Decoding Dark Matter: Specialized Sparse Autoencoders for Interpreting Rare Concepts in Foundation Models

Aashiq Muhamed[1], Mona Diab[1], Virginia Smith[2]
{amuhamed, mdiab, smithv}@andrew.cmu.edu
[1] Language Technologies Institute, [2] Machine Learning Department
Carnegie Mellon University

Abstract

Understanding and mitigating the potential risks associated with foundation models (FMs) hinges on developing effective interpretability methods. Sparse Autoencoders (SAEs) have emerged as a promising tool for disentangling FM representations, but they struggle to capture rare, yet crucial concepts in the data. We introduce Specialized Sparse Autoencoders (SSAEs), designed to illuminate these elusive *dark matter* features by focusing on specific subdomains. We present a practical recipe for training SSAEs, demonstrating the efficacy of dense retrieval for data selection and the benefits of Tilted Empirical Risk Minimization as a training objective to improve concept recall. Our evaluation of SSAEs on standard metrics, such as downstream perplexity and L_0 sparsity, show that they effectively capture subdomain tail concepts, exceeding the capabilities of general-purpose SAEs. We showcase the practical utility of SSAEs in a case study on the Bias in Bios dataset, where SSAEs achieve a 12.5% increase in worst-group classification accuracy when applied to remove spurious gender information. SSAEs provide a powerful new lens for peering into the inner workings of FMs in subdomains.

1 Introduction

Interpretability is crucial for ensuring the safety and reliability of foundation models (FMs) (Bommasani et al., 2021). A key challenge in interpretability research is to scalably explain the myriad unanticipated behaviors in FMs. Sparse Autoencoders (SAEs) have recently emerged as a promising tool for disentangling the complex, high-dimensional representations within FMs into meaningful, human-interpretable features without supervision (Cunningham et al., 2023; Gao et al., 2024; Braun et al., 2024; Bricken et al., 2023). However, even massively wide SAEs, trained on vast amounts of data, may only capture a fraction of the concepts embedded within these models (Templeton et al., 2024). A significant portion of rare or highly specific concepts remain essentially invisible due to their infrequent activation. These elusive features, akin to *dark matter* in the universe of interpretability, pose a significant challenge for understanding and mitigating potential risks associated with FMs. While larger SAEs did exhibit some features for rarer concepts, Templeton et al. (2024) found compelling evidence suggesting a vast amount of *dark matter* features were still being missed. For example, they found features for some of San Francisco's neighborhoods, but their model still lacked features for smaller entities like coffee shops or street intersections. They observed that if a concept is present only once every billion tokens, we may need a billion-feature SAE to capture it reliably. This raises a critical question: can we develop more efficient methods than simply scaling SAE width to capture the tail concepts we are interested in?

This paper introduces Specialized Sparse Autoencoders (SSAEs), a novel approach designed to address this challenge. Instead of aiming to capture all concepts, as in current SAE practices, we propose SSAEs as an unsupervised targeted method for efficiently extracting rare features related to specific subdomains. By focusing on a particular subdomain, we can train SSAEs to learn features representing tail concepts without needing to scale to billions of features. Furthermore, instead of relying solely on scaling, we investigate whether Tilted Empirical Risk Minimization (TERM), which approximates minimax risk at large tilt parameters, can further improve the representation of tail concepts within SSAEs. Our key contributions are:

1. **Specialized Sparse Autoencoders:** An unsupervised method for efficiently extracting rare, subdomain-specific features. We demonstrate empirically that SSAEs capture a greater proportion of tail concepts than standard SAEs trained

Findings of the Association for Computational Linguistics: NAACL 2025, pages 1604–1635
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

on general-purpose data, achieving a 12.5% increase in worst-group classification accuracy on the Bias in Bios dataset when used to remove spurious gender information.

2. **Subdomain Data Selection Strategies:** A practical recipe for training SSAEs, starting with a small seed dataset and leveraging various data selection strategies to identify relevant training data from the FM's pretraining corpus. We find that Dense retrieval is particularly effective while TracIn reranking can offer further improvements.
3. **Tilted Empirical Risk Minimization for SAEs:** A novel training objective for SAEs designed to improve concept recall. At large tilt values, TERM encourages more balanced learning of head and tail concepts. We show that TERM-trained SSAEs are more interpretable, exhibit improved concept detection, while maintaining comparable downstream perplexity.

We envision SSAEs as versatile tools for concept detection and control across domains where identifying rare features is crucial, such as AI safety (detecting deception), healthcare (identifying outliers), and fairness (recognizing underrepresented groups). See Appendix M for additional examples.

Related Work Much interpretability research focuses on analyzing coarse-grained model components like induction heads and MLP modules (Olsson et al., 2022; Elhage et al., 2022b; Geva et al., 2023; Meng et al., 2022; Nanda et al., 2023b), or fine-grained units like linear probes (Kim et al., 2018; Belinkov, 2022; Geiger et al., 2023; Zou et al., 2023). Both have limitations. The inherent polysemanticity of coarse-grained components complicates interpretation. Fine-grained analysis, while potentially more precise, is constrained by reliance on curated datasets that isolate behavior, limiting generalizability to unknown mechanisms. Feature disentanglement methods, such as SAEs (Bricken et al., 2023; Cunningham et al., 2023), offer a promising unsupervised alternative, aiming to identify human-interpretable directions in an FM's latent space. For additional work see Appendix A.

2 Methodology

2.1 Sparse Autoencoders (SAE)

The superposition hypothesis in FMs suggests that a limited number of neurons encode a much larger number of concepts, leading to complex and overlapping representations (Elhage et al., 2022b). Superposition, while efficient, makes it challenging to interpret individual neuron representations or directions in representation space. Sparse autoencoders (SAEs) offer a potential solution by learning to reconstruct FM representations at a layer using a sparse set of features in a higher-dimensional space, disentangling superposed features and revealing more interpretable representations (Elhage et al., 2022a; Olshausen and Field, 1997). In a well-trained SAE, individual features in the hidden dimension align with underlying sparse, semantically meaningful features (Donoho, 2006).

SAEs decompose a model's activation $x \in \mathbb{R}^n$ into a sparse, linear combination of feature directions: $x \approx x_0 + \sum_{i=1}^{M} f_i(x) d_i$, where d_i are $M \gg n$ latent unit-norm feature directions, and the sparse coefficients $f_i(x) \geq 0$ are the corresponding feature activations for x. The right-hand side of this equation has the structure of an autoencoder: an input activation x is encoded into a (sparse) feature activations vector $f(x) \in \mathbb{R}^M$, which is then linearly decoded to reconstruct x. We parameterize a single-layer autoencoder $(f, \hat{x})$ as follows: $f(x) := \text{ReLU}(W_{enc}(x) + b_{enc})$ and $\hat{x}(f) := W_{dec} f + b_{dec}$ where $W_{enc} \in \mathbb{R}^{M \times n}$ and $W_{dec} \in \mathbb{R}^{n \times M}$ are the encoding and decoding weight matrices, and $b_{enc} \in \mathbb{R}^M$ and $b_{dec} \in \mathbb{R}^n$ are the bias vectors. The training objective combines a reconstruction loss and a sparsity penalty:

$$L(x) = \|x - \hat{x}(f(x))\|_2^2 + \lambda \|f(x)\|_1 \quad (1)$$

where $\lambda > 0$ is a hyperparameter controlling the trade-off between reconstruction fidelity and sparsity. We constrain the columns of W_{dec} to have unit norm during training (Bricken et al., 2023).

In existing work, SAEs for FMs are trained on the same large, general-purpose dataset used to train the underlying FM (Bricken et al., 2023; Cunningham et al., 2023; Rajamanoharan et al., 2024; Gao et al., 2024). This approach ensures that the SAE captures a wide array of concepts present in the general language domain. However, this can result in the SAE learning features that are frequent in the pretraining data but miss concepts within specific domains of interest, especially those that are rare by frequency in the pretraining data.

2.2 Specialized Sparse Autoencoders (SSAE)

Specialized Sparse Autoencoders are designed to learn features representing rare concepts within specific subdomains. To train SSAEs, our approach begins with a small seed concept dataset, comprising either a specific concept or limited data from the target subdomain (e.g., toxicity). We then expand this seed dataset using a high-recall retrieval strategy that leverages the seed data to identify and retrieve subdomain-relevant examples from the base FM's pretraining corpus. We then finetune a pretrained general-purpose SAE (GSAE) on this curated subdomain data using Equation 1. The GSAE is initially trained to reconstruct activations on a large, general-purpose dataset, enabling it to capture a broad range of concepts. Finetuning on the subdomain data allows the SAE to specialize and learn features that may be infrequent in the general domain but prevalent within the target subdomain.

To evaluate the quality of the trained SAEs, we use L_0 and Perplexity with SAE (Bricken et al., 2023). L_0 measures the sparsity of the SAE and is defined as the average number of active features on a given input, i.e. $\mathbb{E}_{x \sim D} \|f(x)\|_0$. Perplexity with SAE measures the reconstruction fidelity of the SAE and is the average cross-entropy loss of the language model on an evaluation dataset, when the SAE's reconstructions are spliced into it. A better SAE recovers more of the base model's performance. All other things being equal, a better SAE needs fewer features (L_0) to explain model performance on a given datapoint. Unlike existing works that evaluate SAEs on subsampled training data, we evaluate SSAE generalization using both in-distribution and out-of-distribution test sets drawn from the same subdomain. This dual evaluation approach assesses the SSAE's ability to both accurately capture concepts within the specific training data distribution and generalize to unseen data, reflecting the capability to learn broader subdomain concepts. Additionally, we perform automated interpretability scoring and qualitative analysis, to verify the interpretability of the learned features.

2.3 Subdomain Data Selection Strategies

SSAE effectiveness depends on the quality and relevance of the selected subdomain data used for finetuning. We study several selection strategies to identify data points from a larger corpus (FM's pretraining data) most relevant to the seed data:

Sparse Retrieval: Okapi BM25 (Robertson and Zaragoza, 2009), a TF-IDF variant, ranks documents based on query relevance, considering term frequency, inverse document frequency, and document length. We use the seed dataset as query to retrieve relevant documents from the larger corpus.

Dense Retrieval: Contriever (Izacard et al., 2022), a dual-encoder dense retriever, generates semantically meaningful embeddings for queries and documents. We embed the seed dataset and candidate documents, using cosine similarity to retrieve documents most similar to the seed concepts.

SAE TracIn: Training data Influence Score (TracIn) (Pruthi et al., 2020) quantifies training examples' influence on model predictions. We adapt TracIn to SAEs by calculating the dot product of the loss gradients with respect to the training data and seed data: $\mathrm{TracIn}\,(z, z') = \nabla L_w\,(z) \cdot \nabla L_w\,(z')$ where z is a training data point, z' is the seed dataset, w are the pretrained SAE weights, and $L_w(\cdot)$ is the SAE loss (Equation 1). We use a two-stage approach to identify influential data: Initial Filtering with Sparse/Dense retrieval, then TracIn Reranking to select points for SSAE training.

2.4 Tilted Empirical Risk Minimization for Enhanced Detection

Finetuning with Empirical Risk Minimization (ERM) tends to prioritize learning features for the most frequent *head* concepts in the subdomain data. However, for many applications such as safety, capturing rare *tail* concepts is often crucial. These rare features may represent potential risks or safety violations and are often overlooked by standard ERM as it focuses on minimizing the average loss. To capture these rare, potentially dangerous features we need an objective that is not minimizing the average loss, but rather minimizing the maximum risk. Tilted Empirical Risk Minimization (TERM) (Li et al., 2020; Beirami et al., 2018) provides a framework for approximating this minimax risk, encouraging the model to learn features that better represent these tail concepts.

TERM modifies the standard ERM objective by introducing a tilt parameter (t) that controls the emphasis on different parts of the loss distribution: $\tilde{L}(t; w) = \frac{1}{t} \log \left(\frac{1}{N} \sum_{i \in [N]} e^{t \cdot L_w(z_i)} \right)$ where $L_w(z_i)$ is the standard SAE loss (Equation 1) for data point z_i in a minibatch with N points and SAE parameters w. TERM generalizes ERM as the 0-tilted loss recovers the average loss, while it also recovers other alternatives such as the max-loss ($t \to +\infty$) and min-loss ($t \to -\infty$). In this work we use large tilt parameters ($t \gg 0$) to effectively minimize the maximum loss, encouraging the SAE to learn features that better represent the tail of the data distribution at a given sparsity. Minimax losses are also known to enhance robustness to OOD data, which is relevant for detecting rare concepts often underrepresented in training data (Ye et al., 2021; Sagawa et al., 2019).

Incorporating TERM during finetuning leads to a more balanced representation of both head and tail concepts within the subdomain. This shift reflects a fundamental trade-off between precision and recall in TERM-trained and ERM-trained SAEs. Standard ERM prioritizes precision, yielding highly specialized features that allow for fine-grained control over concepts but may miss rare ones. TERM prioritizes recall, sacrificing some control for broader concept coverage, particularly of rare concepts, making it advantageous for detecting potentially harmful behaviors. TERM encourages the SAE to learn compositional features leading to more interpretable representations (see Appendix L for a formal argument).

3 Experiments And Results

3.1 Specialized Sparse Autoencoders (SSAEs)

3.1.1 Data Selection Strategies

In this section, we evaluate the effectiveness of various data selection strategies for training SSAEs.

Experimental Setup We use the pretrained Gemma-2b (Team et al., 2024) residual stream GSAE (gemma-2b-res-jb checkpoint at blocks.12.hook_resid_post layer) (Bloom, 2024). These SAEs have feature width 16384 and were pretrained on OpenWebText (OWT) (Gokaslan et al., 2019). For the Pareto front, we sweep 8 L1 penalty coefficients, selecting the best model on validation for each L1 value, then evaluating on the held-out test split. SAEs are trained using Adam (Kingma and Ba, 2015) with lr 5e-5, token batch size 4096, data shuffled within a batch buffer of size 4, and linear lr decay over the last 1000 steps. Experiments complete in under 12 hours using 4 A6000 GPUs. We use SAELens (Bloom and Chanin, 2024) for training and analysis.

Computational Requirements A key advantage of our approach is its scalability and accessibility. The computational overhead of indexing the pretraining data is a one-time cost that can be amortized across many different subdomain datasets. Furthermore, indexing a large corpus using a dense retriever is significantly more efficient than pretraining a wide SAE. Using a relatively compact model like Contriever (100M params), the primary computational expense lies in indexing the corpus; this takes approximately 2 hours on a single A6000 GPU and costs roughly $5. The retrieval process itself is highly efficient, requiring less than an hour on CPUs, leveraging the FAISS library for fast approximate nearest neighbor search. This low computational barrier makes our method accessible to researchers even with limited access to high-performance computing resources, such as those using consumer-grade GPUs. In stark contrast, pretraining a Gemma-2B SAE from scratch is considerably more demanding, requiring an estimated 3-4 days on A100 GPUs, with an associated cost of approximately $200. We anticipate that this difference in computational cost will become even more pronounced as we consider wider SAEs and larger base models.

SSAE for Physics We start with a seed concept dataset (Validation) consisting of 9.2K tokens sampled from the arXiv Physics dataset (Anonymous, 2024). Using BM25, Dense Retrieval, and SAE TracIn, we expand this to 13.9M tokens from OWT. The SSAE is trained by finetuning the GSAE for 1000 iterations on this expanded set. For SAE TracIn, we first reduce OWT to 1% using BM25 or Dense retrieval, then rerank using TracIn scores and select 13.9M tokens. We call these methods BM25 TracIn and Dense TracIn, respectively.

We train an SSAE for each strategy and compare its performance to a baseline SAE finetuned on the full OWT dataset across various sparsity coefficients (λ). We evaluate the models on two test splits: 4.8M tokens from arXiv Physics (in-distribution) and 700K tokens from Physics instruction tuning (Group, 2024)(out-of-distribution). Testing on instruction data helps measure whether the SAEs are overfitting to the specific template of the text as opposed to identifying concepts. Figure 1 and 9 show the patched perplexity vs. L_0 curves for these experiments.

We measure performance using area under the curve for a range of L_0 from 60 to 140 i.e., a selection strategy with lower perplexity (SSAE spliced in) is better. Our findings show Dense TracIn and BM25 TracIn achieve comparable performance, surpassing Dense retrieval alone, which in turn outperforms BM25 retrieval. Training on the full OWT dataset yields the lowest performance. We observe: (a) Dense retrieval consistently outperforms BM25. SSAEs trained with Dense Retrieval achieve lower perplexity for a given L_0 than those with BM25,

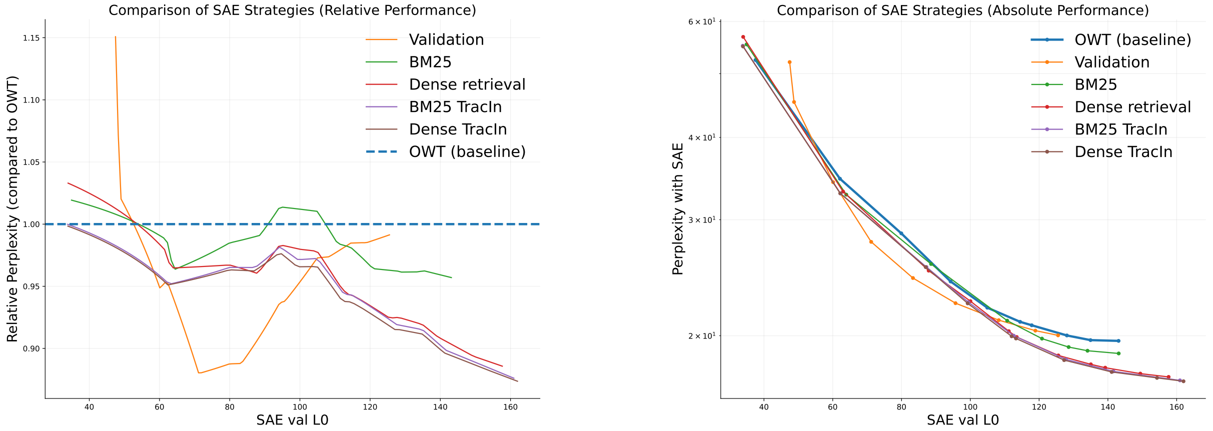


Figure 1: Pareto curves for Physics SSAE trained with various data selection strategies as the sparsity coefficient λ is varied on arXiv Physics test data. We plot (Left) Perplexity with spliced in SSAE relative to GSAE baseline and (Right) Absolute Perplexity with spliced in SSAE. Dense TracIn and BM25 TracIn achieve comparable performance, performing slightly better than Dense retrieval, which outperforms BM25 retrieval and OWT Baseline. All curves are averaged over 3 SAE training seeds.

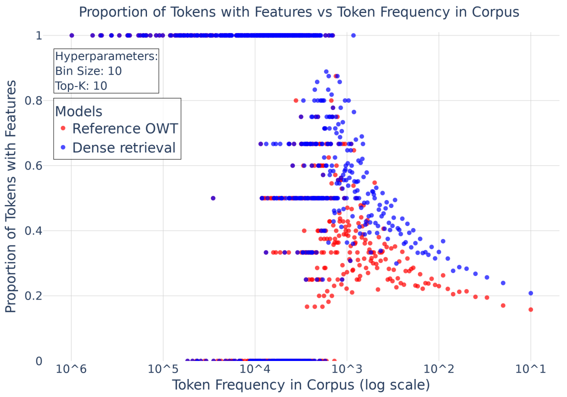


Figure 2: Proportion of tokens with SAE features vs. Token frequency in Physics arXiv data. SSAE trained with dense retrieval captures more tail tokens (concepts) in its features.

both in and out of distribution. (b) BM25 exhibits poor out-of-distribution generalization. While BM25 performs reasonably well in-distribution, its performance degrades significantly on the out-of-distribution test set. (c) Multiple passes on seed data (Validation) during SSAE training improve in-distribution performance but degrade out-of-distribution performance. This suggests multiple passes can overfit to the structure or template of the seed dataset. (d) While TracIn reranking after Dense retrieval yields a marginal performance gain, Dense retrieval alone remains highly competitive.

SSAE for Toxicity We observe similar results when repeating the experiment on the Pile Toxicity dataset (Korbak, 2024) in Appendix C. Dense retrieval outperforms BM25 and TracIn shows a marginal improvement over Dense retrieval alone.

3.1.2 Probing Tail Concept Learning

To probe tail concept learning we use convergent validity (Campbell and Fiske, 1959) with the Logit Lens (Bloom and Lin, 2024). Figure 2, uses the unembedding matrix as a logit lens to analyze the top-10 token logits associated with each SSAE feature. For each frequency bucket in the Physics arXiv test data, we calculate the percentage of tokens that appear among the top-10 logits for at least one feature. Assuming the logit lens correctly interprets the token-level representations of each feature, this measures the extent to which SSAE features represent tokens across different frequency ranges.

We compare two SSAEs at test L_0 of 100: one finetuned on full OWT dataset, another using Dense retrieval. The Dense retrieval finetuned SSAE captures a significantly higher proportion of tail tokens in its features compared to the OWT finetuned SSAE. Moreover, these captured tail tokens often correspond to physics-specific concepts, suggesting that SSAEs are indeed learning to represent rare, domain-relevant concepts. Similar results are obtained for toxicity data in Figure 11.

3.1.3 Case study: Removing Spurious Features in Bias in Bios Classifier

Having shown the effectiveness of ERM-trained SSAEs in capturing tail concepts for finer control, we now apply SSAEs to Spurious Human-interpretable Feature Trimming (SHIFT) (Marks et al., 2024). SHIFT addresses the issue of FM classifiers relying on unintended signals (e.g., spurious features) by modifying their generalization through feature circuit editing. Unlike approaches that rely on disambiguated labeled data, SHIFT operates even when such data is unavailable (Zech et al., 2018; Ngo et al., 2022; Casper et al., 2023; Hase et al., 2024). We show that replacing the GSAE with our SSAE in SHIFT further enhances its editing capabilities.

Method. SHIFT operates as follows, given labeled training data $D = (x_i, y_i)$, classifier C trained on D, and SAEs for components of C:

1. Compute a feature circuit (see Appendix H) explaining C's accuracy on inputs $(x, y) \sim D$ (using metric $m = -\log C(y|x)$).
2. Manually or automatically inspect and evaluate each feature's task-relevancy.
3. Ablate features deemed task-irrelevant to obtain a modified classifier C'.
4. (Optional) Finetune (retrain) C' on data from D to potentially restore performance.

Experimental Setup. We use the Bias in Bios dataset (BiB) (De-Arteaga et al., 2019) to illustrate SHIFT with SSAEs. BiB contains professional biographies and the task is to classify an individual's profession, with gender being a spurious feature. Two subsets are created from BiB: the *ambiguous set* (male professors and female nurses) and the *balanced set* (equal numbers of male professors, male nurses, female professors, and female nurses) (Marks et al., 2024). The ambiguous set represents a worst-case scenario where the unintended signal (gender) perfectly predicts training labels (profession). Our goal is to achieve accurate profession classification on the balanced set using only the ambiguous set for training.

Our base model is a Pythia-70M linear classifier (Biderman et al., 2023), trained on the ambiguous set (training details in subsection G.1). SHIFT is applied by discovering a circuit using the zero-ablation variant (Appendix H). Instead of using human judgement to ablate features, we employ Feature skyline (Marks et al., 2024), sweeping across 1-200 circuit features most causally implicated in spurious feature accuracy on the balanced set. The number of features to ablate is chosen based on best profession classification performance on the dev set.

We use GSAEs (width 32768) for the MLP output, attention output, and residual stream for each layer, pretrained on 2B tokens (first 128 tokens of random documents) from The Pile (Gao et al., 2020). The SSAE is trained by retrieving 8M tokens from The Pile using a dense retriever, guided by 5 BiB examples, and finetuning all the GSAEs in every layer on this data for one epoch. We use $\lambda = 0.1$ and learning rate 10^{-4} throughout.

We also conduct a *Compression* experiment, where we slice the GSAE to width 4096 by taking only the first 4096 rows of the decoder (Comp. GSAE). This examines a worst-case scenario where the GSAE may not capture all relevant subdomain features. Comp. SSAE is initialized with Comp. GSAE before finetuning on the retrieved tokens.

In addition to the Oracle (trained on ground-truth labels from the balanced set) and Original (trained on ground-truth labels from the ambiguous set) classifiers, we include the following baselines:

- Concept Bottleneck Probing (CBP): Adapted from Yan et al. (2023) (see subsection G.2).
- Neuron skyline: Sweeps over number of neurons to ablate (1-200) and mean-ablates those most implicated in spurious feature accuracy.

Method	Accuracy		
	↑Prof.	↓Gen.	↑Worst
Original	61.9	87.4	24.4
CBP	83.3	60.1	67.7
Neuron skyline	75.5	73.2	41.5
GSAE SHIFT	88.5	54.0	76.0
SSAE SHIFT	90.2	53.4	88.5
GSAE SHIFT+retrain	93.1	52.0	89.0
SSAE SHIFT+retrain	**93.4**	**51.9**	**89.5**
Comp. GSAE SHIFT	80.5	68.2	48.6
Comp. SSAE SHIFT	89.6	52.2	78.8
Comp. GSAE SHIFT+retrain	80.0	68.8	57.1
Comp. SSAE SHIFT+retrain	**93.2**	**52.1**	**88.5**
Oracle	93.0	49.4	91.9

Table 1: Balanced set accuracies for intended (profession) and unintended (gender) labels. *Worst* refers to lowest profession accuracy among male professors, male nurses, female professors, and female nurses. Comp.: Compressed SAE (sliced to 1/8th width). Best results per method category are bolded.

Results. As shown in Table 1, GSAE SHIFT effectively reduces the classifier's dependence on gender compared to baselines such as CBP, with Step 3 (feature ablation) providing the most substantial improvement. Applying SHIFT with neurons (Neuron skyline) performs worse than SHIFT with SAEs, likely due to the polysemantic nature of individual neurons (Marks et al., 2024).

SHIFT with SSAEs further improves classifier performance on the balanced set, achieving a 1.7% increase in profession accuracy, a 12.5% increase in worst-group accuracy, and a decrease in spurious gender accuracy, demonstrating its superiority to GSAEs in fine-grained control. These gains persist even after retraining the classifier probe, albeit to a smaller extent. The improvement can be attributed to the SSAE activating more sundomain-relevant features. For instance, at an activation threshold of 0.01, the SSAE activates 908 features compared to 602 in the GSAE. These additional features, as explored in Appendix H, play a crucial role in the sparse feature circuits of the classifier, explaining more of the variance previously attributed to error nodes by the GSAE.

In the Compression experiment, the performance of the Comp. GSAE with missing features drops significantly compared to the GSAE. Retraining the classifier probe fails to mitigate this performance loss, and SHIFT is ineffective at removing spurious features with Comp. GSAE. However, the Comp. SSAE recovers most of this lost performance, even surpassing GSAE SHIFT by 1.1% in profession accuracy. Retraining the probe with Comp. SSAE restores nearly all lost performance.

3.2 Tilted ERM for Enhanced Detection

3.2.1 Motivating Example: TERM-trained GSAEs on TinyStories

ERM-trained GSAEs prioritize learning frequent concepts in the data. In this section, we examine features in TERM-trained GSAEs, showing that TERM improves feature recall at the expense of feature control.

Experimental Setup. We use the 8-layer, 1M parameter base model `TinyStories-1M` (Eldan and Li, 2023). SAEs of width 64 are trained on the residual stream of the 7th layer using both ERM and TERM (tilt=10^9). We use batch size 64, lr 10^{-3}, $\lambda = 0.01$, and train for 1 epoch on the `roneneldan/TinyStories` dataset. This dataset allows us to interpret nearly all features while demonstrating the benefits of the TERM loss. We report results on checkpoints with L_0 of 16.

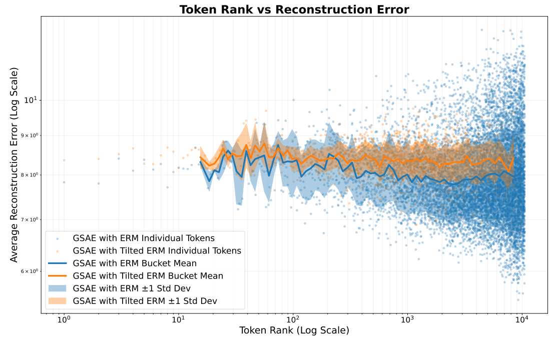


Figure 3: Reconstruction error vs. token rank for TERM-trained and ERM-trained GSAEs. TERM exhibits lower error variance and maximum error for tail tokens.

Results Figure 3 plots the reconstruction error for tokens ranked by frequency, showing that TERM reduces reconstruction error and error variance for tail tokens compared to ERM. Similarly, from the the distribution of reconstruction error for the TERM-trained GSAE (Figure 22), we see that TERM minimizes max error at the cost of slightly higher average error.

We also analyze decoder feature vector coverage using three approaches. A UMAP visualization of token activations and decoder features for both GSAEs, reveals a greater dispersion of decoder directions for the TERM-trained GSAE, indicating broader coverage (Figure 19). Similarly the distribution of cosine similarities between decoder directions, with the TERM-trained GSAE showing lower overall similarity, suggesting greater coverage (Figure 20). The TERM-trained GSAE also requires more PCA components to explain variance in decoder feature directions (40) compared to the ERM-trained GSAE (21) (Figure 21). Taken together, this shows that TERM-trained SAEs cover a wider range of features than ERM-trained SAEs.

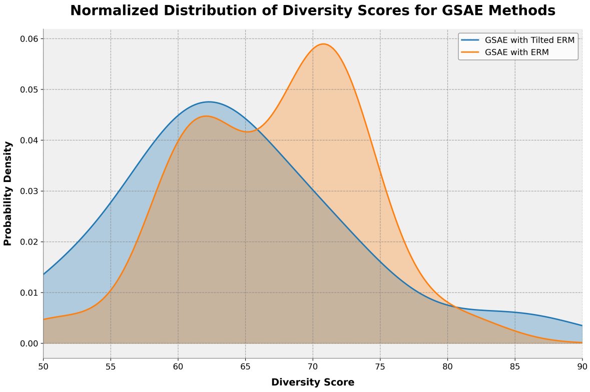


Figure 4: Feature diversity score distributions for TERM-trained and ERM-trained GSAEs. TERM leads to both higher and lower diversity features. Lower diversity features specialize in tail concepts, while higher diversity features capture a broader range of concepts.

Figure 4 presents diversity score distributions for TERM- and ERM-trained GSAE feature explanations, capturing the variety of examples explainable by each feature using Claude 3.5 Sonnet (examples in Section Appendix Q and N.6). TERM-trained GSAEs exhibit both higher and lower diversity features compared to ERM, with lower diversity features specializing in tail concepts and higher diversity features capturing a broader range of concepts, both frequent and rare.

Figure 5 shows that TERM-trained GSAE features exhibit stronger activations and lower entropy compared to ERM-trained GSAE on the data. This, combined with their high recall, suggests a strategy for rare concept detection: tag features strongly associated with rare concepts during pretraining,

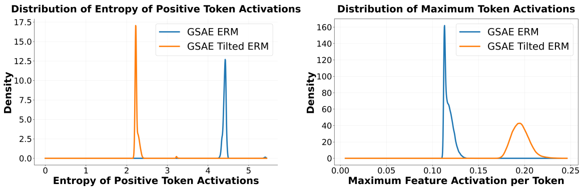


Figure 5: TERM feature activation patterns. (Left) TERM token activation entropy is lower, suggesting more specialized features. (Right) TERM max feature activations per token are higher. These characteristics, from minimizing max risk, contribute to TERM's enhanced tail concept detection.

and at test time, strong activation of these tagged features triggers further investigation. This is more effective than using SAE error with ERM-trained SAEs for rare concept detection, as error nodes do not disambiguate types of rare features.

3.2.2 TERM-trained SSAE Performance

While ERM-trained SSAEs improve tail concept coverage compared to GSAEs, they still prioritize learning frequent subdomain concepts. TERM-trained GSAEs could potentially offer better tail concept representation, but training SAEs from scratch is computationally expensive (Lieberum et al., 2024). Therefore, we investigate whether finetuning SSAEs with TERM on the retrieved data (using hyperparameters from Sec 3.1.1) can achieve similar properties to TERM-trained GSAEs.

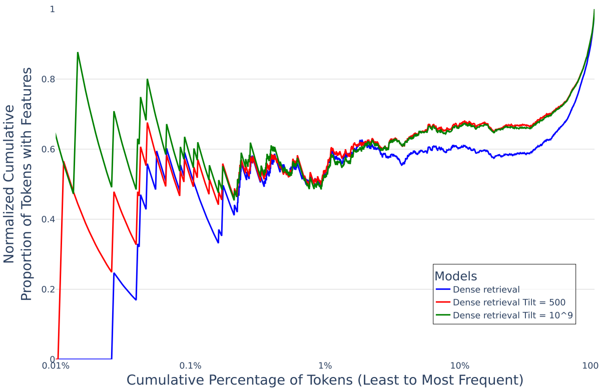


Figure 6: Cumulative proportion of tokens with SAE features vs. cumulative percentage of tokens in Physics arXiv data, normalized per model so that the cumulative proportion of tokens with features is 1 over the entire dataset. SSAE trained with dense retrieval and larger tilt captures more tail tokens (concepts) in its features.

Enhanced Tail Concept Capture with TERM Figure 6 plots the cumulative proportion of tokens with SSAE features (identified using the logit lens approach) versus the cumulative percentage of tokens in the Physics arXiv data for different SSAEs. We normalize the curves per model at a validation L_0 of 100, so that the cumulative proportion of tokens with features is 1 over the entire dataset. Results show that SSAEs trained with Dense retrieval and tilt capture a greater proportion of tail tokens on the low frequency end (on the left) compared to Dense retrieval alone, with this effect increasing with tilt. Figure 14 shows a similar trend for the Toxicity dataset.

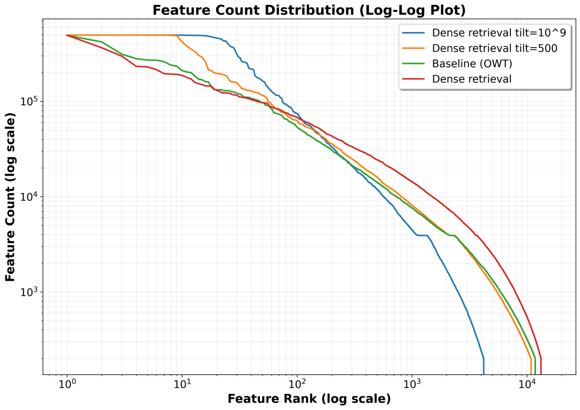


Figure 7: **Feature activation count vs. feature rank** for SSAEs trained on the Physics arXiv dataset using different strategies: full OWT, Dense retrieval, and Dense retrieval with tilt. Tilt encourages the learning of more broadly activating features, indicating increased concept coverage and recall.

Feature activation counts Figure 7 plots feature activation count vs. feature rank, showing that TERM with large tilt encourages learning more broadly activating features with increased concept recall. This represents a fundamentally different mechanism for feature learning compared to standard ERM, promoting more compositional features that capture tail concepts.

We see similar trends in the distribution of differences in feature activation counts between SSAEs (ERM and TERM-trained) and the OWT baseline on the Physics arXiv test set. A peak at 0 indicates that SSAEs retain some similarity to the baseline in their activation patterns. The ERM-trained SSAE exhibits greater probability mass on the right, indicating a focus on frequent concepts, while the TERM-trained SSAEs shift probability mass leftward as tilt increases, suggesting a stronger emphasis on representing domain-specific tail concepts (See Figures 17 and 18).

Downstream perplexity We also find that TERM-finetuned SSAEs achieve comparable downstream perplexity to ERM-finetuned SSAEs within the typical L_0 regime used (see Figures 12 and 13). However at very large or low L_0, training with Adam can lead to higher average risk or many inactive features. Adaptive penalty schemes offer a promising solution to this challenge (more discussion in Appendix E).

3.2.3 Automated Interpretability

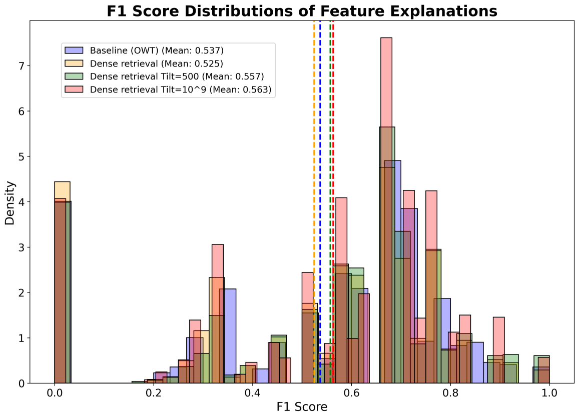


Figure 8: **Automated interpretability**: F1 score distributions for predicting feature activation on Physics arXiv, using only FM-generated explanations. An LM is given examples activating a feature and asked to generate an explanation, which is then used to predict activations on new examples. Dense retrieval with tilt produces more predictive explanations than both the OWT baseline and Dense retrieval alone.

We employ a sequence-level classification task to evaluate interpretability (Bills et al., 2023; Templeton et al., 2024). Instead of predicting feature activation at each token, an FM is tasked with identifying whether entire sequences contain a given feature. This simplifies the task, producing reliable scores even with smaller, faster FMs (Juang et al., 2024). Using Claude 3.5 Sonnet (Anthropic, 2024) as both the *Interpreter* and the *Predictor*, our framework tasks the Interpreter with generating explanations for each feature based on the top 10 activating examples (see Appendix J). The Predictor then receives these explanations along with 10 examples (5 activating, 5 non-activating) and predicts whether each example activates the feature (see Appendix K for prompts). We measure explanation interpretability using the F1 score between the Predictor's predictions and the true feature activations.

Fig 8 shows that TERM-trained SSAEs achieve higher F1 scores than the OWT baseline and ERM-trained SSAEs, indicating their explanations are more effective in predicting activation on new examples. Interestingly, despite superior downstream perplexity vs. L_0, ERM-trained SSAEs did not yield more interpretable explanations than the baseline. This aligns with findings in O'Neill et al. (2024), where interpretability decreased with increasing SAE width attributed to less interpretable fine-grained features. As TERM encourages coarser, more compositional features, its explanations are more readily interpretable.

4 Discussion

Our work focuses on an automated retrieval-based approach for data selection, rather than relying on large, expert-curated datasets. While a sufficiently large, carefully curated dataset (human-generated or synthetic) could potentially yield excellent results, this approach presents significant practical challenges. The creation of such datasets, particularly in specialized domains or for safety-relevant tasks, is often prohibitively expensive and time-consuming. For instance, the WMDP dataset, consisting of only 3,668 questions, required over $200,000 to develop (Li et al., 2024b). Our results demonstrate that a high-recall retrieval strategy, using dense retrieval with a similarity threshold for relevance verification, provides a practical and effective alternative particularly for academic research labs with limited resources. Moreover, our experiments (e.g., Figure 1) show that even small, carefully selected seed datasets, when combined with our retrieval method, can outperform models trained on larger, randomly sampled datasets, and avoid the overfitting issues associated with multiple passes over a small, fixed dataset.

5 Conclusion and Future Work

This work introduces SSAEs for interpreting rare, subdomain features in FMs. SSAEs trained with Dense retrieval and TERM, outperform standard SAEs in capturing tail concepts and yield more interpretable features. Future work could explore their application to targeted concept unlearning.

6 Acknowledgements

This work was supported in part by the National Science Foundation grants IIS2145670 and CCF2107024, and funding from Amazon, Apple, Google, Intel, Meta, the CyLab Security and Privacy Institute, the Anthropic Researcher Access Program, and AI Safety Support. The project originated during Aashiq's participation in the ML Alignment and Theory Scholars (MATS) program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of these funding agencies.

7 Limitations

While our work demonstrates the effectiveness of SSAEs in enhancing interpretability and tail concept capture across diverse domains like Physics and Toxicity, there are several areas for further exploration:

Computational Efficiency of TERM. Training SAEs with TERM, while effective in enhancing concept recall and yielding more interpretable features, can be computationally more demanding than standard ERM. The TERM objective requires computing the exponent of the loss for each data point, which is more computationally intensive than in ERM. This can potentially lead to numerical instability and slower convergence, particularly at high tilt values. The benefits of TERM in improving interpretability and fairness encourage further research to reduce its computational cost for broader adoption and scalability.

Dependence on Seed Data. The effectiveness of SSAEs hinges on the quality and representativeness of the initial seed dataset used for retrieval. While we demonstrate strong results even with remarkably small seed datasets (see subsubsection 3.1.1), low-quality, unrepresentative, or extremely limited seed data could lead to SSAEs that fail to capture the full scope of relevant subdomain concepts or, worse, exhibit biases present in the seed data. Specifically, a seed dataset that is too narrow or focuses on only the most common aspects of a subdomain might cause the retrieval process to miss important, rarer concepts. For instance, a single-sentence physics seed containing only common terms like “energy” or “force” would likely be insufficient. Mitigating this limitation requires careful seed selection and, ideally, validation of the retrieved data’s diversity and relevance.

Generalizability Across Domains and Applications. Our experiments with the Physics, Toxicity, Bias in Bios, and TinyStories datasets demonstrate the effectiveness of SSAEs across diverse domains. While we have no reason to believe our findings won’t generalize, further empirical validation across an even broader range of tasks and datasets would strengthen our conclusions. We are particularly interested in evaluating SSAEs in settings where rare concepts play a crucial role, such as AI safety, healthcare, and fairness. These applications would further solidify SSAEs as powerful and versatile tools for enhancing interpretability and control in foundation models.

8 Ethical Considerations

The ability to interpret and analyze rare concepts within foundation models, particularly those related to sensitive attributes, carries significant ethical implications that warrant careful consideration.

Potential for Misuse and Dual-Use Concerns. The techniques presented in this work, while intended for enhancing interpretability, safety, and fairness, could be misused for malicious purposes. The capability to identify and manipulate rare features, especially those associated with sensitive attributes like gender, race, or political affiliation, could be exploited to amplify existing biases, generate harmful or misleading content, or manipulate model behavior in ways that perpetuate or exacerbate societal inequalities. Addressing these dual-use concerns requires proactive efforts to develop safeguards, promote responsible use guidelines, and engage in open discussions about the potential risks associated with these powerful tools.

Bias Amplification. While SSAEs aim to improve the representation of rare and potentially underrepresented concepts, they are not inherently immune to bias. Biases present in the underlying foundation model and its training data can be inherited and potentially amplified by SSAEs, even when tailored to focus on specific subdomains or sensitive attributes. Mitigating this risk requires careful attention to data curation, development of robust bias detection and mitigation techniques during both FM and SSAE training, and ongoing monitoring and evaluation of SSAE features to ensure they do not perpetuate or exacerbate existing biases.

Data Privacy and Responsible Use. The datasets used in this work are publicly available and widely used within the NLP research community (see Appendix O). These datasets have undergone accepted privacy practices at their creation time. We have strictly adhered to the license terms of these datasets, ensuring responsible and ethical handling.

References

Evan Anders and Joseph Bloom. 2024. Examining language model performance with reconstructed activations using sparse autoencoders.

Anonymous. 2024. arxiv physics dataset. https://huggingface.co/datasets/anonymousdatasets/arxiv-physics.

Anthropic. 2024. Claude 3.5 sonnet. https://www.anthropic.com or https://claude.ai. AI model.

Ahmad Beirami, Robert Calderbank, Mark M Christiansen, Ken R Duffy, and Muriel Médard. 2018. A characterization of guesswork on swiftly tilting curves. *IEEE Transactions on Information Theory*, 65(5):2850–2871.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Yoshua Bengio. 2013. Deep learning of representations: Looking forward. In *International conference on statistical language and speech processing*, pages 1–37. Springer.

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. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. Technical report.

John Bloom. 2024. Gemma-2b-residual-stream-saes. https://huggingface.co/jbloom/Gemma-2b-Residual-Stream-SAEs.

Joseph Bloom and David Chanin. 2024. Saelens. https://github.com/jbloomAus/SAELens.

Joseph Bloom and Johnny Lin. 2024. Understanding SAE features with the Logit Lens.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. 2024. Identifying functionally important features with end-to-end sparse dictionary learning. *arXiv preprint arXiv:2405.12241*.

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. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Donald T Campbell and Donald W Fiske. 1959. Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological bulletin*, 56(2):81.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31.

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

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. 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*, pages 120–128.

David L Donoho. 2006. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Neel Nanda, Tom Henighan, Scott Johnston, Sheer ElShowk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Andy Jones, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. 2022a. Softmax linear units. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/solu/index.html.

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

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12.

Eoin Farrell. 2024. Experiments with an alternative method to promote sparsity in sparse autoencoders.

Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. 2023. Interpreting clip's image representation via text-based decomposition. *arXiv preprint arXiv:2310.05916*.

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

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*.

Atticus Geiger, Chris Potts, and Thomas Icard. 2023. Causal abstraction for faithful model interpretation. *arXiv preprint arXiv:2301.04709*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767*.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Algorithmic Research Group. 2024. arxiv physics instruct tune 30k dataset. `https://huggingface.co/datasets/AlgorithmicResearchGroup/arxiv-physics-instruct-tune-30k`.

Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegreffe. 2024. The unreasonable effectiveness of easy training data for hard tasks. *arXiv preprint arXiv:2401.06751*.

Stefan Heimersheim and Jett Janiak. 2023. A circuit for python docstrings in a 4-layer attention-only transformer. *URL: https://www. alignmentforum. org/posts/u6KXXmKFbXfWzoAXn/acircuit-for-python-docstrings-in-a-4-layer-attention-only*.

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. In *Transactions of the Association for Computational Linguistics*, volume 10, pages 726–741. MIT Press.

Adam Jermyn, Adly Templeton, Joshua Batson, and Trenton Bricken. 2024. Tanh penalty in dictionary learning.

Caden Juang, Gonçalo Paulo, Jacob Drori, and Nora Belrose. 2024. Open source automated interpretability for sparse autoencoder features. `https://blog.eleuther.ai/autointerp/`. EleutherAI Blog.

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

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. In *International conference on machine learning*, pages 2649–2658. PMLR.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Kishore Konda, Roland Memisevic, and David Krueger. 2014. Zero-bias autoencoders and the benefits of co-adapting features. *arXiv preprint arXiv:1402.3337*.

Tomek Korbak. 2024. Pile toxicity balanced dataset. `https://huggingface.co/datasets/tomekkorbak/pile-toxicity-balanced`.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. 2024. Atp*: An efficient and scalable method for localizing llm behaviour to components. *arXiv preprint arXiv:2403.00745*.

Quoc V Le. 2013. Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8595–8598. IEEE.

Honglak Lee, Chaitanya Ekanadham, and Andrew Ng. 2007. Sparse deep belief net model for visual area v2. *Advances in neural information processing systems*, 20.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024a. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Maximilian Li, Xander Davies, and Max Nadeau. 2023. Circuit breaking: Removing model behaviors with targeted ablation. *arXiv preprint arXiv:2309.05973*.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. 2024b. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*.

Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. 2020. Tilted empirical risk minimization. *arXiv preprint arXiv:2007.01162*.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *arXiv preprint arXiv:2408.05147*.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Julien Mairal, Francis Bach, Jean Ponce, et al. 2014. Sparse modeling for image and vision processing. *Foundations and Trends® in Computer Graphics and Vision*, 8(2-3):85–283.

Aleksandar Makelov, George Lange, and Neel Nanda. 2024. Towards principled evaluations of sparse autoencoders for interpretability and control. *arXiv preprint arXiv:2405.08366*.

Stéphane G Mallat and Zhifeng Zhang. 1993. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2024. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*.

Emile Mathieu, Tom Rainforth, Nana Siddharth, and Yee Whye Teh. 2019. Disentangling disentanglement in variational autoencoders. In *International conference on machine learning*, pages 4402–4412. PMLR.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Dan Mossing, Steven Bills, Henk Tillman, Tom Dupré la Tour, Nick Cammarata, Leo Gao, Joshua Achiam, Catherine Yeh, Jan Leike, Jeff Wu, et al. 2024. Transformer debugger.

Neel Nanda. 2023. Attribution patching: Activation patching at industrial scale. *URL: https://www. neelnanda. io/mechanistic-interpretability/attribution-patching*.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023a. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*.

Neel Nanda, S Rajamanoharan, J Kramár, and R Shah. 2023b. Fact finding: Attempting to reverse-engineer factual recall on the neuron level. In *AI Alignment Forum, 2023c. URL https://www. alignmentforum. org/posts/iGuwZTHWb6DFY3sKB/fact-finding-attempting-to-reverse-engineer-factual-recall*, page 19.

Richard Ngo, Lawrence Chan, and Sören Mindermann. 2022. The alignment problem from a deep learning perspective. *arXiv preprint arXiv:2209.00626*.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001.

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

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

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Charles O'Neill, Christine Ye, Kartheik Iyer, and John F Wu. 2024. Disentangling dense embeddings with sparse autoencoders. *arXiv preprint arXiv:2408.00657*.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pages 19920–19930.

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

Logan Riggs and Jannik Brinkmann. 2024. Improving sae's by sqrt()-ing l1 and removing lowest activating features.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

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

Lee Sharkey, Dan Braun, and Beren Millidge. 2022. Taking features out of superposition with sparse autoencoders. In *AI Alignment Forum*, volume 6, pages 12–13.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. Technical report, Anthropic.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural nlp: The case of gender bias. *arXiv preprint arXiv:2004.12265*.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*.

Benjamin Wright and Lee Sharkey. 2024. Addressing feature suppression in saes. In *AI Alignment Forum*, page 16.

An Yan, Yu Wang, Yiwu Zhong, Zexue He, Petros Karypis, Zihan Wang, Chengyu Dong, Amilcare Gentili, Chun-Nan Hsu, Jingbo Shang, et al. 2023. Robust and interpretable medical image classifiers via concept bottleneck models. *arXiv preprint arXiv:2310.03182*.

Haotian Ye, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang. 2021. Towards a theoretical framework of out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34:23519–23531.

Zeyu Yun, Yubei Chen, Bruno A Olshausen, and Yann LeCun. 2021. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. *arXiv preprint arXiv:2103.15949*.

John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric K Oermann. 2018. Confounding variables can degrade generalization performance of radiological deep learning models. *arXiv preprint arXiv:1807.00431*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Related Work

This work intersects with several research areas, including mechanistic interpretability, sparse coding, feature disentanglement, and evaluation methods for Sparse Autoencoders. We contextualize our contributions within this broader landscape.

A.1 Mechanistic Interpretability

Mechanistic Interpretability (MI) aims to decipher the internal workings of neural networks by reverse engineering their computational processes (Olah et al., 2020; Elhage et al., 2021). This approach conceptualizes model computations as collections of circuits – narrow, task-specific algorithms. Recent circuit analyses of Foundation Models (FMs) have focused on mapping these circuits to specific model components like attention heads and MLP layers (Wang et al., 2022; Heimersheim and Janiak, 2023).

Building upon this component-level understanding, the linear representation hypothesis proposes that component activations can be further decomposed into (sparse) linear combinations of meaningful feature vectors. This concept underpins our work on SSAEs. Unlike previous research that sought to identify individual subspaces representing specific concepts (Geiger et al., 2023; Nanda et al., 2023a; Tigges et al., 2023), SAEs aim to provide a more complete picture by fully decomposing activations into interpretable features.

MI has shown promise in various downstream tasks, including modifying model behavior to remove toxic outputs (Li et al., 2023), altering encoded factual knowledge (Meng et al., 2022), improving truthfulness (Li et al., 2024a), analyzing gender bias mechanisms (Vig et al., 2020), and mitigating spurious correlations (Gandelsman et al., 2023). Our work with SSAEs seeks to advance these applications by providing refined tools for detecting, interpreting, and modifying model behavior, particularly concerning rare or underrepresented concepts.

A.2 Sparse Coding, Dictionary Learning, and Sparse Autoencoders

Our work draws inspiration from the foundational concepts of sparse coding with over-complete dictionaries (Mallat and Zhang, 1993) and unsupervised dictionary learning from data (Olshausen and Field, 1996). These ideas, impactful in image processing (Mairal et al., 2014), evolved into the development of sparse autoencoders (SAEs) through their integration with autoencoder architectures (Hinton and Salakhutdinov, 2006; Lee et al., 2007; Le, 2013; Konda et al., 2014).

Recently, SAEs have been applied to language models (Yun et al., 2021; Sharkey et al., 2022; Bricken et al., 2023; Cunningham et al., 2023), with successful implementations on smaller open-source language models (Marks et al., 2024; Bloom and Chanin, 2024; Mossing et al., 2024). We build upon this research trajectory, addressing specific limitations and extending the approach to capture rare, domain-specific features more effectively.

A.3 Challenges, Improvements, and Evaluation of Sparse Autoencoders

Despite their potential, SAEs face several challenges. For example, Anders and Bloom (2024) observed that SAE features trained on language models with specific context lengths fail to generalize to activations from longer contexts. Wright and Sharkey (2024) and Jermyn et al. (2024) osberved feature suppression, a phenomenon where SAE feature activations systematically underestimate true activation values due to sparsity penalties.

Various solutions have been proposed to tackle these challenges, including post-training finetuning (Wright and Sharkey, 2024), alternative sparsity penalties (Jermyn et al., 2024; Riggs and Brinkmann, 2024; Farrell, 2024), and architectural modifications such as Gated SAEs (Rajamanoharan et al., 2024). Our work focuses on overcoming the limitations of SAEs in representing tail concepts and proposes SSAEs to ensure a more balanced representation of both frequent and rare concepts.

Evaluating SAE performance is further complicated by the absence of ground truth labels for the features they learn. Existing research has employed diverse metrics, including comparison with ground truth features in toy data, activation reconstruction loss, L1 loss, number of alive dictionary elements, feature similarity across seeds and dictionary sizes (Sharkey et al., 2022), L0 sparsity, KL divergence upon causal interventions (Cunningham et al., 2023), reconstructed negative log likelihood (Cunningham et al., 2023; Bricken et al., 2023), feature interpretability (Bills et al., 2023), and task-specific comparisons (Makelov et al., 2024).

Our work utilizes a combination of these metrics, including L0 sparsity, reconstruction error, downstream perplexity, and automated interpretability evaluations. We also introduce new metrics specifically designed to assess the effectiveness of SSAEs in capturing rare, domain-specific concepts.

A.4 Disentangled Representations

Our research also connects to the broader field of disentanglement in representation learning (Bengio, 2013). While traditional disentanglement methods often rely on enforcing priors on learned representations (Chen et al., 2018; Kim and Mnih, 2018; Mathieu et al., 2019), SAEs aim to decompose the representation space of a pretrained language model into a sparse linear combination of an overcomplete basis. This approach aligns with the theory that language models implicitly learn disentangled representations of data with specific structures, which we seek to recover using sparse autoencoders.

B Evaluating SSAE for Physics on OOD data

Figure 9 depicts Pareto curves for SSAE trained with various data selection strategies as the sparsity coefficient is varied on the OOD Physics instruction test data. We find that both BM25 retrieval and training on the validation data generalize poorly when tested out of domain.

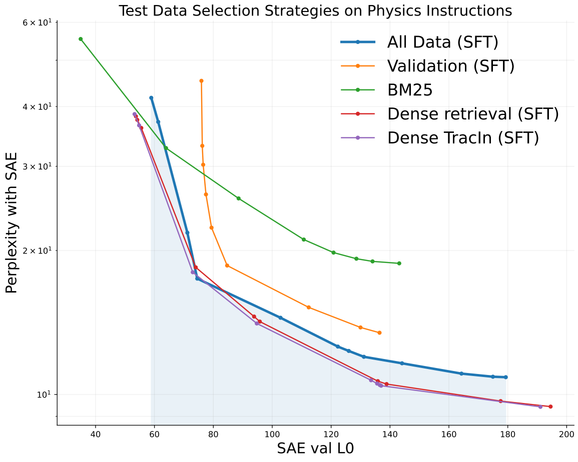
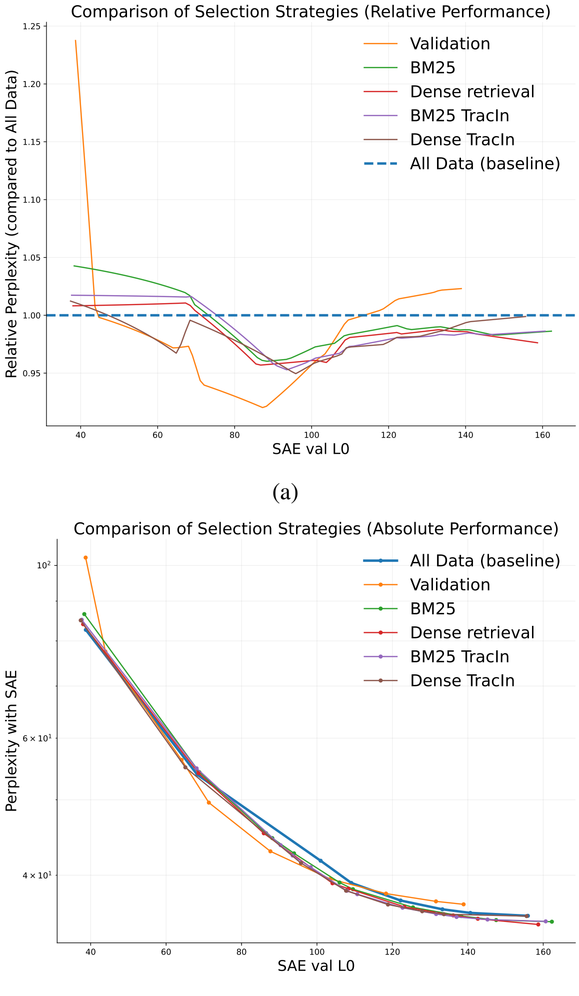


Figure 9: Pareto curves for SSAE trained with various data selection strategies as the sparsity coefficient is varied on Physics instruction test data. We plot absolute perplexity with the spliced in SSAE. We find that both BM25 retrieval and training on the validation data generalize poorly when tested out of domain. All curves are averaged over three SAE training run seeds.

C Evaluating Data Selection Strategies for Toxicity SSAEs

We use a seed concept dataset of 4072 tokens from the Pile Toxicity dataset (Korbak, 2024). We retrieve 5.25M tokens from OWT using the same strategies as before and train SSAEs on this data for 500 iterations. We then evaluate the models on a test split of 3.357M tokens from the Pile Toxicity dataset (in-distribution). Appendix Figure 10 displays the patched perplexity versus L_0 curves for these experiments. The results largely align with the physics experiment, with Dense retrieval outperforming BM25 and TracIn offering a marginal improvement over Dense retrieval alone.



(a)

(b)

Figure 10: Pareto curves for Toxicity SSAE trained with various data selection strategies as the sparsity coefficient is varied on Pile toxicity test data. We plot (a) Perplexity with spliced in SAE relative to a GSAE (Baseline) (b) Absolute Perplexity with the spliced in SSAE. Dense TracIn achieves the best performance, followed by Dense retrieval, BM25 TracIn, BM25 and OWT baseline. All curves are averaged over three SAE training run seeds.

D Probing SSAE Tail Concept Learning for Toxicity

Figure 11 shows the proportion of tokens with SAE features vs. Token frequency in Toxicity data using the Logit Lens approach. We leverage the unem-

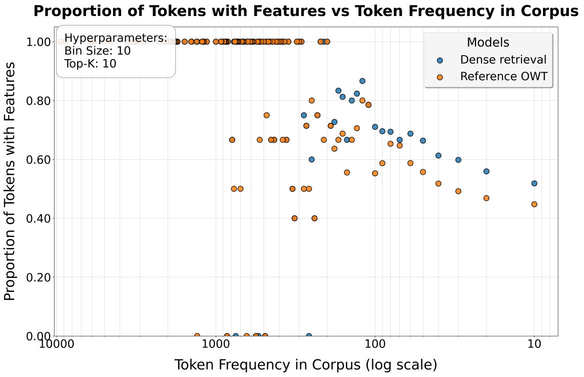


Figure 11: Proportion of tokens with SAE features vs. Token frequency in Toxicity data. SSAE trained with dense retrieval captures more tail tokens (concepts) in its features.

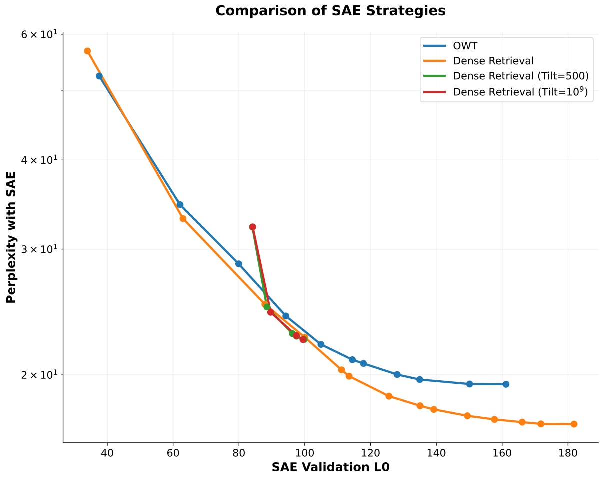


Figure 12: Pareto curves for SSAEs finetuned on the **Physics arXiv dataset** using different strategies: full OpenWebText (OWT), Dense retrieval, and Dense retrieval with Tilted Empirical Risk Minimization (TERM, tilt=500 and TERM, tilt=10^9). TERM-finetuned SSAEs achieve competitive performance with Dense retrieval alone within the L_0 range of 85-100. Outside this range, our current training methodology results in higher reconstruction errors. All curves are averaged over three SAE training run seeds.

bedding matrix as a logit lens to analyze the top-10 token logits associated with each SSAE feature. For each frequency bucket in the Toxicity dataset, we calculate the percentage of tokens that appear among the top-10 logits for at least one feature. This analysis allows us to assess the extent to which SSAE features represent tokens across different frequency ranges. SSAE trained with dense retrieval captures more tail tokens (concepts) in its features compared to the baseline.

E Pareto curves for Tilted ERM trained SSAE

Figure 12 evaluates SSAEs trained with Tilted ERM on the Physics arXiv dataset, displaying Pareto curves where the x-axis represents L_0 and the y-axis shows downstream perplexity with patched-in SSAE. TERM-finetuned SSAEs achieve competitive performance with Dense retrieval alone within the L_0 range of 85-100.

Figure 13 shows similar Pareto curves on the Pile toxicity dataset where TERM-finetuned SSAEs achieve competitive performance with Dense retrieval within the L_0 range of 100-140.

Our experiments demonstrate that TERM-trained SSAEs consistently maintain L_0 within this desired range, ensuring both sparsity and accurate reconstruction of subdomain concepts.

Improving L_0 Control at Extreme Values Adaptive penalty schemes are much better than Adam at precisely controlling L_0 at extreme values. This approach dynamically adjusts the sparsity penalty λ during training based on the current L_0. We found that increasing λ when L_0 exceeds a target range and decreasing it when L_0 falls below helped maintain the desired level of sparsity across a wider range of L_0 values. This also prevented the emergence of inactive features at low L_0 values.

F TERM-trained SSAE enhances Tail Concept Capture in Toxicity data

Figure 14 shows the cumulative proportion of tokens with SAE features vs. cumulative percentage of tokens in Toxicity data, normalized per model so that the cumulative proportion of tokens with features is 1 over the entire dataset. SSAE trained with dense retrieval and larger tilt captures more tail tokens (concepts) in its features.

G Implementation Details for Bias-in-Bios Classification Experiments

We follow the methodology in Marks et al. (2024) for Spurious Human-interpretable Feature Trimming (SHIFT), which we summarize here for completeness. All models can be trained on a single A100 in under a day.

G.1 Classifier Training

Here we describe our approach to training a classifier on Pythia-70M for the Bias in Bios (BiB) task. To mimic a realistic application setting, we conducted a hyperparameter search to train high-performing baseline and oracle classifiers (using the ambiguous and balanced datasets, respectively).

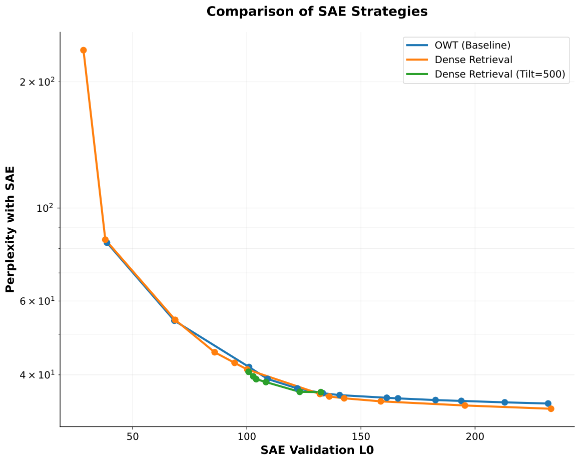


Figure 13: Pareto curves for SSAEs finetuned on the **Toxicity dataset** using different strategies: full OpenWebText (OWT), Dense retrieval, and Dense retrieval with Tilted Empirical Risk Minimization (TERM, tilt=500). TERM-finetuned SSAEs achieve competitive performance with Dense retrieval alone within the L_0 range of 100-140. All curves are averaged over three SAE training run seeds.

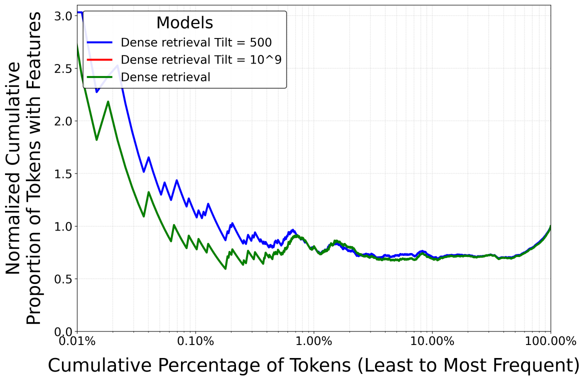


Figure 14: Cumulative proportion of tokens with SAE features vs. cumulative percentage of tokens in Toxicity data, normalized per model so that the cumulative proportion of tokens with features is 1 over the entire dataset. SSAE trained with dense retrieval and larger tilt captures more tail tokens (concepts) in its features. Note that the curves at tilt 500 and tilt 10^9 overlap.

Hyperparameters were not selected with the aim of strong SHIFT performance.

The inputs to our classifier are residual stream activations from the penultimate layer of Pythia-70M. We apply mean-pooling over (non-padding) tokens from the context. In our initial experiments, we found that extracting representations over only the final token led to slightly worse baseline and oracle performance. Similarly, using activations from Pythia-70M's final layer yielded slightly poorer results.

We then fit a linear probe to these representations using logistic regression. For optimization, we employ AdamW (Loshchilov, 2017) with a learning rate of 0.01, training for a single epoch. When retraining after SHIFT, we finetune only this linear probe, leaving the full model unchanged.

Like Marks et al. (2024), we encountered difficulties when attempting to fit a probe with greater-than-chance accuracy using logistic regression on final layer representations. This observation led us to opt for penultimate layer representations in our main approach.

G.2 Implementation for Concept Bottleneck Probing

Our implementation of Concept Bottleneck Probing (CBP) draws from Yan et al. (2023). The process is as follows:

1. First, we select $N = 20$ keywords related to the intended prediction task. Our keyword set includes: nurse, healthcare, hospital, patient, medical, clinic, triage, medication, emergency, surgery, professor, academia, research, university, tenure, faculty, dissertation, sabbatical, publication, and grant.
2. We obtain concept vectors $c_1, \ldots, c_N$ for each keyword by extracting Pythia-70M's penultimate layer representation over the final token of each keyword, then subtracting the mean concept vector. This normalization step proved crucial, as we found that without it, concept vectors exhibited very high pairwise cosine similarities.
3. Given an input with representation x (obtained via the mean-pooling procedure described earlier), we construct a concept bottleneck representation $z \in \mathbb{R}^N$ by computing the cosine similarity with each c_i.
4. Finally, we train a linear probe on these concept bottleneck representations z using logistic regression, following the approach outlined in the Classifier Training subsection.

As in Marks et al. (2024), we decided to normalize concept vectors but not input representations, as this approach yielded stronger performance. We also explored the alternative of computing cosine similarities before mean pooling.

H Sparse Feature Circuits for Bias in Bios Classifer

In this section, we generate sparse feature circuits, which are computational sub-graphs that explain model behaviors in terms of SAE features and error terms, using the methodology in Marks et al.

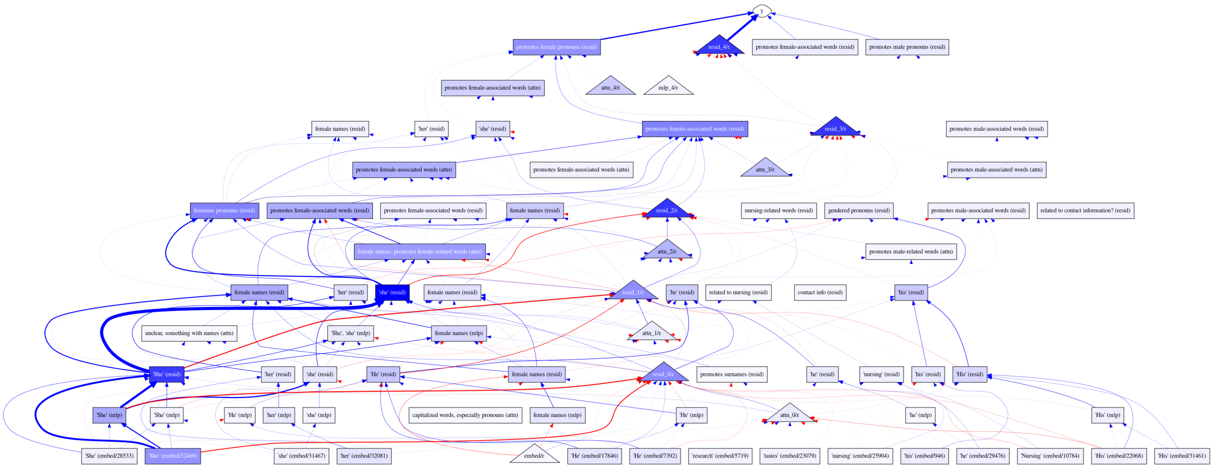


Figure 15: The full annotated feature circuit discovered for the Bias in Bios classifier with the **GSAE patched in**. The circuit was discovered using $T_N = 0.1$ and $T_E = 0.01$. We observe that the circuit contains many nodes that simply detect the presence of gendered pronouns or gendered names. A few features attend to profession information, including one which activates on words related to nursing, and another that activates on passages relating to science and academia.

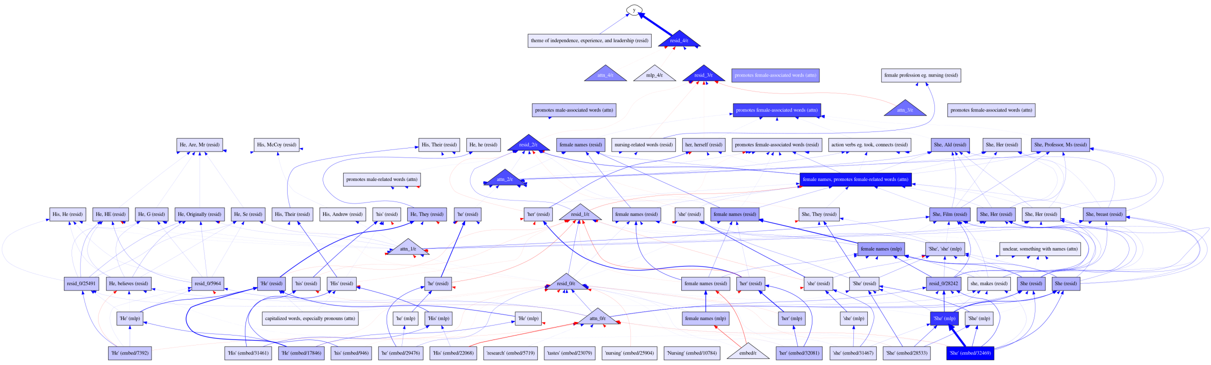


Figure 16: The full annotated feature circuit for the Bias in Bios classifier with the **finetuned SSAE patched in**. The circuit was discovered using $T_N = 0.1$ and $T_E = 0.01$. This circuit is much larger due to newly activated features in the SSAE that detect the presence of gendered pronouns and gendered names, as well as features for profession information such as nursing and academia.

(2024). We begin by describing the process of generating these circuits.

Given a language model M, SAEs for various submodules of M (e.g., attention outputs, MLP outputs, and residual stream vectors), a dataset D consisting of either contrastive pairs $(x_{\text{clean}}, x_{\text{patch}})$ of inputs or single inputs x, and a metric m depending on M's output when processing data from D, we can construct these circuits. The idea is to treat SAE features as part of the model. By applying the decomposition to various hidden states x in the LM, we can view the feature activations f_i and SAE errors ε as integral parts of the LM's computation. This allows us to represent the model as a computation graph G where nodes correspond to feature activations or SAE errors at particular token positions.

To approximate the Indirect Effect (IE) of each node, we compute $\hat{IE}(m; a; x)$ for each node a in G and input $x \sim D$, where $\hat{IE}$ is either $\hat{IE}_{\text{atp}}$ or $\hat{IE}_{\text{ig}}$. We then apply a node threshold T_N to select nodes with a large (absolute) IE. Consistent with prior work (Nanda, 2023; Kramár et al., 2024), we find that $\hat{IE}_{\text{atp}}$ accurately estimates IEs for SAE features and errors, except for nodes in the layer 0 MLP and early residual stream layers. For these components, $\hat{IE}_{\text{ig}}$ significantly improves accuracy, so we employ it in our experiments.

We also compute the average IE of edges in the computation graph using an analogous linear approximation. After computing these IEs, we filter for edges with absolute IE exceeding some edge threshold T_E.

For templatic data where tokens in matching positions play consistent roles, we take the mean effect of nodes/edges across examples. For non-templatic data, we first sum the effects of corresponding nodes/edges across token positions before taking the example-wise mean (Marks et al., 2024).

Figure 15 presents the full annotated feature circuit for the Bias in Bios linear classifier based on Pythia-70M with the pretrained GSAE patched in. The annotations are from human inspection of examples that activate features. Many nodes simply detect the presence of gendered pronouns or gendered names. A few features attend to profession information, including one which activates on words related to nursing, and another which activates on passages relating to science and academia.

Similarly, Figure 16 displays the full annotated feature circuit for the Bias in Bios linear classifier based on Pythia-70M with the finetuned SSAE patched in. This circuit, discovered using $T_N = 0.1$ and $T_E = 0.01$, is much larger due to newly activated features in the SSAE that detect the presence of gendered pronouns and gendered names, as well as features for profession information such as nursing and academia. This is responsible for the improved classification performance with the SSAE.

In each circuit, sparse features are shown in rectangles, whereas causally relevant error terms not yet captured by our SAEs are shown in triangles. Nodes shaded in darker colors have stronger effects on the target metric m. Blue nodes and edges are those which have positive indirect effects (i.e., are useful for performing the task correctly), whereas red nodes and edges are those which have counterproductive effects on m (i.e., cause the model to consistently predict incorrect answers).

I Relative Feature Activation Distribution

Figures 17 and 18 analyze the distribution of differences in feature activation counts between the same features in specialized SAEs (both ERM and TERM-trained) and the OWT baseline on the Physics arXiv test set. The difference is quantified as the log ratio of feature activation counts: $\log_2(\frac{M+1}{B+1})$, where M represents the SSAE and B the OWT baseline. Positive values indicate features activating on more data points in the specialized SAEs relative to the baseline SAE.

Finetuning on the subdomain with ERM leads to an increase in feature activation counts overall, as evidenced by the positive probability mass. This adaptation reflects the SSAE features specializing towards concepts prevalent in the Physics arXiv dataset.

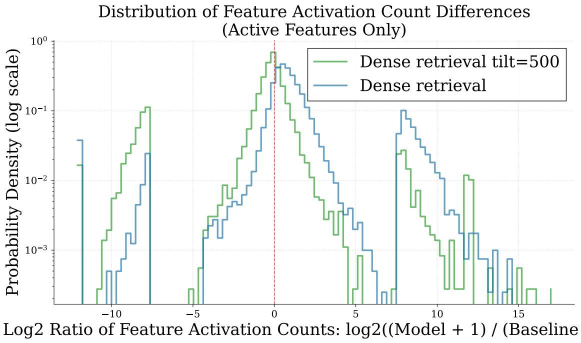


Figure 17: Distribution of log-ratio feature activation count differences between specialized SAEs and the OWT baseline on the Physics arXiv test set, normalized per SAE model. Blue represents the ERM-trained SSAE with Dense retrieval, orange represents the TERM-trained SSAE with tilt=500. The ERM-trained SSAE exhibits more probability mass on the right, indicating an emphasis on representing common concepts, while the TERM-trained SSAE's shift towards the left suggests a greater focus on representing domain-specific tail concepts.

Training SSAEs with TERM, which minimizes worst-case performance, distinctly alters feature activation patterns. Compared to standard ERM, TERM-trained SAEs concentrate more probability mass on the distribution's left side, indicating many features are less activated relative to the baseline. This leftward shift aligns with the theoretical underpinnings of TERM, which encourages robustness to distribution shift and tail events. By upweighting worse-performing examples, TERM promotes the activation of features crucial for capturing tail concepts. The TERM-trained SAE redistributes its capacity, with numerous features specializing in tail concepts (low-level activations), while others become more general activating on a wider range of concepts. This shift towards negative relative counts intensifies with increasing tilt, suggesting that higher tilt values further prioritize the representation of tail concepts.

J Automated Intepretability Explanations

Boxes 1, 2, and 3 show the Interpreter's explanations for the active features among the first ten features (by count) of the pretrained GSAE, the ERM-trained SSAE, and the TERM-trained SSAE, respectively, on the arXiv Physics test set. We observe a clear distinction in how these models specialize and represent concepts. While the ERM-trained SSAE activates more features than the GSAE, reflecting its focus on frequent concepts within the domain, its explanations are more complex and less readily interpretable. Conversely, the TERM-trained SSAE, despite activating fewer fea-

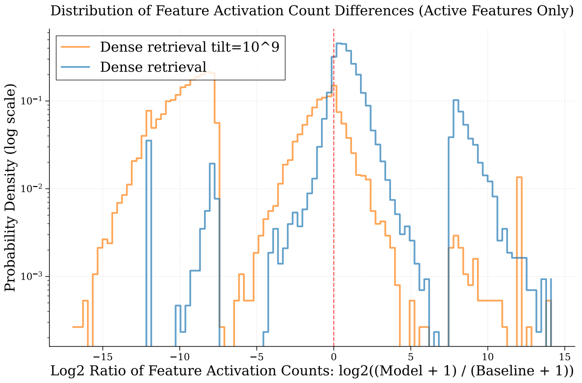


Figure 18: Distribution of log-ratio feature activation count differences on the Physics arXiv test set, normalized per SAE model. Blue represents the ERM-trained SSAE with Dense retrieval, orange represents the TERM-trained SSAE with tilt=10^9. The intensified leftward shift of probability mass with higher tilt demonstrates that TERM increasingly prioritizes representing tail concepts compared to standard ERM-trained SSAE, which focuses more on frequent concepts.

tures overall, produces explanations that are easier to understand. This suggests that TERM learns features that are compositional and encourages a balanced representation of both frequent and rare concepts. The lower number of active features for the TERM-trained SSAE could be attributed to the potential absence of many tail concepts in the test set.

K Automated Interpretability Prompts

In this section, we present the Interpreter and Predictor prompts used with Claude 3.5 Sonnet (`claude-3-5-sonnet-20240620`) in our automated interpretability pipeline. We note that all AutoInterp experiments cost less than $1000 to run.

K.1 Interpreter Prompt

The Interpreter prompt in Box 4 is designed to analyze SAE feature activations and explain what causes a specific feature to activate. It is given a list of text examples where the feature activates, with the activating tokens highlighted.

K.1.1 Example Application of Interpreter Prompt

Box 5 provides an example of how the Interpreter prompt is applied.

K.2 Predictor Prompt

The Predictor prompt in Box 6 is used to predict given a feature explanation whether the given text examples activate the feature. It returns a binary classification label for each example.

K.2.1 Example Application of Predictor Prompt

Box 7 provides an example of how the Predictor prompt is applied.

L Proof of Lower Description Length under Tilted ERM

We prove that training a Sparse Autoencoder (SAE) using Tilted ERM leads to a lower total description length compared to standard ERM under specific conditions, suggesting Tilted ERM produces more interpretable features according to the Minimum Description Length (MDL) principle.

L.1 Problem Setup and Assumptions

We consider a dataset $\mathcal{D} = \{x_i\}_{i=1}^N$, where each $x_i \in \mathbb{R}^d$ is generated from a mixture of two Gaussian distributions: a majority cluster (Cluster A) and a minority cluster (Cluster B). Cluster A has mean $\boldsymbol{\mu}_A = \mathbf{0}$, covariance $\Sigma_A = \sigma^2\mathbf{I}$, and proportion $q_A = N_A/N$. Cluster B has mean $\boldsymbol{\mu}_B = \boldsymbol{\delta}$ (where $\boldsymbol{\delta} = \delta\mathbf{1}$, $\delta > 0$), covariance $\Sigma_B = \sigma^2\mathbf{I}$, and proportion $q_B = N_B/N = 1 - q_A$. We assume $q_A \gg q_B$, reflecting a significant class imbalance often encountered in real-world scenarios.

The SAE consists of an encoder $h_i = Wx_i$ and a decoder $\hat{x}_i = W^\top h_i$, where $W \in \mathbb{R}^{k\times d}$ is the weight matrix and $h_i \in \mathbb{R}^k$ is the latent representation. Sparsity is enforced through an L_1 penalty in the loss function, defined as $L(x_i; W) = \|x_i - \hat{x}_i\|^2 + \lambda\|h_i\|_1$, where $\lambda > 0$ controls the trade-off between reconstruction error and sparsity. Assume the nonlinearity is always activated i.e., the identity function.

We compare two training objectives: standard ERM, which minimizes the average loss $\frac{1}{N}\sum_{i=1}^N L(x_i; W)$, and Tilted ERM, which approximates the minimization of the maximum loss through the objective $\frac{1}{\tau}\log(\sum_{i=1}^N e^{\tau L(x_i;W)})$ for large $\tau > 0$.

We make several simplifying assumptions. First, we assume binary latent codes, where $h_{ij} \in \{0, 1\}$. This assumption, while a simplification of continuous-valued activations, allows for a clearer analysis of feature interpretability through the lens of information theory. Second, we assume that features are activated independently, which, while not always true in practice, provides a tractable framework for our analysis. Lastly, we assume uniform activation probabilities across features within each cluster, which simplifies our calculations while still

Box 1: Generalized SAE

0. The token "0" appearing in scientific notation, journal article citations, or encoded ASCII representations, often in the context of physics or chemistry literature references.
5. This neuron appears to activate on mathematical and scientific notation, particularly symbols, equations, and specialized formatting in technical documents. It may play a role in recognizing and processing scientific or mathematical content within text.
7. The neuron appears to activate on punctuation marks, particularly commas and quotation marks, when they are used to separate or enclose items in a list, mathematical expressions, or technical notation in scientific or mathematical text. It may play a role in parsing and understanding the structure of complex technical writing.
8. This neuron appears to activate on tokens that are part of or follow noun phrases, often in technical or academic contexts. It seems to be sensitive to words that introduce or refer to specific objects, concepts, or pieces of information within a larger text. The neuron may play a role in tracking referential elements or key pieces of information in complex, information-dense text.
9. The token "," appearing after complex scientific or technical phrases, often preceding conjunctions or additional clauses that provide further explanation or context in academic or scientific writing.
10. This neuron appears to activate on abbreviated references to academic or scientific sources, particularly in bibliographies or citation lists. It responds to: 1. Abbreviated journal names (e.g. "NY", "APS", "Euro") 2. Abbreviated organization names (e.g. "SIAM", "INSPEC") 3. URL components of online references (e.g. "citeseer", "philsci", "biology-") 4. Abbreviated publisher names (e.g. "TERRAPUB") The neuron seems to play a role in recognizing citation patterns.

Box 2: Specialized SAE

0. The token "0" appearing in scientific paper citations, journal volume numbers, or ASCII code representations, often in the context of physics or mathematics literature.
4. This neuron appears to activate on tokens related to academic and scientific writing, particularly in the context of physics, science education, and the philosophy of science. It frequently activates on words like "universities", "science", "class", "theories", and other academic terminology. The neuron may be involved in recognizing and generating text related to scientific discourse and academic writing.
5. This neuron appears to activate on scientific and mathematical notation, particularly superscripts, subscripts, and special characters used in equations and formulas. It may play a role in processing and understanding technical or scientific text.
7. The token "by" often appears before introducing a variable, parameter, or label in mathematical or scientific text. It is frequently used to define or denote specific elements in equations, models, or experimental setups.
8. The neuron appears to activate on numerical digits, particularly the digit "4", within scientific or technical contexts such as citations, measurements, or equipment specifications. This suggests the neuron may play a role in identifying or processing numerical information in academic or technical writing.
9. The token "," after various phrases in scientific or technical writing, often used to separate clauses or elements in a list. This neuron may be detecting punctuation patterns in formal, academic-style text.
10. This neuron appears to activate on abbreviations and short identifiers in academic or scientific references, particularly those related to publications, databases, or online resources. Examples include "cites", "NY", "ZIN", "TER", "SI", "e-", "cond", "Compustat", "ASP", "IN", "CAS", "Physics", "Pren", "ourworld", "compuserve", and "APS". These often appear in bibliographic entries, URLs, or other citation-related contexts in academic writing.

Box 3: Specialized SAE with Tilt 500

0. The token "0" appearing in scientific notation, particularly in journal citations, volume numbers, and page numbers. This neuron may be involved in recognizing and processing numerical information in academic or scientific contexts.
5. This neuron appears to activate on mathematical and scientific notation, particularly equations, variables, and symbols. It seems to be sensitive to complex mathematical expressions, physical constants, and scientific formulas across various fields including physics, chemistry, and engineering. The neuron may play a role in processing and generating technical scientific content.
7. The neuron appears to activate on punctuation marks, particularly commas and angle brackets, when used to separate or enclose items in mathematical or scientific notation. It may play a role in parsing and understanding the structure of technical or mathematical text.
9. The token "," after phrases or clauses, often used to separate elements in scientific or technical writing. This neuron may be detecting punctuation patterns in formal, academic text.

Box 4: Interpreter Prompt

```
SYSTEM = """You are a meticulous AI researcher conducting an important investigation into a
↪  certain neuron in a language model. Your task is to analyze the neuron and explain what
↪  causes the neuron to activate.
{prompt}
Guidelines:
You will be given a list of text examples on which the neuron activates. The specific tokens
↪  which cause the neuron to activate will appear between delimiters like <<this>>. If a
↪  sequence of consecutive tokens all cause the neuron to activate, the entire sequence of
↪  tokens will be contained between delimiters <<just like this>>.
- You must produce a concise final description. Simply describe the text features that
↪  activate the neuron, and what its role might be based on the tokens it predicts.
- The last line of your response must be the formatted explanation.
- Think carefully about the patterns in the text examples and the tokens that activate the
↪  neuron. Pay attention to detail.
{subject_specific_instructions}"""
```

Box 5: Interpreter Example

```
EXAMPLE_1 = """
Example 1:  and he was <<over the moon>> to find
Example 2:  we'll be laughing <<till the cows come home>>! Pro
Example 3:  thought Scotland was boring, but really there's more
<<than meets the eye>>! I'd
"""
EXAMPLE_1_EXPLANATION = """
[EXPLANATION]: Common idioms in text conveying positive sentiment.
"""
```

Box 6: Predictor Prompt

```
DSCORER_SYSTEM_PROMPT = """You are an intelligent and
meticulous linguistics researcher.
You will be given a certain feature of text, such as
"male pronouns" or "text with negative sentiment".
You will then be given several text examples. Your task
is to determine which examples possess the feature.
For each example in turn, return 1 if the sentence is
correctly labeled or 0 if the tokens are mislabeled. You
must return your response in a valid Python list. Do not
return anything else besides a Python list.
"""
```

Box 7: Predictor Example

```
DSCORER_EXAMPLE_1 = """Feature explanation: "of" before words that start
with a capital letter.
Text examples:
Example 0: climate, Tomblinâ Chief of Staff Charlie Lorensen said.
Example 1: no wonderworking relics, no true Body and Blood of Christ,
no true Baptism
Example 2:Deborah Sathe, Head of Talent Development and Production
at Film London,
Example 3: It has been devised by Director of Public Prosecutions (DPP)
Example 4: and fair investigation not even include the Director of
Athletics? Finally, we believe the
"""
DSCORER_RESPONSE_1 = "[1,1,1,1,1]"
```

capturing the essential dynamics of the system.

L.2 Description Length and Feature Activation Probabilities

The total description length is given by $DL_{\text{total}} = DL_{\text{model}} + DL_{\text{data}}$. Since DL_{model} is the same for both ERM and Tilted ERM (assuming identical model capacity), we focus our analysis on DL_{data}, which represents the description length of the latent representations $\{h_i\}$. For a binary latent vector h_i, the description length is given by:

$$DL(h_i) = \sum_{j=1}^{k} -\Big(h_{ij}\log_2 P(h_{ij}=1) + (1-h_{ij})\log_2 P(h_{ij}=0)\Big) \tag{2}$$

Given our assumption of independent features, the expected description length per data point from Cluster C ($C \in \{A, B\}$) is:

$$DL_C = k \cdot H(p_C) \tag{3}$$

where p_C is the activation probability for features in Cluster C, and $H(p)$ is the binary entropy function: $H(p) = -p\log_2 p - (1-p)\log_2(1-p)$. The total description length for the data is thus:

$$\begin{aligned} DL_{\text{data}} &= N_A DL_A + N_B DL_B \\ &= N_A k H(p_A) + N_B k H(p_B) \end{aligned} \tag{4}$$

Our goal is to show that under certain conditions, $DL_{\text{data}}^{\text{Tilted}} < DL_{\text{data}}^{\text{ERM}}$.

L.3 Analysis of ERM vs. Tilted ERM

Under standard ERM, the SAE focuses on minimizing the average loss, which is dominated by Cluster A due to its larger size. This leads to features being optimized primarily to represent Cluster A well. For Cluster B, the reconstruction error is typically higher, leading to less sparse representations (higher p_B). This occurs because the network attempts to compensate for poor reconstruction by activating more features, even if they're not ideally suited to the minority cluster's characteristics.

In contrast, Tilted ERM focuses on minimizing the maximum loss, giving more attention to Cluster B. This approach leads to features being adjusted to better represent both clusters. As a result, we expect a slight increase in activation probabilities for Cluster A (p_A increases slightly) as the network makes minor adjustments to accommodate Cluster B. Importantly, we anticipate a significant decrease in activation probabilities for Cluster B (p_B decreases significantly) as the features become more tailored to its characteristics, allowing for sparser and more efficient encoding.

The relationship between feature activation probabilities and reconstruction error is key to understanding the dynamic between ERM and TERM. Lower reconstruction error is associated with lower activation probabilities, as the network can more efficiently encode the input data. Conversely, higher reconstruction error often leads to higher activation probabilities as the network *struggles* to represent the data, activating more features in an attempt to reduce the error.

L.4 Quantitative Analysis

To formalize this analysis, let us denote the activation probabilities under ERM as $p_A^{\text{ERM}} = p_A$, $p_B^{\text{ERM}} = p_B$; and under Tilted ERM as $p_A^{\text{Tilted}} = p_A + \Delta p_A$, $p_B^{\text{Tilted}} = p_B - \Delta p_B$. Here, $\Delta p_A > 0$ is small, reflecting the minor adjustments made to Cluster A's representation, while $\Delta p_B > 0$ is significant, capturing the substantial improvement in Cluster B's encoding. The difference in total description length is then:

$$\begin{aligned} \Delta DL &= DL_{\text{data}}^{\text{ERM}} - DL_{\text{data}}^{\text{Tilted}} \\ &= N_A k \left(H(p_A^{\text{ERM}}) - H(p_A^{\text{Tilted}})\right) \\ &\quad + N_B k \left(H(p_B^{\text{ERM}}) - H(p_B^{\text{Tilted}})\right) \end{aligned} \tag{5}$$

Defining $\Delta H_A = H(p_A^{\text{ERM}}) - H(p_A^{\text{Tilted}})$ and $\Delta H_B = H(p_B^{\text{ERM}}) - H(p_B^{\text{Tilted}})$, we can express this as:

$$\Delta DL = k\left(N_A \Delta H_A + N_B \Delta H_B\right) \tag{6}$$

Our aim is to show that $\Delta DL > 0$ under specific conditions.

L.5 Conditions for Lower Description Length under Tilted ERM

For Tilted ERM to yield a lower total description length, we require:

$$N_A \Delta H_A + N_B \Delta H_B > 0 \tag{7}$$

Given that $N_A \gg N_B$, ΔH_A is small and negative (since p_A increases slightly), while ΔH_B is large and positive (since p_B decreases significantly), this condition can be satisfied if:

$$\frac{\Delta H_B}{|\Delta H_A|} > \frac{N_A}{N_B} \tag{8}$$

This inequality summarizes the core of our argument: if the decrease in entropy for Cluster B (per data point) is sufficiently large compared to the increase in entropy for Cluster A, weighted by their respective sample sizes, then Tilted ERM will lead to a lower total description length.

L.6 Numerical Illustration

To illustrate this condition, consider a scenario where $p_A^{\text{ERM}} = 0.1$, $p_A^{\text{Tilted}} = 0.12$, $p_B^{\text{ERM}} = 0.5$, and $p_B^{\text{Tilted}} = 0.1$, with $N = 1000$, $N_A = 900$, and $N_B = 100$. Computing the binary entropy values and their differences, we find:

$$\Delta H_A = H(0.1) - H(0.12) = -0.031 \text{ bits}$$
$$\Delta H_B = H(0.5) - H(0.1) = 0.531 \text{ bits}$$

The total difference in description length is then:

$$\begin{aligned}\Delta DL &= k\,(900 \times (-0.031) + 100 \times 0.531) \\ &= k \times 25.2 \text{ bits}\end{aligned}$$

This positive value of ΔDL demonstrates that, in this example, the total description length is indeed lower under Tilted ERM.

L.7 Implications

This proof demonstrates that under specific conditions—namely, when Tilted ERM significantly reduces the activation probabilities for the minority cluster while only slightly increasing them for the majority cluster—the total description length is lower under Tilted ERM compared to standard ERM. According to the MDL principle, which posits that models with lower total description length are preferable, this result implies that Tilted ERM leads to more interpretable features.

The key insight is that Tilted ERM's focus on minimizing the maximum loss allows it to develop features that more efficiently encode both the majority and minority clusters. While this may come at the cost of a slight increase in description length for the majority cluster, the substantial decrease in description length for the minority cluster more than compensates, leading to an overall improvement in feature interpretability.

It's important to note that this analysis relies on several simplifying assumptions, including binary latent codes, independent features, and uniform activation probabilities within clusters. In practice, the actual changes in activation probabilities will depend on the specific data distribution and optimization dynamics. Nonetheless, this theoretical result provides valuable insight into how Tilted ERM can lead to models with enhanced interpretability, particularly in scenarios involving imbalanced datasets.

In future work we could focus on relaxing these assumptions, exploring the implications of continuous-valued latent representations, and investigating the relationship between feature interpretability and other metrics of model performance and fairness. Empirical studies could provide further validation of these theoretical findings across a range of real-world datasets and tasks.

TERM, at high values of the tilt parameter t, can be viewed as minimizing the maximum loss across all data points. Under the assumption that the loss function is proportional to the negative log-likelihood, this becomes equivalent to minimizing the maximum description length in the Minimum Description Length (MDL) framework. In other words, at high tilt, Tilted ERM minimizes the maximum description length for the most poorly represented data points, ensuring that no single data point incurs an excessively long encoding.

This is particularly important in safety-critical applications, such as the detection of rare but hazardous features or circuits. In such cases, these rare features may be infrequent in the dataset and thus underrepresented when training with standard ERM, leading to high description lengths that make detection more difficult. By minimizing the maximum description length through Tilted ERM, these rare safety-relevant features are represented more efficiently, leading to more compact encodings that facilitate their detection and analysis. This improves both the interpretability and reliability of the model, enabling more robust identification of critical features in safety audits or interpretability studies, where compact and clear representations are essential for ensuring that important safety-related circuits are not overlooked.

M Applications of Tilted ERM SAEs in Capturing Tail Concepts

Sparse Autoencoders trained using ERM focus on minimizing the average reconstruction error across all inputs, leading to strong performance on frequent patterns but poor representation of rare or difficult-to-reconstruct activations. In contrast, SAEs trained via Tilted ERM emphasize reducing the reconstruction error of high-error examples, enabling better capture of rare concepts, improved handling of fine-grained detection tasks, and enhanced performance in high-stakes applications where edge cases are critical. Some applications of TERM-trained SSAEs include:

M.1 Capturing Tail Concepts in Multilingual Models

TERM-trained SAEs offer a significant advantage in capturing *rare linguistic patterns*, such as those found in multilingual or dialect-rich datasets. Foundation models (FMs) trained on predominantly English data often struggle to accurately represent less common languages or dialects. While an ERM-trained SAE might prioritize frequent language patterns, a Tilted ERM-trained SAE focuses on reducing reconstruction error for high-error examples, including these rare language patterns.

This approach is particularly important in *multilingual models* used in global applications, where inclusivity and fairness across languages are essential. For example, in a multilingual chatbot, Tilted ERM ensures that features for *low-frequency languages* such as Swahili or Icelandic are reconstructed more accurately, providing a better user experience for speakers of these languages.

M.2 Fine-Grained Anomaly Detection in High-Stakes Applications

TERM-trained SAEs excel in *fine-grained anomaly detection*, where identifying subtle deviations from normal behavior is crucial, especially in high-stakes applications such as security, finance, or medical diagnosis. While an ERM-trained SAE might flag anomalies due to their higher-than-average reconstruction error, it is less likely to represent these rare events with sufficient accuracy or disambiguate them from other errors for further analysis.

A Tilted ERM SAE allocates more capacity to rare, high-error cases, ensuring better reconstruction of these outliers. This improves both *detection* and *interpretability* of rare anomalies. For instance, in a financial fraud detection system, a Tilted ERM SAE can better capture the subtle patterns that differentiate fraudulent transactions from normal activity, leading to more effective interventions.

M.3 Improved Coverage of Rare Concepts in Fairness and Bias Mitigation

Tilted ERM improves fairness by ensuring that *underrepresented groups* or *tail concepts* are well represented in the model. In many real-world datasets, certain demographic or cultural groups may be underrepresented, leading to *bias in language models*. Tilted ERM ensures that rare patterns—including those associated with underrepresented languages, cultural references, or peoples—are better captured.

This approach leads to a more inclusive model that provides fairer representations across different groups. For example, a Tilted ERM SAE trained on a language model used in customer support applications can better represent *minority dialects* or *regional slang*, reducing bias in customer interactions and improving service for all users.

M.4 Robustness in Safety-Critical Systems

In safety-critical systems, such as *autonomous vehicles* or *aviation control*, rare but dangerous events must be handled with high accuracy. Tilted ERM, by focusing on minimizing the reconstruction error for the most difficult cases, ensures that the model is better equipped to handle rare, high-risk scenarios.

For example, in an autonomous driving system, rare but critical inputs such as *uncommon weather conditions* or *unusual road hazards* are more likely to be accurately captured by a Tilted ERM SAE. This improved representation helps the system react more reliably to rare but potentially dangerous situations, enhancing overall safety.

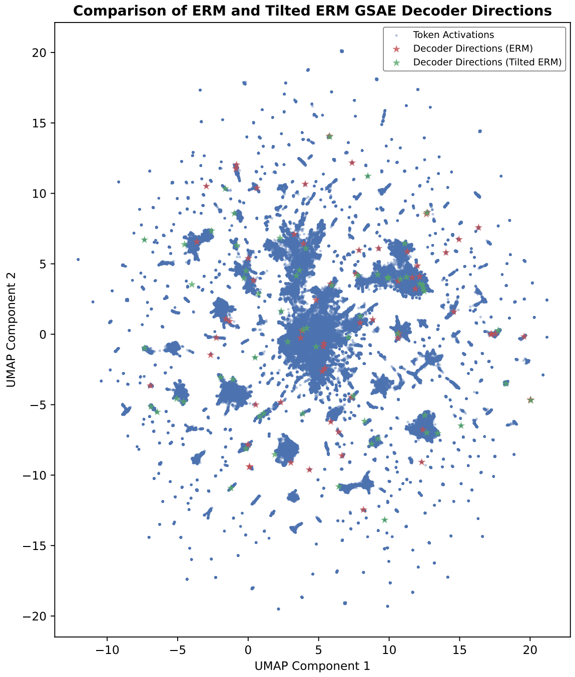


Figure 19: UMAP visualization of token activations and decoder features for a TERM-trained and ERM-trained GSAE. Decoder directions for TERM-trained GSAE appear more spread out, suggesting the SAE has wider coverage than the ERM-trained GSAE.

N TERM-trained GSAE Features on TinyStories

N.1 UMAP Plot of Decoder Directions

Figure 19 plots the UMAP visualization of token activations and decoder features for a TERM-

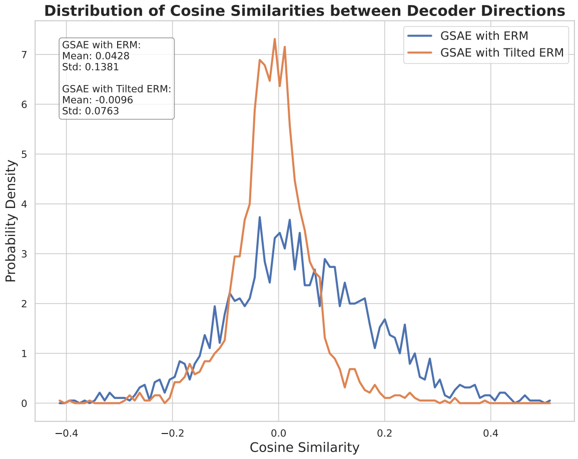


Figure 20: Distribution of cosine similarities between decoder directions of TERM-trained and ERM-trained GSAEs. TERM-trained GSAE shows lower similarity between decoder feature directions implying greater coverage.

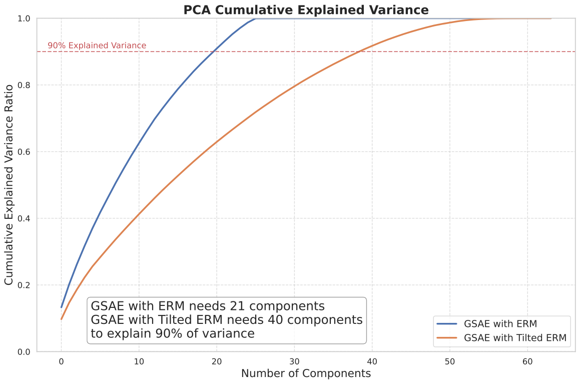


Figure 21: Number of PCA components required to explain variance in decoder feature directions of TERM-trained and ERM-trained GSAEs. TERM-trained GSAE shows greater variance in decoder feature directions implying greater coverage.

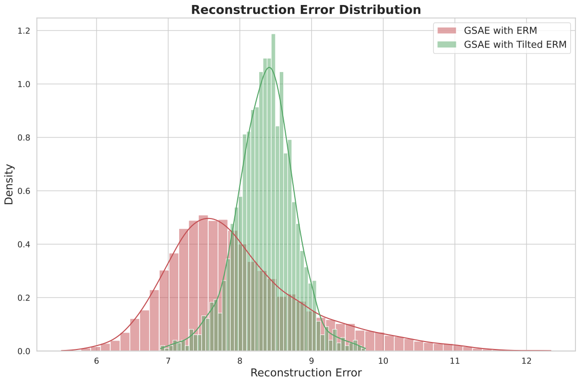


Figure 22: Reconstruction error distribution of TERM-trained and ERM-trained GSAE. TERM-trained GSAE minimizes the maximum error at the cost of average error.

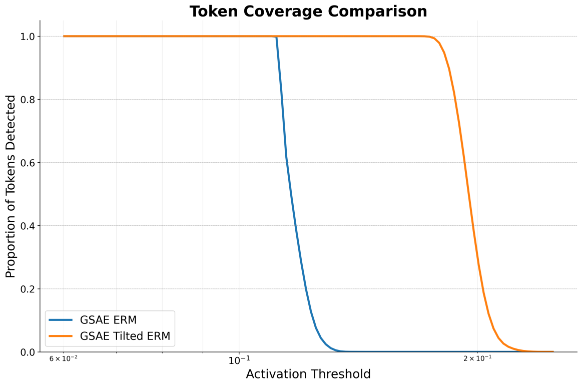


Figure 23: Proportion of tokens detected vs. activation threshold for TERM-trained and ERM-trained GSAEs. TERM-trained features exhibit stronger activations.

trained and ERM-trained GSAE.

N.2 Decoder Cosine Similarities

Figure 20 plots the distribution of cosine similarities between decoder directions of TERM-trained and ERM-trained GSAEs.

N.3 PCA Components to Explain Variance

Figure 21 plots the number of PCA components required to explain variance in decoder feature directions of TERM-trained and ERM-trained GSAEs. TERM-trained GSAE shows greater variance in decoder feature directions implying greater coverage.

N.4 Reconstruction Error

Figure 22 plots the reconstruction error distribution of TERM-trained and ERM-trained GSAE on 5M tokens sampled from TinyStories. TERM-trained GSAE minimizes the maximum error at the cost of average error.

N.5 Feature Activation Threshold for TERM-trained and ERM-trained GSAEs

Figure 23 plots the proportion of tokens detected vs. activation threshold on 5M tokens from TinyStories for TERM-trained and ERM-trained GSAEs.

N.6 Feature Diversity

When comparing the feature diversity score distribution of TERM-trained GSAE with ERM-trained GSAE in Figure 4, we observe that TERM-trained GSAE induces some features to specialize in tail concepts, while others generalize to represent a broader range of concepts relative to the ERM-trained GSAE.

To generate this plot, we first extract explanations for features based on the input examples that

Box 8: Feature Explanation Aggregation Prompt

```
You are an AI assistant tasked with unifying multiple explanations for a single feature in a language
model. These features are from the TinyStories dataset, which consists of short stories using simple
vocabulary. Your goal is to create a concise explanation that captures the essence of all the
individual explanations.

Individual explanations:
{chr(10).join(f”{i+1}. {exp}” for i, exp in enumerate(explanations))}

Please provide a unified explanation that:
1. Provides a clear and concise description of the feature's function or role in the context of the
TinyStories dataset and the language model. Include 2-3 brief examples of how this feature
might manifest in the stories.
```

Box 9: Diversity Score Generation Prompt

```
You are an AI assistant tasked with unifying multiple explanations for a single feature in a language
model. These features are from the TinyStories dataset, which consists of short stories using simple
vocabulary. Your goal is to create a concise explanation that captures the essence of all the
individual explanations.

Individual explanations:
{chr(10).join(f”{i+1}. {exp}” for i, exp in enumerate(explanations))}

Please provide a unified explanation that:
1. Provides a clear and concise description of the feature's function or role in the context of the
TinyStories dataset and the language model. Include 2-3 brief examples of how this feature might
manifest in the stories.
2. Scores the diversity of the feature's activations on a scale of 1 to 100, where:
  - 1-20: Very low diversity (e.g., a specific feature that only activates for a specific character
    name like ”Tom”)
  - 21-40: Low diversity (e.g., a less generic feature that activates for different character names,
    but only names)
  - 41-60: Moderate diversity (e.g., a generic feature that activates for various types of objects
    found in a home)
  - 61-80: High diversity (e.g., a generic feature that activates for different types of actions, both
    physical and verbal)
  - 81-100: Very high diversity (e.g., a generic feature that activates across various story elements:
    characters, actions, settings, emotions, dialogue)
Note: Consider the full range of possibilities within the TinyStories dataset. Don't hesitate to use
the full scale from 1 to 100 based on your analysis even if they all pertain to children's stories
since this is the dataset we are evaluating.

Unified explanation:
[Your unified explanation with 2-3 examples]

Diversity Score: [1-100]
Justification:[Brief justification for the score, considering the context of the TinyStories dataset]
```

activate them, using the prompt detailed in Box 4. To understand the behavior of all features, particularly those representing tail concepts, we cannot use random sampling of the TinyStories dataset, as employed in prior work since this would not capture tail concepts effectively. Instead, we process the entire TinyStories dataset in chunks of 5 million data points, and generate explanations by sampling uniformly from the top 50% examples that activate a feature. We then aggregate the explanations from each chunk using the explanation aggregation prompt provided in Box 8.

After aggregating feature explanations across dataset chunks, we derive the diversity score. This score is obtained using the score generation prompt presented in Box 9, implemented with Claude 3.5 Sonnet (`claude-3-5-sonnet-20240620`).

O Dataset Details

All datasets used in this study are in English. Below are the details for each dataset:

OpenWebText (OWT) A large-scale, diverse corpus of web content derived from URLs shared on Reddit. We use a single split comprising approximately 8 million documents and over 40GB of text data.[1]

Pile We utilize 2B tokens from the Pile dataset, a large-scale curated corpus designed for language model training. This subset contains 10.8M examples across various domains including academic writing, code, and web content.[2]

TinyStories A dataset of simple, coherent stories generated specifically for language model research. It consists of a single split containing 2.12M training examples, designed to be semantically meaningful while using limited vocabulary.[3]

arXivPhysics A collection of physics papers from arXiv. We use the first five examples, comprising 4.8M tokens. The full dataset contains 15.8k rows, split into 60% train, 20% validation, and 20% test, representing a broad range of physics topics.[4]

Physics Instruction Tuning A specialized dataset for physics-related instruction tuning. We use all 700K tokens from this dataset, which contains 30k examples of physics questions, explanations, and problem-solving instructions.[5]

Pile Toxicity A curated subset of the Pile dataset focusing on toxic content, designed for studying and mitigating harmful language in language models. We employ a 60-20-20 train-validation-test split to ensure balanced evaluation.[6]

Bias in Bios A dataset of online biographies used to study gender bias in machine learning models. It contains 257k training examples, 39.6k validation examples, and 99.1k test examples, providing a rich source for analyzing gender representation in professional contexts.[7]

P Computational Resources

Our experiments were conducted using modest computational resources, showing the accessibility of our approach. All experiments, including:

- Finetuning SSAEs on OpenWebText, Physics-arXiv, Toxicity data, and Pile datasets
- Training the Pythia-70M classifier and other baselines for the Bias in Bios task
- Pretraining GSAEs on the TinyStories dataset

were completed using 4 NVIDIA A100 GPUs or A6000 GPUs in less than 24 hours.

Q TERM-trained and ERM-trained GSAE features on TinyStories

We present a qualitative analysis of the feature explanations derived from both TERM-trained and ERM-trained GSAEs on the TinyStories dataset. Four explanations for each SSAE are shown in Tables 2 and 3.

TERM-trained GSAE TERM-trained GSAEs exhibit a fascinating mix of features, some capturing broad conceptual themes while others specialize in highly specific linguistic patterns. This duality stems from TERM's objective of minimizing the maximum loss, encouraging the SAE to learn features that can effectively reconstruct both frequent and rare examples.

[1] https://huggingface.co/datasets/Skylion007/openwebtext

[2] https://huggingface.co/datasets/NeelNanda/pile-small-tokenized-2b

[3] https://huggingface.co/datasets/roneneldan/TinyStories

[4] https://huggingface.co/datasets/anonymousdatasets/arxiv-physics

[5] https://huggingface.co/datasets/AlgorithmicResearchGroup/arxiv-physics-instruct-tune-30k

[6] https://huggingface.co/datasets/tomekkorbak/pile-toxicity-balanced

[7] https://huggingface.co/datasets/LabHC/bias_in_bios

Feature h.7_feature17, for example, is remarkably broad, described as processing “text related to children’s stories, simple narratives, and basic concepts in children’s literature.” This wide scope allows it to represent various story elements, from character actions and emotions to dialogue and sensory experiences, reflecting TERM’s focus on capturing the full spectrum of data patterns.

In contrast, feature h.7_feature8 demonstrates TERM’s ability to learn highly specific features. It activates exclusively on the indefinite article “an” when introducing new elements in a story, suggesting its role in recognizing a distinct grammatical pattern within the TinyStories dataset. This specific feature might capture a unique characteristic of the data or potentially represent a less frequent but important narrative element.

ERM-trained GSAE ERM-trained GSAE features, on the other hand, tend towards greater specificity, reflecting ERM’s focus on minimizing the average reconstruction error. This leads to features that accurately represent the most common patterns in the data but might struggle to capture tail concepts effectively.

For instance, feature h.7_feature3 in the ERM-trained GSAE is tailored to recognizing “narrative structures in simple, moralistic children’s stories.” While it encompasses a range of story elements, its scope remains constrained to a specific type of narrative common within the TinyStories dataset. This contrasts with the broader TERM feature h.7_feature17, which captures the essence of children’s stories more generally.

Implications This qualitative analysis suggests that TERM, by balancing broad and specific features, encourages the learning of more compositional representations, potentially improving the SAE’s ability to detect and interpret a wider variety of concepts, including rare or underrepresented ones. ERM’s emphasis on specificity, while effective for frequent patterns, may limit the SAE’s capacity to accurately represent the full spectrum of data patterns, particularly those found in the tail of the distribution.

Table 2: ERM-trained GSAE Features

Feature	**Explanation**
h.7_feature3	Unified explanation: This neuron recognizes narrative structures in simple, moralistic children’s stories. It activates on new story segments, character introductions, settings, conflicts, and dialogue. Frequent themes include lessons on kindness, honesty, and sharing. Examples: 1. "Lily woke up early on Saturday morning. ’Mom, can I go play with my friend Jenny?’ she asked." 2. "Once upon a time, there was a little boy named Tommy who loved to play with his toys but never wanted to share." 3. "After school, Timmy came home feeling sad. ’What’s wrong?’ his mom asked. ’I got in trouble for not telling the truth,’ Timmy replied." Diversity Score: 71 Justification: Activates on diverse narrative elements in children’s stories, including dialogue, character introductions, settings, events, emotions, and moral lessons. High diversity within the genre of educational stories for young audiences.
h.7_feature5	Unified explanation: This neuron activates on language patterns associated with conveying moral lessons, advice, and guidance on appropriate behavior in children’s stories or parental scenarios. It frequently fires on modal verbs like "should" and "can" when characters are learning about right and wrong actions, facing consequences, or being instructed on proper conduct. Examples: 1. "You should not take things that don’t belong to you," said Mom, after catching Timmy taking a candy bar from the store. 2. "The little boy learned that he can be kind to others by sharing his toys." 3. "If you can’t say something nice, you should not say anything at all," advised the teacher to the rowdy class. Diversity Score: 68 Justification: While specializing in moral lessons and guidance, the range of potential lessons, advice, and behavioral instructions is quite broad. It activates across various story elements and moral themes, encompassing a diverse array of instructional language in children’s literature.
h.7_feature6	Unified Explanation: This neuron activates when "<|endoftext|>" is followed by the beginning of a short, simple story or narrative, often with a moral lesson, cautionary tale, or tragic ending. These stories frequently feature children or animals as main characters, written in a style suitable for young readers. Examples: 1. "<|endoftext|> Once upon a time, there was a little girl who loved to play in the forest. One day, she wandered too far from home and got lost..." 2. "<|endoftext|> A group of young animals decided to explore the old abandoned barn, despite their parents’ warnings. But it was too late when they realized the danger inside..." 3. "<|endoftext|> Tommy was a curious boy who couldn’t resist the temptation of the old well in his backyard. He leaned over too far and..." Diversity Score: 61 Justification: While specific to children’s stories, the diversity is high, involving various characters, settings, actions, and themes. It captures a range of narrative elements, including plot structure, character archetypes, and common literary devices.
h.7_feature12	Unified explanation: This neuron activates at the beginning of short stories or narratives aimed at children. The consistent trigger is the token "<|endoftext|>", indicating the start of a new text sample. It recognizes the opening of simple narrative structures, often involving young protagonists, animal characters, moral lessons, or elements of danger or misfortune. Examples: 1. "<|endoftext|> Once upon a time, there was a little girl named Lily who loved to explore the enchanted forest near her home." 2. "<|endoftext|> In a cozy burrow, a family of rabbits lived happily until a hungry fox threatened their safety." 3. "<|endoftext|> Tommy the turtle was always in a hurry, but his impatience nearly cost him his life when he wandered too far from home." Diversity Score: 71 Justification: While focused on children’s stories, the range of possible stories and themes is quite diverse, involving different characters, settings, plots, and outcomes.

Table 3: TERM-trained GSAE Features

Feature	Explanation
h.7_feature8	Unified explanation: This feature detects the indefinite article "an" when introducing new or significant elements in children's stories or simple narratives. It activates when "an" precedes a noun at the beginning of a sentence or clause, signaling a novel element important to the plot. Examples: 1. "An old man lived in a tiny house by the forest." 2. "One day, an unexpected visitor arrived at the village." 3. "Deep in the ocean, an ancient treasure awaited discovery." Diversity Score: 65 Justification: High diversity in types of elements introduced (characters, objects, concepts) within children's stories, but limited to narrative contexts.
h.7_feature13	Unified explanation: This feature captures interjections or exclamations in children's stories or dialogues expressing surprise, excitement, or drawing attention to something noteworthy. Tokens like "Wow" or "Look" often appear at the beginning of quoted speech or exclamations. Examples: 1. "Wow! Look at that giant castle!" a child might exclaim upon seeing an impressive structure. 2. "Look, the caterpillar turned into a butterfly!" a character might say, pointing out a transformation. 3. "Wow, that was a close one!" someone might remark after narrowly avoiding danger. Diversity Score: 71 Justification: While specific to interjections, these can be used across a wide range of contexts and story elements, reflecting a high degree of diversity within children's stories and dialogues.
h.7_feature14	Unified explanation: This neuron predicts words related to pleasant or appetizing food experiences in children's stories or simple narratives. It activates on the first few letters of words like "yummy", "candy", "crumbs", and "celery", generating vocabulary associated with tasty treats, cooking, or domestic activities. Examples: 1. "The little girl licked her lips as she stared at the yummy chocolate cake." 2. "After playing outside, the kids ran to the kitchen for a snack of celery and peanut butter." 3. "Mom swept up the crumbs from the cookies the children had enjoyed earlier." Diversity Score: 53 Justification: While primarily focused on food-related words, it recognizes a range of vocabulary including adjectives, nouns, and verbs related to food experiences in children's stories.
h.7_feature17	Unified explanation: This neuron processes text related to children's stories, simple narratives, and basic concepts in children's literature. It responds to character names, diminutives, dialogue markers, sensory experiences, emotions, onomatopoeias, common objects, food items, childhood experiences, simple actions, and basic vocabulary. Examples: 1. "Ducky waddled over to the lollipop on the ground. 'Yum!' he exclaimed, gobbling it up." 2. "Ow, ow, ow! Timmy had scraped his knee on the rough sand. Mom kissed it better and gave him a sausage to cheer him up." 3. "Bark, bark! Spidey's new puppy was digging in the garden, scattering the soil everywhere. 'No, no, pup!' scolded Spidey." Diversity Score: 85 Justification: Displays very high diversity within children's literature, responding to a wide range of elements including characters, emotions, actions, objects, sensory experiences, and dialogue patterns.