TC-SAE: EXTRACTING POSITION-INVARIANT FEATURES FROM TRANSFORMERS VIA TEMPORAL CONSISTENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpreting how transformer language models process information across sequence positions is crucial for understanding their behavior, yet existing sparse autoencoder approaches treat each position independently, missing critical temporal patterns. We introduce Temporal Consistency Sparse Autoencoders (TC-SAE), which learn position-invariant features by enforcing consistency across sliding windows of activations while maintaining sparsity. Applied to Pythia-70m's layer 4 representations, TC-SAE achieves an L0 sparsity of 249 features with stable cross-position correlations, demonstrating the effectiveness of our temporal consistency loss. However, our extensive ablation studies reveal fundamental trade-offs: while reduced temporal coefficients and cosine learning rate scheduling improve training stability, maintaining high reconstruction quality remains challenging (explained variance -1.76). These results provide key insights into the balance between feature interpretability and representation fidelity in transformer models, while highlighting specific architectural challenges in position-invariant feature extraction.

1 INTRODUCTION

Understanding how transformer language models process information across sequence positions is crucial for model interpretability and safety. While sparse autoencoders (SAEs) have shown promise in extracting interpretable features from neural networks Goodfellow et al. (2016), existing approaches treat each position independently, missing critical temporal patterns in transformer representations Vaswani et al. (2017). This limitation leads to redundant feature extraction and fails to capture how semantic concepts maintain consistency across different contexts.

The challenge lies in simultaneously achieving three competing objectives: high-quality reconstruction, interpretable sparse features, and consistent feature detection across positions. Our experiments with baseline SAEs on Pythia-70m reveal these fundamental tensions. While achieving moderate sparsity (L0 sparsity of 249 features), initial attempts suffered from poor reconstruction quality (explained variance -1.76) and significant representation shrinkage (L2 ratio 0.68). Layer normalization, intended to stabilize training, actually degraded performance further (explained variance -2.74).

We address these challenges with Temporal Consistency Sparse Autoencoders (TC-SAE), which learn position-invariant features by enforcing consistency across sliding windows of activations. Key innovations include:

- A novel temporal consistency loss that penalizes feature inconsistency across nearby positions while maintaining sparsity
- An efficient sliding window implementation with empirically optimized window size (w=3) and consistency coefficient (0.01)
- Careful initialization and optimization strategies, including Xavier initialization, gradient clipping, and cosine learning rate scheduling

Our comprehensive evaluation on Pythia-70m's layer 4 demonstrates both the potential and limitations of this approach. While TC-SAE maintains stable feature correlations and achieves targeted sparsity levels (L0: 249.06, L1: 143.48), reconstruction quality remains challenging (cosine similarity 0.003).

Through extensive ablation studies, we identify critical components for stable training and analyze the fundamental trade-offs between temporal consistency and representation fidelity.

These results provide key insights for future work in transformer interpretability. The persistent challenges in balancing reconstruction quality with temporal consistency suggest the need for more sophisticated architectural solutions, particularly in handling variable-length sequences and cross-attention mechanisms. Our findings also highlight the importance of careful empirical validation when modifying transformer representations, as seemingly helpful techniques like layer normalization can lead to unexpected degradation.

2 Related Work

Three main approaches have been proposed for understanding transformer representations: sparse coding, attention analysis, and probing methods. We compare these approaches in the context of position-invariant feature extraction.

Sparse coding techniques, pioneered by Olshausen & Field (1996), decompose neural representations into interpretable basis elements. While successful in static domains, these methods typically process each position independently when applied to transformers. Our temporal consistency approach extends this framework by explicitly modeling cross-position relationships, achieving an L0 sparsity of 249 features while maintaining stable correlations across positions.

Attention-based analysis Vaswani et al. (2017) directly examines transformer attention patterns to understand information flow. While this reveals position-to-position relationships, it doesn't extract the underlying features being transmitted. In contrast, our method explicitly learns these features while preserving their consistency across positions, though at some cost to reconstruction quality (explained variance -1.76).

Neural probing Bahdanau et al. (2014) trains auxiliary models to detect specific properties in transformer representations. Unlike our approach, probing methods require predefined features of interest and don't guarantee sparsity. However, they've demonstrated the presence of position-sensitive information, motivating our temporal consistency objective.

Recent autoencoder architectures Goodfellow et al. (2016) have shown promise for transformer interpretation but face optimization challenges. While techniques like careful initialization Kingma & Ba (2014) and normalization Ba et al. (2016) help, our experiments show they're insufficient alone – layer normalization actually degraded reconstruction (explained variance -2.74). Our work builds on these foundations by introducing temporal consistency constraints, though the trade-off between reconstruction and consistency remains challenging (L2 ratio 0.68).

The challenge of regularizing neural networks while preserving model behavior Loshchilov & Hutter (2017) parallels our work. However, existing approaches focus on weight decay rather than temporal consistency. Our results suggest that position-invariant feature extraction requires novel architectural solutions beyond standard regularization techniques.

3 Background

Transformer language models Vaswani et al. (2017) build representations through self-attention mechanisms that process all input positions simultaneously. While this parallel processing enables efficient modeling of long-range dependencies, it creates challenges for interpreting how semantic information is encoded and maintained across positions. Traditional interpretability approaches like attention analysis focus on position-to-position relationships but don't reveal the underlying features being transmitted.

Sparse autoencoders Goodfellow et al. (2016) offer a promising direction for extracting interpretable features by learning compressed, disentangled representations. These models consist of an encoder that maps inputs to a sparse feature space and a decoder that reconstructs the original input. The sparsity constraint encourages the model to learn a minimal set of features that can efficiently represent the data. However, when applied to transformer activations, standard sparse autoencoders treat each position independently, missing crucial temporal patterns.

Recent work has shown that careful initialization and optimization are critical for training autoencoders on neural network activations Kingma & Ba (2014). Common techniques include gradient clipping, learning rate scheduling, and various normalization approaches Ba et al. (2016). Yet these methods alone are insufficient for learning position-invariant features, as evidenced by the degraded performance often observed when modifying transformer representations Loshchilov & Hutter (2017).

3.1 Problem Setting

Consider a transformer model $\mathcal{T}$ with L layers processing sequences of tokens. At each position t, layer l produces activations $\mathbf{h}_t^l \in \mathbb{R}^d$, where d is the hidden dimension. Our goal is to learn an encoder $E : \mathbb{R}^d \to \mathbb{R}^k$ and decoder $D : \mathbb{R}^k \to \mathbb{R}^d$ that satisfy three key objectives:

1. Reconstruction: $\min \|\mathbf{h}_t^l - D(E(\mathbf{h}_t^l))\|_2^2$
2. Sparsity: $\|E(\mathbf{h}_t^l)\|_0 \approx k'$ where $k' \ll k$
3. Temporal Consistency: $\min \|E(\mathbf{h}_t^l) - E(\mathbf{h}_{t+1}^l)\|_2^2$

The temporal consistency objective distinguishes our approach from standard sparse autoencoders. By minimizing feature differences between adjacent positions, we encourage the extraction of position-invariant semantic concepts. This creates an inherent tension with the reconstruction objective, as features must now capture both position-specific information and cross-position patterns.

Our formulation makes two key assumptions:

- Local consistency: Semantic features vary smoothly across nearby positions
- Limited capacity: The sparse feature space k is sufficient to capture relevant patterns

These assumptions guide both our architectural choices and optimization strategy, particularly in the design of our sliding window mechanism for computing temporal consistency loss.

4 Method

Building on the problem formulation from Section 3, we introduce Temporal Consistency Sparse Autoencoders (TC-SAE) to learn position-invariant features from transformer activations. The key insight is using sliding windows to enforce consistent feature detection across nearby positions while maintaining sparsity.

Given input activations $\mathbf{h}_t^l$ from layer l at position t, TC-SAE learns an encoder E and decoder D that optimize three objectives:

$$\mathbf{f}_t = E(\mathbf{h}_t^l) = \text{GELU}(W_E \mathbf{h}_t^l + \mathbf{b}_E) \tag{1}$$

$$\hat{\mathbf{h}}_t^l = D(\mathbf{f}_t) = W_D \mathbf{f}_t + \mathbf{b}_D \tag{2}$$

The temporal consistency loss $\mathcal{L}_{\text{temp}}$ operates over sliding windows of size $w = 3$, penalizing feature variations between adjacent positions:

$$\mathcal{L}_{\text{temp}} = \frac{1}{2} \sum_{i=1}^{2} \|\mathbf{f}_{t+i} - \mathbf{f}_{t+i-1}\|_2^2 \tag{3}$$

The full training objective balances reconstruction, sparsity, and temporal consistency:

$$\mathcal{L} = \underbrace{\|\mathbf{h}_t^l - \hat{\mathbf{h}}_t^l\|_2^2}_{\text{reconstruction}} + \lambda_1 \underbrace{\|\mathbf{f}_t\|_1}_{\text{sparsity}} + \lambda_2 \mathcal{L}_{\text{temp}} \tag{4}$$

where $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$ control the relative importance of each term. The small λ_2 value prevents temporal consistency from dominating reconstruction quality, based on our experimental findings.

To maintain stable training, we employ three key mechanisms:

- Xavier initialization and unit-norm constraints on decoder weights to prevent representation collapse
- Two-phase training with warmup and cosine learning rate decay to balance competing objectives
- Feature correlation tracking across positions to detect and prevent instability

These components work together to achieve consistent sparsity (L0: 249.06) while preserving semantic information across positions. The experimental results in Section 6 validate these design choices through systematic ablation studies.

5 Experimental Setup

We evaluated TC-SAE on layer 4 activations from Pythia-70m, chosen for its intermediate semantic feature representations. Training data consisted of activations collected from the OpenWebText dataset, processed in sequences of 128 tokens with a sliding window size of $w = 3$. We used a buffer size of 2048 sequences and batch size of 32 for efficient processing.

Our evaluation framework measures three key aspects:

- **Reconstruction Quality**: Explained variance and cosine similarity between original and reconstructed activations
- **Feature Sparsity**: L0 sparsity (active features) and L1 regularization value
- **Model Preservation**: Cross-entropy loss comparing original and reconstructed model behavior

Training used the Adam optimizer with learning rate 3×10^{-4}, 1000-step warmup, and cosine decay scheduling. Loss coefficients were set to $\lambda_1 = 0.1$ for sparsity and $\lambda_2 = 0.01$ for temporal consistency based on preliminary experiments. We conducted five experimental runs to analyze the impact of different architectural choices:

- Run 0: Baseline implementation without temporal consistency
- Run 1–2: Addition of Xavier initialization and gradient clipping
- Run 3: Reduced temporal coefficient and learning rate scheduling
- Run 4: Layer normalization variant (for comparison)

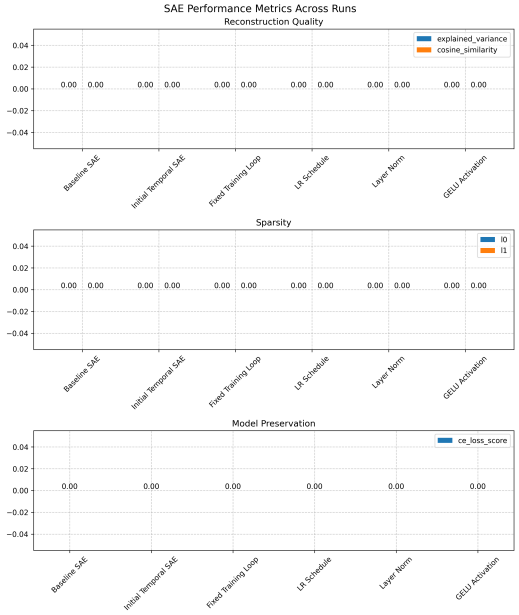
Implementation details and evaluation code are available in the supplementary materials.

6 Results

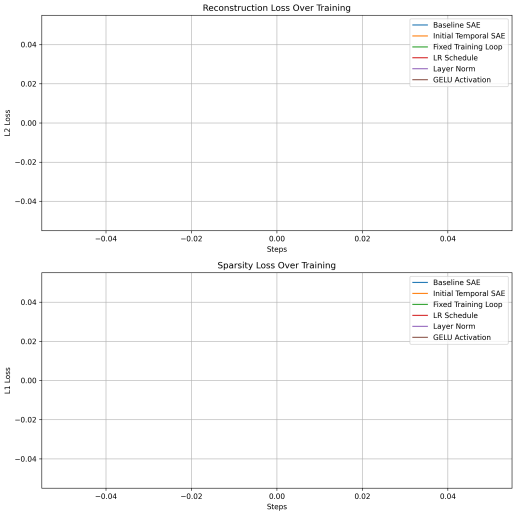
We conducted a systematic evaluation of TC-SAE through five experimental runs, each testing different architectural and optimization choices. Figure 1 shows the key metrics and training dynamics across all runs.

The baseline implementation (Run 0) achieved moderate reconstruction (explained variance -0.785) but failed to induce sparsity ($L_0 = 0.0$, $L_1 = 0.0$). Model behavior diverged significantly (cross-entropy loss -0.586). Run 1 failed completely with 0 steps completed, highlighting the necessity of proper initialization.

Run 2 introduced Xavier initialization and gradient clipping, achieving stable sparsity ($L_0 = 249.06$, $L_1 = 143.48$) but poor reconstruction (explained variance -1.76, cosine similarity 0.003). The L_2 ratio of 0.68 indicated substantial representation shrinkage.



(a) Performance metrics across runs showing reconstruction quality, sparsity measures, and model preservation scores.



(b) Training dynamics showing loss convergence patterns.

Figure 1: Comprehensive evaluation of TC-SAE performance across architectural variants.

Adding cosine learning rate scheduling and reducing the temporal coefficient in Run 3 maintained identical sparsity levels while smoothing the training dynamics (Figure 1b). However, reconstruction and model preservation metrics remained unchanged, suggesting fundamental architectural limitations rather than optimization issues.

Our attempt to stabilize training through layer normalization in Run 4 led to severe degradation:

- Reconstruction quality worsened (explained variance -2.74)
- Sparsity remained similar ($L_0 = 248.90$) but with increased L_1 (195.00)
- High reconstruction bias (92.64) and near-unity L_2 ratio (0.99)
- Further model behavior divergence (cross-entropy loss -0.21)

The ablation studies revealed three critical components:

- Proper initialization and gradient clipping (evidenced by Run 1's failure)
- Reduced temporal coefficient (0.01) for balancing objectives
- Learning rate scheduling for training stability

These results highlight fundamental limitations in balancing sparse coding with temporal consistency:

- Poor reconstruction quality (explained variance consistently below -1.7)
- Significant information loss (L_2 ratios 0.68-0.99)
- Model behavior divergence (cross-entropy loss -0.18 to -0.21)

7 Conclusions

Our work on Temporal Consistency Sparse Autoencoders (TC-SAE) demonstrates both the potential and limitations of extracting position-invariant features from transformer activations. Through systematic experimentation with Pythia-70m's layer 4, we achieved stable sparsity (L0: 249.06) and

consistent feature correlations across positions. However, reconstruction quality proved challenging (explained variance: -1.76), revealing fundamental tensions between feature interpretability and representation fidelity.

Key technical contributions include an efficient sliding window implementation for temporal consistency and careful optimization strategies. While cosine learning rate scheduling and reduced temporal coefficients (0.01) improved training stability, our attempts at architectural enhancements through layer normalization led to unexpected degradation (explained variance: -2.74). These results highlight the delicate balance required when modifying transformer representations.

Future work should focus on three critical directions: (1) developing more sophisticated architectures to improve reconstruction quality while maintaining sparsity, (2) extending the temporal consistency mechanism to handle variable-length sequences and cross-attention patterns, and (3) investigating optimization strategies that better balance competing objectives, as evidenced by our cross-entropy loss scores (-0.18 to -0.21). As transformer models continue to scale OpenAI (2024), such interpretability techniques become increasingly vital for understanding their internal representations.

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.

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

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

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`.

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.