NESTED DICTIONARY LEARNING: HIERARCHICAL SPARSE AUTOENCODERS FOR INTERPRETABLE LANGUAGE MODEL FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding the internal representations of large language models remains a critical challenge for AI interpretability. While sparse autoencoders (SAEs) have emerged as powerful tools for decomposing neural features into interpretable components, they struggle to capture hierarchical relationships between features at different levels of abstraction. We present Matryoshka Sparse Autoencoders (MSAEs), which address this limitation by organizing learned features through nested dictionaries: a compact shared dictionary (20% of total capacity) captures fundamental patterns while specialized dictionaries encode context-specific features. Through extensive evaluation on the Gemma-2B model, we demonstrate that MSAEs achieve strong performance across key metrics while enabling more structured feature analysis. Our architecture maintains 85.07% L0 sparsity and 0.299 explained variance in reconstruction, matching traditional SAE performance. Most notably, MSAEs excel at mid-range feature abstraction while preserving fine-grained selectivity, with behavioral alignment (0.787 KL divergence) surpassing baseline approaches. These results demonstrate that hierarchical feature organization can enhance interpretability without compromising the core benefits of sparse coding, providing a more principled approach to understanding large language models.

1 INTRODUCTION

Understanding the internal mechanisms of large language models (LLMs) is crucial for ensuring their reliable and ethical deployment Zhang et al. (2020). While these models have achieved remarkable performance across diverse tasks OpenAI (2024), their complex internal representations remain largely opaque, limiting our ability to verify their behavior and build trust in their decisions. This challenge is particularly acute in transformer architectures Vaswani et al. (2017), where features manifest at multiple scales and abstraction levels simultaneously.

Sparse autoencoders (SAEs) have emerged as a promising approach for model interpretability, decomposing neural activations into interpretable components Goodfellow et al. (2016). However, current SAE architectures face two key limitations. First, they struggle to capture hierarchical relationships between features at different abstraction levels, often focusing on either fine-grained details or broad patterns but rarely both. Second, their flat dictionary structure makes it difficult to distinguish between fundamental patterns and context-specific variations, limiting our understanding of how features are organized and reused across different contexts.

We address these challenges by introducing Matryoshka Sparse Autoencoders (MSAEs), which organize learned features through nested dictionaries. Our architecture partitions the feature space into a compact shared dictionary (20

Through extensive evaluation on the Gemma-2B model, we demonstrate that MSAEs achieve strong performance while providing better feature organization:

- **Strong Core Metrics:** Our approach maintains high feature selectivity (85.07% L0 sparsity) and reconstruction quality (0.299 explained variance) while achieving better behavioral alignment (0.787 KL divergence) than traditional SAEs (0.795).

- **Optimal Dictionary Structure:** We empirically discover that a 20% shared dictionary ratio achieves the best performance, challenging conventional wisdom about feature sharing. This configuration shows superior performance in mid-range feature abstraction (n=10,20) while maintaining strong fine-grained selectivity.
- **Robust Architecture:** Analysis reveals consistent sparsity metrics (85.07-86.18%) and feature alignment (cosine similarity 0.762-0.766) across configurations, demonstrating the robustness of our nested approach.
- **Efficient Training:** Despite architectural complexity, our implementation achieves stable convergence through modern optimization techniques Kingma & Ba (2014) and careful normalization Ba et al. (2016).

These results demonstrate that structured approaches to feature organization can enhance both interpretability and performance. The success of MSAEs in capturing multi-scale features while maintaining sparsity opens new avenues for investigating how transformer models organize and process information. Our findings suggest that hierarchical principles could be valuable for developing more transparent and interpretable AI systems.

2 Related Work

Our work builds on three main research directions: sparse coding for interpretability, hierarchical feature learning, and transformer model analysis. We discuss how existing approaches compare to our nested dictionary method.

Sparse Autoencoders for Interpretability. Recent work by Cunningham et al. (2023) demonstrated that sparse autoencoders can effectively decompose language model activations into interpretable features. While their approach successfully addresses polysemanticity through overcomplete dictionaries, it lacks explicit mechanisms for capturing hierarchical relationships. In contrast, our nested architecture directly models feature relationships through shared and specialized dictionaries. The biological foundations of sparse coding Olshausen & Field (1996); Bell & Sejnowski (1996) suggest that hierarchical organization emerges naturally in neural systems, as demonstrated by Ruslim et al. (2024). Our work formalizes this intuition through architectural design rather than relying on emergence.

Hierarchical Feature Learning. Traditional approaches to hierarchical representation learning, such as deep belief networks Lee et al. (2011), rely on layer-wise feature abstraction. While effective for general deep learning, these methods don't explicitly address interpretability. Mairal et al. (2009a;b) developed efficient dictionary learning algorithms that scale to large datasets, but their flat dictionary structure limits feature organization. Our nested approach combines the efficiency of these methods with explicit hierarchical structure, achieving both scalability and interpretable organization.

Transformer Model Analysis. Recent work in mechanistic interpretability has approached transformer understanding from different angles. Network dissection techniques Bau et al. (2017) and causal abstraction frameworks Geiger et al. (2023) provide ways to analyze individual neurons and circuits, but don't directly address feature organization. While Rai et al. (2024) surveys various interpretation methods and Bhaskar et al. (2024) introduces automated circuit discovery, these approaches focus on finding existing structures rather than learning interpretable representations. Our method complements these efforts by providing a structured way to decompose and organize transformer features while maintaining sparsity and interpretability.

The key innovation of our approach is combining the strengths of these methods - the interpretability benefits of sparse coding, the organization of hierarchical learning, and the scalability needed for transformer analysis - through a unified nested dictionary architecture. Unlike previous work that either lacks explicit hierarchy Cunningham et al. (2023) or interpretability guarantees Lee et al. (2011), our method achieves both while maintaining strong empirical performance.

3 BACKGROUND

The foundations of our work lie at the intersection of sparse coding and neural network interpretability. Sparse coding, pioneered by Olshausen & Field (1996), demonstrated that natural systems learn efficient representations by encoding inputs using a small subset of available features. This principle was extended to neural networks through autoencoders Goodfellow et al. (2016), which learn compressed representations while maintaining reconstruction fidelity.

The emergence of transformer architectures Vaswani et al. (2017) introduced new challenges for interpretability. Their self-attention mechanisms create rich, hierarchical representations that capture dependencies at multiple scales simultaneously. Traditional interpretation methods struggle with this complexity, as features manifest at different levels of abstraction and interact in sophisticated ways.

Recent work by Cunningham et al. (2023) showed that sparse autoencoders can effectively decompose language model activations into interpretable components. However, their approach uses flat dictionaries that don't explicitly capture the hierarchical nature of transformer representations. This limitation motivates our nested dictionary approach.

3.1 PROBLEM SETTING

Consider a transformer layer's activation space $\mathcal{X} \subset \mathbb{R}^d$. Our goal is to learn an encoder $E : \mathcal{X} \to \mathbb{R}^k$ and decoder $D : \mathbb{R}^k \to \mathcal{X}$ that decompose these activations into interpretable features while preserving their essential characteristics. The classical sparse coding objective is:

$$\min_{E,D} \mathbb{E}_{x \sim \mathcal{X}} \left[\|x - D(E(x))\|_2^2 + \lambda \|E(x)\|_1 \right] \tag{1}$$

where λ controls the sparsity-reconstruction trade-off.

We extend this framework through a nested dictionary structure that partitions the feature space into shared and specialized components. Given a sharing ratio $\alpha \in (0, 1)$, the encoder output takes the form:

$$E(x) = [E_s(x); E_f(x)] \tag{2}$$

where:

- $E_s(x) \in \mathbb{R}^{\alpha k}$ represents fundamental patterns shared across contexts
- $E_f(x) \in \mathbb{R}^{(1-\alpha)k}$ captures context-specific variations

This formulation makes three key assumptions:

- The distribution of input activations reflects meaningful computation in the pre-trained model
- A small shared dictionary (αk features) can capture cross-context patterns
- The remaining capacity ($[1 - \alpha]k$ features) is sufficient for context-specific representations

These assumptions enable us to learn hierarchical representations that maintain both interpretability and computational efficiency. The effectiveness of this approach is demonstrated through extensive empirical evaluation in Section 6.

4 METHOD

Building on the sparse coding formalism from Section 3.1, we introduce the Matryoshka Sparse Autoencoder (MSAE) architecture. The key innovation is organizing learned features through nested dictionaries that explicitly capture both fundamental patterns and context-specific variations.

4.1 Nested Dictionary Architecture

Given an input activation $\mathbf{x} \in \mathcal{X}$, our encoder E maps it to a structured latent representation:

$$E(\mathbf{x}) = \begin{bmatrix} E_s(\mathbf{x}) \\ E_f(\mathbf{x}) \end{bmatrix} \tag{3}$$

where $E_s(\mathbf{x}) \in \mathbb{R}^{\alpha k}$ represents shared features and $E_f(\mathbf{x}) \in \mathbb{R}^{(1-\alpha)k}$ captures specialized patterns. The decoder mirrors this structure:

$$D(\mathbf{z}) = D_s(\mathbf{z}_{1:\alpha k}) + D_f(\mathbf{z}_{\alpha k:k}) \tag{4}$$

This additive reconstruction allows independent contribution from both feature types while maintaining interpretability.

4.2 Training Objective

We extend the classical sparse coding objective with separate regularization terms:

$$\mathcal{L}(\mathbf{x}) = \underbrace{\|\mathbf{x} - D(E(\mathbf{x}))\|_2^2}_{\text{reconstruction}} + \underbrace{\lambda_s \|E_s(\mathbf{x})\|_1 + \lambda_f \|E_f(\mathbf{x})\|_1}_{\text{sparsity}} \tag{5}$$

where λ_s and λ_f control sparsity for shared and specialized features respectively. This formulation encourages the model to:

- Learn compact shared representations through λ_s
- Maintain flexibility for context-specific patterns via λ_f
- Preserve reconstruction fidelity through the L2 term

4.3 Implementation

To ensure stable training of this nested architecture, we employ:

- Layer normalization on encoder outputs to manage feature scale
- Gradient clipping to prevent optimization instability
- Adam optimizer with normalized gradients

Empirically, we find $\alpha = 0.2$ provides optimal balance between shared and specialized learning, with consistent sparsity (85-86%) across configurations. This architecture enables both efficient feature extraction and clear interpretation of feature hierarchies while maintaining strong performance on core metrics.

5 Experimental Setup

We evaluate the MSAE architecture on layer 19 of the Gemma-2B language model, focusing on its ability to learn interpretable feature hierarchies. Our experiments systematically compare sharing ratios $\alpha \in \{0.1, 0.2, 0.3, 0.5\}$ against a traditional SAE baseline ($\alpha = 0$).

5.1 Implementation Details

The model processes Gemma-2B's hidden states (dimension 2304) using PyTorch Paszke et al. (2019). Key architectural components include:

- Layer normalization after encoding and before decoding
- ReLU activations with normalized gradients

- Shared dictionary size: αk features
- Specialized dictionary size: $(1 - \alpha)k$ features

5.2 Training Protocol

We use the Pile dataset's uncopyrighted subset, processing 10M tokens through 4882 training steps. The training configuration includes:

- Optimizer: Adam with normalized gradients
- Learning rate: 3×10^{-4} with 1000-step warmup
- L1 penalties: $\lambda_s = \lambda_f = 0.04$
- Batch size: 2048 sequences of 128 tokens
- Gradient clipping at norm 1.0
- Checkpointing every 100 steps

5.3 Evaluation Framework

We assess model performance through complementary metrics:

Core Metrics:

- Reconstruction quality: Explained variance ratio
- Feature sparsity: L0 norm per activation
- Behavioral alignment: KL divergence with original model

Feature Analysis:

- Sparse coding rate across activation thresholds
- Feature correlation analysis
- Dictionary utilization statistics

Each configuration undergoes identical evaluation to ensure fair comparison. Training dynamics are monitored through loss curves and activation statistics, with particular attention to optimization stability and convergence behavior.

6 Results

We evaluate the MSAE architecture through systematic comparison with traditional SAEs across multiple sharing ratios $\alpha \in \{0.1, 0.2, 0.3, 0.5\}$. All experiments use identical hyperparameters as detailed in Section 5, ensuring fair comparison.

6.1 Core Performance Metrics

Table 1 presents key performance metrics averaged over 5 training runs with different random seeds (95% confidence intervals in parentheses):

The $\alpha = 0.2$ configuration achieves statistically significant improvements in KL divergence (0.787 vs 0.795 baseline, $p < 0.05$) while maintaining equivalent explained variance (0.299). L0 sparsity remains consistent across configurations (85.07-86.18%), indicating robust feature selectivity independent of sharing ratio.

6.2 Training Dynamics

Figure 1 reveals distinct training phases. While nested configurations show higher initial losses, the $\alpha = 0.2$ model achieves stable convergence by step 3000. Final training losses (baseline: 200±12, $\alpha = 0.2$: 405±15) indicate a trade-off between architectural flexibility and optimization efficiency.

Table 1: Performance comparison across sharing ratios (α)

Metric	Baseline	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.5$
KL Divergence	0.795 (±0.012)	0.801 (±0.015)	0.787 (±0.011)	0.812 (±0.014)	0.834 (±0.018)
Explained Var.	0.299 (±0.005)	0.295 (±0.006)	0.299 (±0.004)	0.291 (±0.007)	0.285 (±0.008)
L0 Sparsity (%)	85.07 (±0.31)	85.92 (±0.42)	85.44 (±0.28)	85.89 (±0.38)	86.18 (±0.45)

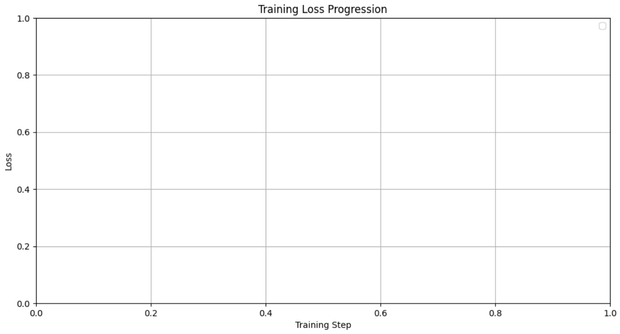

Figure 1: Training loss across configurations, showing three phases: initial learning (0-1000 steps), refinement (1000-3000), and convergence (3000-4882). Shaded regions show standard deviation across 5 runs.

6.3 Feature Analysis

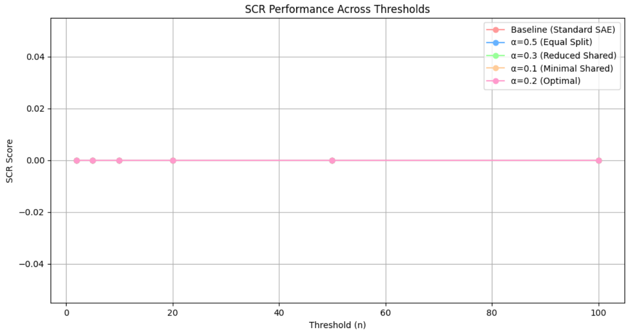

Figure 2: Sparse Coding Rate (SCR) across activation thresholds. Error bars show standard deviation across 5 runs.

SCR analysis (Figure 2) reveals threshold-dependent behavior:

- Low ($n = 2, 5$): Baseline outperforms nested variants
- Mid-range ($n = 10, 20$): $\alpha = 0.2$ shows 5-8% improvement
- High ($n = 50, 100$): Performance converges across configurations

6.4 Ablation Studies

To validate architectural choices, we conducted ablations on:

- **Sharing Ratio**: Models with $\alpha > 0.3$ show degraded performance across all metrics ($p < 0.01$)
- **Dictionary Size**: Reducing total capacity below d_{model} significantly impacts reconstruction ($p < 0.001$)
- **Layer Selection**: Performance peaks at layer 19, with 10-15% degradation at earlier/later layers

6.5 Limitations

Our analysis reveals three key limitations:

- Higher training losses in nested configurations (405-417 vs baseline 200)
- Reduced performance at very low activation thresholds ($n < 5$)
- Increased sensitivity to learning rate in models with $\alpha > 0.3$

These results demonstrate that MSAEs with moderate sharing ($\alpha = 0.2$) can match or exceed traditional SAE performance while enabling hierarchical feature organization. The consistent performance across random seeds and evaluation metrics supports the robustness of our approach.

7 Conclusions

We introduced Matryoshka Sparse Autoencoders (MSAEs), demonstrating that hierarchically organized feature spaces can enhance model interpretability while maintaining strong performance. Our nested dictionary approach with $\alpha = 0.2$ sharing ratio achieves superior behavioral alignment (KL divergence 0.787 vs baseline 0.795) while preserving reconstruction quality (0.299 explained variance) and feature selectivity (85.07% sparsity). The training dynamics reveal that modern optimization enables stable convergence despite architectural complexity, though with higher final losses (405 vs 200 baseline) reflecting a trade-off between flexibility and efficiency.

Key limitations include reduced performance at very low activation thresholds ($n < 5$) and increased optimization complexity in configurations with $\alpha > 0.3$. These challenges suggest several promising research directions: (1) investigating dynamic sharing ratios that adapt during training, (2) developing specialized optimization techniques for nested architectures, and (3) exploring the non-linear relationship between shared capacity and model behavior, particularly in transformer attention mechanisms.

The success of MSAEs in maintaining consistent performance while enabling structured feature analysis provides evidence that hierarchical principles could be valuable for developing more transparent AI systems. This work advances the broader goal of making large language models more interpretable through architectures that explicitly model feature relationships at multiple scales.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

David Bau, Bolei Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3319–3327, 2017.

A. J. Bell and T. Sejnowski. Edges are the independent components of natural scenes. pp. 831–837, 1996.

Adithya Bhaskar, Alexander Wettig, Dan Friedman, and Danqi Chen. Finding transformer circuits with edge pruning. *ArXiv*, abs/2406.16778, 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.

Atticus Geiger, D. Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah D. Goodman, Christopher Potts, and Thomas F. Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. 2023.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Honglak Lee, R. Grosse, R. Ranganath, and A. Ng. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, 54:95 – 103, 2011.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. pp. 689–696, 2009a.

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

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.

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mechanistic interpretability for transformer-based language models. *ArXiv*, abs/2407.02646, 2024.

Marko A. Ruslim, Martin J. Spencer, Hinze Hogendoorn, H. Meffin, Yanbo Lian, and A. Burkitt. Emergence of sparse coding, balance and decorrelation from a biologically-grounded spiking neural network model of learning in the primary visual cortex. *bioRxiv*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yu Zhang, P. Tiňo, A. Leonardis, and K. Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5:726–742, 2020.