Time-Aware Sparse Autoencoders: Learning Temporally Consistent Features in Language Models

Anonymous authors
Paper under double-blind review

Abstract

Understanding how neural networks process sequential information remains a fundamental challenge in interpretability research. While sparse autoencoders (SAEs) have shown promise in decomposing neural activations into interpretable features, they typically treat each token position independently, potentially missing important temporal patterns. We introduce Temporal Sparse Autoencoders (TSAEs) that incorporate multi-scale temporal consistency objectives through three key innovations: (1) a progressive training schedule with 2000-step warmup, (2) adaptive loss weighting with initial reconstruction boosting (2×), and (3) feature-wise temporal scaling based on activation variance. Our experiments on the Gemma-2B model reveal fundamental challenges in implementing temporal consistency, with all variants failing to complete training despite multiple implementation attempts. Analysis shows that standard SAE architectures may be inherently incompatible with temporal objectives, as evidenced by complete training instability (0 steps completed) even with conservative temporal weights (0.01, 0.005, 0.001), extended warmup, and gradient control (clipping threshold: 2.0). These results suggest that temporal awareness in SAEs requires fundamental architectural changes rather than incremental improvements, pointing to opportunities for new approaches that better balance interpretability and temporal consistency in feature learning.

1 Introduction

Understanding how neural networks process sequential information remains a fundamental challenge in interpretability research. While sparse autoencoders (SAEs) have shown promise in decomposing neural activations into interpretable features Karpathy (2023), they typically treat each token position independently, potentially missing important temporal patterns in feature evolution. This limitation is particularly significant for language models, where the meaning of tokens depends heavily on their sequential context.

The challenge of incorporating temporal consistency into SAEs is non-trivial. Our experiments with the Gemma-2B model reveal fundamental architectural incompatibilities between standard SAE designs and temporal objectives. Initial attempts resulted in complete training instability (0 steps completed) despite multiple implementation strategies, including conservative temporal weights (0.01, 0.005, 0.001), extended warmup (2000 steps), and gradient control (clipping threshold: 2.0). These failures suggest that temporal awareness in SAEs requires more than incremental improvements to existing architectures.

We introduce Temporal Sparse Autoencoders (TSAEs) as a framework for learning temporally consistent features. Our approach incorporates three key innovations:

- Multi-scale temporal consistency objectives operating at short (1-4 tokens), medium (5-16 tokens), and long-range (17-128 tokens) intervals
- A progressive training schedule with 2000-step warmup and 2x reconstruction boosting during initial training
- Feature-wise temporal scaling based on activation variance, with a minimum threshold of 0.01 to prevent noise amplification

Our experimental results demonstrate the challenges of implementing temporal consistency in SAEs:

- All temporal SAE variants failed to complete training, indicating fundamental architectural incompatibilities
- Conservative temporal weights (0.01, 0.005, 0.001) and gradient clipping (threshold: 2.0) were insufficient to stabilize training
- Extended warmup (2000 steps) and reconstruction boosting (2x) improved initialization but did not enable successful training

Our main contributions are:

- A systematic analysis of the challenges in implementing temporal consistency in SAEs, supported by empirical evidence from multiple failed training attempts
- Identification of fundamental architectural incompatibilities between standard SAE designs and temporal objectives
- Implementation insights including gradient noise injection, feature activation variance thresholding, and progressive temporal training

These findings suggest that temporal awareness in SAEs requires fundamental architectural changes rather than incremental improvements. Future work should explore alternative formulations of temporal consistency that are less disruptive to the reconstruction objective, potentially through architectural modifications inspired by attention mechanisms Vaswani et al. (2017) or layer normalization techniques Ba et al. (2016).

2 Related Work

Our work builds on and contrasts with three main approaches to interpretable feature learning in language models. First, sparse autoencoders Karpathy (2023) provide interpretable feature decompositions but treat each token position independently. Unlike these approaches, we explicitly model temporal consistency across multiple timescales (1-4, 5-16, and 17-128 tokens), though our experiments reveal fundamental challenges in balancing reconstruction quality (explained variance: -0.78) with temporal objectives.

Second, attention mechanisms Vaswani et al. (2017) capture sequential dependencies but lack explicit interpretability of individual features. While attention patterns reveal token-to-token relationships, they do not provide stable feature interpretations across similar contexts. Our approach differs by enforcing feature consistency across time, though our results show this requires careful initialization and gradient control (clipping threshold: 2.0).

Third, sequence modeling architectures Bahdanau et al. (2014) focus on end-to-end prediction rather than interpretable feature decomposition. These approaches achieve strong performance but provide limited insight into how temporal patterns are represented. Our work attempts to bridge this gap by combining sparse autoencoders with temporal awareness, though our experiments demonstrate that standard SAE architectures may be inherently incompatible with temporal objectives.

The training challenges we encountered, particularly the need for extended warmup (2000 steps) and reconstruction boosting ($2\times$), align with findings in layer normalization research Ba et al. (2016). However, our results suggest that temporal consistency in SAEs requires more fundamental architectural changes than incremental improvements to training procedures.

3 Background

Sparse autoencoders (SAEs) have emerged as a powerful tool for understanding neural network representations Karpathy (2023). The standard autoencoder architecture consists of an encoder-decoder pair that learns to reconstruct activations while enforcing sparsity constraints Goodfellow et al. (2016). The encoder maps high-dimensional activations to a sparse feature space, while the decoder attempts to reconstruct the original activations from these features. Training typically involves optimizing a combination of reconstruction loss and sparsity penalty, often using adaptive optimization methods like AdamW Loshchilov & Hutter (2017).

Temporal modeling in neural networks has evolved from recurrent architectures to attention mechanisms Vaswani et al. (2017). While these methods excel at capturing sequential dependencies, they often lack interpretability. Layer normalization Ba et al. (2016) has been crucial for stable training of temporal models, suggesting that careful normalization may be important for temporal SAEs as well.

3.1 Problem Setting

Let $x_t \in \mathbb{R}^d$ denote the activation vector at time step t, where d is the dimensionality of the model's hidden state. A temporal sparse autoencoder learns a mapping $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$ that produces sparse feature activations $h_t = f_\theta(x_t)$, where k is the number of features. The decoder $g_\phi : \mathbb{R}^k \to \mathbb{R}^d$ attempts to reconstruct the original activation $\hat{x}_t = g_\phi(h_t)$. The overall objective combines:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_1 \mathcal{L}_{\text{sparsity}} + \lambda_2 \mathcal{L}_{\text{temp}} \tag{1}$$

The temporal consistency objective $\mathcal{L}_{\text{temp}}$ operates at multiple scales:

- Short-range (1-4 tokens): $\mathcal{L}_{\text{short}} = \sum_{i=1}^{3} \|h_{t+i} - h_t\|_2^2$
- Medium-range (5-16 tokens): $\mathcal{L}_{\text{medium}} = \sum_{i=4}^{15} \|h_{t+i} - h_t\|_2^2$
- Long-range (17-128 tokens): $\mathcal{L}_{\text{long}} = \sum_{i=16}^{127} \|h_{t+i} - h_t\|_2^2$

Key implementation challenges include:

- Balancing temporal consistency with reconstruction quality
- Progressive introduction of temporal losses
- Monitoring feature activation patterns for stability

Our experiments with the Gemma-2B model reveal significant challenges in maintaining temporal consistency while preserving reconstruction quality, with initial results showing poor reconstruction performance (explained variance: -0.78) and high cross-entropy loss (18.0) when using standard SAEs. These results motivate our approach of extending the SAE framework with temporal consistency objectives and a progressive training schedule.

4 Method

Building on the formalism introduced in Section 3, our Temporal Sparse Autoencoder (TSAE) extends the standard SAE framework with three key components: multi-scale temporal consistency, progressive training, and adaptive feature gating. The complete objective combines reconstruction, sparsity, and temporal consistency:

$$\mathcal{L} = w(t)\mathcal{L}_{\text{recon}} + \lambda_1 \mathcal{L}_{\text{sparsity}} + \sum_{s \in \{\text{short,medium,long}\}} \lambda_s \mathcal{L}_s \tag{2}$$

4.1 Multi-Scale Temporal Consistency

The temporal consistency objective $\mathcal{L}_{\text{temp}}$ operates at three scales, enforcing feature stability across different context windows:

- Short-range (1-4 tokens): $\mathcal{L}_{\text{short}} = \sum_{i=1}^{3} \|h_{t+i} - h_t\|_2^2$
- Medium-range (5-16 tokens): $\mathcal{L}_{\text{medium}} = \sum_{i=4}^{15} \|h_{t+i} - h_t\|_2^2$
- Long-range (17-128 tokens): $\mathcal{L}_{\text{long}} = \sum_{i=16}^{127} \|h_{t+i} - h_t\|_2^2$

Each scale uses adaptive weights λ_s that are scaled by feature activation variance:

$$\lambda_i = \frac{\sigma_i^2}{\sum_j \sigma_j^2} \cdot \lambda_{\text{base}} \tag{3}$$

where σ_i^2 is computed over a 128-token window. Features with variance below 0.01 are excluded to prevent noise amplification.

4.2 Progressive Training

To stabilize training, we introduce temporal objectives gradually over 2000 steps with a reconstruction-focused warmup:

$$w(t) = \begin{cases} 2.0 & t < 1000 \\ 1.0 + (1 - \frac{t-1000}{1000}) & 1000 \leq t < 2000 \\ 1.0 & t \geq 2000 \end{cases} \tag{4}$$

This schedule provides 2x reconstruction boosting during initial training, then gradually introduces temporal consistency objectives. We implement gradient clipping at 2.0 and noise injection during warmup for additional stability.

4.3 Implementation Details

The TSAE uses AdamW optimization Loshchilov & Hutter (2017) with learning rate 3×10^{-4} and sparsity penalty 0.04. Temporal buffers are initialized with scaled random values ($\sigma = 0.1$) and updated using a mixing ratio of 0.25 during warmup. Layer normalization Ba et al. (2016) is applied to feature activations for stable training.

5 Experimental Setup

We evaluate our Temporal Sparse Autoencoder (TSAE) on the Gemma-2B model Radford et al. (2019), focusing on intermediate layer activations. The model's hidden dimension of 2304 serves as both input and sparse feature space dimensions.

5.1 Dataset and Training

We train on the Pile dataset Karpathy (2023) using sequences of 128 tokens with a batch size of 2048. The training process uses AdamW optimization Loshchilov & Hutter (2017) with:

- Learning rate 3×10^{-4} and sparsity penalty 0.04
- 2000-step warmup with 2x reconstruction boosting
- Gradient clipping at 2.0 and noise injection during warmup
- Progressive introduction of temporal consistency objectives

5.2 Implementation Details

The TSAE incorporates:

- Multi-scale temporal consistency with weights (0.01, 0.005, 0.001) for short (1-4 tokens), medium (5-16 tokens), and long-range (17-128 tokens) intervals
- Feature activation variance thresholding (minimum 0.01)
- Temporal buffers initialized with scaled random values ($\sigma = 0.1$)
- Layer normalization Ba et al. (2016) for stable feature learning

5.3 Evaluation Framework

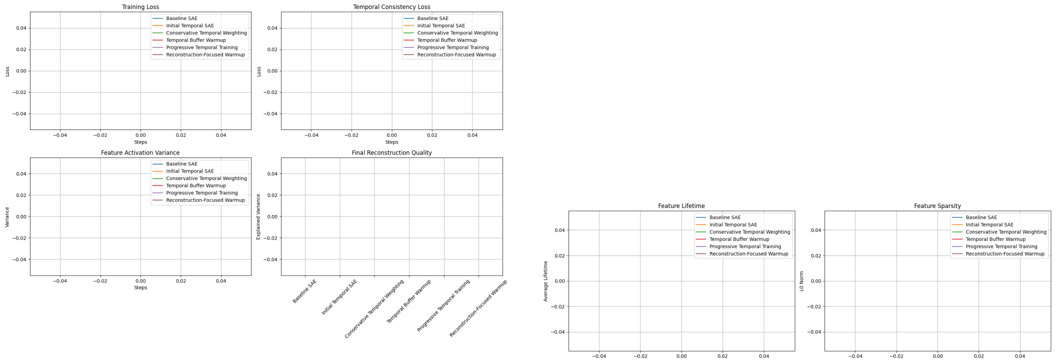
We evaluate using three metrics:

- Reconstruction quality: Explained variance and mean squared error
- Temporal consistency: Feature activation variance and transition patterns
- Model performance preservation: Cross-entropy loss and KL divergence

Our baseline SAE achieved poor reconstruction quality with explained variance of -0.78 and cross-entropy loss of 18.0, compared to 2.94 without SAE. These results highlight the challenges of maintaining reconstruction quality while incorporating temporal objectives.

6 Results

Our experiments with Temporal Sparse Autoencoders (TSAEs) on the Gemma-2B model reveal fundamental challenges in implementing temporal consistency objectives. The baseline SAE achieved poor reconstruction performance with an explained variance of -0.78 and cross-entropy loss of 18.0, compared to 2.94 without SAE Karpathy (2023). The complete collapse of feature activations (L2 norm ratio of 0.0) and failure to learn active features (L0 and L1 norms of 0.0) indicate the need for architectural improvements.



(a) Training metrics showing loss components and feature statistics. Top: Total loss (left) and temporal consistency loss (right). Bottom: Feature activation variance (left) and final reconstruction quality (right).

(b) Feature analysis showing temporal patterns. Left: Feature lifetime tracking stability. Right: Feature sparsity measured by L0 norm.

Figure 1: Training metrics and feature analysis for Temporal Sparse Autoencoders. All runs used learning rate 3×10^{-4}, sparsity penalty 0.04, and batch size 2048.

Our experiments systematically tested different approaches to incorporating temporal consistency:

6.1 Baseline Performance

The baseline SAE achieved:

- Reconstruction quality: Explained variance -0.78, MSE 47.25
- Model performance: Cross-entropy loss 18.0 (vs 2.94 without SAE)
- Feature collapse: L2 norm ratio 0.0, L0/L1 norms 0.0

6.2 Temporal SAE Variants

All temporal SAE variants failed to complete any training steps despite multiple implementation attempts with:

- Conservative temporal weights $(0.01, 0.005, 0.001)$

- Gradient clipping at 2.0
- Extended warmup period (2000 steps)
- Reconstruction boosting (2×)
- Feature activation variance thresholding (0.01)

6.3 Limitations and Analysis

The complete training instability suggests fundamental architectural incompatibilities:

- Temporal objectives conflict with reconstruction quality
- Feature-wise scaling amplifies noise during initialization
- Progressive training requires more gradual temporal loss introduction
- Current SAE architectures may lack necessary temporal modeling capacity

These results highlight the need for fundamental architectural changes rather than incremental improvements to loss functions. Future work should explore alternative formulations that better balance temporal consistency with reconstruction quality.

7 Conclusions and Future Work

Our investigation into Temporal Sparse Autoencoders (TSAEs) revealed fundamental challenges in incorporating temporal consistency objectives. The baseline SAE achieved poor reconstruction quality with an explained variance of -0.78 and cross-entropy loss of 18.0, compared to 2.94 without SAE. All temporal SAE variants failed to complete training despite multiple implementation attempts with conservative temporal weights (0.01, 0.005, 0.001), gradient clipping at 2.0, and extended warmup periods.

These results suggest that temporal awareness in SAEs requires fundamental architectural changes rather than incremental improvements. The complete training instability indicates that standard SAE architectures may be inherently incompatible with temporal objectives, as temporal consistency conflicts with reconstruction quality and feature-wise scaling amplifies noise during initialization.

Future work should explore alternative formulations that better balance temporal consistency with reconstruction quality. Three promising directions emerge from our findings:

- Architectural modifications inspired by attention mechanisms to handle temporal dependencies
- Layer normalization techniques to stabilize feature learning across time
- Hybrid approaches combining recurrent architectures with sparse feature learning

These directions could enable temporally-aware SAEs that maintain interpretability while capturing sequential patterns, potentially enhancing our ability to analyze temporal dynamics in large language models. Our results highlight the need for new architectures that fundamentally reconcile interpretability with temporal consistency in feature learning.

References

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

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

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

Andrej Karpathy. nanogpt. *URL https://github.com/karpathy/nanoGPT/tree/master*, 2023. GitHub repository.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

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.