

# Resurrecting the Salmon: Rethinking Mechanistic Interpretability with Domain-Specific Sparse Autoencoders

Charles O'Neill<sup>1</sup>, Mudith Jayasekara<sup>1</sup> and Max Kirkby<sup>1</sup>

<sup>1</sup>Parsed, London, UK

Sparse autoencoders (SAEs) decompose large language model (LLM) activations into latent features that reveal mechanistic structure. Conventional SAEs train on broad data distributions, forcing a fixed latent budget to capture only high-frequency, generic patterns. This often results in significant linear “dark matter” in reconstruction error and produces latents that fragment or absorb each other, complicating interpretation. We show that restricting SAE training to a well-defined domain (medical text) reallocates capacity to domain-specific features, improving both reconstruction fidelity and interpretability. Training JumpReLU SAEs on layer-20 activations of Gemma-2 models using 195k clinical QA examples, we find that domain-confined SAEs explain up to 20% more variance, achieve higher loss recovery, and reduce linear residual error compared to broad-domain SAEs. Automated and human evaluations confirm that learned features align with clinically meaningful concepts (e.g., “taste sensations” or “infectious mononucleosis”), rather than frequent but uninformative tokens. These domain-specific SAEs capture relevant linear structure, leaving a smaller, more purely nonlinear residual. We conclude that domain-confinement mitigates key limitations of broad-domain SAEs, enabling more complete and interpretable latent decompositions, and suggesting the field may need to question “foundation-model” scaling for general-purpose SAEs.

## 1. Introduction

Sparse autoencoders (SAEs) are employed in mechanistic interpretability to decompose the hidden activations of large language models (LLMs) into sparse latent representations that ideally correspond to semantically distinct, causal features. In these models, an SAE learns to reconstruct input activations using a fixed latent budget and a sparsity constraint, thereby forcing the network to represent each activation as a sparse linear combination of latent directions. Prior work has applied SAEs to models such as GPT-4 and Claude to identify interpretable circuits and uncover the features underlying model behaviour (Bricken et al., 2023; Cunningham et al., 2023).

Several studies indicate that when SAEs are trained on broad data distributions, a number of limitations arise. When trained on a broad dataset, the fixed latent budget forces the SAE to capture only the most common, high-frequency patterns, leaving little capacity for fine-grained, domain-specific features. For instance, despite increasing coverage of concepts as the capacity of the SAE is increased, there is evidence that even in the largest SAEs, the set of features uncovered is an incomplete description of the model’s internal representations (Templeton et al., 2024).<sup>1</sup> A large portion of SAE error is linearly predictable from input activations, suggesting a multitude of unlearned features (Engels et al., 2024). This residual error leads to substantial downstream substitution loss when the SAE reconstruction is reinserted into the language model (Gao et al., 2024).

<sup>1</sup>For example, Templeton et al. (2024) confirmed that Claude 3 Sonnet can list all of the London boroughs when asked, and in fact can name tens of individual streets in many of the areas. However, features corresponding to approximately 60% of the boroughs were found in the 34M SAE. This suggests that the model contains many more features than they have found, which may be extracted with even larger SAEs.

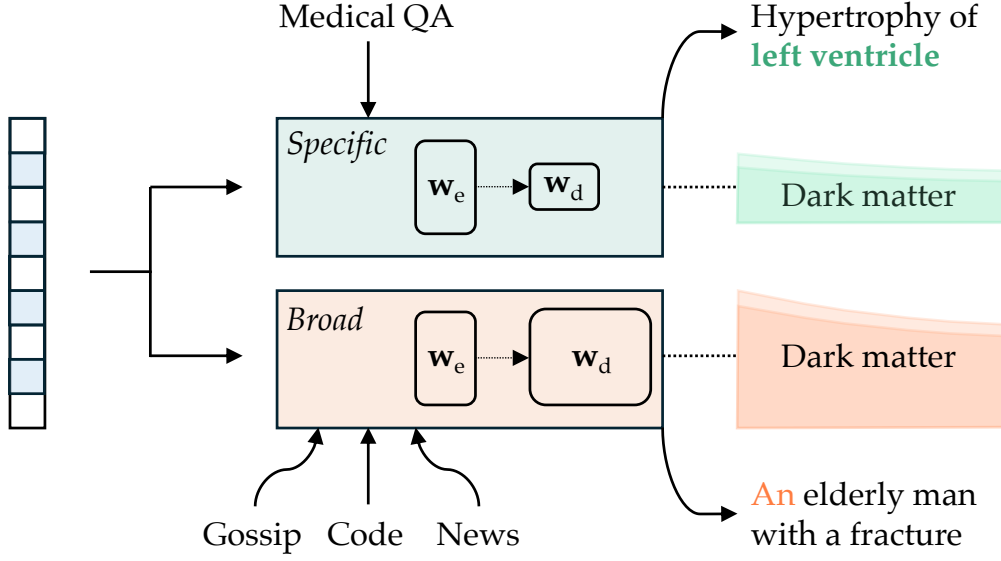


Figure 1 | Pareto curves of fraction of variance explained across different models (gemma-2-2b and gemma-2-9b from Team et al. (2024)), SAE widths, and sparsities, comparing our SAEs to the GemmaScope (Lieberum et al., 2024) SAEs.

Studies also demonstrate that SAEs can exhibit feature splitting, where a latent that should represent a single interpretable concept instead fragments into multiple, more specific latents. Moreover, feature absorption occurs when token-aligned latents “absorb” an expected feature direction, causing the intended latent to fail to activate in some contexts and reducing overall recall (Chanin et al., 2024). Till (2024) argues that the L1 regularisation used to enforce sparsity can drive the SAE to learn common combinations of features rather than the atomic “true features” that causally mediate the model’s computations, although this has been ameliorated somewhat with newer architectures and optimisation techniques (Gao et al., 2024; Rajamanoharan et al., 2024a,b). Recent work has also shown that SAEs trained on the same model and data, differing only in their random initialisation, learn substantially different feature sets (Paulo and Belrose, 2025), indicating that SAE decomposition is not unique but rather a pragmatic artifact of training conditions.

Collectively, these findings indicate that the conventional “more is better” scaling paradigm for language models does not effectively translate to mechanistic interpretability using SAEs. Broad-domain training produces latent representations that are generic, inconsistent, and vulnerable to issues such as nonlinear error and feature absorption. We hypothesise that narrowing the input domain could encourage the SAE’s fixed latent capacity to selectively represent only high-fidelity, task-relevant features. This reallocation of capacity is expected to reduce downstream substitution error and yield latent representations that more accurately reflect the causal circuitry of the target model (Chanin et al., 2024; Gao et al., 2024).

In this paper, we argue that mechanistic interpretability via SAEs requires a shift from broad-domain scaling to domain-specific training. To support this claim, we conduct a thorough study of SAEs applied to medical text data, and demonstrate empirically and theoretically the benefits of training domain-specific SAEs. Specifically, we train JumpReLU SAEs with the same capacity as various GemmaScope SAEs (Lieberum et al., 2024) and determine how training on a well-defined but narrow domain addresses many, if not all, of the concerns with SAEs outlined above through *unsupervised evaluation*. We then demonstrate that the features learned by our domain-specific SAEs are not only much more specific, but more *interpretable* than GemmaScope features pertaining to medicine. Finally,

we show how the minimal linear and nonlinear variance in our domain-specific SAEs leads us to conclude that reasonably sized domain-specific SAEs can truly learn all required features for said domain.

Taken together, this paper provides evidence that SAEs applied to specific domains are the most promising direction forward for the current paradigms of mechanistic interpretability.

## 2. Background

### 2.1. Sparse Autoencoder (SAE) Architectures

Our goal is to decompose a model’s activation  $\mathbf{x} \in \mathbb{R}^n$  into a sparse linear combination of learned feature directions. Intuitively, we express:

$$\mathbf{x} \approx \mathbf{x}_0 + \sum_{i=1}^M f_i(\mathbf{x}) \mathbf{d}_i, \quad (1)$$

where  $\mathbf{d}_i$  are  $M \gg n$  latent unit-norm feature directions and the coefficients  $f_i(\mathbf{x}) \geq 0$  denote the activation strength for each feature.

In a sparse autoencoder (SAE), the encoder and decoder are defined as:

$$\mathbf{f}(\mathbf{x}) := \sigma(\mathbf{W}_{\text{enc}} \mathbf{x} + \mathbf{b}_{\text{enc}}), \quad (2)$$

$$\hat{\mathbf{x}}(\mathbf{f}) := \mathbf{W}_{\text{dec}} \mathbf{f} + \mathbf{b}_{\text{dec}}, \quad (3)$$

where  $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^M$  is a sparse, non-negative vector, and the columns of  $\mathbf{W}_{\text{dec}}$ , denoted  $\mathbf{d}_i$ , form the dictionary. For convenience, we define the encoder’s pre-activations as:

$$\boldsymbol{\pi}(\mathbf{x}) := \mathbf{W}_{\text{enc}} \mathbf{x} + \mathbf{b}_{\text{enc}}. \quad (4)$$

While standard activations such as ReLU (Bricken et al., 2023; Templeton et al., 2024) and TopK (Gao et al., 2024) are commonly used, in our work we adopt the *JumpReLU* activation function (Erichson et al., 2019; Rajamanoharan et al., 2024b):

$$\text{JumpReLU}_{\theta}(z) := z H(z - \theta), \quad (5)$$

where  $H(z)$  is the Heaviside step function and  $\theta \in \mathbb{R}_+$  is a learnable threshold. The additional parameter  $\theta$  enables the network to decide whether a feature is active prior to estimating its magnitude.

### 2.2. Loss Functions for SAEs

A typical loss function for language model SAEs is formulated as:

$$\mathcal{L}(\mathbf{x}) := \underbrace{\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{f}(\mathbf{x}))\|_2^2}_{\mathcal{L}_{\text{reconstruct}}} + \underbrace{\lambda S(\mathbf{f}(\mathbf{x}))}_{\mathcal{L}_{\text{sparsity}}} + \mathcal{L}_{\text{aux}}, \quad (6)$$

where  $S$  is a sparsity penalty such as L1 and  $\lambda$  balances reconstruction fidelity and sparsity.

A JumpReLU SAE modifies the standard sparse autoencoder architecture (Equations 2 and 3) by replacing the usual activation function with a JumpReLU. In this model, the encoder is defined as:

$$\mathbf{f}(\mathbf{x}) := \text{JumpReLU}_{\theta}(\mathbf{W}_{\text{enc}} \mathbf{x} + \mathbf{b}_{\text{enc}}), \quad (7)$$

where  $\theta \in \mathbb{R}_+^M$  is a vector of positive thresholds, one per feature. Unlike the standard ReLU, this extra parameter  $\theta$  sets a minimum required value for each encoder preactivation to be considered active. Thus, the JumpReLU SAE clearly separates the decision of whether a feature is active from the estimation of its magnitude.

In JumpReLU SAEs, we employ an L0 sparsity penalty:

$$\mathcal{L}(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{f}(\mathbf{x}))\|_2^2 + \lambda \|\mathbf{f}(\mathbf{x})\|_0. \quad (8)$$

Since the L0 norm simply counts non-zero entries, it can be expressed via the Heaviside step function:

$$\|\mathbf{f}(\mathbf{x})\|_0 = \sum_{i=1}^M H(\pi_i(\mathbf{x}) - \theta_i), \quad (9)$$

where  $\pi_i(\mathbf{x})$  is the  $i^{\text{th}}$  component of  $\pi(\mathbf{x})$ .

### 2.3. Straight-Through Estimators (STEs)

The discontinuities inherent in the L0 penalty and the threshold parameter  $\theta$  result in zero gradients under standard backpropagation. To overcome this, we adopt *straight-through estimators* (STEs) that define pseudo-derivatives for the non-differentiable functions (Bengio et al., 2013).

Specifically, we define:

$$\frac{\partial}{\partial \theta} \text{JumpReLU}_{\theta}(z) := -\frac{\theta}{\varepsilon} K\left(\frac{z - \theta}{\varepsilon}\right), \quad (10)$$

$$\frac{\partial}{\partial \theta} H(z - \theta) := -\frac{1}{\varepsilon} K\left(\frac{z - \theta}{\varepsilon}\right), \quad (11)$$

where  $K$  is a kernel function (e.g. the rectangle function defined as  $\text{rect}(z) := H(z + 1/2) - H(z - 1/2)$ ) and  $\varepsilon > 0$  is a small bandwidth parameter.

These STEs allow gradient information to pass through the discontinuities. Importantly, they can be interpreted as yielding a kernel density estimation (KDE) for the true gradient (see Appendix B).

## 3. Methodology

In this section, we outline our experimental approach. We begin by describing the construction of our medical text dataset, then detail the training procedure for our SAEs, and finally introduce the evaluation metrics used to assess the quality of these SAEs compared to GemmaScope.

### 3.1. Dataset

Our first step is to create a domain-specific corpus that captures the essential features relevant to clinical tasks such as differential diagnosis. To achieve this, we combine multiple publicly available datasets, summarised in Table 1.

Each source dataset was standardised into a uniform format, preserving the question text, multiple-choice options, correct answers, and, when available, explanations and contextual details. For PubMedQA entries, we also include the relevant medical context and detailed answer explanations. All fields are concatenated into a single string per example. Preserving a uniform structure allows us

Dataset	Examples	Type	Description & Citation
MedQA	10,200	Multiple Choice	USMLE questions from professional board exams ( <a href="#">Jin et al., 2020</a> ).
MedMCQA	183,000	Multiple Choice	Questions from real-world medical entrance exams ( <a href="#">Pal et al., 2022</a> ).
MMLU College Medicine	173	Multiple Choice	College-level medical knowledge questions ( <a href="#">Hendrycks et al., 2021</a> ).
MMLU Clinical Knowledge	265	Multiple Choice	Questions evaluating clinical concepts and practices.
MMLU Professional Medicine	272	Multiple Choice	Questions assessing professional medical knowledge.
PubMedQA	450	Question Answering	Biomedical Q&A with research questions, abstracts (without conclusions), long answers, and a yes/no/maybe summary ( <a href="#">Jin et al., 2019</a> ).

Table 1 | Summary of the medical datasets used in our study.

to remove the need to learn a significant number of features that correspond simply to the structure of the text rather than its contents.

The final combined dataset comprises approximately 195,000 examples and roughly 50 million tokens (as determined by the Gemma-2 tokeniser ([Team et al., 2024](#))). The complete dataset is publicly available on HuggingFace as [irisai/medical-qa-combined](#).

With our unified dataset in hand, we now describe the training setup used to extract latent features from large language models.

### 3.2. Training

We trained Sparse Autoencoders (SAEs) on the post-MLP output residual stream (layer 20) of both Gemma-2-2b and Gemma-2-9b models. For each model-width combination, we employed a JumpReLU architecture with a bandwidth parameter of 0.001, trained using MSE reconstruction loss plus a quadratic penalty on deviations from the target sparsity level. Training was conducted using the Adam optimiser with a learning rate of  $7e-5$  and  $\beta$  parameters (0.0, 0.999), following the implementation in the original JumpReLU paper ([Rajamanoharan et al., 2024b](#)). We normalised all activation vectors to have unit mean squared norm before training, following the recommendations in [Cunningham et al. \(2023\)](#), and scaled the dictionary weights appropriately after training.

Our activation buffer implementation maintained approximately 30 000 contexts (each of length 1,024) of activations in memory at any given time, refreshing the buffer when it became half empty. The buffer collected activations by processing text in batches of 4 contexts and yielded training batches of 2,048 tokens. We used the `medical-qa-combined` dataset for training.

For Gemma-2-2b, we trained SAEs with dictionary sizes of  $2^{14}$  and  $2^{16}$  features. Target L0 sparsities (average number of active features per activation vector) were set to 20 for both dictionary sizes. For Gemma-2-9b, we explored a broader range of dictionary sizes:  $2^{14}$ ,  $2^{15}$ ,  $2^{16}$ ,  $2^{17}$ ,  $2^{18}$ ,  $2^{19}$ , and  $2^{20}$  features. Target L0 sparsities were scaled with dictionary width, ranging from 20 to approximately 650.

All models were trained for 49 million tokens with a linear learning rate warmup over the first 1,000 steps and sparsity warmup over the first 5,000 steps. We employed gradient clipping with a maximum norm of 1.0 to ensure training stability. The sparsity penalty coefficient was set to 1.0 throughout training, with the quadratic penalty term scaled by the ratio of actual to target sparsity. Training was conducted using mixed precision (bfloat16) for computational efficiency, with model weights stored in float32 precision.

### 3.3. Evaluation

After training our SAEs, we evaluate their performance through a series of quantitative and qualitative metrics.

**Unsupervised Metrics** We evaluate the quality of trained SAEs using three unsupervised metrics: the L0 sparsity, the fraction of variance explained, and the loss recovered. The L0 sparsity is defined as the expected number of active features per input:

$$\mathbb{E}_{\mathbf{x} \sim \mathcal{D}} [\|\mathbf{f}(\mathbf{x})\|_0]. \quad (12)$$

Reconstruction fidelity is measured by the fraction of variance explained:

$$1 - \frac{\text{Var}(\mathbf{x} - \hat{\mathbf{x}}(\mathbf{f}(\mathbf{x})))}{\text{Var}(\mathbf{x})}, \quad (13)$$

where  $\text{Var}(\mathbf{x})$  denotes the sum of variances across all dimensions. In addition, we assess reconstruction quality via the loss recovered metric. Let  $\text{CE}(\phi)$  be the average cross-entropy loss of the language model when a function  $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is spliced into the model at the SAE insertion point. If  $\hat{\mathbf{x}} \circ \mathbf{f}$  is the autoencoder function,  $\zeta : \mathbf{x} \mapsto \mathbf{0}$  the zero-ablation function, and  $\text{Id} : \mathbf{x} \mapsto \mathbf{x}$  the identity function, then the loss recovered is given by:

$$1 - \frac{\text{CE}(\hat{\mathbf{x}} \circ \mathbf{f}) - \text{CE}(\text{Id})}{\text{CE}(\zeta) - \text{CE}(\text{Id})}.$$

By definition, a SAE that always outputs the zero vector achieves a loss recovered of 0%, while perfect reconstruction yields 100%.

Baseline SAEs trained with an L1 sparsity penalty tend to underestimate feature activations, a phenomenon known as shrinkage. This bias leads to reconstructions whose norms are lower than those of the inputs. To quantify shrinkage, we define the relative reconstruction bias  $\gamma$  as the optimal multiplicative factor that minimises the L2 reconstruction loss:

$$\gamma := \arg \min_{\gamma'} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[ \left\| \frac{\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x})}{\gamma'} - \mathbf{x} \right\|_2^2 \right].$$

An unbiased SAE satisfies  $\gamma = 1$ , whereas  $\gamma < 1$  indicates shrinkage. The derivation of the analytical solution to  $\gamma$  in this metric is provided in Appendix C.

**Interpretability** Beyond raw reconstruction metrics, it is important to understand whether the latent features are meaningful. We evaluate interpretability by generating explanations for each feature and measuring their fidelity.

We evaluate feature interpretability using the method of Paulo et al. (2024), which involves two stages: (1) generating an interpretation for each feature and (2) assessing the fidelity of that interpretation to the network’s true behaviour. If a feature does not support a clear interpretation, the auto-interpretability pipeline will return low evaluation scores. These evaluations can indicate when an SAE produces generally low-quality feature decompositions.

*Generating interpretations.* For each feature, we uniformly sample examples from each activation decile to ensure that explanations are robust across both strong and weak activations. In every example, we mark the tokens with maximum activation using designated delimiters and report their

activation magnitudes. In addition, we compute the logit weights via the path expansion  $W_U W_D[f]$  (where  $W_U$  is the model unembedding matrix and  $W_D[f]$  is the decoder direction for feature  $f$ ). The top promoted tokens from this expansion capture the feature’s causal effects, thereby sharpening the resulting explanation—an approach equivalent to using a logit lens (Joseph Bloom, 2024).

*Assessing faithfulness.* We measure faithfulness by treating the interpretation as a classifier that predicts whether a feature will activate in a given context. A faithful interpretation should exhibit both high recall (capturing most activating text) and high precision (distinguishing between activating and non-activating text). We employ two methods.

The first is detection: we prompt a language model to determine if an entire sequence activates a given SAE latent based on its interpretation. By including both activating and non-activating contexts, this method evaluates precision and recall without requiring token-level localisation, and it leverages token probabilities to gauge classification confidence. The second is embedding. Here, the interpretation acts as a query to retrieve contexts where the feature is active. We embed both activating and non-activating contexts using an encoding transformer and use the similarity between the query and these contexts to classify them. The resulting classifier is evaluated using the AUROC.

Further details on our automated interpretability pipeline are provided in Appendix D.

**Dark Matter** Inspired by the framework of Engels et al. (2024), we further analyse the reconstruction error of our sparse autoencoders (SAEs) by quantifying linear predictability. For an input activation  $\mathbf{x}$ , the SAE produces a reconstruction  $\hat{\mathbf{x}} = \text{SAE}(\mathbf{x})$ , yielding an error defined as:

$$\text{SaeError}(\mathbf{x}) = \mathbf{x} - \text{SAE}(\mathbf{x}).$$

We hypothesise that a significant portion of this error arises from unlearned, linearly structured features – “dark matter”.

To probe this hypothesis, we adopt a three-fold approach. First, we assess the linear predictability of the error norm by learning an optimal scalar probe that maps  $\mathbf{x}$  to the squared error norm  $\|\text{SaeError}(\mathbf{x})\|_2^2$ . Formally, we solve for:

$$\mathbf{a}^* = \arg \min_{\mathbf{a} \in \mathbb{R}^{d+1}} \|\mathbf{a}^T \cdot [\mathbf{x}; 1] - \|\text{SaeError}(\mathbf{x})\|_2^2\|_2^2,$$

where the augmentation  $[\mathbf{x}; 1]$  accounts for a bias term, and the quality of the fit is measured by the coefficient of determination  $R^2$ .

Second, we predict the full error vector by seeking an optimal linear transformation that maps the activation  $\mathbf{x}$  to  $\text{SaeError}(\mathbf{x})$ . Specifically, we solve:

$$\mathbf{b}^* = \arg \min_{\mathbf{b} \in \mathbb{R}^{(d+1) \times d}} \|\mathbf{b} \cdot [\mathbf{x}; 1] - \text{SaeError}(\mathbf{x})\|_2^2.$$

The average  $R^2$  computed across activation dimensions quantifies the extent to which the residual error is linearly predictable. A higher  $R^2$  means the residual is more linearly structured – suggesting the SAE missed certain linear features that remain in the error.

Finally, we evaluate the nonlinear fraction of variance unexplained (FVU) by examining how well the combination of the SAE reconstruction and the linear error prediction accounts for the original activation. Defining:

$$\tilde{\mathbf{x}} = \text{SAE}(\mathbf{x}) + \mathbf{b}^* \cdot \mathbf{x},$$

we compute:

$$\text{FVU}_{\text{nonlinear}} = 1 - R^2(\mathbf{x}, \tilde{\mathbf{x}}).$$



This metric captures the remaining variance in the activations that is not explained by the sum of the SAE output and the optimal linear prediction of its error. A *larger*  $FVU_{\text{nonlinear}}$  indicates that even after accounting for a linear error term, more variance in  $x$  remains unexplained, i.e. the residual has a greater *nonlinear* component.

For all analyses, activations are extracted from transformer layer 20 after filtering to include only tokens beyond a fixed position within each context, thereby mitigating potential confounds due to token position (Lieberum et al., 2024).

## 4. Results

Here we present the results of our analysis comparing domain-specific SAEs with foundational SAEs, across reconstruction fidelity, interpretability and dark matter analysis.

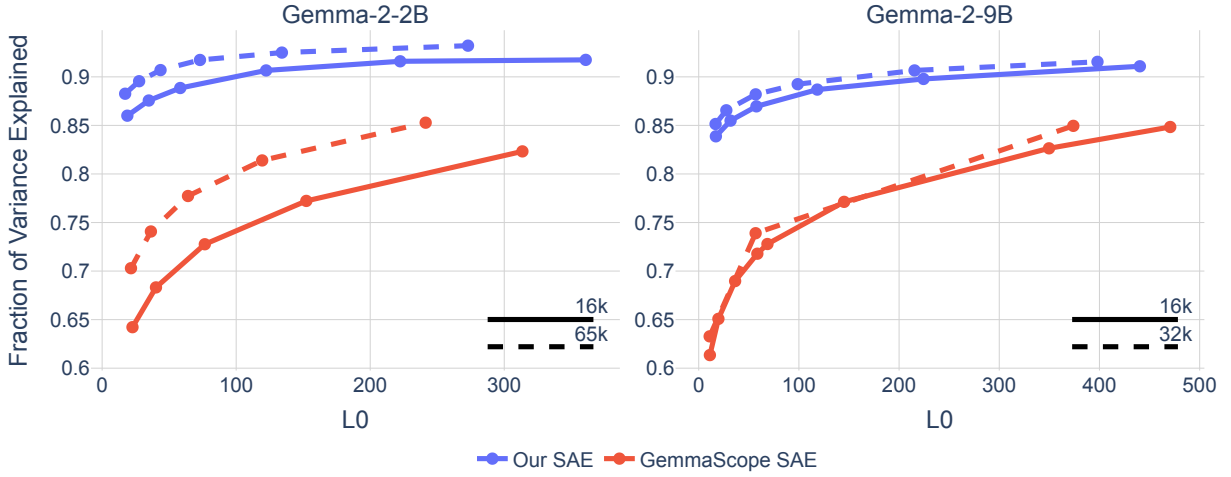


Figure 2 | Pareto curves of fraction of variance explained across different models (gemma-2-2b and gemma-2-9b from Team et al. (2024)), SAE widths, and sparsities, comparing our SAEs to the GemmaScope (Lieberum et al., 2024) SAEs.

### 4.1. Unsupervised Evaluations

The fraction of variance explained for our SAEs is consistently approximately 15-20% higher than the accompanying GemmaScope SE, across all L0s (Figure 2). Similarly, the loss recovered (compared to the original Gemma model) when substituting in our SAE activations is also consistently higher (Figure 3).

We show additional evaluation metrics in Appendix E. We find all SAEs, both ours and GemmaScope, to have close to perfect relative reconstruction bias, meaning there is minimal shrinkage (Appendix C, Figure 7). We also show improvements of our SAEs over GemmaScope in cosine similarity between the model activations and SAE reconstructions (Appendix F, Figure 6).

### 4.2. Interpretability

Some analysis and visualisation of feature similarities, both within SAEs and between ours and GemmaScope, are given in Appendix F.



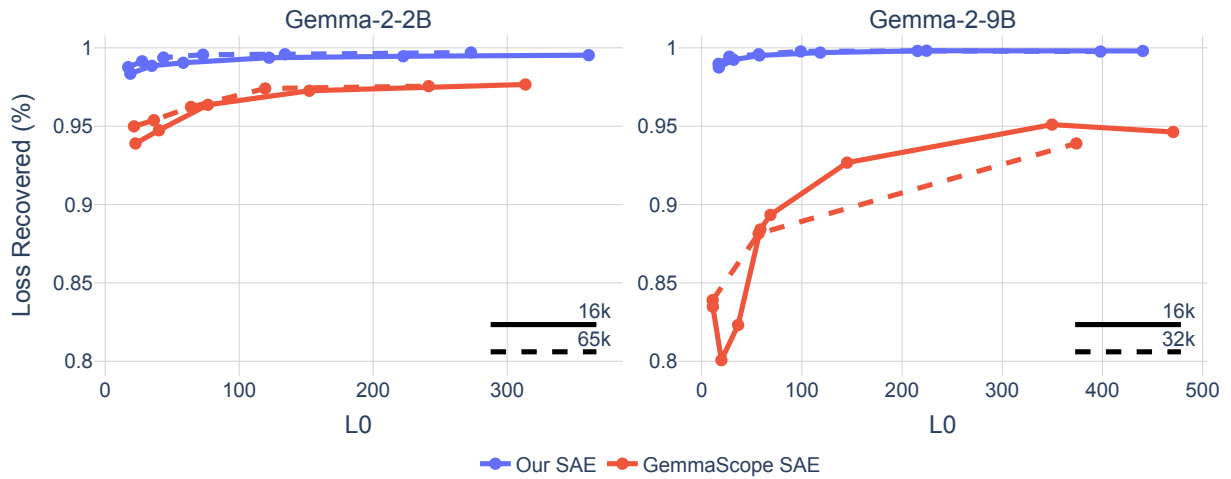


Figure 3 | Pareto curves of loss recovered when substituting SAE reconstructions into the model, across different models (gemma-2-2b and gemma-2-9b), SAE widths, and sparsities, comparing our SAEs to the GemmaScope SAEs.

Feature Explanation	F1 Score	Representative Examples
Taste Sensations	0.94	"smell, or <b>taste</b> of food, instead of the previous normal salivary secretion by the parotid gland" "Loss of <b>taste</b> sensation in anterior 2/3 of tongue due to chorda tympani involvement"
Specificity in Diagnostic Testing	0.94	"High <b>specificity</b> of the test ensured minimal false positives" "The initial laboratory test was notably <b>specific</b> in confirming the diagnosis"
Infectious Mononucleosis	1.00	"Positive Paul Bunnell test confirmed <b>mononucleosis</b> in the patient" "Detection of heterophile antibodies was indicative of <b>mononucleosis</b> "
Enzyme Catalase in Differentiating Bacterial Species	0.91	"Staphylococci were determined to be <b>catalase</b> positive using the standard test" "The <b>catalase</b> test differentiated between bacterial species based on enzyme activity"
Negation or Lack of Association	0.97	"Water remains <b>unchanged</b> in a burn patient, indicating no effect on fluid balance" "The drug did <b>not alter</b> the resting membrane potential of the cells"
Femur in Orthopedic Injuries	1.00	"An elderly woman sustained a fracture of the <b>femur</b> following a fall" "A fracture at the <b>femoral neck</b> was observed on X-ray imaging"
Atria and Ventricles in Cardiovascular Anatomy	0.93	"ECG revealed enlargement of the <b>right atrium</b> suggesting atrial dilation" "Echocardiography showed hypertrophy of the <b>left ventricle</b> consistent with chronic pressure overload"
Conjunctions/Prepositions for Exceptions/Inclusions	0.92	"All diagnostic criteria were met, <b>except</b> for one minor finding" "The patient improved <b>although</b> some laboratory values remained borderline"
Hair Cells in Auditory System	0.91	"Damage to the <b>hair cells</b> in the cochlea can lead to significant hearing loss" "The study focused on the function of the <b>outer hair cells</b> in auditory transduction"
Lapse Feature	0.97	"Imaging revealed a clear <b>prolapse</b> of the mitral valve, associated with Marfan syndrome" "A uterine <b>prolapse</b> was noted during the pelvic examination"

Table 2 | Clinically relevant interpretable features with their F1 scores and representative examples. Activating tokens are highlighted in green.

Feature Explanation	F1 Score	Representative Examples
Image and Imaging in Medical Contexts	0.913	“Periapical <b>image</b> reveals bone destruction similar to periodontal disease around the lateral incisor” “Acute leukemia: <b>image</b> shows blast cells, suggesting acute leukemia”
The term “often”	0.980	“Nausea, vomiting, and abdominal guarding are <b>often</b> seen in the patient” “Colicky abdominal pain is <b>often</b> accompanied by the passage of blood and mucus per rectum”
Frequent Use of “get” and Its Variations	0.990	“She notes that her symptoms <b>get</b> much worse when exposed to sunlight” “The fragments <b>get</b> nipped between the condyles of tibia and femur, preventing full extension”
“Overall” for Summarisation	0.913	“When one mole of O <sub>2</sub> binds, it causes a shift in the <b>overall</b> conformation of the protein” “Adenocarcinoma of the breast now has an <b>overall</b> 5-year survival rate of 60–70%”
Articles “a” and “an”	0.943	“ <b>A</b> left shift was noted in the complete blood count” “ <b>An</b> irregular mass protruding from the vaginal wall was observed on examination”
Use of “nothing,” “anything,” and “something”	0.913	“The patient reports that there is <b>nothing</b> abnormal on examination” “She insists that there is <b>something</b> wrong in her head despite normal tests”
Frequent References to “block,” “blocking,” and “obstruction”	0.936	“The electrical impulse is <b>blocked</b> , necessitating an accessory pacemaker” “A large bolus of air caused an <b>obstruction</b> in the right atrium and ventricle”
Frequent Use of “used” and Its Variations	0.990	“Penicillin 250 mg 12-hourly may be <b>used</b> if the patient is allergic to penicillin” “Contrast <b>used</b> for MRI provided enhanced visualization of the lesion”
Frequent References to Statistical Measures (Mean, Average, Age)	0.969	“The <b>average</b> age of onset was 60 years with a standard deviation of 5” “The <b>mean</b> corpuscular volume was calculated to be 90 $\mu\text{m}^3$ ”

Table 3 | Clinically relevant interpretable features for the GemmaScope 32k width SAE with their F1 scores and representative examples. Activating tokens are highlighted in green.

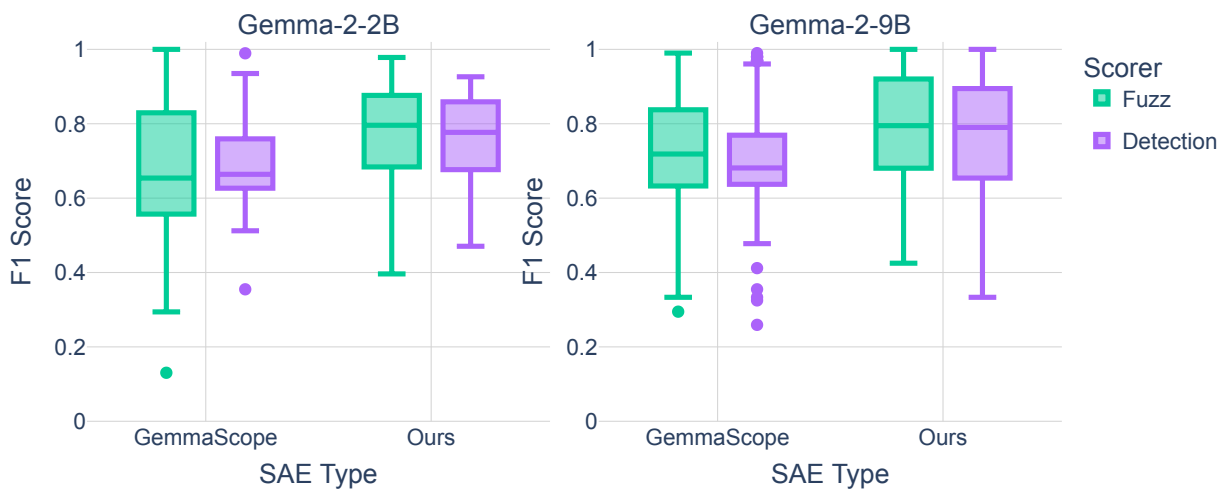


Figure 4 | F1 scores of detection and fuzzing evaluations used automated interpretability on our SAE and GemmaScope SAE, on Gemma-2-9B (width 65k latents) and Gemma-2-9B (32k latents).

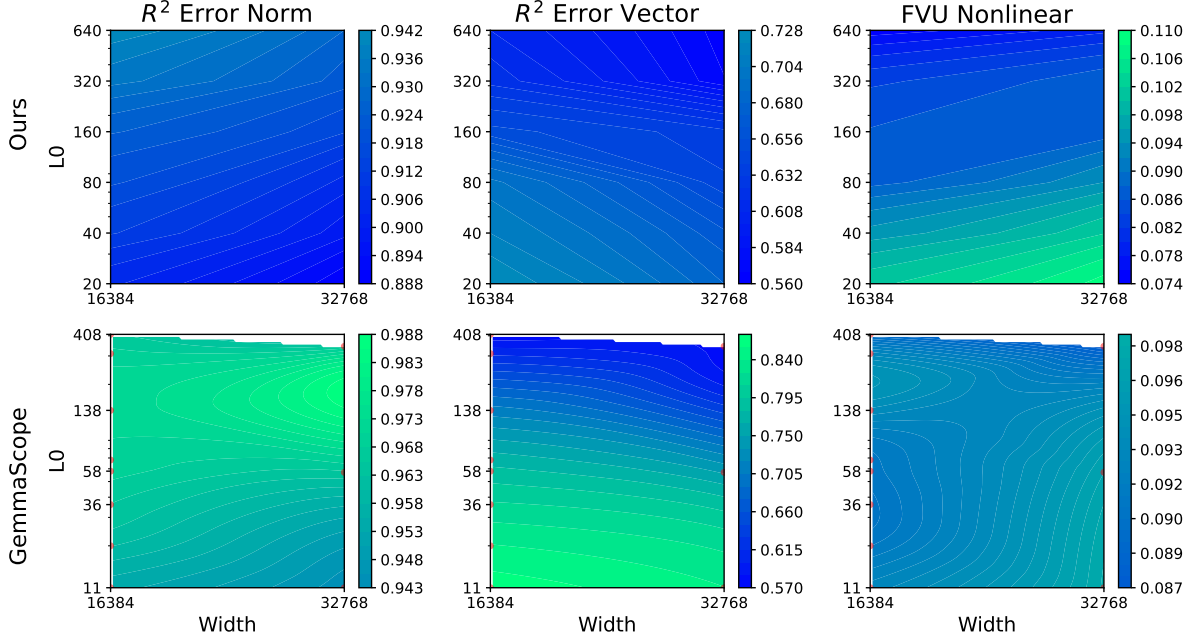


Figure 5 | **Dark matter analysis** of reconstruction error for domain-specific SAEs (top row) vs. GemmaScope SAEs (bottom row), measured at layer 20 of Gemma-2-9b. **Left:**  $R^2$  of a scalar probe predicting the error norm from  $\mathbf{x}$ . **Centre:**  $R^2$  of a linear map predicting the entire error vector. **Right:** Fraction of variance unexplained ( $FVU_{\text{nonlinear}}$ ) after combining the SAE reconstruction with the best linear approximation of its error.

### 4.3. Dark Matter

To gain insight into *what* the SAE fails to reconstruct, we follow the “dark matter” framework of Engels et al. (2024) and ask: *How much of the SAE’s residual error is itself just unlearned **linear** structure, and how much is fundamentally nonlinear?*

Across different widths and sparsity levels for Gemma-2-9b, we consistently observe that our **domain-specific** SAEs (top row of Figure 5) achieve *lower error norm predictability*, *lower linear predictability* of the error itself (i.e.  $R^2$  error vector is not substantially inflated), and *higher nonlinear fraction* ( $FVU_{\text{nonlinear}}$ ) in the residual.

In other words, once our domain-specific SAEs capture the main *linear* features relevant to the clinical text domain, the *remaining* error is smaller in magnitude but more purely nonlinear. By contrast, the GemmaScope SAEs (bottom row, Figure 5) tend to leave behind more linearly predictable “dark matter” – consistent with the interpretation that broad-domain SAEs underutilise their capacity for fine-grained domain features. Put differently, GemmaScope’s residual still contains substantial *linear* structure that could, in principle, have been represented in the SAE’s dictionary, whereas our domain-specific SAEs allocate capacity to those linear patterns and thus push most leftover error into a harder-to-capture (and smaller) nonlinear remainder.

## 5. Related Work

*Failure Modes of SAEs.* SAEs suffer from substantial reconstruction errors that degrade model performance. Inserting a 16M-latent SAE into GPT-4 resulted in a language modelling loss equivalent to a model trained on only 10% of GPT-4’s compute (Gao et al., 2024), while using SAE reconstructions in

GPT-2 small led to performance drops of 10% on task-specific data and 40% on general data (Makelov et al., 2024). Expanding dictionary size and sparsity can mitigate errors but introduces computational costs and compromises interpretability by creating near one-to-one latent mappings. Sparse dictionary learning (SDL) methods with “error nodes” (Marks et al., 2024) offer partial improvements, yet much of the reconstruction error remains linearly predictable, indicating unlearned structured features (Engels et al., 2024).

Despite their design, SAEs do not guarantee interpretable latents. Feature splitting, absorption, and compositional artifacts (Chanin et al., 2024; Till, 2024) suggest that excessive sparsity constraints can distort feature representations. Alternative optimisation objectives, such as minimising description length, may yield more meaningful decompositions (Ayonrinde et al., 2024). Furthermore, SAEs trained on pretraining data often fail to capture task-specific latents, missing key functional concepts needed for downstream applications. For instance, SAEs trained on pretraining corpora fail to encode features for refusal behaviour, while those trained on chat data do (Kissane et al., 2024). These issues highlight the broader limitations of SAEs in mechanistic interpretability, as outlined in Sharkey et al. (2025).

*Domain-Specific SAEs.* SAEs have been applied across diverse domains beyond LLMs, uncovering interpretable features in genomics, proteomics, and neuroscience. In Evo 2, Batch-TopK SAEs trained on layer-26 activations reveal biologically meaningful genomic features like exon–intron boundaries and transcription factor binding motifs (Brixi et al., 2024; Bussmann et al., 2024). Similarly, SAEs applied to ESM-2 embeddings extract latent dimensions corresponding to protein binding sites and structural motifs (Garcia and Ansuini, 2025; Simon and Zou, 2024). In AI safety, SAEs enable controlled knowledge removal by selectively downscaling activations of biology-related concepts (Farrell et al., 2024). Genomic applications extend to sparse convolutional denoising autoencoders for linkage disequilibrium detection (Chen and Shi, 2019) and stacked SAE architectures for cancer classification from gene expression data (Zenbout et al., 2020). SAEs also show promise in network anomaly detection (Mazadu et al., 2022), radiology diagnosis using vision transformers (Abdulaal et al., 2024), and scientific literature analysis with neural embedding models (O’Neill et al., 2024). In neuroscience, SAEs have been used to model neuronal responses in the visual cortex (Geadah et al., 2024), analyse neural recordings (Almuqhim and Saeed, 2021; Theodosis and Ba, 2023), and extract interpretable features from visual neurons (Klindt et al., 2023).

## 6. Discussion

In this work, we have demonstrated that domain — specific SAEs—trained exclusively on medical text — yield markedly improved performance compared to broad-domain, foundational SAEs. Our experiments show that these SAEs achieve 15–20% higher fraction of variance explained, superior loss recovery, and improved cosine similarity between activations and reconstructions. Furthermore, our interpretability analyses reveal that the latent features extracted by domain-specific SAEs align more closely with clinically meaningful concepts, while dark matter analysis indicates that our models capture the majority of linear structure, leaving a smaller and more nonlinear residual. Finally, our steering evaluations confirm that interventions based on these latent features lead to more precise and effective control over domain-specific model behaviour.

A detailed examination of our results reinforces these findings. In unsupervised evaluations, the enhanced reconstruction fidelity of domain-specific SAEs is evidenced by increased variance explained and loss recovery, which are critical for faithful mechanistic interpretability. The interpretability pipeline further shows that the latent features not only carry higher automated F1 scores but also encapsulate nuanced clinical phenomena (e.g., taste sensations, diagnostic specificity, and mononucle-

osis) more robustly than features from broad-domain models. Our dark matter analysis reveals that while foundational SAEs leave behind a significant fraction of linearly predictable error (Engels et al., 2024), our domain-specific models more effectively allocate capacity to capturing linear structure, relegating residual errors to a smaller, predominantly nonlinear component. In addition, steering experiments demonstrate that these refined latent features facilitate more targeted interventions, enhancing the model’s controllability—a promising sign for downstream applications in clinical settings.

Several failure modes in broad-domain SAEs further underscore the necessity of domain-specific training. Feature splitting and absorption, for example, are exacerbated in wider SAEs, where latents often fragment into overly specialised components or absorb token-aligned signals, reducing their interpretability (Chanin et al., 2024; Till, 2024). This is likely driven by the sparsity penalty, which, as argued by Anders et al. (2024), biases SAEs toward learning frequent feature compositions rather than atomic, semantically distinct latents. Additionally, attempts to mitigate reconstruction errors by increasing dictionary size—while effective in broad SAEs—lead to computationally expensive models that approach one-to-one mappings between latents and activations, undermining their usefulness for mechanistic interpretability (Gao et al., 2024; Makelov et al., 2024). Furthermore, broad SAEs trained on pretraining distributions frequently fail to capture task-critical concepts, such as refusal mechanisms in chat models (Kissane et al., 2024), highlighting the misalignment between general-purpose training and specialised interpretability needs.

As discussed in Sharkey et al. (2025), these issues are compounded by evaluation methodologies that rely on generic behavioural probes rather than assessing the meaningfulness of individual features in specific domains. For instance, SAE Bench evaluates feature absorption by using features for “word starts with x”, which is not useful for evaluating domain-specific feature absorption.

Domain-specific SAEs force the model to allocate its latent capacity to high-fidelity, task-relevant features, mitigating feature fragmentation, improving reconstruction fidelity, and enhancing interpretability. These results suggest that shifting from a foundation-model scaling paradigm to domain-constrained SAE training is crucial for achieving accurate and actionable decompositions of model activations.

## 6.1. Future Work

Looking ahead, several promising avenues remain for extending this research. One primary direction is to scale our approach by training SAEs on a vastly larger corpus of medical text—including bioRxiv papers, extensive clinical textbooks, and upwards of 2 billion medicine-specific tokens—to further enhance feature granularity and robustness. Additionally, exploring methodological extensions such as alternative optimisation targets (e.g., minimising description length rather than solely enforcing sparsity (Ayonrinde et al., 2024)) may yield even more interpretable latent decompositions. There is also significant potential in applying domain-specific SAE methods to other modalities or in the context of crosscoders, which could broaden the impact of our findings on mechanistic interpretability. Finally, further experiments are needed to assess the downstream benefits of these refined latent features — such as in the construction of causal subcircuits using sparse feature circuits (Marks et al., 2024) — to validate their utility in real-world applications.

## References

A. Abdulaal, H. Fry, N. Montaña-Brown, A. Ijishakin, J. Gao, S. Hyland, D. C. Alexander, and D. C. Castro. An x-ray is worth 15 features: Sparse autoencoders for interpretable radiology report

- generation, 2024. URL <https://arxiv.org/abs/2410.03334>.
- F. Almuqhim and F. Saeed. Asd-saenet: a sparse autoencoder, and deep-neural network model for detecting autism spectrum disorder (asd) using fmri data. *Frontiers in Computational Neuroscience*, 15:654315, 2021.
- E. Anders, C. Neo, J. Hoelscher-Obermaier, and J. N. Howard. Sparse autoencoders find composed features in small toy models. <https://www.lesswrong.com/posts/a5wwqza2cY3W7L9cj/sparse-autoencoders-find-composed-features-in-small-toy>, 2024.
- K. Ayonrinde, M. T. Pearce, and L. Sharkey. Interpretability as compression: Reconsidering sae explanations of neural activations with mdl-saes, 2024. URL <https://arxiv.org/abs/2410.11179>.
- Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.
- T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.
- G. Brix, M. G. Durrant, J. Ku, M. Poli, G. Brockman, D. Chang, G. A. Gonzalez, S. H. King, D. B. Li, A. T. Merchant, M. Naghipourfar, E. Nguyen, C. Ricci-Tam, D. W. Romero, G. Sun, A. Taghibakshi, A. Vorontsov, B. Yang, M. Deng, L. Gorton, N. Nguyen, N. K. Wang, E. Adams, S. A. Baccus, S. Dillmann, S. Ermon, D. Guo, R. Ilango, K. Janik, A. X. Lu, R. Mehta, M. R. Mofrad, M. Y. Ng, J. Pannu, C. Ré, J. C. Schmok, J. St. John, J. Sullivan, K. Zhu, G. Zynda, D. Balsam, P. Collison, A. B. Costa, T. Hernandez-Boussard, E. Ho, M.-Y. Liu, T. McGrath, K. Powell, D. P. Burke, H. Goodarzi, P. D. Hsu, and B. L. Hie. Genome modeling and design across all domains of life with evo 2, 2024. Preprint. Available at <https://arcinstitute.org/tools/evo/evo-designer>.
- B. Bussmann, P. Leask, and N. Nanda. Batchtopk sparse autoencoders, 2024. URL <https://arxiv.org/abs/2412.06410>.
- D. Chanin, J. Wilken-Smith, T. Dulka, H. Bhatnagar, and J. Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *arXiv preprint arXiv:2409.14507*, 2024.
- J. Chen and X. Shi. Sparse convolutional denoising autoencoders for genotype imputation. *Genes*, 10 (9):652, 2019.
- H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.
- J. Engels, L. Riggs, and M. Tegmark. Decomposing the dark matter of sparse autoencoders. *arXiv preprint arXiv:2410.14670*, 2024.
- N. B. Erichson, Z. Yao, and M. W. Mahoney. Jumprelu: A retrofit defense strategy for adversarial attacks. *arXiv preprint arXiv:1904.03750*, 2019.
- E. Farrell, Y.-T. Lau, and A. Conmy. Applying sparse autoencoders to unlearn knowledge in language models, 2024. URL <https://arxiv.org/abs/2410.19278>.
- L. Gao, T. D. la Tour, H. Tillman, G. Goh, R. Troll, A. Radford, I. Sutskever, J. Leike, and J. Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.



- E. N. V. Garcia and A. Ansuini. Interpreting and steering protein language models through sparse autoencoders. *arXiv preprint arXiv:2502.09135*, 2025.
- V. Geadah, G. Barello, D. Greenidge, A. S. Charles, and J. W. Pillow. Sparse-coding variational autoencoders. *Neural Computation*, 36(12):2571–2601, 2024.
- D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding, 2021. URL <https://arxiv.org/abs/2009.03300>.
- D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams, 2020. URL <https://arxiv.org/abs/2009.13081>.
- Q. Jin, B. Dhingra, Z. Liu, W. W. Cohen, and X. Lu. Pubmedqa: A dataset for biomedical research question answering, 2019. URL <https://arxiv.org/abs/1909.06146>.
- J. L. Joseph Bloom. Understanding sae features with the logit lens. <https://www.lesswrong.com/posts/qykrYY6rXXM7EEs8Q/understanding-sae-features-with-the-logit-lens>, 2024.
- C. Kissane, R. Krzyzanowski, N. Nanda, and A. 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>.
- D. Klindt, S. Sanborn, F. Acosta, F. Poitevin, and N. Miolane. Identifying interpretable visual features in artificial and biological neural systems, 2023. URL <https://arxiv.org/abs/2310.11431>.
- T. Lieberum, S. Rajamanoharan, A. Conmy, L. Smith, N. Sonnerat, V. Varma, J. Kramár, A. Dragan, R. Shah, and N. Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*, 2024.
- A. Makelov, G. Lange, and N. Nanda. Towards principled evaluations of sparse autoencoders for interpretability and control, 2024. URL <https://arxiv.org/abs/2405.08366>.
- S. Marks, C. Rager, E. J. Michaud, Y. Belinkov, D. Bau, and A. Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024.
- I. J. Mazadu, J. J. Abayomi, E. Agu, A. Oloyade, G. Aimufua, and O. J. Egwom. An improved deep sparse autoencoder driven network intrusion detection system (idsae-nids). *Available at SSRN* 4331553, 2022.
- C. O’Neill, C. Ye, K. Iyer, and J. F. Wu. Disentangling dense embeddings with sparse autoencoders, 2024. URL <https://arxiv.org/abs/2408.00657>.
- C. O’Neill, A. Gumran, and D. Klindt. Compute optimal inference and provable amortisation gap in sparse autoencoders, 2025. URL <https://arxiv.org/abs/2411.13117>.
- A. Pal, L. K. Umapathi, and M. Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR, 2022.
- G. Paulo and N. Belrose. Sparse autoencoders trained on the same data learn different features. *arXiv preprint arXiv:2501.16615*, 2025.



- G. Paulo, A. Mallen, C. Juang, and N. Belrose. Automatically interpreting millions of features in large language models. *arXiv preprint arXiv:2410.13928*, 2024.
- S. Rajamanoharan, A. Conmy, L. Smith, T. Lieberum, V. Varma, J. Kramar, R. Shah, and N. Nanda. Improving sparse decomposition of language model activations with gated sparse autoencoders. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.
- S. Rajamanoharan, T. Lieberum, N. Sonnerat, A. Conmy, V. Varma, J. Kramár, and N. Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024b.
- L. Sharkey, B. Chughtai, J. Batson, J. Lindsey, J. Wu, L. Bushnaq, N. Goldowsky-Dill, S. Heimersheim, A. Ortega, J. Bloom, et al. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*, 2025.
- E. Simon and J. Zou. Interplm: Discovering interpretable features in protein language models via sparse autoencoders. *bioRxiv*, pages 2024–11, 2024.
- G. Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé, J. Ferret, P. Liu, P. Tafti, A. Friesen, M. Casbon, S. Ramos, R. Kumar, C. L. Lan, S. Jerome, A. Tsitsulin, N. Vieillard, P. Stanczyk, S. Girgin, N. Momchev, M. Hoffman, S. Thakoor, J.-B. Grill, B. Neyshabur, O. Bachem, A. Walton, A. Severyn, A. Parrish, A. Ahmad, A. Hutchison, A. Abdagic, A. Carl, A. Shen, A. Brock, A. Coenen, A. Laforge, A. Paterson, B. Bastian, B. Piot, B. Wu, B. Royal, C. Chen, C. Kumar, C. Perry, C. Welty, C. A. Choquette-Choo, D. Sinopalnikov, D. Weinberger, D. Vijaykumar, D. Rogozińska, D. Herbison, E. Bandy, E. Wang, E. Noland, E. Moreira, E. Senter, E. Eltyshv, F. Visin, G. Rasskin, G. Wei, G. Cameron, G. Martins, H. Hashemi, H. Klimczak-Plucińska, H. Batra, H. Dhand, I. Nardini, J. Mein, J. Zhou, J. Svensson, J. Stanway, J. Chan, J. P. Zhou, J. Carrasqueira, J. Iljazi, J. Becker, J. Fernandez, J. van Amersfoort, J. Gordon, J. Lipschultz, J. Newlan, J. yeong Ji, K. Mohamed, K. Badola, K. Black, K. Millican, K. McDonell, K. Nguyen, K. Sodhia, K. Greene, L. L. Sjoesund, L. Usui, L. Sifre, L. Heuermann, L. Lago, L. McNealus, L. B. Soares, L. Kilpatrick, L. Dixon, L. Martins, M. Reid, M. Singh, M. Iverson, M. Görner, M. Velloso, M. Wirth, M. Davidow, M. Miller, M. Rahtz, M. Watson, M. Risdal, M. Kazemi, M. Moynihan, M. Zhang, M. Kahng, M. Park, M. Rahman, M. Khatwani, N. Dao, N. Bardoliwalla, N. Devanathan, N. Dumai, N. Chauhan, O. Wahltinez, P. Botarda, P. Barnes, P. Barham, P. Michel, P. Jin, P. Georgiev, P. Culliton, P. Kuppala, R. Comanescu, R. Merhej, R. Jana, R. A. Rokni, R. Agarwal, R. Mullins, S. Saadat, S. M. Carthy, S. Cogan, S. Perrin, S. M. R. Arnold, S. Krause, S. Dai, S. Garg, S. Sheth, S. Ronstrom, S. Chan, T. Jordan, T. Yu, T. Eccles, T. Hennigan, T. Kocisky, T. Doshi, V. Jain, V. Yadav, V. Meshram, V. Dharmadhikari, W. Barkley, W. Wei, W. Ye, W. Han, W. Kwon, X. Xu, Z. Shen, Z. Gong, Z. Wei, V. Cotruta, P. Kirk, A. Rao, M. Giang, L. Peran, T. Warkentin, E. Collins, J. Barral, Z. Ghahramani, R. Hadsell, D. Sculley, J. Banks, A. Dragan, S. Petrov, O. Vinyals, J. Dean, D. Hassabis, K. Kavukcuoglu, C. Farabet, E. Buchatskaya, S. Borgeaud, N. Fiedel, A. Joulin, K. Kenealy, R. Dadashi, and A. Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL <https://arxiv.org/abs/2408.00118>.
- A. Templeton, T. Conerly, J. Marcus, J. Lindsey, B. C. Trenton Bricken, A. Pearce, E. A. Craig Citro, A. Jones, H. Cunningham, N. L. Turner, C. McDougall, M. MacDiarmid, A. Tamkin, T. H. Esin Durmus, F. Mosconi, D. Freeman, T. R. Sumers, E. Rees, J. Batson, S. C. Adam Jermyn, C. Olah, and T. Henighan. Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. <https://transformer-circuits.pub/2024/scaling-monosemanticity/>, 2024. [Accessed 18-02-2025].

- E. Theodosios and D. Ba. Learning silhouettes with group sparse autoencoders. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.
- D. Till. Do sparse autoencoders find "true features", 2024. URL <https://www.lesswrong.com/posts/QoR8noAB3Mp2KBA4B/do-sparse-autoencoders-find-true-features>.
- I. Zenboud, A. Bouramoul, and S. Meshoul. Stacked sparse autoencoder for unsupervised features learning in pancancer mirna cancer classification. *CEUR Workshop Proceedings*, 2020.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Sparse Autoencoder (SAE) Architectures . . . . .	3
2.2	Loss Functions for SAEs . . . . .	3
2.3	Straight-Through Estimators (STEs) . . . . .	4
<b>3</b>	<b>Methodology</b>	<b>4</b>
3.1	Dataset . . . . .	4
3.2	Training . . . . .	5
3.3	Evaluation . . . . .	6
<b>4</b>	<b>Results</b>	<b>8</b>
4.1	Unsupervised Evaluations . . . . .	8
4.2	Interpretability . . . . .	8
4.3	Dark Matter . . . . .	11
<b>5</b>	<b>Related Work</b>	<b>11</b>
<b>6</b>	<b>Discussion</b>	<b>12</b>
6.1	Future Work . . . . .	13
<b>A</b>	<b>Derivation of the Expected Loss Derivative</b>	<b>18</b>
<b>B</b>	<b>STE Pseudo-Derivatives and the KDE Connection</b>	<b>18</b>
B.1	Probability-Based Intuition . . . . .	19
<b>C</b>	<b>Derivation of the Relative Reconstruction Bias Metric</b>	<b>19</b>

<b>D Automated Interpretability Details</b>	<b>21</b>
<b>E Additional SAE Evaluations</b>	<b>21</b>
<b>F Feature Similarity Analysis</b>	<b>21</b>

## A. Derivation of the Expected Loss Derivative

Here we derive the gradient of the expected loss with respect to the threshold parameter  $\theta_i$ . Starting from

$$\mathcal{L}_\theta(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{f}(\mathbf{x}))\|_2^2 + \lambda \sum_{i=1}^M H(\pi_i(\mathbf{x}) - \theta_i),$$

one may use the Leibniz integral rule and properties of the Dirac delta function to show that

$$\frac{\partial \mathbb{E}_{\mathbf{x}}[\mathcal{L}_\theta(\mathbf{x})]}{\partial \theta_i} = \left( \mathbb{E}_{\mathbf{x}}[I_i(\mathbf{x}) \mid \pi_i(\mathbf{x}) = \theta_i] - \lambda \right) p_i(\theta_i), \quad (14)$$

where

$$I_i(\mathbf{x}) := 2\theta_i \mathbf{d}_i \cdot (\mathbf{x} - \hat{\mathbf{x}}(\mathbf{f}(\mathbf{x}))),$$

and  $p_i(\theta_i)$  is the probability density of the pre-activation  $\pi_i(\mathbf{x})$  at  $\theta_i$ .

## B. STE Pseudo-Derivatives and the KDE Connection

This appendix explains how the pseudo-derivatives defined in Equations 10 and 11 naturally yield a kernel density estimator (KDE) for the gradient of the expected loss with respect to the threshold parameter  $\theta_i$ .

Recall that when a function is discontinuous (as in the case of the Heaviside step function), standard backpropagation fails to provide meaningful gradient information. To address this, we replace the true derivative with a pseudo-derivative defined via a kernel function  $K$ . In particular, the pseudo-derivative for the JumpReLU function is given by:

$$\frac{\partial}{\partial \theta} \text{JumpReLU}_\theta(z) \approx -\frac{\theta}{\varepsilon} K\left(\frac{z - \theta}{\varepsilon}\right),$$

and similarly for the Heaviside function,

$$\frac{\partial}{\partial \theta} H(z - \theta) \approx -\frac{1}{\varepsilon} K\left(\frac{z - \theta}{\varepsilon}\right),$$

where  $\varepsilon > 0$  is a small bandwidth parameter.

The kernel  $K$  serves as a smooth approximation to the Dirac delta function. Its role is to “smear” the discontinuity over a small interval, allowing gradients to propagate through points where the original function is not differentiable.

In kernel density estimation, given samples  $x_1, \dots, x_N$ , the density at a point  $x$  is estimated as:

$$\hat{p}_X(x) = \frac{1}{N\varepsilon} \sum_{\alpha=1}^N K\left(\frac{x - x_\alpha}{\varepsilon}\right). \quad (15)$$

In our setting, the gradient of the expected loss with respect to  $\theta_i$  involves contributions from the reconstruction error and the sparsity penalty. For each sample  $\mathbf{x}_\alpha$ , these contributions are modulated by the kernel  $K\left(\frac{\pi_i(\mathbf{x}_\alpha) - \theta_i}{\varepsilon}\right)$ , which localises the gradient estimation around the threshold  $\theta_i$ . Concretely, the batch-wise gradient approximation is given by:

$$\frac{1}{N\varepsilon} \sum_{\alpha=1}^N \left[ I_i(\mathbf{x}_\alpha) - \lambda \right] K\left(\frac{\pi_i(\mathbf{x}_\alpha) - \theta_i}{\varepsilon}\right), \quad (16)$$

where

$$I_i(\mathbf{x}_\alpha) := 2\theta_i \mathbf{d}_i \cdot (\mathbf{x}_\alpha - \hat{\mathbf{x}}(\mathbf{f}(\mathbf{x}_\alpha)))$$

captures the contribution from the reconstruction loss, and  $\lambda$  scales the contribution from the sparsity penalty.

This expression is analogous to the KDE estimator in Equation 15, but with each sample weighted by the term  $[I_i(\mathbf{x}_\alpha) - \lambda]$ . In other words, the STE-based gradient calculation performs a local averaging (or smoothing) over the mini-batch, effectively estimating the gradient by aggregating contributions from samples whose pre-activations  $\pi_i(\mathbf{x}_\alpha)$  lie near the threshold  $\theta_i$ .

Alternative kernel choices (such as triangular, Gaussian, or Epanechnikov kernels) can be used in place of the one presented here, and they all yield a similar smoothing effect in the gradient estimation. This connection to KDE provides an intuitive interpretation of the STE: it estimates the gradient of the expected loss by effectively performing a density-weighted average of the sample-wise contributions near the point of interest.

### B.1. Probability-Based Intuition

Another perspective is to view the activations as random variables. The expected L0 penalty is given by:

$$\mathbb{E}_{\mathbf{x}} \|\mathbf{f}(\mathbf{x})\|_0 = \sum_{i=1}^M \mathbb{P}(\pi_i(\mathbf{x}) > \theta_i),$$

which is differentiable with respect to  $\theta_i$ . In fact,

$$\frac{d}{d\theta_i} \mathbb{P}(\pi_i(\mathbf{x}) > \theta_i) = -p_i(\theta_i).$$

Thus, the STE approximates this derivative by replacing the Dirac delta function with a smoothed kernel  $K$ . This probabilistic interpretation underlies the KDE connection discussed in Appendix B.

## C. Derivation of the Relative Reconstruction Bias Metric

We aim to determine the optimal multiplicative factor

$$\gamma := \arg \min_{\gamma'} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[ \left\| \frac{\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x})}{\gamma'} - \mathbf{x} \right\|_2^2 \right],$$

where  $\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x})$  is the SAE's reconstruction of  $\mathbf{x}$ . Define the error function:

$$E(\gamma') = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[ \left\| \frac{\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x})}{\gamma'} - \mathbf{x} \right\|_2^2 \right].$$

For each  $\mathbf{x}$ , expanding the squared L2 norm gives:

$$\left\| \frac{\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x})}{\gamma'} - \mathbf{x} \right\|_2^2 = \frac{1}{\gamma'^2} \|\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x})\|_2^2 - \frac{2}{\gamma'} \hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x}) \cdot \mathbf{x} + \|\mathbf{x}\|_2^2.$$

Define

$$A = \mathbb{E} \left[ \|\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x})\|_2^2 \right], \quad B = \mathbb{E} \left[ \hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x}) \cdot \mathbf{x} \right], \quad C = \mathbb{E} \left[ \|\mathbf{x}\|_2^2 \right].$$

Thus,

$$E(\gamma') = \frac{A}{\gamma'^2} - \frac{2B}{\gamma'} + C.$$

Minimising  $E(\gamma')$  with respect to  $\gamma'$ , we differentiate:

$$\frac{dE}{d\gamma'} = -\frac{2A}{\gamma'^3} + \frac{2B}{\gamma'^2} = 0.$$

Multiplying by  $\gamma'^3/2$  (which is positive) yields:

$$-A + B\gamma' = 0 \quad \implies \quad \gamma' = \frac{A}{B}.$$

Hence, the optimal scaling factor is:

$$\gamma = \frac{A}{B} = \frac{\mathbb{E} \left[ \|\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x})\|_2^2 \right]}{\mathbb{E} \left[ \hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x}) \cdot \mathbf{x} \right]}.$$

We now express  $B$  in terms of the mean squared reconstruction error. Noting that

$$\|\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x}) - \mathbf{x}\|_2^2 = \|\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x})\|_2^2 + \|\mathbf{x}\|_2^2 - 2\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x}) \cdot \mathbf{x},$$

taking expectations gives:

$$D = \mathbb{E} \left[ \|\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x}) - \mathbf{x}\|_2^2 \right] = A + C - 2B.$$

Solving for  $B$ :

$$B = \frac{A + C - D}{2}.$$

Substituting back, we obtain:

$$\gamma = \frac{A}{\frac{A+C-D}{2}} = \frac{2A}{A+C-D}.$$

Thus, the relative reconstruction bias is given by

$$\gamma = \frac{2 \mathbb{E} \left[ \|\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x})\|_2^2 \right]}{\mathbb{E} \left[ \|\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x})\|_2^2 \right] + \mathbb{E} \left[ \|\mathbf{x}\|_2^2 \right] - \mathbb{E} \left[ \|\hat{\mathbf{x}}_{\text{SAE}}(\mathbf{x}) - \mathbf{x}\|_2^2 \right]}.$$

**Additional Remarks:** The derivation uses the identity

$$2 \mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 - \|\mathbf{a} - \mathbf{b}\|_2^2.$$

An unbiased reconstruction (perfect SAE) yields  $\gamma = 1$ ; however, in practice, L1 regularization may induce shrinkage ( $\gamma < 1$ ). This metric thus quantifies the extent to which the SAE's decoder underestimates the input norm.

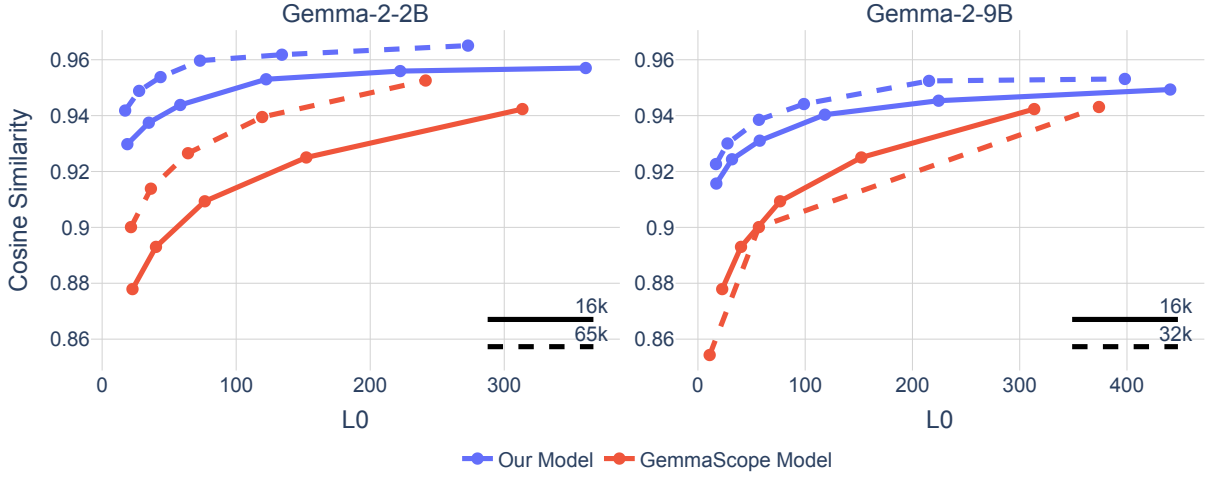


Figure 6 | Pareto curves of cosine similarity, between the model activations and SAE reconstructions, across different models (gemma-2-2b and gemma-2-9b), SAE widths, and sparsities, comparing our SAEs to the GemmaScope SAEs.

## D. Automated Intepretability Details

In our evaluation, we employ the unified `charlieoneill/medical-qa-combined` dataset, processing a total of 49 million tokens. Each activation context is constrained to 256 tokens, and the activations are processed in batches of 8 examples. For each latent feature, our sampling procedure ensures that only those with at least 250 examples are considered, while we cap the number of examples per latent at 5 000. During training, we sample 40 examples per latent using a quantile-based method, and for evaluation, 50 test examples per latent are selected. In addition, non-activating examples are randomly sampled from contexts that do not trigger the latent, which provides a balanced set of inputs for our detection pipeline.

To limit the computational load, our evaluation is restricted to a maximum of 1 000 latent features. The LLM explainer is configured to use OpenAI’s `gpt-4o-mini` model, with the maximum context length set to 4208 tokens. This setting ensures that the explainer has sufficient context to generate high-quality explanations without exceeding model limitations.

The entire evaluation pipeline is implemented asynchronously to manage the large-scale data processing efficiently. This asynchronous design spans activation caching, latent example construction, and both detection and fuzzing scoring.

## E. Additional SAE Evaluations

## F. Feature Similarity Analysis

We applied the Hungarian algorithm to match each feature vector from Gemma’s decoder and encoder to the nearest feature vector in our SAE based on cosine similarity (O’Neill et al., 2025; Paulo and Belrose, 2025). Each point in Figure 8 corresponds to one GemmaScope feature, with the x-axis showing the cosine similarity in the decoder space and the y-axis showing the cosine similarity in the encoder space. Points are coloured according to whether both matches (decoder and encoder) mapped to the same index in our SAE. Most points with high decoder similarity also have high encoder similarity, suggesting that these features align consistently in both representations. A smaller

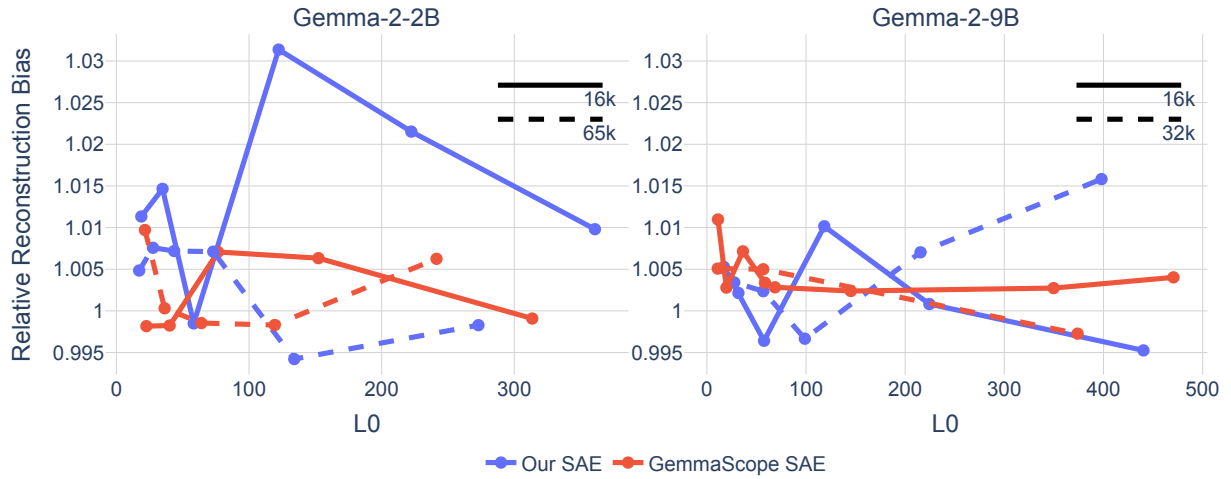


Figure 7 | Pareto curves of relative reconstruction bias across different models (gemma-2-2b and gemma-2-9b), SAE widths, and sparsities, comparing our SAEs to the GemmaScope SAEs.

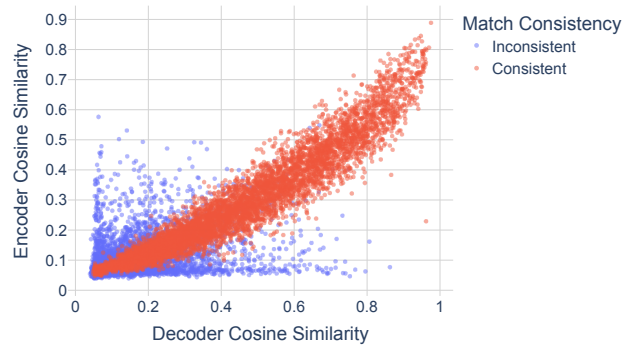


Figure 8 | Matched feature similarities for Gemma and our SAE decoders and encoders. Each point represents a Gemma feature vector, with its matched cosine similarity in decoder space on the x-axis and encoder space on the y-axis. Points are coloured by whether the matched index in our SAE is consistent for both decoder and encoder.

group of inconsistent matches indicates features that do not map to the same index across the two spaces.



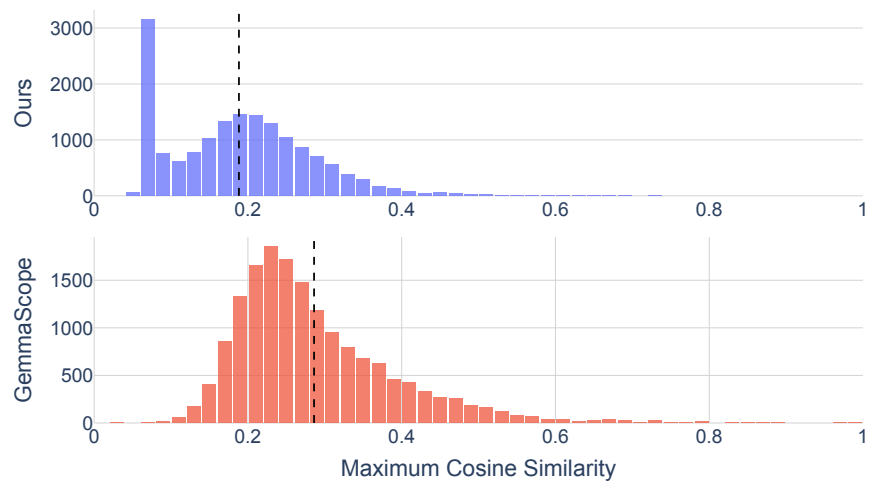


Figure 9 | Distribution of maximum cosine similarities for the learned SAE features