatterji and Peter L. Bartlett. Alternating minimization for dictionary learning with random initialization. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:1528126.

Maheep Chaudhary and Atticus Geiger. Evaluating open-source sparse autoencoders on disentangling factual knowledge in gpt-2 small, 2024.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research*, 23(189):1–59, 2022.

Julien Colin, Lore Goetschalckx, Thomas FEL, Victor Boutin, Jay R Gopal, Thomas Serre, and Nuria M Oliver. Local vs distributed representations: What is the right basis for interpretability?, 2025. URL https://openreview.net/forum?id=fmWVPbRGC4.

Róbert Csordás, Christopher Potts, Christopher D Manning, and Atticus Geiger. Recurrent neural networks learn to store and generate sequences using non-linear representations. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 248–262, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.17. URL https://aclanthology.org/2024.blackboxnlp-1.17/.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 120–128, 2019.

Esin Durmus, Alex Tamkin, Jack Clark, Jerry Wei, Jonathan Marcus, Joshua Batson, Kunal Handa, Liane Lovitt, Meg Tong, Miles McCain, Oliver Rausch, Saffron Huang, Sam Bowman, Stuart Ritchie, Tom Henighan, and Deep Ganguli. Evaluating feature steering: A case study in mitigating social biases, 2024. URL https://anthropic.com/research/evaluating-feature-steering.

Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. doi: 10.1109/TIP.2006.881969.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Joshua Engels, Logan Riggs Smith, and Max Tegmark. Decomposing the dark matter of sparse autoencoders, 2024. URL https://openreview.net/forum?id=5IZfo98rqr.

Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=d63a4AM4hb.

Eoin Farrell, Yeu-Tong Lau, and Arthur Conmy. Applying sparse autoencoders to unlearn knowledge in language models, 2025. URL https://openreview.net/forum?id=ZtvRqm6oBu.

Thomas Fel. Sparks of explainability: recent advancements in explaining large vision models. 2024.

Thomas Fel. Sparks of explainability: Recent advancements in explaining large vision models. (lueurs d'explicabilité : Avancées récentes dans l'explication des grands modèles de vision). *ArXiv*, abs/2502.01048, 2025. URL https://api.semanticscholar.org/CorpusID:276094390.

Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2711–2721, 2023.

Simon Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on numerical analysis*, 49(6):2543–2563, 2011.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

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 *ICLR*, 2025.

Edith Natalia Villegas Garcia and Alessio Ansuini. Interpreting and steering protein language models through sparse autoencoders. *arXiv preprint arXiv:2502.09135*, 2025.

Dar Gilboa, Sam Buchanan, and John Wright. Efficient dictionary learning with gradient descent. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:53871245.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pp. 399–406, 2010.

Yuzhou Gu, Zhao Song, Junze Yin, and Lichen Zhang. Low rank matrix completion via robust alternating minimization in nearly linear time. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=N0gT4A0jNV.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=JYs1R9IMJr.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015. ISBN 1498712169.

Thomas Heap, Tim Lawson, Lucy Farnik, and Laurence Aitchison. Sparse autoencoders can interpret randomly initialized transformers. *arXiv preprint arXiv:2501.17727*, 2025.

Praveen Hegde. Effectiveness of sparse autoencoder for understanding and removing gender bias in LLMs. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. URL https://openreview.net/forum?id=IxEMXN5hCg.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Sai Sumedh R. Hindupur, Ekdeep Singh Lubana, Thomas Fel, and Demba E. Ba. Projecting assumptions: The duality between sparse autoencoders and concept geometry. In *ICML 2025 Workshop on Methods and Opportunities at Small Scale*, 2025. URL https://openreview.net/forum?id=AKaoBzhIIF.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? a case study in sparse probing. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=rNfzT8YkgO.

Adam Karvonen, Can Rager, Samuel Marks, and Neel Nanda. Evaluating sparse autoencoders on targeted concept erasure tasks. *arXiv preprint arXiv:2411.18895*, 2024.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Isaac Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum Stuart McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=qrU3yNfX0d.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Connor Kissane, Robert Krzyzanowski, Neel Nanda, and Arthur Conmy. Saes are highly dataset dependent: A case study on the refusal direction. Alignment Forum, 2024. URL https://www.alignmentforum.org/posts/rtp6n7Z23uJpEH7od/saes-are-highly-dataset-dependent-a-case-study-on-the.

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

Jinqi Luo, Tianjiao Ding, Kwan Ho Ryan Chan, Darshan Thaker, Aditya Chattopadhyay, Chris Callison-Burch, and René Vidal. Pace: Parsimonious concept engineering for large language models. *Advances in Neural Information Processing Systems*, 37:99347–99381, 2024.

Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):791–804, 2011.

Benoît Malézieux, Thomas Moreau, and Matthieu Kowalski. Understanding approximate and unrolled dictionary learning for pattern recovery. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=rI0LYgGeYaw.

Yongqiang Mao, Zonghao Guo, Xiaonan Lu, Zhiqiang Yuan, and Haowen Guo. Bidirectional feature globalization for few-shot semantic segmentation of 3d point cloud scenes. *2022 International Conference on 3D Vision (3DV)*, pp. 505–514, 2022. URL https://api.semanticscholar.org/CorpusID:251564029.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=I4e82CIDxv.

Fabio Valerio Massoli, Christos Louizos, and Arash Behboodi. Variational learning ISTA. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=AQk0UsituG.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. 2024.

Abhinav Menon, Manish Shrivastava, Ekdeep Singh Lubana, and David Krueger. Analyzing (in)abilities of SAEs via formal languages. In *MINT: Foundation Model Interventions*, 2024. URL https://openreview.net/forum?id=FDn8YNn6U6.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013. URL https://api.semanticscholar.org/CorpusID:5959482.

Thomas Moreau and Joan Bruna. Understanding trainable sparse coding with matrix factorization. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJGPL9Dex.

Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

Charles O'Neill, Alim Gumran, and David Klindt. Compute optimal inference and provable amortisation gap in sparse autoencoders. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=8forr1FkvC.

Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

Kenny Peng, Rajiv Movva, Jon Kleinberg, Emma Pierson, and Nikhil Garg. Use sparse autoencoders to discover unknown concepts, not to act on known concepts. *arXiv preprint arXiv:2506.23845*, 2025.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162/.

Tomaso A. Poggio and Thomas Serre. Learning a dictionary of shape-components in visual cortex: comparison with neurons, humans and machines. 2006. URL https://api.semanticscholar.org/CorpusID:19261154.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, Janos Kramar, Rohin Shah, and Neel Nanda. Improving sparse decomposition of language model activations with gated sparse autoencoders. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=zLBlin2zvW.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, Janos Kramar, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumpreLU sparse autoencoders, 2025. URL https://openreview.net/forum?id=mMPaQzgzAN.

Gustavo Silva and Paul Rodriguez. Efficient consensus model based on proximal gradient method applied to convolutional sparse problems. *arXiv preprint arXiv:2011.10100*, 2020.

Elana Simon and James Zou. Interplm: Discovering interpretable features in protein language models via sparse autoencoders. *bioRxiv*, pp. 2024.11.14.623630, 2024.

Daniel A Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. In *Conference on Learning Theory*, pp. 37–1. JMLR Workshop and Conference Proceedings, 2012.

Yuanming Suo, Minh Dao, Umamahesh Srinivas, Vishal Monga, and Trac D Tran. Structured dictionary learning for classification. *arXiv preprint arXiv:1406.1943*, 2014.

Wen Tang, Émilie Chouzenoux, Jean-Christophe Pesquet, and Hamid Krim. Deep transform and metric learning network: Wedding deep dictionary learning and neural networks. *ArXiv*, abs/2002.07898, 2020. URL https://api.semanticscholar.org/CorpusID:211171394.

Gemma Team. Gemma. 2024. doi: 10.34740/KAGGLE/M/3301. URL https://www.kaggle.com/m/3301.

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. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Harrish Thasarathan, Julian Forsyth, Thomas Fel, Matthew Kowal, and Konstantinos G. Derpanis. Universal sparse autoencoders: Interpretable cross-model concept alignment. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=UoaxRN88oR.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Bahareh Tolooshams and Demba E. Ba. Stable and interpretable unrolled dictionary learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=e3S0Bl2RO8.

Rheeya Uppaal, Apratim Dey, Yiting He, Yiqiao Zhong, and Junjie Hu. Model editing as a robust and denoised variant of DPO: A case study on toxicity. In *Neurips Safe Generative AI Workshop 2024*, 2024. URL https://openreview.net/forum?id=2I821Dypa2.

Mengru Wang, Ziwen Xu, Shengyu Mao, Shumin Deng, Zhaopeng Tu, Huajun Chen, and Ningyu Zhang. Beyond prompt engineering: Robust behavior control in llms via steering target atoms. *arXiv preprint arXiv:2505.20322*, 2025.

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. *arXiv preprint arXiv:2501.17148*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Xudong Zhu, Jiachen Jiang, Mohammad Mahdi Khalili, and Zhihui Zhu. From emergence to control: Probing and modulating self-reflection in language models. *arXiv preprint arXiv:2506.12217*, 2025.

A Related Works

Sparse Dictionary Learning

The convergence of classical dictionary learning is well-studied (Bao et al., 2014; Hastie et al., 2015; Bao et al., 2016), with theoretical guarantees for exact recovery via factorization (Spielman et al., 2012) and semidefinite programming (Barak et al., 2015), alongside popular algorithms like K-SVD (Elad & Aharon, 2006; Aharon et al., 2006) and alternating minimization (Chatterji & Bartlett, 2017; Gu et al., 2024).

Subsequent inquiries have explored gradient-based methods (Beck & Teboulle, 2009; Bauschke & Combettes, 2011; Arora et al., 2015), but analyses of deep, unrolled networks such as LISTA (Gregor & LeCun, 2010; Tang et al., 2020; Massoli et al., 2024) have predominantly focused on encoder convergence speed under the assumption of a fixed dictionary (Suo et al., 2014; Moreau & Bruna, 2017). While some studies address gradient stability (Gilboa et al., 2018; Tolooshams & Ba, 2022; Malézieux et al., 2022), their scope remains limited to local error characterization.

Diverging from this focus on encoder convergence, the present research re-examines SAEs' nonlinearities through the lens of proximal theory. This perspective reveals a direct correspondence between activation functions and the proximal mappings of sparse regularizers, thereby situating SAEs within the broader framework of dictionary learning. Building upon this connection, a novel operator, *AbsTopK*, is introduced. By abrogating the non-negativity constraint common in prior work, AbsTopK capacitates a single dictionary feature to encode bidirectional semantic axes. The principal contribution is therefore the formal alignment of SAE architectural design with the intrinsic geometry of semantic representation, distinct from classical theories centered on signal recovery.

Mechanistic interpretability

Sparse autoencoders (SAEs) have emerged as a central tool in mechanistic interpretability, serving as a dictionary learning approach for concept-level explainability (Kim et al., 2018). A variety of architectures have been proposed, including ReLU SAEs (Bricken et al., 2023), TopK SAEs (Gao et al., 2025), JumpReLU SAEs (Rajamanoharan et al., 2025), gated SAEs (Rajamanoharan et al., 2024), Batch TopK SAEs (Bussmann et al., 2024), and ProLU SAEs (O'Neill et al., 2025), among others. Indeed, these models have demonstrated a remarkable ability to capture a diverse spectrum of interpretable concepts within their latent representations, ranging from abstract notions like refusal, gender, and writing script (Bricken et al., 2023; Templeton et al., 2024; Hegde, 2024), to visual elements such as foreground/background separation (Thasarathan et al., 2025), and even the fundamental structures of proteins (Simon & Zou, 2024).

Despite these successes, a growing body of work has highlighted limitations of the SAE paradigm. Simple prompting baselines have been shown to outperform SAE interventions in model control (Wu et al., 2025; Bhalla et al., 2025), with similar conclusions reported in formal language settings (Menon et al., 2024). Other critiques question the linear representation hypothesis underlying classical SAE design, showing that features can be multidimensional or nonlinear (Engels et al., 2024; 2025; Peng et al., 2025; Wang et al., 2025). Moreover, multiple studies demonstrate severe algorithmic instability: two SAEs trained on the same data but with different random seeds may yield divergent feature dictionaries, leading to inconsistent interpretations (Ayonrinde et al., 2024; Kissane et al., 2024; Colin et al., 2025). These observations suggest that, while SAEs provide a promising path toward interpretability, their current formulation suffers from fragility and non-canonical representations.

Our work responds to this need by placing SAE nonlinearities in the broader dictionary learning framework, using a proximal perspective. This unifying view clarifies the connection between popular activation functions and sparse regularizers and motivates our proposed *AbsTopK*, which enables bidirectional semantic representation.

B Experimental Setup

In this appendix, we describe the architecture and training setup of our SAEs. For all experiments, we trained on the `monology/pile-uncopyrighted` (Gao et al., 2020) dataset.

Architecturally, the SAEs are comprised of a single, overcomplete hidden layer which incorporates a sparsifying nonlinearity. The encoder component projects residual activations into a latent space of higher dimensionality, while the decoder component reconstructs the original residual dimension from these latent representations. A fixed expansion factor of 16 was uniformly applied across all models.

For comparative analysis, three distinct variants of the SAE were trained: TopK, AbsTopK, and JumpReLU. In the TopK and AbsTopK configurations, exact k-sparsity was enforced upon the latent representation, with the sparsity hyperparameter, k, and the specific layers targeted for intervention being systematically selected for each foundational model:

- `EleutherAI/pythia-70m` (Biderman et al., 2023): $k = 51$, layers: 3, 4.
- `google/gemma-2-2b` (Team, 2024): $k = 230$, layers: 12, 16.
- `Qwen/Qwen3-4B` (Yang et al., 2025): $k = 256$, layers: 18, 20.
- `openai-community/gpt2` (Radford et al., 2019): $k = 76$, layers: 6, 8.

Here, k was set to approximately one-tenth of the hidden dimension for each model, and the intervention layers were selected from the middle of the network to capture representative latent features (Arditi et al., 2024). In contrast, the JumpReLU models adopted the same configuration as in prior work (Rajamanoharan et al., 2025; Bussmann et al., 2024).

The optimization for all models was performed using the Adam algorithm over a duration of 30,000 training steps, with a consistent batch size of 4096. A learning rate of 3e-4 was configured, complemented by Adam's momentum parameters, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. And we used a bandwidth parameter of 0.001 across all experiments.

C Proof of Lemma 1

Proof. We prove the result by deriving the proximal operator corresponding to each regularizer separately.

Case I: ReLU. Note that $R(\boldsymbol{z})$ is separable as

$$R(\boldsymbol{z}) = \|\boldsymbol{z}\|_1 + \iota_{\{\boldsymbol{z}\geq 0\}}(\boldsymbol{z}) = \sum_i \left(|z_i| + \iota_{\{z_i \geq 0\}}(z_i)\right),$$

which implies that the proximal operator is also separable, i.e., $(\mathrm{prox}_{\lambda R}(\boldsymbol{u}))_i$ is equivalent to the following scalar proximal problem

$$\begin{aligned}
\mathrm{prox}_{\lambda R}(u) &= \arg\min_{z\in\mathbb{R}} \frac{1}{2}(z-u)^2 + \lambda|z| + \iota_{\{z\geq 0\}}(z) \\
&= \arg\min_{z\geq 0} \frac{1}{2}(z-u)^2 + \lambda z \\
&= \max\{u-\lambda,\, 0\}.
\end{aligned}$$

Therefore, the proximal operator induces the ReLU operator, with a shift by λ:

$$\boxed{(\mathrm{prox}_{\lambda R}(\boldsymbol{u})) \;=\; \max\{\boldsymbol{u}-\lambda,\, 0\},}$$

which reduces to the standard ReLU when $\lambda \to 0$. In this case, however, the operator no longer encourages sparsity. When $\lambda > 0$, the effect is equivalent to introducing a bias term that suppresses small activations and thereby promotes sparsity. In practice, this restriction can be relaxed: during training, gradient descent can learn a separate bias parameter for each entry.

Case II: JumpReLU. Similarly, $R(\boldsymbol{z})$ is also separable as

$$R(\boldsymbol{z}) = \|\boldsymbol{z}\|_0 + \iota_{\boldsymbol{z}\geq 0}(\boldsymbol{z}) = \sum_i \left(\mathbf{1}(z_i \neq 0) + \iota_{\{z_i \geq 0\}}(z_i)\right).$$

where $\mathbf{1}(z_i \neq 0) = \begin{cases} 1, z_i \neq 0, \\ 0, z_i = 0. \end{cases}$ Thus, it suffices to first consider the following scalar proximal operator

$$\begin{aligned} \operatorname{prox}_{\lambda R}(u) &= \arg\min_{z \in \mathbb{R}} \frac{1}{2}(z-u)^2 + \lambda \mathbf{1}(z \neq 0) + \iota_{\{z \geq 0\}}(z) \\ &= \arg\min_{z \geq 0} \underbrace{\frac{1}{2}(z-u)^2 + \lambda \mathbf{1}(z \neq 0)}_{\xi(z)}. \end{aligned}$$

Note that within the region $z \geq 0$, ξ achieve its minimum at either 0 or u. Setting $\xi(u) = \lambda = \xi(0) = \frac{1}{2}u^2$ yields $u = \sqrt{2\lambda}$. One can verify that ξ achives its minimum at u when $u \geq \sqrt{2\lambda}$, and at 0 otherwise. Hence, the proximal operator induces the JumReLU with parameter $\sqrt{2\lambda}$:

$$\boxed{(\operatorname{prox}_{\lambda R}(\boldsymbol{u}))_i = \begin{cases} u, & u \geq \sqrt{2\lambda}, \\ 0, & u < \sqrt{2\lambda}. \end{cases}}$$

Case III: TopK. For this case, the corresponding proximal operator reduces to a Euclidean projection onto the feasible set:

$$\operatorname{prox}_{\lambda R}(\boldsymbol{u}) = \arg\min_{\boldsymbol{z} \in \mathbb{R}^d} \tfrac{1}{2}\|\boldsymbol{u} - \boldsymbol{z}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{z}\|_0 \leq k,\ \boldsymbol{z} \geq 0. \tag{14}$$

Given the quadratic objective and the non-negativity constraint, the optimal choice on any candidate support S with $|S| \leq k$ is

$$z_i = \begin{cases} \max\{u_i, 0\}, & i \in S, \\ 0, & i \notin S. \end{cases} \tag{15}$$

Thus, the minimization problem reduces to selecting the index set S that captures the k largest nonnegative entries of $\boldsymbol{u}$. Formally, letting $\mathcal{T}_k(\boldsymbol{z})$ denote the set of indices corresponding to the k largest entries of $\boldsymbol{z}$, the proximal operator becomes

$$\boxed{[\operatorname{prox}_{\lambda R}(\boldsymbol{u})]_i = \begin{cases} \max\{u_i, 0\}, & i \in \mathcal{T}_k(\boldsymbol{z}), \\ 0, & i \notin \mathcal{T}_k(\boldsymbol{z}). \end{cases}} \tag{16}$$

□

D Unsupervised Metrics on All Models

This section presents the unsupervised metrics from our model evaluations. We tested each model with a specific set of k values. For the Pythia model, we used k-values of 10, 20, 30, 40, and 50. The evaluation of the Gemma model involved k values of 30, 50, 100, 200, and 230. For the GPT model, the k values were 10, 30, 50, 60, and 76. Lastly, the Qwen model was tested with k values of 30, 50, 100, 200, and 256.

As shown in Figure 4, across the majority of evaluated models, we observe that AbsTopK achieves lower training MSE, reduced normalized reconstruction error, and better preservation of language modeling performance relative to both TopK and JumpReLU. This consistent advantage across these metrics provides evidence for the effectiveness and robustness of the AbsTopK method.. In particular, while TopK and JumpReLU sometimes exhibit competitive performance in isolated settings, AbsTopK maintains robustness across architectures and layers, thereby demonstrating the superiority of our proposed formulation.

E Steering and Probe Task on All Models

E.1 Task Description

We provide an overview of the tasks employed in the SAEBench evaluation for SAEs. We do not utilize the Automated Interpretability (AutoInterp) evaluation, as its reliability has been ques-

Table 2: **Performance comparison of SAE variants across tasks on all other models and layers.** For all tasks, higher scores indicate better performance; the Unlearning and Absorption scores have been transformed as 1−original score to maintain this consistency.

Model	Method	Unlearning	Absorption	SCR	TPP	RAVEL	Sparse Probing
Gemma2-2B L12	AbsTopK	0.93	0.73	0.27	0.34	0.73	0.76
	TopK	0.88	0.76	0.20	0.29	0.70	0.71
	JumpReLU	0.90	0.75	0.22	0.30	0.71	0.73
Gemma2-2B L14	AbsTopK	0.91	0.70	0.27	0.42	0.71	0.70
	TopK	0.89	0.68	0.21	0.36	0.74	0.67
	JumpReLU	0.94	0.69	0.23	0.39	0.72	0.69
Pythia-70M L3	AbsTopK	0.75	0.54	0.20	0.22	0.64	0.66
	TopK	0.71	0.47	0.15	0.14	0.62	0.60
	JumpReLU	0.73	0.50	0.17	0.21	0.63	0.61
Pythia-70M L4	AbsTopK	0.79	0.53	0.21	0.23	0.68	0.57
	TopK	0.72	0.50	0.16	0.21	0.69	0.61
	JumpReLU	0.77	0.51	0.17	0.20	0.61	0.62
GPT2-small L6	AbsTopK	0.74	0.66	0.18	0.22	0.60	0.54
	TopK	0.80	0.63	0.14	0.19	0.57	0.50
	JumpReLU	0.77	0.65	0.15	0.20	0.58	0.52
GPT2-small L8	AbsTopK	0.75	0.67	0.23	0.28	0.51	0.59
	TopK	0.71	0.67	0.15	0.20	0.48	0.55
	JumpReLU	0.73	0.67	0.18	0.23	0.49	0.57
Qwen3-4B L18	AbsTopK	0.95	0.79	0.35	0.36	0.81	0.83
	TopK	0.91	0.77	0.26	0.31	0.79	0.82
	JumpReLU	0.93	0.78	0.28	0.30	0.80	0.78
Qwen3-4B L20	AbsTopK	0.95	0.80	0.32	0.45	0.85	0.81
	TopK	0.92	0.77	0.27	0.36	0.76	0.84
	JumpReLU	0.93	0.78	0.29	0.39	0.81	0.83

tioned (Heap et al., 2025). For detailed methodology, we refer readers to the original SAEBench paper (Karvonen et al., 2025).

E.1.1 Feature Absorption

Sparsity incentives can cause a SAE to engage in feature absorption, a phenomenon where correlated features are merged into a single latent representation. This process arises when a direct implication exists between two concepts, such that concept A always implies concept B. To reduce the number of active latents, the SAE might absorb the feature for A into the latent for B. For example, a feature for "starts with S" could be absorbed into a more general latent for "short." While this merging improves computational efficiency, it compromises interpretability by creating gerrymandered features that represent multiple, distinct concepts.

To quantify feature absorption, we employ a first-letter classification task, following the methodology of previous studies (Chanin et al., 2025). First, a supervised logistic regression probe is trained on tokens containing only English letters to establish ground-truth feature directions. Next, K-sparse probing is applied to the SAE's latents to identify the primary latent corresponding to each feature, using a threshold of $\tau_{\text{fs}} = 0.03$ to account for potential feature splits. For test set tokens where main latents fail but the probe succeeds, additional SAE latents are included if they satisfy cosine similarity with the probe of at least $\tau_{\text{ps}} = 0.025$ and a projection fraction of at least $\tau_{\text{pa}} = 0.4$. All parameter values are chosen following the original SAEBench settings (Karvonen et al., 2025). To make the results more interpretable and such that higher values indicate stronger unlearning, we present the final scores as $1 -$ original value.

E.1.2 UNLEARNING

SAEs are evaluated on their ability to selectively remove knowledge while maintaining performance on unrelated tasks (Farrell et al., 2025). We use the WMDP-bio dataset (Li et al., 2024) for unlearning and MMLU (Hendrycks et al., 2021) to assess general abilities.

The intervention methodology involves clamping selected WMDP-bio SAE feature activations to negative values whenever the corresponding features activate during inference. To evaluate broader model effects, we also measure performance on the MMLU benchmark (Hendrycks et al., 2021). The final evaluation reports the highest unlearning effectiveness on WMDP-bio while ensuring MMLU accuracy remains above 0.99, thereby quantifying optimal unlearning performance under constrained side effects. To make the results more interpretable and such that higher values indicate stronger unlearning, we present the final scores as $1 -$ original value.

E.1.3 SPURIOUS CORRELATION REMOVAL (SCR)

SCR (Karvonen et al., 2024) evaluates the ability of SAEs to disentangle latents corresponding to distinct concepts. We conduct experiments on datasets known for spurious correlations, such as Bias in Bios (De-Arteaga et al., 2019) and Amazon Reviews (Hou et al., 2024), which contain two binary gender labels. For each dataset, we create a balanced set containing all combinations of profession (professor/nurse) and gender (male/female), as well as a biased set including only male+professor and female+nurse combinations. A biased classifier C is first trained on the biased set and then debiased by ablating selected SAE latents.

We quantify SCR using the normalized evaluation score:

$$S_{\text{SHIFT}} = \frac{A_{\text{abl}} - A_{\text{base}}}{A_{\text{oracle}} - A_{\text{base}}}, \tag{17}$$

where A_{abl} is the probe accuracy after SAE feature ablation, A_{base} is the baseline accuracy before ablation, and A_{oracle} is the skyline accuracy obtained by a probe trained directly on the desired concept. Higher S_{SHIFT} values indicate more effective removal of spurious correlations. This score represents the proportion of improvement achieved through ablation relative to the maximum possible improvement, enabling fair comparison across classes and models.

E.1.4 TARGETED PROBE PERTURBATION (TPP)

TPP (Marks et al., 2025) extends the SHIFT methodology to multiclass natural language processing datasets. For each class c_i in a dataset, we select the most relevant SAE latents L_i. We then evaluate the causal effect of ablating L_i on linear probes C_j trained to classify each class c_j.

Let A_j denote the accuracy of probe C_j before ablation, and $A_{j\setminus i}$ the accuracy after ablating L_i. We define the accuracy change as

$$\Delta A_{j\setminus i} = A_{j\setminus i} - A_j. \tag{18}$$

The TPP score is then

$$S_{\text{TPP}} = \mathbb{E}_{i=j}\left[\Delta A_{j\setminus i}\right] - \mathbb{E}_{i\neq j}\left[A_{j\setminus i}\right], \tag{19}$$

which measures the extent to which ablating latents for class i selectively degrades the corresponding probe while leaving other probes unaffected. A high TPP score thus indicates effective disentanglement of SAE latents.

E.1.5 RAVEL

RAVEL (Chaudhary & Geiger, 2024) evaluates the ability of SAEs to disentangle features by testing whether individual latents correspond to distinct factual attributes. The dataset spans five entity types (cities, Nobel laureates, verbs, physical objects, and occupations), each with 400–800 instances and 4–6 attributes (e.g., cities have country, continent, and language), probed with 30–90 natural language and JSON prompt templates.

Evaluation proceeds in three stages: (i) filtering entity and attribute pairs that the model predicts reliably, (ii) identifying attribute and specific features using probes trained on latent representations, and (iii) computing a disentanglement score that averages *cause* and *isolation* metrics. The *cause*

score measures whether intervening on a feature for attribute A (e.g., setting Paris's country to Japan) correctly changes the prediction of A, while the *isolation* score verifies that other attributes B (e.g., language = French) remain unaffected. A higher final score indicates stronger disentanglement of features.

E.1.6 PROBING EVALUATION

We assess whether SAEs capture interpretable features through targeted probing tasks across five domains: profession classification, sentiment and product categorization , language identification, programming language classification, and topic categorization. Each dataset is partitioned into multiple binary classification tasks, yielding a total of 35 evaluation tasks.

For each task, we encode inputs with the SAE, apply mean pooling over non-padding tokens, and select the topk latents via maximum mean difference. A logistic regression probe is then trained on these representations and evaluated on held-out test data. To ensure comparability across tasks, we sample 4,000 training and 1,000 test examples per task, truncate inputs to 128 tokens, and, for GitHub, exclude the first 150 characters following Gurnee et al. (2023). We also compare mean and max pooling, finding mean pooling slightly superior. Datasets with more than two classes are subsampled into balanced binary subsets while maintaining a positive class ratio of at least 0.2.

E.2 TASK PERFORMANCE

As shown in Table 2, we find that the AbsTopK methodology exhibits a superior level of performance relative to the comparative TopK and JumpReLU techniques across the evaluated models and layers.

In particular, the AbsTopK operator performs best on the majority of the evaluation metrics. While its performance is more competitive in a few areas, its dominant strength in the other key areas makes it a robust and highly effective sparsity operator according to these results. The method's strength appears to be model-agnostic, showcasing its general applicability.

F STEERING METHODS FOR DIM AND SAES

In this section, we present methods for controlling specific concepts in model representations. For DiM, we introduce two intervention strategies: *activation addition*, to amplify a concept's effect, and *directional ablation*, to remove it from intermediate activations. For the HarmBench experiments, we specifically employ the activation addition method. Following this, we describe how similar steering can be achieved in SAEs through latent feature manipulation and ablation.

Activation addition. Given a concept vector $\boldsymbol{d}^{(l)}$ extracted from layer l, we can modulate the corresponding feature via a simple linear intervention. Concretely, for a specific input, we add the vector to the layer activations with the strength α to shift them toward the concept activation, thereby inducing the given concept:

$$\boldsymbol{x}^{(l)\prime} \leftarrow \alpha \boldsymbol{d}^{(l)} + \boldsymbol{x}^{(l)}. \tag{20}$$

This intervention is applied only at layer l and across all token positions.

Directional ablation. To study the role of a particular direction $\boldsymbol{d}$ in the model's computation, we can remove it from the representations using directional ablation. Specifically, we zero out the component along $\boldsymbol{d}$ for every residual stream activation $\boldsymbol{x}$:

$$\boldsymbol{x}^{(l)\prime} \leftarrow \boldsymbol{x}^{(l)} - \alpha \boldsymbol{d}\boldsymbol{d}^{\top}\boldsymbol{x}^{(l)}. \tag{21}$$

This operation is applied to every activation $\boldsymbol{x}^{(l)}$, across all layers l, effectively preventing the model from encoding this direction in its residual stream.

SAE Latent feature clamping. For a target latent feature z_i in the SAE feature vector $\boldsymbol{z}$, we can modulate its influence on model behavior by clamping it to a constant $c \in \mathbb{R}$. Denote a feature vector $\boldsymbol{z}$, and let $\boldsymbol{z}_{i,c}$ be the modified vector with z_i replaced by c.

Define the clamping function $C_{i,c}$ as

$$[C_{i,c}(\boldsymbol{z})]_k = \begin{cases} z_k & \text{if } k \neq i, \\ c & \text{if } k = i, \end{cases} \tag{22}$$

so that $C_{i,c}(\boldsymbol{z}) = \boldsymbol{z}_{i,c}$.

In conventional SAEs, this clamping strategy can be interpreted as a directional control: setting c to a negative value suppresses the corresponding concept, while a positive c encourages it. We adopt a similar approach to perform steering in our framework, using clamping to directly modulate individual latent features and thereby control the presence or absence of specific semantic concepts in the reconstructed representation.

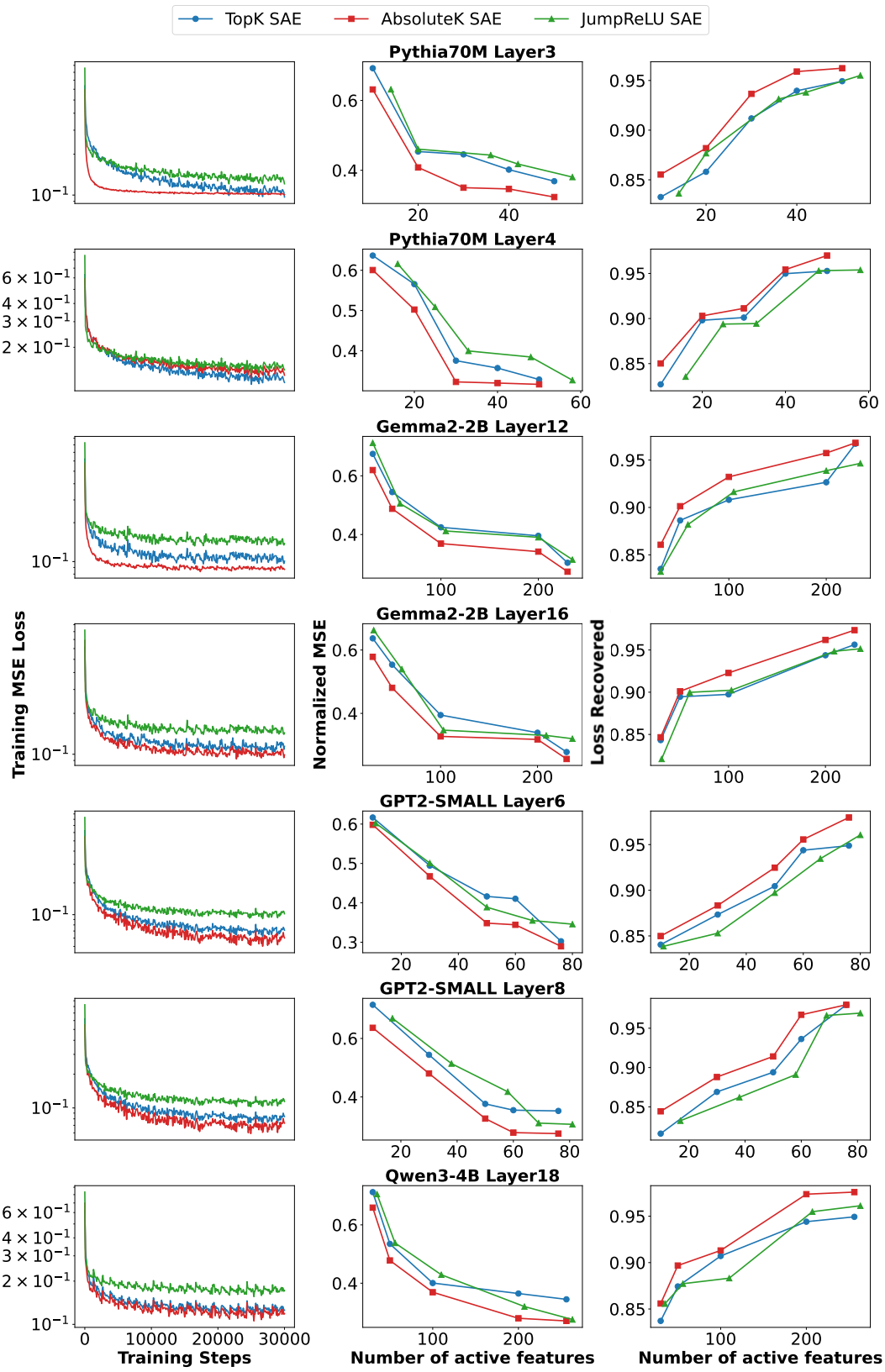

Figure 4: **Performance comparison of JumpReLU, TopK, and AbsTopK SAEs on all other models and layers**, showing **(a)** MSE Training Loss, **(b)** Normalized MSE, and **(c)** Loss Recovered.