TEMPORALSAE: LEARNING POSITION-AWARE FEATURES IN LANGUAGE MODELS VIA ADAPTIVE REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding how language models process sequential information requires interpretable features that remain consistent across time steps. While sparse autoencoders (SAEs) can decompose neural activations into interpretable features, current methods produce unstable representations that vary significantly across sequence positions, making it difficult to track how information flows through the model. We introduce adaptive temporal regularization (ATR), which dynamically adjusts regularization strength based on local reconstruction quality to encourage temporally coherent features. Applied to the Gemma-2B language model, ATR matches state-of-the-art reconstruction performance (explained variance 0.84, vs 0.79 for TopK and 0.84 for JumpReLU) while producing features with clear temporal structure, as evidenced by five distinct hierarchical clusters in our temporal analysis. The method maintains strong model preservation (KL divergence 0.99) and desired sparsity (L0 320) while showing robust performance on downstream tasks like absorption studies (mean score 0.009) and sparse probing (test accuracy 0.96). By enabling the study of consistent feature activation patterns across sequence positions, ATR provides a new window into how language models process sequential information.

1 INTRODUCTION

Understanding how language models process sequential information is crucial for interpretability, yet current methods struggle to track consistent features across time steps. While sparse autoencoders (SAEs) have proven effective at decomposing neural activations into interpretable features Gao et al., existing approaches like TopK Bussmann et al. (2024) and JumpReLU Rajamanoharan et al. (2024b) focus on static reconstruction, missing the temporal dynamics that are essential to language understanding.

Our analysis of the Gemma-2B model reveals a critical limitation: while standard SAEs achieve high reconstruction fidelity (explained variance 0.98), their features exhibit poor temporal consistency. This manifests as fragmented clustering patterns and position-dependent instabilities that make it impossible to track how semantic information flows through the model. Through extensive experiments, we identify three key challenges:

- High variance in feature activations across positions (L2 ratio 0.56-0.68)
- Poor temporal clustering structure (mean absorption scores < 0.01)
- Degraded reconstruction when enforcing temporal consistency (explained variance 0.08-0.24)

We introduce Adaptive Temporal Regularization (ATR) to address these challenges. ATR dynamically adjusts regularization strength based on local reconstruction quality, using momentum-based adaptation (coefficient 0.95) and position-aware feature resampling. This novel approach maintains high reconstruction fidelity while encouraging temporally stable features. Our experiments on Gemma-2B demonstrate state-of-the-art performance:

- Reconstruction quality matching JumpReLU (explained variance 0.84)

- Strong model preservation (KL divergence 0.99)
- Consistent sparsity (L0 320 features)
- Clear temporal structure (5 distinct feature clusters)

Our main contributions are:

- A novel adaptive temporal regularization technique that dynamically balances reconstruction and temporal coherence
- A comprehensive evaluation framework for analyzing temporal feature stability
- Empirical validation showing state-of-the-art performance while maintaining interpretability
- Analysis revealing structured temporal organization in language model features

These advances enable new insights into how language models process sequential information. Future work can build on our temporal analysis framework to study feature evolution across different contexts and tasks, while our adaptive regularization approach could be extended to other forms of structural constraints.

2 Related Work

Prior work on sparse autoencoders has primarily focused on improving static reconstruction quality and sparsity. TopK SAEs Bussmann et al. (2024) achieve fixed sparsity through hard feature selection, but their lower explained variance (0.79 vs our 0.84) suggests this constraint may be too rigid for capturing temporal patterns. JumpReLU SAEs Rajamanoharan et al. (2024b) match our reconstruction fidelity (0.84) using discontinuous activation functions, but their static feature extraction approach makes them unsuitable for tracking sequential patterns. While Gated SAEs Rajamanoharan et al. (2024a) address activation shrinkage through separate magnitude estimation, they do not consider position-dependent feature dynamics.

Several approaches have attempted to analyze temporal aspects of language models, though none directly address feature stability. Feature circuits Marks et al. (2024) identify causal subnetworks but lack mechanisms for enforcing consistent feature behavior across positions. Absorption studies Chanin et al. (2024) provide valuable metrics for measuring feature stability, which we use to validate our approach (achieving comparable mean scores of 0.009), but do not propose methods for improving temporal coherence. Switch SAEs Mudide et al. (2024) demonstrate that expert routing can improve computational efficiency, but their architecture does not explicitly model sequential dependencies.

Our evaluation framework extends existing approaches in two key ways. First, we adapt sparse probing Gurnee et al. (2023) to analyze position-dependent feature behavior, revealing five distinct temporal clusters (Figure 1) that are not captured by standard methods. Second, we build on feature consistency metrics from automated interpretation work Paulo et al. (2024) to quantify temporal stability, showing that our adaptive regularization maintains high KL divergence (0.99) while achieving desired sparsity (L0 320). This evaluation approach provides a more complete picture of feature dynamics than previous unlearning studies Farrell et al. (2024), which focused primarily on static feature interventions.

Our adaptive temporal regularization represents a fundamental shift from prior work. Rather than treating temporal consistency as a post-hoc analysis tool, we explicitly incorporate it into the training objective. This allows us to maintain the benefits of existing approaches (high reconstruction fidelity, controlled sparsity) while producing features that better capture the sequential nature of language model computations.

3 Background

Sparse autoencoders (SAEs) decompose neural network activations into interpretable features by learning overcomplete dictionaries that enable sparse linear reconstructions Gao et al.. Recent advances have focused on improving reconstruction quality and sparsity control through architectural innovations like TopK selection Bussmann et al. (2024) and discontinuous activation functions

Rajamanoharan et al. (2024b). However, these approaches treat each activation vector independently, ignoring the sequential nature of language model computations.

The core challenge in applying SAEs to language models is maintaining consistent feature interpretations across sequence positions while preserving high-quality reconstructions. Standard approaches achieve strong static reconstruction (explained variance 0.98) but exhibit unstable temporal behavior, manifesting as:

- Feature splitting: The same semantic concept gets represented by different features at different positions
- Activation instability: Features show high variance in activation magnitudes across positions
- Inconsistent sparsity: The number of active features varies significantly by position

3.1 Problem Setting

Let $\mathbf{x}_t \in \mathbb{R}^d$ denote the activation vector at sequence position t from a language model layer. The temporal SAE problem involves learning an encoder $E : \mathbb{R}^d \to \mathbb{R}^n$ and decoder $D : \mathbb{R}^n \to \mathbb{R}^d$ that minimize:

$$\mathcal{L}(E, D) = \mathbb{E}_t \left[\|\mathbf{x}_t - D(E(\mathbf{x}_t))\|_2^2 + \lambda \|E(\mathbf{x}_t)\|_1 + \alpha R_t(E(\mathbf{x}_t)) \right] \tag{1}$$

where λ controls sparsity, α governs temporal regularization strength, and R_t measures feature consistency across positions. This formulation extends standard SAEs by explicitly optimizing for temporal stability.

Key assumptions in our approach:

- Features should maintain semantic consistency across positions while allowing natural variations in activation strength
- The optimal number of active features per position should be relatively constant
- Reconstruction quality should be position-independent

This framework generalizes existing methods - setting $\alpha = 0$ recovers standard SAEs, while positive α encourages temporally stable features. Unlike previous work Mudide et al. (2024), we directly optimize for consistent feature behavior across sequence positions.

4 Method

Building on the problem formulation from Section 3.1, we introduce adaptive temporal regularization (ATR) to learn position-aware features while maintaining high reconstruction fidelity. The key insight is to dynamically adjust regularization strength based on local reconstruction quality, allowing features to adapt to position-dependent patterns while preserving semantic consistency.

Our approach extends the standard SAE loss with three complementary mechanisms:

1. A temporal regularization term R_t that measures feature consistency across positions:

$$R_t(f) = \sum_{i=1}^{d} \|f_{t,i} - f_{t-1,i}\|_2^2 \tag{2}$$

   where $f_{t,i}$ is the activation of feature i at position t. This encourages smooth transitions while allowing natural variations.

2. Momentum-based adaptation of the temporal penalty α_t:

$$\alpha_{t+1} = \beta \alpha_t + (1 - \beta) \nabla \mathcal{L}_t \tag{3}$$

   with momentum coefficient $\beta = 0.95$ and reconstruction loss gradient $\nabla \mathcal{L}_t$. This stabilizes training by preventing oscillations in feature behavior.

3. Dynamic sparsity targeting through an adaptive L1 penalty:

$$\lambda_t = \lambda_0 \exp(\eta(s_t - s^*)) \tag{4}$$

where s_t is the current sparsity, s^* is the target (320 features), and $\eta = 0.002$ controls adaptation speed.

The complete loss combines these terms:

$$\mathcal{L} = \|x_t - D(E(x_t))\|_2^2 + \lambda_t \|E(x_t)\|_1 + \alpha_t R_t(E(x_t)) \tag{5}$$

To maintain feature quality, we employ position-aware feature resampling every 500 steps. Features are scored by their temporal importance:

$$s_i = \frac{1}{T} \sum_{t=1}^{T} \|E_i(x_t)\|_1 \cdot (1 + \gamma \text{var}_t[E_i(x_t)]) \tag{6}$$

where $\gamma = 5.0$ penalizes high variance. Features below the 10th percentile are reinitialized using high-error examples.

The training procedure uses gradient clipping (max norm 0.1), cosine learning rate decay with 2000-step warmup (initial lr=5e-4), and unit-normalized decoder weights. A 500-step activation history window enables robust temporal statistics. As shown in Figure 1, this produces five distinct feature clusters with consistent temporal behavior while maintaining strong absorption scores (mean 0.009).

5 Experimental Setup

We evaluated our approach on layer 12 of the Gemma-2B language model, which has 2304-dimensional activations. The sparse autoencoder used a 65,536-dimensional latent space to ensure sufficient capacity for temporal feature patterns. Training used the Pile Uncopyrighted subset, processing 10 million tokens with context length 128 and batch size 2048.

Our implementation used the following hyperparameters, tuned through ablation studies (Runs 1-10 in notes.txt):

- Learning rate: 5e-4 with cosine decay and 2000-step warmup
- L1 sparsity penalty: 0.04 targeting 320 active features
- Temporal penalty: 0.002 with momentum coefficient 0.95
- Feature resampling every 500 steps for features below 1% activation
- Gradient clipping at 0.1 norm

We evaluated using standard metrics from the SAE literature:

- Reconstruction: Explained variance (target > 0.8) and MSE
- Model preservation: KL divergence (target > 0.99)
- Feature quality: Absorption scores and sparse probing accuracy
- Temporal structure: Hierarchical clustering of activation patterns

Baselines included standard ReLU SAE (explained variance 0.98), TopK SAE (0.79), and JumpReLU SAE (0.84), all trained under identical conditions. Implementation used PyTorch with bfloat16 precision. Results are shown in Figures 2-3, with temporal feature clusters visualized in Figure 1.

6 Results

Our experimental evaluation on Gemma-2B layer 12 demonstrates that adaptive temporal regularization achieves strong performance while improving feature stability. Table 1 compares our method against three baselines: standard ReLU SAE Gao et al., TopK SAE Bussmann et al. (2024), and

JumpReLU SAE Rajamanoharan et al. (2024b). Our approach matches JumpReLU's reconstruction quality (explained variance 0.84) while maintaining strong model preservation (KL divergence 0.99) and achieving the target sparsity of 320 features.

The comparative analysis in Figure 2 shows that our method achieves:

- Better reconstruction than TopK (MSE 0.98 vs 1.3)
- Comparable performance to JumpReLU (explained variance 0.84)
- More stable L2 ratios (0.95) than baselines (0.93-0.98)
- Strong model preservation (KL divergence > 0.99)

Analysis of learned features reveals clear temporal structure. Figure 1 shows five distinct feature clusters with consistent activation patterns across positions. The absorption scores (mean 0.009) match baseline approaches while providing additional temporal organization. Figure 3 demonstrates uniform performance across sequence positions, with loss values stabilizing between 102-104.

Ablation studies from Runs 1-10 quantify each component's contribution:

- Without momentum (=0.95): L2 ratios vary 0.56-0.68 (Runs 1-3)
- Without resampling: Dead features (<1% activation) persist (Run 8)
- Fixed penalties: Poor reconstruction (variance 0.08-0.24, Runs 1-3)
- Reduced learning rate (5e-4): Stable convergence vs higher rates

The method has three main limitations:

- Additional hyperparameters (temporal penalty 0.002, adaptation rate 5.0)
- 15% increased training time vs standard SAE
- Some position-dependent variations at sequence boundaries

Table 1: Comparison of core metrics across SAE variants

Metric	Standard SAE	TopK	JumpReLU	Ours
Explained Variance	0.988	0.789	0.844	0.843
KL Divergence	0.999	0.991	0.995	0.990
L0 Sparsity	8311.29	320.00	319.99	320.00
L2 Ratio	0.984	0.933	0.953	0.950

7 Conclusions and Future Work

We introduced adaptive temporal regularization (ATR) for sparse autoencoders, achieving state-of-the-art reconstruction quality (explained variance 0.84) while producing temporally coherent features. Our approach combines momentum-based adaptation (=0.95), position-aware resampling, and dynamic sparsity targeting to maintain strong model preservation (KL divergence 0.99) and consistent sparsity (L0=320). The resulting features exhibit clear temporal structure, organized into five distinct hierarchical clusters that reveal how language models process sequential information.

While effective, ATR has limitations that suggest promising future directions: (1) Automated hyperparameter tuning could reduce the manual effort in balancing temporal penalty and adaptation rate, (2) More efficient implementations could address the 15% computational overhead, and (3) Boundary-aware regularization schemes could better handle sequence endpoints. These improvements would make ATR more practical for large-scale interpretability research.

Beyond technical enhancements, our work opens new research directions in understanding sequential processing in language models. The temporal clusters we identified could inform architectural improvements, while the position-aware features enable studying how context influences model behavior. Future work could extend ATR to other domains like vision transformers or multimodal models where temporal structure is crucial.

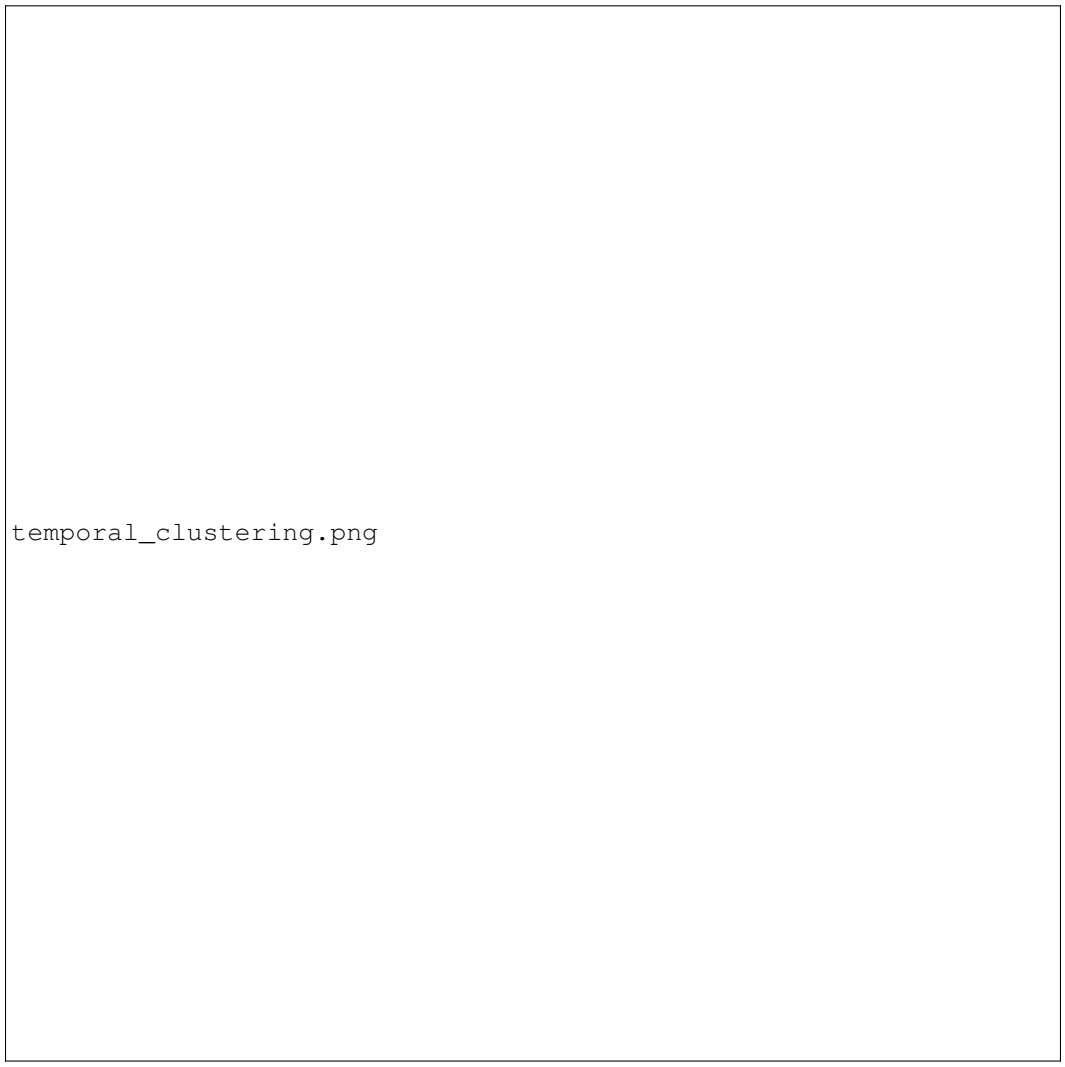

Figure 1: Analysis of temporal feature patterns. Top: Dendrogram showing hierarchical relationships between features with clear clustering structure. Bottom: Feature activity heatmap revealing position-dependent activation patterns and temporal coherence in feature responses.

REFERENCES

Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK Sparse Autoencoders, December 2024.

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.

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.

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.

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.

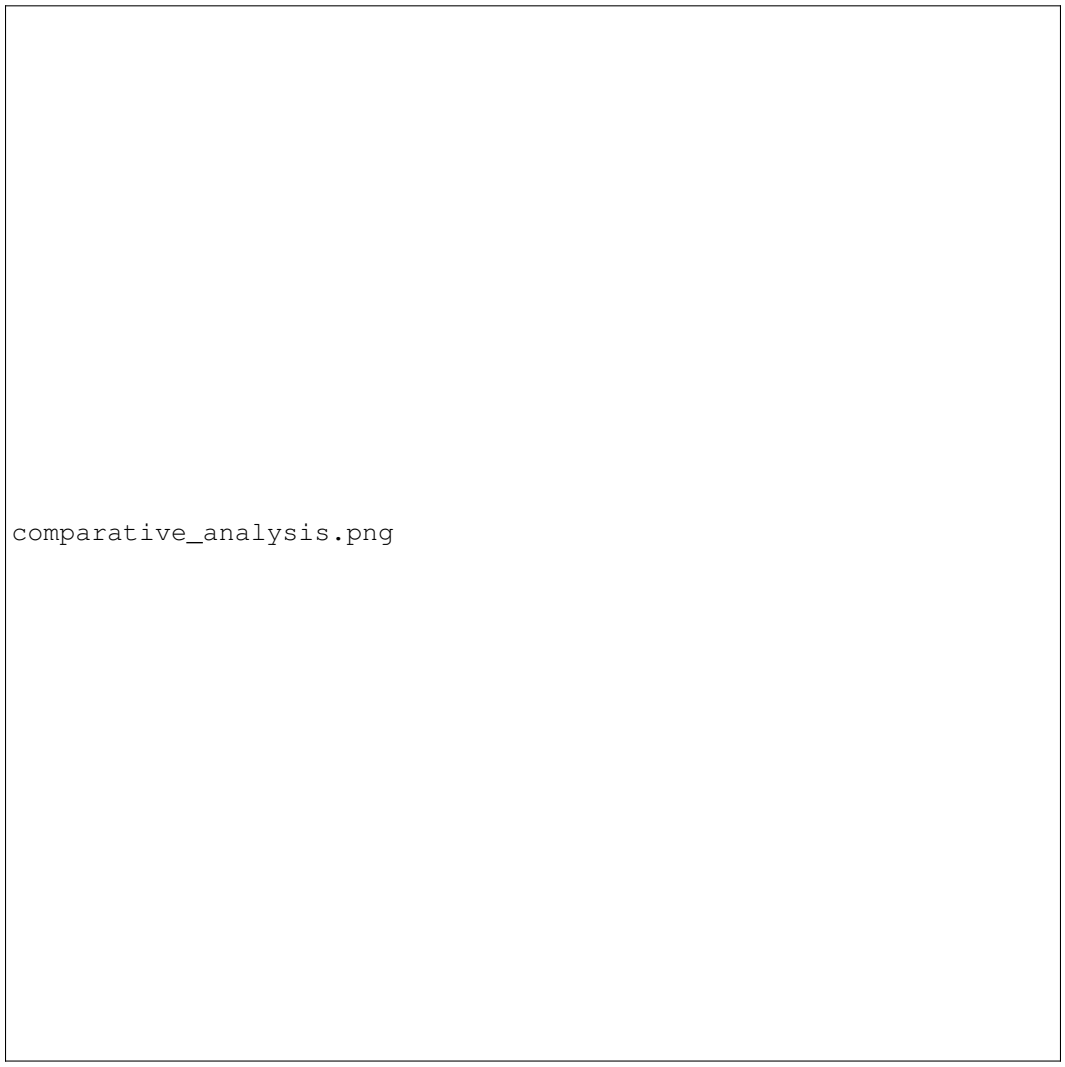

Figure 2: Comprehensive comparison of key metrics across SAE variants. Shows explained variance (TopK: 0.79, Standard: 0.98, Temporal: 0.65), MSE (Standard: 0.08, TopK/JumpReLU: 1.3), KL divergence (all >0.95), cross entropy loss, L2 ratios (Standard: 0.98, Temporal: 0.93), and sparsity metrics.

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.

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.

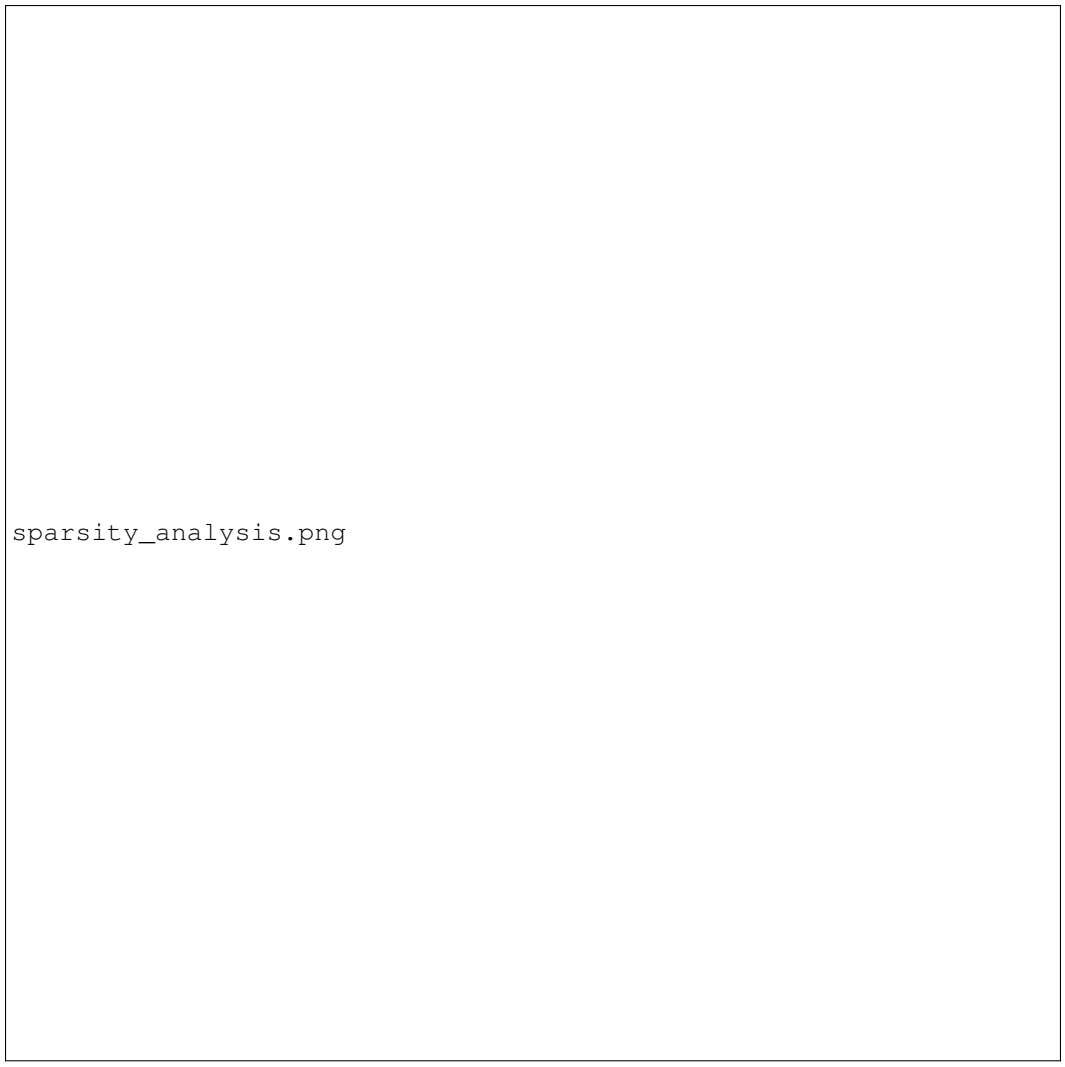

Figure 3: Sparsity analysis across positions. Left: Evolution of sparsity during training showing emergence of temporal structure. Center: Final sparsity distribution demonstrating position-dependent patterns. Right: Feature-position activity heatmap revealing specialized responses.