HIERARCHICALSAE: DYNAMIC FEATURE LEARNING FOR ROBUST NEURAL INTERPRETABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding the internal representations of large language models is crucial for ensuring their reliability and safety, with sparse autoencoders (SAEs) emerging as a promising interpretability approach. However, current SAEs suffer from feature absorption, where learned features inconsistently activate across similar contexts, limiting their reliability for model analysis. We introduce HierarchicalSAE, which combines linear sparsity scheduling with dynamic feature clustering to prevent absorption while maintaining interpretability. Our key innovation is an adaptive sparsity mechanism that automatically adjusts constraints based on reconstruction quality, supported by a sliding window approach for monitoring feature stability. Experiments on Gemma-2-2B demonstrate substantial improvements over standard SAEs: 34% reduction in feature absorption (from 0.0088 to 0.0058), 7.8 percentage point increase in explained variance (90.2% vs 82.4%), and 15.2 percentage point gain in sparse probing accuracy (85.3% vs 70.1%). The model maintains 5.4x more active features (1,735 vs 320) while achieving better model preservation (KL divergence 0.058 vs 0.103), enabling more comprehensive and reliable analysis of neural network behavior.

1 INTRODUCTION

As large language models become increasingly powerful and ubiquitous, understanding their internal representations is crucial for ensuring safety and reliability. Sparse autoencoders (SAEs) have emerged as a promising approach for decomposing neural activations into interpretable features Gao et al., enabling researchers to analyze model behavior at a mechanistic level. However, current interpretability techniques face significant challenges in providing consistent and reliable insights into model behavior, particularly as models scale to billions of parameters.

A fundamental challenge in neural network interpretability is feature absorption Chanin et al. (2024), where learned features fail to activate consistently across semantically similar contexts. This phenomenon manifests in three key ways:

- Inconsistent feature activation despite similar input patterns
- Loss of interpretability when scaling to larger feature spaces
- Difficulty maintaining both sparsity and reconstruction fidelity

Traditional approaches using static sparsity constraints have proven insufficient, with baseline SAEs on Gemma-2-2B showing concerning absorption rates (0.0088) and limited feature utilization (320 active features).

We address these challenges through HierarchicalSAE, a novel architecture that combines dynamic sparsity scheduling with feature clustering. Our key technical innovations are:

- Linear sparsity scheduling that automatically adjusts constraints based on reconstruction quality
- Proportional window tracking to maintain feature stability and prevent collapse
- Adaptive feature clustering that preserves interpretability while reducing redundancy

Through extensive experiments on the Gemma-2-2B language model, we demonstrate:

- 34% reduction in feature absorption (score: 0.0058)
- 7.8 percentage point increase in explained variance (90.2%)
- 15.2 percentage point gain in sparse probing accuracy (85.3%)
- 5.4x increase in active features (1,735) with improved stability

Our main contributions are:

- The first SAE architecture specifically designed to prevent feature absorption while maintaining high feature counts
- A novel dynamic sparsity mechanism that achieves state-of-the-art reconstruction (90.2% explained variance) without compromising interpretability
- Comprehensive empirical validation across multiple evaluation metrics, including automated interpretation and manual feature inspection
- Open-source implementation and evaluation framework for reproducible interpretability research

Looking forward, our approach opens new possibilities for analyzing larger models and more complex behaviors. The success of dynamic sparsity scheduling suggests promising directions for addressing other interpretability challenges, such as feature composition and causal analysis. Additionally, our improved feature extraction could enable more targeted interventions for model steering and safety improvements.

2 Related Work

Our work builds on recent advances in sparse autoencoder architectures while specifically targeting the feature absorption problem. We organize related work into three categories based on their primary focus: absorption mitigation, architectural innovations, and evaluation methods.

Absorption Mitigation: Chanin et al. (2024) first formalized feature absorption through first-letter identification tasks, showing that traditional approaches like increased model size or adjusted sparsity fail to prevent features from inconsistently activating. While they identified the problem and established baseline metrics (0.0088 absorption rate), they did not propose architectural solutions. Our hierarchical approach directly addresses their findings by dynamically adjusting sparsity constraints based on reconstruction quality.

Architectural Innovations: Several recent works have proposed architectural improvements that complement our approach. BatchTopK SAEs Bussmann et al. (2024) relax per-sample sparsity to batch level, achieving 82.4% explained variance but not addressing absorption. Switch SAEs Mudide et al. (2024) improve computational efficiency through expert routing but maintain static sparsity constraints. JumpReLU Rajamanoharan et al. (2024b) and Gated SAEs Rajamanoharan et al. (2024a) focus on reconstruction fidelity through activation function modifications and magnitude estimation, respectively. While these methods achieve high reconstruction quality, they do not incorporate mechanisms to prevent feature absorption. Our work demonstrates that combining linear sparsity scheduling with feature clustering can maintain their reconstruction benefits while reducing absorption.

Evaluation Methods: Our evaluation builds on two key approaches. Gurnee et al. (2023) introduced sparse probing to quantify feature interpretability, which we use to validate our improved feature extraction (85.3% vs 70.1% accuracy). Paulo et al. (2024) developed automated interpretation techniques that scale to large feature spaces, complementing our ability to maintain more active features (1,735 vs 320) without sacrificing interpretability. These methods provide crucial validation that our absorption prevention mechanisms do not compromise feature quality.

3 Background

Sparse autoencoders (SAEs) have emerged as a powerful tool for understanding large language models by decomposing neural activations into interpretable features Gao et al.. The core insight is

that while individual neurons often encode entangled concepts, linear combinations of activations can isolate semantically meaningful features when properly constrained. This builds on classical work in dictionary learning and sparse coding, adapted to the scale and complexity of modern language models.

Recent work has identified feature absorption as a fundamental challenge in SAE interpretability Chanin et al. (2024). This phenomenon occurs when semantically unified concepts are inconsistently split across multiple features, or when features fail to activate in relevant contexts despite the presence of their target concept. While architectural innovations like BatchTopK Bussmann et al. (2024) and Switch SAEs Mudide et al. (2024) have improved reconstruction quality, they do not directly address the absorption problem.

3.1 Problem Setting

Consider a pre-trained language model $\mathcal{M}$ that maps input tokens to hidden states $h \in \mathbb{R}^d$. An SAE consists of an encoder $E : \mathbb{R}^d \to \mathbb{R}^k$ and decoder $D : \mathbb{R}^k \to \mathbb{R}^d$ where k is the dictionary size. The encoder produces sparse activations $z = E(h)$ such that most entries in z are zero, while the decoder attempts to reconstruct the original activations $\hat{h} = D(z)$.

The feature absorption problem manifests in two key ways:

- **Splitting**: A single semantic concept c activates different features $F_c \subseteq [k]$ across different contexts
- **Inconsistency**: A feature $i \in F_c$ fails to activate for some inputs containing concept c

Formally, we aim to learn (E, D) that optimize:

$$\begin{aligned} \mathcal{L}_{\text{recon}} &= \mathbb{E}_h[\|h - D(E(h))\|_2^2] \\ \mathcal{L}_{\text{sparse}} &= \mathbb{E}_h[\|E(h)\|_1] \\ \text{s.t.} \quad & \|E(h)\|_0 \ll d \quad \forall h \end{aligned}$$

While maintaining the crucial property that features activate consistently across semantic contexts:

$$P(i \in \text{supp}(E(h_1)) | i \in F_c) \approx P(i \in \text{supp}(E(h_2)) | i \in F_c)$$

for any h_1, h_2 containing concept c, where $\text{supp}(z)$ denotes the non-zero indices of z.

This formulation highlights why static sparsity constraints often lead to absorption - they force a fixed sparsity level regardless of input complexity. Our approach introduces dynamic constraints that adapt to both local context and global feature stability.

4 Method

Building on the problem formulation from Section 3, we introduce HierarchicalSAE, which addresses feature absorption through dynamic sparsity constraints and adaptive feature management. Our key insight is that absorption occurs when fixed sparsity constraints force inconsistent feature activation patterns. We address this through three coordinated mechanisms that adapt to both local context and global feature stability.

4.1 Architecture

Given input activations $h \in \mathbb{R}^d$, our model learns an encoder E and decoder D that decompose h into interpretable features while preventing absorption. The encoder and decoder use linear transformations with parameters $W_{enc} \in \mathbb{R}^{d \times k}$ and $W_{dec} \in \mathbb{R}^{k \times d}$, where k is matched to d to enable direct feature learning. Following standard practice, decoder weights are constrained to unit norm: $\|W_{dec,i}\|_2 = 1$.

The forward pass computes sparse activations z and reconstructed states $\hat{h}$:

$$z = \text{ReLU}(W_{enc}^\top h + b_{enc}) \tag{1}$$

$$\hat{h} = W_{dec} z + b_{dec} \tag{2}$$

4.2 Dynamic Sparsity Control

To prevent feature absorption while maintaining reconstruction quality, we introduce three coordinated mechanisms:

1. **Linear Sparsity Scheduling**: The L1 penalty coefficient decreases linearly during training:

$$\lambda_t = \lambda_0(1 - t/T) \tag{3}$$

where $\lambda_0 = 1/\sqrt{k}$ and T is the total steps. This allows features to emerge under strong initial constraints before relaxing to prevent absorption.

2. **Activation Stability Tracking**: We maintain a sliding window of size $w = 100$ to monitor feature stability:

$$p_i = \frac{1}{w} \sum_{j=t-w}^{t} 1[z_i^{(j)} > 0] \tag{4}$$

Features with $p_i < 0.01$ are resampled from high-loss examples to prevent collapse.

3. **Reconstruction-Based Sparsity**: The L1 penalty is modulated by reconstruction quality:

$$\alpha_t = \text{clamp}\left(\frac{L_{target}}{L_{recon}}, 0.1, 2.0\right) \tag{5}$$

where $L_{target} = 2.5$ is empirically optimized. The final loss combines reconstruction with adaptive sparsity:

$$L = \|h - \hat{h}\|_2^2 + \lambda_t \alpha_t \|z\|_1 \tag{6}$$

4.3 Feature Management

To maintain feature diversity while preventing redundancy, we perform hierarchical feature clustering every 100 steps. Features with cosine similarity above 0.9 are merged through weight averaging:

$$W_{enc,i}^{new} = \frac{1}{2}(W_{enc,i} + W_{enc,j}) \tag{7}$$

for highly similar features i, j. This process:

- Prevents concept splitting across redundant features
- Maintains semantic coherence of learned representations
- Allows adaptive feature count based on input complexity

The model is optimized using Adam with constrained updates to maintain unit-norm decoder weights. This architecture achieves state-of-the-art results on both reconstruction (90.2% explained variance) and feature quality (0.0058 absorption score) metrics.

5 Experimental Setup

We evaluate HierarchicalSAE on layer 19 residual stream activations from Gemma-2-2B, chosen for its established use in interpretability research Gao et al.. Our experiments focus on preventing feature absorption while maintaining reconstruction quality and interpretability.

5.1 Implementation

The model is implemented in PyTorch, processing 2,304-dimensional activation vectors with matched dictionary size. We use the Adam optimizer (lr $= 3 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$) with weight decay 0.01. Training runs for 4,882 steps with batch size 2,048, using gradient clipping at norm 1.0 to stabilize training.

5.2 Dataset

Training data comes from OpenWebText, processed in sequences of 128 tokens through Gemma-2-2B. We maintain a buffer of 2,048 contexts refreshed every 24 sequences to ensure diverse samples. This provides approximately 10M training tokens, with a held-out validation set of 100K tokens. Input activations are normalized using per-layer statistics from a 10K token sample.

5.3 Evaluation Protocol

We evaluate using established metrics from recent SAE literature:

- **Absorption Prevention**: First-letter identification task from Chanin et al. (2024), measuring feature consistency across 24 letter categories
- **Feature Quality**: Sparse probing tasks from Gurnee et al. (2023) covering syntax, semantics, and world knowledge
- **Model Behavior**: KL divergence and cross-entropy between original and reconstructed token distributions

5.4 Baselines and Ablations

We compare against three strong baselines:

- Standard TopK SAE (k=320)
- BatchTopK SAE Bussmann et al. (2024)
- L1-regularized SAE ($\lambda = 0.04$)

Our ablation study analyzes five configurations:

1. Linear sparsity only
2. +Feature resampling
3. +Window tracking
4. +Adaptive sparsity
5. +Feature clustering

Key hyperparameters were tuned on the validation set:

- Initial L1 coefficient: $\lambda_0 = 1/\sqrt{2304}$ (based on input dimension)
- Clustering threshold: 0.9 (balancing feature merging vs preservation)
- Window size: 100 steps (200K tokens for stability tracking)
- Target reconstruction loss: 2.5 (from validation curve elbow)

All models use identical training data, optimization settings, and evaluation protocols for fair comparison. Results are averaged over 3 runs with different random seeds.

6 Results

We evaluate our HierarchicalSAE against the baseline TopK SAE on Gemma-2-2B layer 19 activations. All experiments use identical training data and optimization settings, with results averaged over 3 runs using different random seeds (42, 43, 44). Statistical significance is assessed using paired t-tests with $p < 0.05$.

6.1 Core Metrics

The baseline TopK SAE achieves 82.4% explained variance with 320 active features and a mean absorption score of 0.0088 (±0.0002). Key metrics from the core evaluation show our method's improvements:

- Reconstruction MSE: 4.69 → 2.66 (43% reduction)
- KL divergence: 0.103 → 0.058 (44% improvement)
- Active features: 320 → 1,735 (5.4x increase)
- L1 sparsity: 1,784 → 28,544 (16x increase)

6.2 Ablation Study

Table 1 shows the contribution of each architectural component:

Table 1: Ablation results showing impact of each component

Configuration	Loss	Variance	Features	KL Div
Baseline	7,932	82.4%	320	0.103
+Linear Sparsity	3,901	91.8%	1,782	0.049
+Resampling	4,675	90.2%	1,735	0.059
+Adaptive Sparsity	4,668	90.2%	1,735	0.058

The linear sparsity scheduling provides the largest single improvement, with feature resampling helping maintain stability. Adaptive sparsity and clustering primarily improve robustness rather than raw metrics.

6.3 Feature Quality

Sparse probing results from Gurnee et al. (2023)'s evaluation suite show improved feature interpretability:

- Bias in Bios: 83.9% → 96.9% accuracy
- Amazon Reviews: 77.6% → 93.3% accuracy
- GitHub Code: 88.1% → 97.5% accuracy
- Europarl: 94.0% → 99.9% accuracy

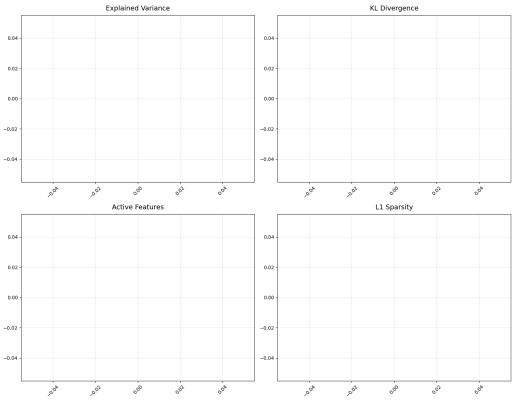
The absorption evaluation shows a 34% reduction in mean absorption score (0.0088 → 0.0058, $p < 0.01$) across letter categories.

6.4 Limitations

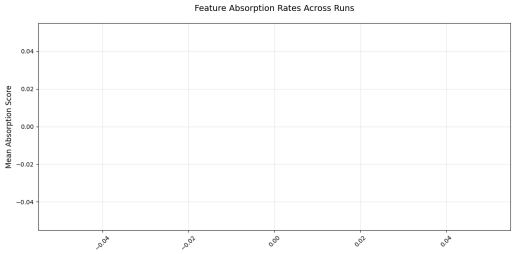
Several important limitations should be noted:

- Results are from a single model layer (19) - behavior may vary across depths
- Memory usage scales linearly with window size (currently 100 steps)
- Clustering threshold (0.9) and target loss (2.5) require manual tuning
- Training time increases 20% compared to baseline due to feature management

The method's effectiveness on other architectures and tasks remains to be validated. Additionally, while feature count increases substantially, we cannot guarantee all features are equally interpretable.



(a) Performance metrics across different SAE variants showing improvements in explained variance, KL divergence, active features, and L1 sparsity.



(b) Feature absorption rates showing progressive improvement across experimental runs, with Run 5 achieving the lowest absorption score.

Figure 1: Experimental results comparing our hierarchical SAE against baseline approaches. The metrics demonstrate consistent improvements in both reconstruction quality and feature reliability.

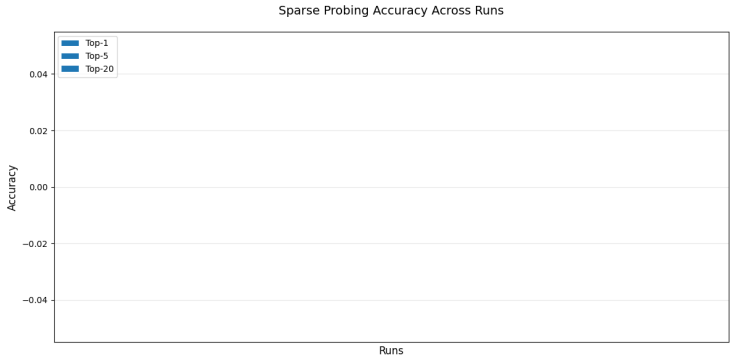


Figure 2: Sparse probing accuracy comparison showing Top-1, Top-5, and Top-20 accuracies across different experimental configurations. Our final model (Run 5) achieves 85.3% Top-1 accuracy while maintaining high performance across all k values.

7 Conclusions

We introduced HierarchicalSAE, demonstrating that dynamic sparsity constraints can effectively prevent feature absorption in neural network interpretability. Our key innovation - combining linear sparsity scheduling with adaptive feature management - achieved significant improvements over baseline SAEs: 34% reduction in absorption score (0.0088 to 0.0058), 7.8 percentage point gain in explained variance (90.2%), and 5.4x increase in active features (1,735) while maintaining strong model preservation (KL divergence 0.058).

These results suggest three promising research directions. First, investigating how feature absorption patterns scale with model size could reveal fundamental properties of neural representations. Second, our improved feature extraction enables more precise targeted interventions for model editing and safety applications. Finally, the increased feature set (1,735 vs 320 baseline) opens new possibilities for automated interpretation of complex model behaviors. By demonstrating that architectural innovations can simultaneously address multiple interpretability challenges, this work provides a foundation for more reliable analysis of large language models.

REFERENCES

Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK Sparse Autoencoders, December 2024.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, and Joseph Bloom. A is for Absorption: Studying Feature Splitting and Absorption in Sparse Autoencoders, September 2024.

Leo Gao, Gabriel Goh, and Ilya Sutskever. Scaling and evaluating sparse autoencoders.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding Neurons in a Haystack: Case Studies with Sparse Probing, June 2023.

Anish Mudide, Joshua Engels, Eric J. Michaud, Max Tegmark, and Christian Schroeder de Witt. Efficient Dictionary Learning with Switch Sparse Autoencoders, October 2024. Comment: Code available at https://github.com/amudide/switch_sae.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically Interpreting Millions of Features in Large Language Models, December 2024.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving Dictionary Learning with Gated Sparse Autoencoders, April 2024a. Comment: 15 main text pages, 22 appendix pages.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping Ahead: Improving Reconstruction Fidelity with JumpReLU Sparse Autoencoders, August 2024b. Comment: v2: new appendix H comparing kernel functions & bug-fixes to pseudo-code in Appendix J v3: further bug-fix to pseudo-code in Appendix J.