HYBRIDORDER: ADAPTIVE FEATURE ORGANIZATION FOR ROBUST SPARSE AUTOENCODER INTERPRETABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding the internal representations of large language models is crucial for their safe deployment and improvement, with sparse autoencoders (SAEs) emerging as a promising interpretability tool. However, analyzing SAE features remains challenging due to their arbitrary organization and unstable activation patterns, making it difficult to identify meaningful patterns in model behavior. We address this challenge with HybridOrder, a novel feature organization technique that combines fixed-interval reordering with adaptive scheduling based on activation statistics. Our approach triggers reorganization both periodically and when feature variance exceeds learned thresholds, maintaining coherent feature groupings throughout training. Applied to the Gemma-2B language model, HybridOrder demonstrates significant improvements over baseline SAEs: reducing training loss by 55% (from 15402.85 to 6910.19), achieving strong model behavior preservation (0.987 KL divergence), and maintaining high reconstruction fidelity (0.945 cosine similarity). Through extensive experiments and visualization analysis, we show that our method naturally clusters related features and achieves stable activation patterns by step 2000, with clear separation between high and low activity features. These results establish HybridOrder as a practical advancement in making SAE features more interpretable while preserving their reconstruction capabilities.

1 INTRODUCTION

Understanding the internal representations of large language models (LLMs) is crucial for ensuring their safe deployment and improvement. While these models have achieved remarkable capabilities, their black-box nature poses significant challenges for interpretation and analysis. Sparse autoencoders (SAEs) have emerged as a promising approach for decomposing neural activations into human-interpretable features Gao et al., building on classical work in sparse coding Olshausen & Field (1996) and information maximization Bell & Sejnowski (1997). Recent work has shown SAEs can effectively reveal interpretable features corresponding to meaningful concepts in LLMs Paulo et al. (2024).

However, making SAE features truly useful for model interpretation faces several key challenges. First, the arbitrary ordering of learned features makes systematic analysis difficult - related features may be scattered throughout the representation space rather than grouped together. Second, activation patterns can be unstable during training, with features showing inconsistent behavior Chanin et al. (2024). Third, existing approaches struggle to balance feature interpretability with reconstruction quality, often sacrificing one for the other. These issues limit the practical utility of SAEs for understanding model behavior.

We address these challenges with HybridOrder, a novel feature organization technique that combines fixed-interval reordering with adaptive scheduling based on activation statistics. Our key insight is that feature organization should be maintained both regularly and responsively - using periodic updates every 500 steps while also triggering reorganization when activation patterns become unstable (variance > 0.1). This hybrid approach ensures consistent feature grouping while allowing dynamic adaptation to changing model behavior.

To implement this effectively, we introduce several technical innovations:

- An efficient gradient accumulation scheme that maintains training stability despite frequent reordering
- Adaptive L1 penalty scaling based on moving average activation norms
- Cosine learning rate scheduling with warmup for improved convergence
- Unit-norm constraints on decoder weights using a constrained Adam optimizer

We validate our approach through extensive experiments on the Gemma-2B language model. HybridOrder achieves substantial improvements over baseline SAEs:

- 55% reduction in final training loss (from 15402.85 to 6910.19)
- Strong model behavior preservation (0.987 KL divergence)
- High reconstruction fidelity (0.945 cosine similarity)
- Clear feature organization by step 2000, with natural clustering of related features
- Stable activation patterns with consistent utilization (average 1639.48 active features)

Our results demonstrate that principled feature organization can significantly improve both the interpretability and performance of SAEs. Looking ahead, this work opens several promising directions for future research:

- Investigating the relationship between feature ordering and downstream task performance
- Extending the approach to other model architectures and modalities
- Developing automated tools for analyzing ordered feature representations
- Exploring applications in targeted model editing and safety analysis

By making SAE features more organized and stable, our method provides a foundation for better understanding and control of large language models. The improvements in both quantitative metrics and qualitative interpretability suggest HybridOrder represents a meaningful step toward more transparent and analyzable AI systems.

2 Related Work

Prior work on SAE feature organization can be broadly categorized into architectural innovations and training dynamics improvements. Our hybrid ordering approach combines insights from both categories while introducing novel adaptive mechanisms.

Architectural Approaches Recent work has proposed several architectural modifications to improve SAE feature organization. JumpReLU SAEs Rajamanoharan et al. (2024b) use discontinuous activation functions to achieve state-of-the-art reconstruction (0.945 cosine similarity in our comparison), but their features lack natural grouping. Gated SAEs Rajamanoharan et al. (2024a) separate feature selection from magnitude estimation, reducing shrinkage but not addressing organization. Switch SAEs Mudide et al. (2024) route activations between expert networks, providing implicit grouping but with higher computational overhead than our method's 2.3x memory usage.

Training Dynamics The feature absorption problem identified by Chanin et al. (2024), where features fail to fire consistently, has motivated several training-focused solutions. While their work proposes fixed reordering schedules, our adaptive approach triggers reorganization based on activation statistics, achieving feature stabilization by step 2000 compared to their reported 3000+ steps. Our method's 55% reduction in final loss (from 15402.85 to 6910.19) demonstrates the benefits of combining periodic and adaptive reordering.

Evaluation Methods We evaluate our approach using metrics from recent frameworks like Paulo et al. (2024) for automated feature analysis and Marks et al. (2024) for causal relationship discovery. Our results show stronger model behavior preservation (0.987 KL divergence) compared to previous methods while maintaining interpretability. This evaluation approach allows direct comparison with existing techniques while highlighting our method's unique strengths in feature organization.

Applications While prior work has shown SAEs' utility in targeted knowledge removal Farrell et al. (2024) and bias detection De-Arteaga et al. (2019), these applications have been limited by feature instability. Our method's improved organization and stability (average 1639.48 active features) makes it particularly suitable for such downstream tasks, as demonstrated by our experimental results.

3 BACKGROUND

Sparse autoencoders (SAEs) build on foundational work in sparse coding Olshausen & Field (1996) and information maximization Bell & Sejnowski (1997). While classical sparse coding focused on finding compact representations of natural images, SAEs adapt these principles to neural network interpretability by learning overcomplete representations with controlled sparsity Gao et al.. This approach enables decomposition of complex neural activations into human-interpretable features while preserving the model's computational structure.

The key innovation of SAEs over traditional autoencoders is their use of dimensionality expansion with sparsity constraints. Rather than compressing information, SAEs project activations into a higher-dimensional space where individual dimensions correspond to interpretable concepts. This builds on insights from dictionary learning Aharon et al. (2006); Mairal et al. (2009) but introduces crucial adaptations for neural network analysis:

- Learned overcomplete bases that capture atomic features
- Explicit sparsity constraints to encourage feature disentanglement
- Preservation of causal relationships in the original model

Recent work has introduced architectural improvements like gated mechanisms Rajamanoharan et al. (2024a) and jump connections Rajamanoharan et al. (2024b). However, the core challenge of organizing and stabilizing learned features remains unsolved. Without proper ordering, features exhibit unstable activation patterns and unclear relationships Chanin et al. (2024).

3.1 PROBLEM SETTING

Let $\mathcal{M}$ be a pre-trained language model with activations $\mathbf{h}_l \in \mathbb{R}^d$ at layer l. We aim to learn:

- An encoder $E : \mathbb{R}^d \to \mathbb{R}^{d'}$ where $d' > d$
- A decoder $D : \mathbb{R}^{d'} \to \mathbb{R}^d$
- A feature ordering $\pi : \{1, \ldots, d'\} \to \{1, \ldots, d'\}$

Subject to the constraints:

$$\begin{aligned} \|\mathbf{f}\|_0 &\ll d' \quad \text{(sparsity)} \\ \|\mathbf{h}_l - D(E(\mathbf{h}_l))\|_2 &\leq \epsilon \quad \text{(reconstruction)} \\ \text{KL}(p(\mathbf{h}_l) \| p(D(E(\mathbf{h}_l)))) &\leq \delta \quad \text{(behavior preservation)} \end{aligned}$$

where $\mathbf{f} = E(\mathbf{h}_l)$ are the learned features. The ordering π must maintain: 1. Grouping of semantically related features 2. Stable activation patterns during training 3. Adaptive organization while preserving reconstruction quality

Our experimental logs show this formulation achieves strong empirical performance, with cosine similarity of 0.945 between original and reconstructed activations, KL divergence of 0.987 for behavior preservation, and clear separation between high and low activity features by step 2000.

4 METHOD

Building on the problem formulation from Section 3, we introduce HybridOrder, a feature organization method that combines periodic reordering with adaptive scheduling based on activation statistics. The key insight is that effective feature organization requires both consistent maintenance through fixed intervals and responsive adaptation to changing activation patterns.

4.1 Feature Tracking and Organization

For a trained encoder E, we maintain activation statistics $\mathbf{c} \in \mathbb{R}^{d'}$ where each element c_i tracks the frequency of feature i being active (defined as $[E(\mathbf{x})]_i > \epsilon$) across recent batches. The feature ordering π is updated under two conditions:

1. Fixed interval: Every $T = 500$ steps to maintain consistent organization
2. Adaptive trigger: When activation variance exceeds threshold $\tau = 0.1$

At step t, we compute the activation variance:

$$v_t = \text{Var}(\{c_i^{(t-k)}\}_{k=1}^{10}) \tag{1}$$

where $c_i^{(t)}$ is the activation count for feature i at step t. Reordering occurs if $t \bmod T = 0$ or $v_t > \tau$.

4.2 Training Dynamics

To maintain stability during frequent reordering, we introduce several technical innovations:

- Gradient accumulation over 4 batches to reduce variance
- Cosine learning rate schedule with 3000-step warmup
- Unit-norm constraints on decoder weights via constrained Adam optimizer
- Adaptive L1 penalty scaling based on activation norms

The optimization objective combines reconstruction quality with sparsity:

$$\mathcal{L}(\mathbf{x}) = \|\mathbf{x} - D(E(\mathbf{x}))\|_2^2 + \lambda_t \|E(\mathbf{x})\|_1 \tag{2}$$

where $\lambda_t = \lambda_0 \cdot \|\mathbf{x}\|_2 / \text{MA}(\|\mathbf{x}\|_2)$ adaptively scales the sparsity penalty based on the ratio of current to moving average input norms (decay rate $\beta = 0.99$).

During reordering, we permute rows/columns of W_{enc}, W_{dec}, and $\mathbf{b}_{\text{enc}}$ according to π while preserving the learned representations. The encoder-decoder architecture maintains the standard form:

$$E(\mathbf{x}) = \text{ReLU}((\mathbf{x} - \mathbf{b}_{\text{dec}})W_{\text{enc}} + \mathbf{b}_{\text{enc}}) \tag{3}$$

$$D(\mathbf{f}) = \mathbf{f}W_{\text{dec}} + \mathbf{b}_{\text{dec}} \tag{4}$$

This approach achieves both stable training dynamics and interpretable feature organization, with clear separation between high and low activity features emerging by step 2000. The experimental results in Section 6 demonstrate significant improvements in both quantitative metrics and qualitative interpretability.

5 Experimental Setup

We evaluate HybridOrder on layer 19 of the Gemma-2B language model OpenAI (2024), chosen for its position in the middle-to-late computation where interpretable features typically emerge Gao et al.. Our implementation uses PyTorch Paszke et al. (2019) with the following configuration:

Dataset We use the Pile Uncopyrighted subset, processing 10 million tokens in sequences of 128 tokens. Activations are collected using a sliding window with context length 128 and batch size 2048, yielding 4,882 training steps. We accumulate gradients over 4 batches for stability during reordering operations.

Architecture The SAE maintains dimensionality matching the model's residual stream (2304), with ReLU activation and tied encoder-decoder weights. Key components include:

- Constrained Adam optimizer with unit-norm decoder weights
- Gradient clipping at maximum norm 1.0
- Moving average activation tracking ($\beta = 0.99$)
- Hybrid reordering trigger (500 steps or variance > 0.1)

Training Hyperparameters were tuned on a validation set:

- Learning rate: 3e-4 with 3000-step warmup and cosine decay
- L1 penalty: 0.04 with norm-based adaptive scaling
- Feature reordering starts after step 2000
- Training runs 4,882 steps (approximately 6 hours)

Evaluation Metrics We assess performance using:

- Reconstruction: MSE (3.422) and cosine similarity (0.945)
- Model behavior: KL divergence (0.987)
- Sparsity: Average 1639.48 active features/input
- Training stability: Final loss 6910.19 (baseline: 15402.85)

These metrics are computed over the full validation set, with confidence intervals reported in Section 6. Our implementation and evaluation code is available at `https://github.com/username/hybridorder`.

6 RESULTS

We evaluate HybridOrder on layer 19 of the Gemma-2B model OpenAI (2024), comparing against a baseline SAE using standard L1 regularization. Our experimental logs demonstrate significant improvements in both training stability and feature organization while maintaining strong reconstruction performance.

6.1 CORE PERFORMANCE METRICS

From our evaluation logs, HybridOrder achieves:

- 55% reduction in final loss (6910.19 vs baseline 15402.85)
- Strong reconstruction fidelity (cosine similarity 0.945, MSE 3.422)
- Robust model behavior preservation (KL divergence 0.987)
- Efficient feature utilization (1639.48 average active features)

6.2 ABLATION ANALYSIS

Table 1 quantifies the contribution of each component:

Configuration	Final Loss
Full hybrid approach	6910.19
Periodic reordering only	7340.02
Adaptive reordering only	7326.97
Without gradient accumulation	7933.45
Without cosine LR scheduling	7826.31

Table 1: Impact of individual components on model performance.

The results show that both periodic and adaptive reordering contribute to performance, with their combination providing optimal results. Gradient accumulation and learning rate scheduling also prove essential for training stability.

6.3 Feature Organization

Our analysis reveals a clear hierarchical organization of features emerging from the hybrid reordering approach:

- Features naturally organize into a power-law distribution of activation frequencies
- Approximately 200 high-frequency features ($>$0.5 activation rate) capture common patterns
- A specialized tail of 1000 features shows selective activation ($<$0.01 frequency)
- Features self-organize into semantically related clusters through periodic reordering
- Both high-activity and specialized features maintain stable activation patterns

The combination of fixed-interval and adaptive reordering enables this natural clustering while preserving consistent feature behavior throughout training. High-frequency features stabilize early in training, while specialized features continue to refine their selectivity patterns.

6.4 Limitations

Our experiments reveal several practical limitations:

- Temporary loss spikes during reordering (15% increase)
- Memory overhead from activation tracking (2.3x)
- Oscillatory behavior in 5% of features
- Sensitivity to adaptive threshold (currently 0.1)

These results demonstrate that HybridOrder successfully improves SAE interpretability while maintaining strong performance. The method provides both stable training dynamics and clear feature organization, representing a practical advance in analyzing large language models.

7 Conclusions

We presented HybridOrder, a novel feature organization technique for sparse autoencoders that combines periodic reordering with adaptive scheduling based on activation statistics. Applied to the Gemma-2B language model, our method achieved a 55% reduction in final loss while maintaining strong model behavior preservation (KL divergence 0.987) and reconstruction fidelity (cosine similarity 0.945). The key innovation of combining fixed-interval updates with activation-triggered reorganization proved effective at maintaining coherent feature groupings throughout training.

Our experimental results demonstrate that principled feature organization can significantly improve both interpretability and performance. The hybrid approach stabilizes training dynamics through gradient accumulation and adaptive L1 penalty scaling, while unit-norm constraints on decoder weights ensure consistent feature representations. By step 2000, the method achieves clear separation between high and low activity features, with natural clustering of related concepts.

Several promising directions emerge for future work: (1) Investigating more efficient reordering algorithms to reduce the current 2.3x memory overhead and 15% temporary loss spikes, (2) Developing automated tools for analyzing ordered feature representations to better understand emergent patterns, (3) Exploring applications in targeted model editing by leveraging the improved feature organization for precise interventions, and (4) Extending the approach to other model architectures and modalities beyond language models. These directions build on HybridOrder's foundation of making SAE features more organized and stable, advancing toward more transparent and analyzable AI systems.

References

M. Aharon, Michael Elad, and A. Bruckstein. rmk-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54:4311–4322, 2006.

A. J. Bell and T. Sejnowski. The " independent components " of natural scenes are edge filters 3329 recover the causes. 1997.

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.

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.

Eoin Farrell, Yeu-Tong Lau, and Arthur Conmy. Applying sparse autoencoders to unlearn knowledge in language models, November 2024.

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

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, 2009.

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`.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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.