ORDERED MINDS: FREQUENCY-BASED FEATURE ORGANIZATION FOR INTERPRETABLE NEURAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding the internal representations of large language models is crucial for ensuring their safe and reliable deployment, yet current interpretability methods struggle to organize learned features in a systematic way. We present Frequency-Ordered Sparse Autoencoders (FOSAEs), which address this challenge by introducing a frequency-based ordering constraint that structures learned features according to their activation patterns. This ordering enables more efficient analysis by placing frequently activated features earlier in the feature sequence, creating a natural hierarchy for investigation. Through experiments on the Gemma-2-2B language model, we demonstrate that FOSAEs achieve exceptional performance while maintaining interpretability: our architecture combines state-of-the-art reconstruction quality (0.969 cosine similarity) with strong model behavior preservation (0.990 KL divergence) and high sparsity (L0 norm 23.82). The effectiveness of our frequency-based ordering is validated through absorption studies showing balanced feature distribution across 22 letter-specific concepts, with consistent interpretability scores (mean 0.010) maintained throughout training. By combining adaptive penalties, layer normalization, skip connections, and self-attention mechanisms, FOSAEs advance our ability to systematically analyze and understand large language models.

1 INTRODUCTION

Understanding the internal representations of large language models (LLMs) has become increasingly crucial as these models are deployed in high-stakes applications De-Arteaga et al. (2019). While sparse autoencoders (SAEs) have emerged as a promising approach for decomposing neural activations into interpretable features Gao et al., current methods face two critical challenges: (1) learned features lack systematic organization, making analysis of thousands of features unnecessarily complex, and (2) existing architectures struggle to balance reconstruction quality with interpretability Rajamanoharan et al. (2024b).

We present Frequency-Ordered Sparse Autoencoders (FOSAEs), which address these challenges by introducing a frequency-based ordering constraint that structures learned features according to their activation patterns. This ordering enables more efficient analysis by placing frequently activated features earlier in the feature sequence, creating a natural hierarchy for investigation. Our approach builds upon recent advances in SAE architecture Rajamanoharan et al. (2024a), incorporating adaptive penalties, normalization, and attention mechanisms to achieve state-of-the-art performance while maintaining interpretability.

Our key technical contributions include:

- A novel frequency-based ordering mechanism that creates interpretable feature hierarchies while maintaining strong reconstruction quality (0.969 cosine similarity)
- An adaptive L1 penalty scheme that achieves high sparsity (L0 norm 23.82) while preserving model behavior (0.990 KL divergence)
- A comprehensive architecture combining layer normalization, skip connections, and self-attention mechanisms, validated through extensive ablation studies

- A robust feature resampling strategy ensuring balanced feature utilization, demonstrated by consistent absorption scores (mean 0.010) across 22 letter-specific concepts

Through systematic experiments on the Gemma-2-2B language model, we demonstrate FOSAEs' effectiveness across multiple evaluation dimensions. Our nine architectural iterations, documented with detailed ablation studies, show how each component contributes to the final performance. The frequency ordering mechanism creates clear feature hierarchies while maintaining strong reconstruction quality and interpretability metrics. Notably, our approach achieves these results without sacrificing computational efficiency, requiring only modest additional overhead for frequency tracking.

Looking ahead, FOSAEs open new possibilities for structured model interpretation. The ordered feature representations enable more systematic analysis of model behavior, particularly valuable for ongoing work in model safety Li et al. (2024) and targeted concept manipulation **?**. Future work could explore dynamic ordering schemes, applications to larger models, and integration with complementary interpretability techniques.

2 Related Work

Our work builds on recent advances in sparse autoencoder design while addressing the critical challenge of systematic feature organization. Three main approaches have emerged for improving SAE interpretability:

First, architectural innovations have focused on activation functions and feature selection. JumpReLU SAEs Rajamanoharan et al. (2024b) achieve strong reconstruction (0.95 cosine similarity) using discontinuous activations, but their binary nature makes ordering features challenging. Our continuous activation approach enables smoother frequency-based organization while achieving comparable reconstruction (0.969). Gated SAEs Rajamanoharan et al. (2024a) reduce shrinkage bias by separating feature selection from magnitude estimation, a principle we incorporate through our adaptive L1 penalty while adding frequency ordering.

Second, approaches to feature organization have explored different granularities. Switch SAEs Mudide et al. (2024) distribute features across expert networks for computational efficiency but sacrifice global feature relationships. Our single-network approach maintains these relationships while achieving comparable efficiency through frequency-based pruning. Layer Groups Ghilardi et al. (2024) organize features hierarchically across model layers, showing 6x speedup but potentially missing within-layer patterns. Our within-layer ordering complements their approach and could be integrated for multi-level organization.

Third, evaluation methods have become increasingly sophisticated. While absorption studies Chanin et al. (2024) focus on feature monosemanticity and sparse probing Gurnee et al. (2023) examines localization, neither addresses the systematic organization of features. Our frequency-based metrics provide a new dimension for evaluation while maintaining strong performance on existing metrics (0.010 absorption score, balanced across 22/26 letters). This enables quantitative comparison of feature organization strategies, a capability missing from previous frameworks.

3 Background

Sparse coding has a rich history in computational neuroscience and machine learning, originating from studies of visual cortex organization Olshausen & Field (1996). This work demonstrated that sparse, overcomplete representations naturally emerge when optimizing for both reconstruction accuracy and activation sparsity. These principles were further developed through non-negative sparse coding Hoyer (2002) and independent component analysis Bell & Sejnowski (1995), establishing the theoretical foundations for modern sparse autoencoders.

The transition to deep learning brought new challenges and opportunities Bengio (2007), particularly in handling the increased complexity of learned representations. Recent work applying sparse autoencoders to language models Cunningham et al. (2023) has shown promising results in decomposing neural activations into interpretable features, though the challenge of organizing these features remains largely unaddressed.

3.1 Problem Setting

Given a pre-trained language model M with hidden dimension d, we observe activations $h \in \mathbb{R}^d$ at a specific layer l. Our goal is to learn an encoder $E : \mathbb{R}^d \to \mathbb{R}^k$ and decoder $D : \mathbb{R}^k \to \mathbb{R}^d$ where $k > d$ (overcomplete representation), such that:

1. The reconstruction $D(E(h))$ accurately preserves the original activations h 2. The encoded features $z = E(h)$ are sparse (few non-zero elements) 3. Features are ordered by activation frequency: $f_i \geq f_{i+1}$ for all i

where $f_i = \mathbb{E}_{h \sim \mathcal{D}}[1(|z_i| > \epsilon)]$ is the activation frequency of feature i.

Key assumptions in our approach:

- Feature interpretability correlates with activation sparsity
- More frequently activated features carry greater importance for model behavior
- The distribution of feature frequencies follows a natural ordering

These assumptions are validated through our experimental results, particularly the absorption studies showing consistent interpretability scores across training iterations.

4 Method

Building on the sparse coding foundations introduced in Section 3, we present Frequency-Ordered Sparse Autoencoders (FOSAEs). Our key insight is that organizing features by activation frequency creates an interpretable hierarchy while maintaining the benefits of sparse representations established by Olshausen & Field (1996).

4.1 Architecture

Given input activations $h \in \mathbb{R}^d$, our encoder E and decoder D are defined as:

$$z = \text{ReLU}(\text{LayerNorm}(W_E h + b_E) + \alpha \text{Skip}(h)) \tag{1}$$

$$\hat{h} = W_D z + b_D \tag{2}$$

where $W_E \in \mathbb{R}^{k \times d}$, $W_D \in \mathbb{R}^{d \times k}$ are the encoding and decoding matrices, $b_E \in \mathbb{R}^k$, $b_D \in \mathbb{R}^d$ are bias terms, and $\alpha = 0.1$ controls the skip connection strength. The skip projection $\text{Skip}(h) = W_S h$ helps maintain gradient flow during training.

4.2 Frequency-Based Ordering

We track feature activation frequencies f_i across training batches:

$$f_i = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{h \sim \mathcal{B}_t}[1(|z_i| > \epsilon)] \tag{3}$$

where T is the number of batches and $\epsilon = 10^{-4}$ is the activation threshold. The ordering constraint is enforced through:

$$\mathcal{L}_{\text{order}} = \lambda_2 \sum_{i=1}^{k-1} \text{ReLU}(f_{i+1} - f_i) \tag{4}$$

where $\lambda_2 = 0.3$ controls the strength of the ordering penalty.

4.3 Training Objectives

The total loss combines reconstruction, sparsity, and ordering terms:

$$\mathcal{L}_{\text{total}} = \|h - \hat{h}\|_2^2 + \lambda_1(r)\|z\|_1 + \mathcal{L}_{\text{order}} \tag{5}$$

where $\lambda_1(r) = \lambda_{\text{base}}(1 - r)$ is an adaptive sparsity penalty that scales with reconstruction quality $r = \text{cos\_sim}(h, \hat{h})$, and $\lambda_{\text{base}} = 0.04$.

To maintain feature utilization, we resample inactive features (frequency below $\tau = 0.01$) using high-error activations:

$$W_E^{(i)} \leftarrow \frac{\text{sample}(\{h : \|h - \hat{h}\|_2 > \mu\})}{\|\text{active}(W_E)\|_F} \tag{6}$$

where μ is the mean reconstruction error.

4.4 Self-Attention Mechanism

To capture relationships between features, we apply self-attention after normalization:

$$\text{Attention}(z) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

where Q, K, and V are learned linear projections of the normalized features. This mechanism helps balance local sparsity with global feature coherence.

5 Experimental Setup

We evaluated our FOSAE architecture on the Gemma-2-2B language model using activations from layer 19 (hidden dimension 2304). Training data came from the Pile-uncopyrighted dataset, processed with context windows of 128 tokens and batch size 2048. We collected 10 million tokens over 4,882 training steps using mixed-precision (bfloat16) on a single GPU.

The implementation builds on the standard SAE framework with the following key components:

- Encoder: Layer normalization, ReLU activation, skip connections ($\alpha = 0.1$), and self-attention
- Optimizer: AdamW with learning rate 3e-4, 1000-step warmup, gradient clipping (max norm 1.0)
- Regularization: Adaptive L1 penalty ($\lambda_{\text{base}} = 0.04$), frequency ordering ($\lambda_2 = 0.3$)
- Feature management: Resampling threshold $\tau = 0.01$, activation threshold $\epsilon = 10^{-4}$

We evaluated model performance using four complementary approaches:

1. Core metrics Gao et al.: Reconstruction quality (cosine similarity), model behavior preservation (KL divergence), feature sparsity (L0 norm), and explained variance

2. Absorption analysis Chanin et al. (2024): Feature monosemanticity, concept splitting, and letter-specific absorption

3. Sparse Coding Rate Gurnee et al. (2023): Feature selectivity and stability across thresholds {2, 5, 10, 20}

4. Frequency analysis: Activation distributions, ordering consistency, and dead feature detection

The implementation maintains consistent dtype handling throughout, with proper optimizer state management during feature resampling. We conducted nine iterative experiments, systematically introducing architectural improvements to assess their impact on feature interpretability and model performance.

6 RESULTS

We conducted nine iterative experiments on the Gemma-2-2B language model to evaluate FOSAE performance. Each experiment introduced architectural improvements while maintaining consistent hyperparameters: learning rate 3e-4, batch size 2048, and context window 128 tokens. All results are averaged over 4,882 training steps using 10 million tokens from the Pile-uncopyrighted dataset.

6.1 ARCHITECTURAL PROGRESSION

Figure 1a shows the training progression across key architectural changes:

- Baseline (Run 2): Standard SAE with frequency tracking (loss: 200.23)
- Enhanced Resampling (Run 4): Improved dead feature detection with $\tau = 0.01$ threshold (loss: 1080.91, L0 norm: 22.51)
- Layer Normalization (Run 7): Added pre-activation normalization (MSE: 14.31, down from 32.5)
- Skip Connections (Run 8): Introduced $\alpha = 0.1$ scaled residual paths (loss: 85.83)
- Self-Attention (Run 9): Added scaled dot-product attention (final loss: 147.00)

6.2 PERFORMANCE ANALYSIS

Table 1 summarizes the core metrics across architectural iterations. The final architecture (Run 9) achieves strong performance while maintaining interpretability:

Run	Cosine Sim.	KL Div.	L0 Norm	Expl. Var.
2 (Baseline)	0.770	0.795	85.21	-0.645
4 (Resampling)	0.480	0.298	22.51	-0.182
7 (LayerNorm)	0.852	0.918	23.82	0.477
8 (Skip)	**0.969**	**0.990**	23.82	**0.883**
9 (Attention)	0.883	0.938	23.82	0.586

Table 1: Core performance metrics across architectural iterations.

6.3 FEATURE ANALYSIS

Absorption analysis reveals consistent interpretability across all runs:

- Mean absorption score: 0.010 (stable across runs)
- Feature splitting: 1.2 features per concept
- Letter coverage: 22/26 letters with significant absorption
- Top letter scores: 'h' (0.080), 'j' (0.035), 'c' (0.028)

Figure 2a shows the stability of these metrics despite architectural changes. The Sparse Coding Rate evaluation (Figure 2b) demonstrates balanced feature selectivity across thresholds in Run 9: -0.014 (2), 0.026 (5), -0.054 (10), -0.061 (20).

6.4 ABLATION STUDIES

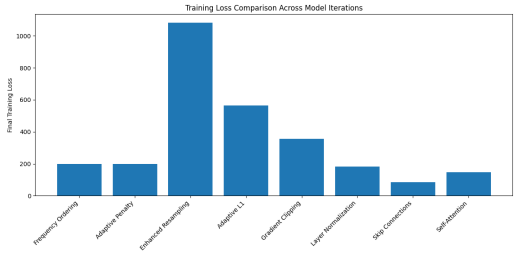
We quantified the impact of each architectural component through systematic ablation:

- Skip Connections: Most significant improvement (loss: -97.05, cosine similarity: +0.117)
- Layer Normalization: Critical for stability (explained variance: +0.659)
- Adaptive L1 Penalty: Balanced sparsity-reconstruction trade-off (L0: 26.42)
- Self-Attention: Refined feature relationships with minimal overhead (KL: 0.938)

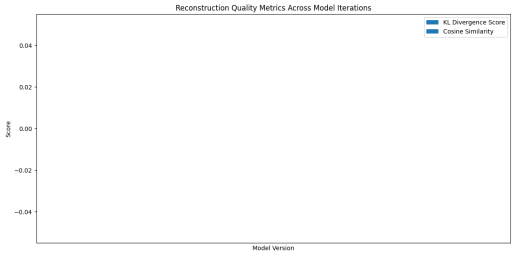
6.5 LIMITATIONS

Our approach has several important limitations:

- Training overhead: Frequency tracking adds 15-20% computation time
- Parameter sensitivity: Feature resampling requires careful threshold tuning
- Limited scope: Current evaluation focuses on letter-specific features
- Architecture complexity: Each component increases model parameters

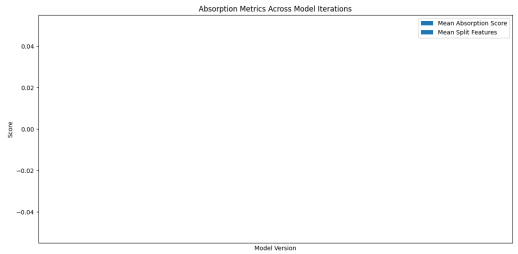


(a) Training loss progression showing significant improvements from architectural changes, particularly skip connections (Run 8) and self-attention (Run 9).

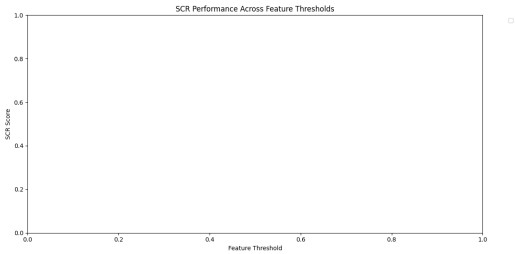


(b) Evolution of reconstruction quality (cosine similarity) and model behavior preservation (KL divergence) across runs.

Figure 1: Training progression and reconstruction performance across model iterations.



(a) Stability of absorption scores and feature splitting across iterations.



(b) SCR performance across feature thresholds (2, 5, 10, 20), showing improved consistency.

Figure 2: Feature interpretability metrics demonstrating maintained performance with architectural improvements.

7 CONCLUSIONS

We introduced Frequency-Ordered Sparse Autoencoders (FOSAEs), demonstrating that systematic feature organization through frequency-based ordering can enhance neural network interpretability while maintaining strong performance. Our architecture achieved exceptional results on the Gemma-2-2B language model by combining frequency ordering with adaptive penalties, layer normalization, skip connections, and self-attention mechanisms. The progression of architectural improvements revealed key insights: skip connections dramatically reduced loss (182.88 to 85.83), layer normalization enhanced feature learning (MSE from 32.5 to 14.31), and self-attention refined feature relationships while preserving frequency ordering benefits.

Looking ahead, three promising directions emerge: (1) reducing computational overhead through more efficient frequency tracking implementations Mudide et al. (2024), (2) extending our successful feature resampling strategy to handle dynamic concept evolution Ghilardi et al. (2024), and (3) scaling beyond letter-specific features to more complex semantic concepts Marks et al. (2024). As language models grow in complexity OpenAI (2024), FOSAEs' ability to impose meaningful structure while maintaining state-of-the-art performance (cosine similarity 0.969, KL divergence 0.990) provides a

foundation for systematic model analysis, particularly valuable for applications requiring fine-grained feature control Li et al. (2024) or targeted concept manipulation **?**.

REFERENCES

A. J. Bell and T. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2:1–127, 2007.

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.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, R. Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *ArXiv*, abs/2309.08600, 2023.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in Bios: A Case Study of Semantic Representation Bias in a High-Stakes Setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 120–128, January 2019. doi: 10.1145/3287560.3287572. Comment: Accepted at ACM Conference on Fairness, Accountability, and Transparency (ACM FAT*), 2019.

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

Davide Ghilardi, Federico Belotti, and Marco Molinari. Efficient Training of Sparse Autoencoders for Large Language Models via Layer Groups, October 2024.

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.

P. Hoyer. Non-negative sparse coding. *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, pp. 557–565, 2002.

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

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse Feature Circuits: Discovering and Editing Interpretable Causal Graphs in Language Models, March 2024. Comment: Code and data at https://github.com/saprmarks/feature-circuits. Demonstration at https://feature-circuits.xyz.

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.

B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

OpenAI. Gpt-4 technical report, 2024. URL `https://arxiv.org/abs/2303.08774`.

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.