FINDING THE ZEITGEIST IN TIME SERIES FOUNDATION MODELS

Felix Divo[1] **Maurice Kraus**[1,2] **Ruben Härle**[1,2]
Patrick Ahrend[3,4] **Kristian Kersting**[1,5,6,7]

[1]AI & ML Group, TU Darmstadt [2]Lab1141 [3]BMW Group [4]TU München [5]hessian.AI
[6]German Research Center for AI (DFKI) [7]Centre for Cognitive Science, TU Darmstadt
{felix.divo,maurice.kraus,ruben.haerle,kersting}@cs.tu-darmstadt.de
patrick.ahrend@tum.de

ABSTRACT

Time series foundation models (TSFMs) achieve strong zero-shot and transfer performance across diverse forecasting tasks, yet their internal representations remain poorly understood. In language and vision models, sparse autoencoders (SAEs) have emerged as a powerful tool for mechanistic interpretability, revealing disentangled and often monosemantic features from high-dimensional residual streams. In this work, we explore whether similar structures can be uncovered in pretrained TSFMs. Our results demonstrate that SAE-based analysis provides a viable and scalable lens into the internal structure of TSFMs, uncovering sparse features that align with coherent temporal patterns. This work represents an initial step toward unsupervised mechanistic interpretability for TSFMs and highlights promising directions for future research.

Track: Research

1 INTRODUCTION

Time series foundation models (TSFMs) have recently emerged as a powerful paradigm for forecasting and representation learning across diverse temporal domains, achieving strong zero-shot and transfer performance by leveraging Transformer- and recurrence-based architectures originally developed for language and vision (Ansari et al., 2024; 2025; Auer et al., 2025; Liu et al., 2025; Das et al., 2024). Despite these empirical successes, the internal representations learned by TSFMs and how they support forecasting behavior remain poorly understood and are an active area of research (Pandey et al., 2025; Park et al., 2026). In contrast, mechanistic interpretability has made substantial progress in large language and vision models by extracting sparse, human-interpretable internal features using sparse autoencoders (SAEs) (Cunningham et al., 2024; Bricken et al., 2023; Stevens et al., 2025) emerging as a particularly effective tool for discovering disentangled, often monosemantic features. In this work, we investigate whether SAE-based analysis can similarly uncover meaningful and interpretable structure within pretrained TSFMs.

2 RELATED WORK

Interpretability for Time Series Foundation Models. Recent work has begun to examine the internal representations of TSFMs, motivated by concerns around transparency and trustworthiness (Park et al., 2026; Steinmann et al., 2024). Several studies analyze their representations using linear probes, representational similarity measures, or targeted interventions, revealing shared structural properties across layers and architectures as well as interpretable temporal concepts such as trend and periodicity (Wiliński et al., 2025; Bao et al., 2026; Pandey et al., 2025). Complementary efforts apply circuit-level or attribution-based analyses to specific time series models (Kalnāre et al., 2025; Queen et al., 2023), or trace failure modes, such as hallucinations, to representational misalignment (Zou et al., 2025). While these works provide valuable insights into what information TSFMs encode, they rely on probes or targeted analyses that require predefined concepts, and do not decompose hidden states into sparse, individually interpretable features amenable to unsupervised feature discovery.

Sparse Autoencoders for Mechanistic Interpretability. SAEs have emerged as a powerful tool for mechanistic interpretability by decomposing the high-dimensional internal model states into sparse, often monosemantic features. Building on earlier dictionary-learning approaches for transformers (Yun et al., 2021; Elhage et al., 2022; Sharkey et al., 2022), recent work has shown that SAEs can recover thousands to millions of interpretable features from language models and can be scaled to frontier architectures (Bricken et al., 2023; Cunningham et al., 2024; Templeton et al., 2024; Gao et al., 2024). Subsequent studies have explored the limits and variants of SAE-based feature discovery and demonstrated their applicability beyond language, including vision-language, protein, and audio models (Härle et al., 2025; Pach et al., 2025; Simon & Zou, 2025; Aparin et al., 2026). In the time series domain, SAEs have been applied to explain black-box forecasters at the input–output level (Oublal et al., 2026), but not to analyze the internal representations of pretrained TSFMs. To our knowledge, this work is the first to apply SAE-based analysis directly to hidden states of time series foundation models.

3 METHOD

Our goal is to use SAEs as a post-hoc lens into the hidden representations of a TSFM. We keep the base model frozen and train SAEs on its internal residual stream states. This section formalizes the setup and highlights adaptations specific to time series. A complete overview of the proposed system is shown in Figure 1.

3.1 SAE FORMULATION

We train an SAE on hidden state vectors $\boldsymbol{x} \in \mathbb{R}^{d_{\text{model}}}$ obtained from a pretrained model. The SAE consists of a linear encoder E, a sparsifying non-linear operator σ, and a linear decoder D:

$$\begin{aligned} \text{SAE}(\boldsymbol{x}) &= D(\sigma(E(\boldsymbol{x}))) \,, \\ \text{with} \quad E(\boldsymbol{x}) = \boldsymbol{W}_{\text{enc}}\, \boldsymbol{x} + \boldsymbol{b}_{\text{enc}} &= \boldsymbol{f}, \quad D(\boldsymbol{h}) = \boldsymbol{W}_{\text{dec}}\, \boldsymbol{h} + \boldsymbol{b}_{\text{dec}} = \hat{\boldsymbol{x}}, \\ \text{and} \quad \sigma(\boldsymbol{f}) &= \boldsymbol{h} \,. \end{aligned} \tag{1}$$

The encoder projects $\boldsymbol{x}$ into an overcomplete latent space of dimension d_{sae}, where typically $d_{\text{sae}} \gg d_{\text{model}}$. This enables the hidden state to be expressed as a sparse linear combination of learned dictionary vectors (the columns of $\boldsymbol{W}_{\text{dec}}$).

To enforce sparsity, we define $\sigma(\cdot)$ as $\sigma(\boldsymbol{f}) = \text{TopK}(\text{ReLU}(\boldsymbol{f}))$, i.e., we first apply ReLU to ensure non-negative activations and then retain only the k largest components per sample, setting all remaining entries to zero. The resulting sparse vector $\boldsymbol{h}$ captures the active latent features used by the decoder to reconstruct the hidden state.

We train the SAE by minimizing $\mathcal{L} = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2 + \lambda \mathcal{L}_{\text{aux}}$, where the first term enforces reconstruction fidelity, and the auxiliary term $\mathcal{L}_{\text{aux}}$ provides additional regularization. Depending on the specific SAE architecture, this term may mitigate dead features, promote balanced feature utilization, or enforce additional sparsity constraints, following standard practice in sparse autoencoder training (Templeton et al., 2024; Gao et al., 2024; Härle et al., 2025).

3.2 HIDDEN STATE EXTRACTION FROM A TSFM

We now describe how hidden states $\boldsymbol{x}$ are obtained. Let $\mathcal{M}$ be a pretrained Transformer model operating on patchified time series inputs. Given a time series window $\boldsymbol{s} \in \mathcal{S} \subseteq \mathbb{R}^T$, we partition it into P non-overlapping patches of length L such that $T = P \cdot L$. After patch embedding and positional encoding, the sequence of patch tokens is processed by multiple Transformer blocks. At layer ℓ, the model produces a sequence of residual stream hidden states $X^{(\ell)}(\boldsymbol{s}) = (\boldsymbol{x}_1^{(\ell)}(\boldsymbol{s}), \ldots, \boldsymbol{x}_P^{(\ell)}(\boldsymbol{s}))$ with $\boldsymbol{x}_p^{(\ell)}(\boldsymbol{s}) \in \mathbb{R}^{d_{\text{model}}}$. Each vector $\boldsymbol{x}_p^{(\ell)}(\boldsymbol{s})$ represents the model's internal encoding of a temporal patch at position p. For SAE training, we treat individual hidden states as independent samples and construct a dataset $\mathcal{D}^{(\ell)} = \{\boldsymbol{x}_p^{(\ell)}(s)\}_{\boldsymbol{s} \in \mathcal{S}, p \in \{1,\ldots,P\}}$ aggregated across time series, windows, and patch positions.

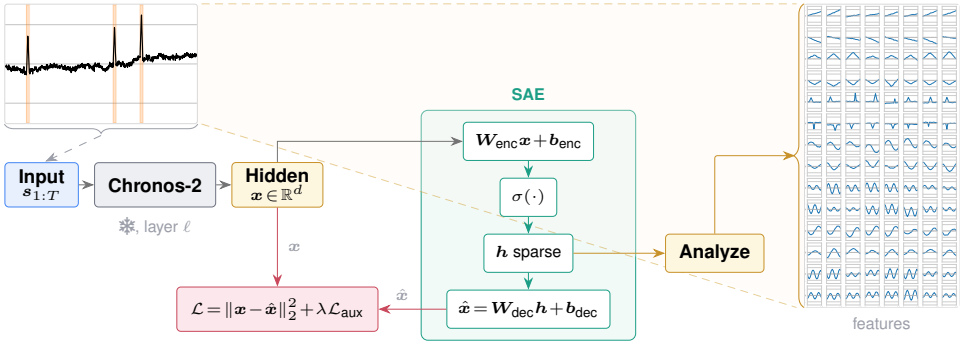


Figure 1: **SAE-based lens into CHRONOS-2 hidden states.** We extract the residual-stream activation $\boldsymbol{x}$ after block ℓ of the frozen model, train an SAE with reconstruction and sparsity loss, and analyze the learned sparse features f.

3.3 INFERENCE AND FEATURE ANALYSIS

Transformer hidden states are high-dimensional and encode information in superposition, making individual coordinates difficult to interpret (Elhage et al., 2022; Park et al., 2024). The trained SAE provides a sparse representation of each hidden state in terms of latent features. Since the SAE is trained solely to reconstruct hidden states under a sparsity constraint, without predefined temporal concepts, the discovered features reflect intrinsic structure in the model's internal representations. We analyze a feature by evaluating its activation across the dataset and inspecting time-series patches that strongly activate it. Mapping these activations back to their original temporal segments enables qualitative assessment of the temporal patterns captured by each feature.

4 EXPERIMENTS

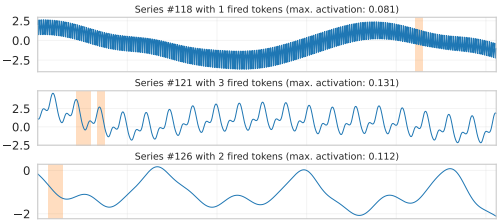
Setup. All experiments use the pretrained CHRONOS-2 model (from Hugging Face), which operates on time series patchified into length-16 tokens. We analyze residual-stream hidden states from layers 2 and 9 (out of 12), with the base model kept frozen throughout. This work focuses on univariate time series. Training data is drawn from KernelSynth-1M (Ansari et al., 2024), consisting of 1M univariate time series of length 1024 with no missing values. **Synthetic Concepts.** To assess whether at least some suspected key temporal structures are encoded in the hidden representations, we construct a controlled probing dataset with 14 synthetic temporal concepts. See Appendix B for details. For each concept, we generate 512 time series. In each series, 1 to 6 non-overlapping patches that match the tokenization boundaries are changed to the corresponding concept pattern. The remaining patches follow the background process, a slow random-walk baseline with additive white noise. **Evaluation.** Across all classifiers, we use a shared one-vs-rest, token-level evaluation per concept c. The positive pool contains all tokens that represent c, while the negative pool combines $B = 4096$ background tokens and $O = 4096$ tokens from other concepts, yielding 8192 negatives per concept. Performance on this imbalanced corpus is reported using precision, recall, and F1 score, summarized via the geometric mean across concepts. **Sparse Autoencoders.** The SAEs are trained on hidden states using BatchTopK with expansion factors 6× and 8×. Full training details are provided in Appendix A.

4.1 VALIDATING THE HIDDEN STATE OF CHRONOS-2 VIA LINEAR PROBES

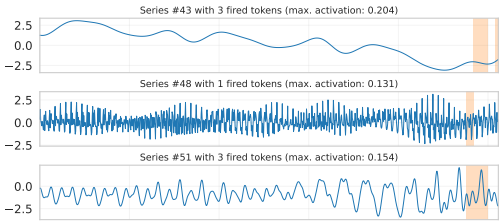
Before applying SAEs, we first verify that intermediate hidden states of CHRONOS-2 encode meaningful temporal structure. Following prior TSFM interpretability work, we train linear probes to predict synthetic temporal concepts directly from frozen hidden states. Specifically, we fit logistic regression classifiers (with class re-balancing) on hidden states extracted from layers 2 and 9. The probe setup follows a standard linear evaluation protocol and does not update the

Table 1: **The 14 suspected concepts can reliably be found in the CHRONOS-2 hidden state.** Geometric mean over all concepts.

Position	Precision	Recall	F1
Layer 2	0.993	1.000	0.996
Layer 9	0.992	1.000	0.996



(a) Neuron 827: Activation following onsets of downward transitions.



(b) Neuron 5478: Sparse activations near terminal low-frequency oscillations.

Figure 2: **The SAE neurons from layer 9 (with 8× expansion) respond to distinct temporal concepts despite being learned without any supervision.** Shading indicates activating patches.

base model. Table 1 summarizes linear probe performance across all evaluated temporal concepts. For completeness, per-concept precision, recall, and F1 scores are reported in Appendix C. These results demonstrate that the selected layers contain structured and linearly accessible temporal concepts. This confirms that the hidden states provide a meaningful substrate for further representational analysis.

4.2 Training SAEs as a Lens into TSFMs

We train SAEs on hidden states of layers 2 and 9 under different expansion factors (6× and 8×) over the foundation model dimension. To quantify how well the learned latent features capture the synthetic temporal concepts, we train shallow decision trees on the SAE feature activations and report the resulting F1 scores. Table 2 summarizes performance across the 14 synthetic concepts (cf. Appendix B) for all evaluated configurations. Detailed per-concept results are visualized in Appendix D. Across configurations, SAE feature representations achieve average F1 scores of up to 0.77, depending on depth and expansion factor. These results indicate that the latent representations capture relevant temporal structure, despite being learned without supervision and optimized solely to reconstruct under sparsity constraints. The performance is consistent with observations in large language models, where SAE features often yield partially disentangled instead of perfectly separable concepts (Härle et al., 2025). In the remainder of this section, we focus on the best-performing SAE (Layer 9, 8× expansion, depth 2) to identify the learned features.

Table 2: **Unsupervised SAEs extract meaningful discrete concepts.** Gemeotric average F1 score, best highlighted in **bold**.

	Depth 1		Depth 2	
Layer	6×	8×	6×	8×
2	0.691	0.715	0.756	0.764
9	0.687	0.661	0.749	**0.768**

4.3 Staring into the abyss

To qualitatively assess the structure captured by individual SAE features, we inspect neurons that fire sparsely but consistently across the dataset (roughly ∼1% of tokens). Figure 2 shows two representative examples. Neuron 827 activates immediately following downward transitions, rather than during the decline itself, suggesting sensitivity to post-transition or stabilization structure. In contrast, Neuron 5478 fires near the terminal occurrence of low-frequency oscillatory patterns, typically close to the end of a time series. Importantly, neither neuron responds to the corresponding primitive pattern in isolation, but instead appears selective for a specific temporal role. These behaviors are not directly aligned with any single synthetic probe concept, indicating that SAE features capture higher-level temporal structure beyond the injected primitives. More extensive examples are provided in Appendix E.

5 Conclusion

We investigated SAEs as an unsupervised lens into the hidden representations of time series foundation models. Applied to frozen Chronos-2 hidden states, SAE features capture substantial temporal structure, achieving strong performance on synthetic probing tasks and exhibiting qualitatively

coherent behaviors such as phase boundaries, terminal patterns, and precursor structure. These findings suggest that TSFMs encode temporal information in an entangled manner and that SAE-based analysis provides a viable path toward mechanistic interpretability in the time series domain.

A current limitation of our study is that qualitative feature discovery is based on manual inspection of individual neurons. Developing automated interpretation pipelines would enable systematic analysis of hundreds or thousands of SAE features and substantially improve scalability (Paulo et al., 2025). We view such automation as a key direction for future work. Similarly, this analysis should eventually compare different layers, multivariate data, and other foundation models, possibly of different base architectures, such as state space (Graf et al., 2025) or recurrent models (Kraus et al., 2025; Auer et al., 2025). Ultimately, the mechanistic interpretability of time series foundation models should extend to multimodal models (Divo et al., 2025; Xie et al., 2025).

ACKNOWLEDGMENTS

This work benefited from the support of the German Federal Ministry for Economic Affairs and Energy (BMWE) through "EU-SAI: Souveräne KI für Europa" (grant number 13IPC040G), and the BMFTR project "XEI" (FKZ 16IS24079B).

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Georgii Aparin, Tasnima Sadekova, Alexey Rukhovich, Assel Yermekova, Laida Kushnareva, Vadim Popov, Kristian Kuznetsov, and Irina Piontkovskaya. AudioSAE: Towards understanding of audio-processing models with sparse autoencoders. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2026.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. TiRex: Zero-Shot Forecasting Across Long and Short Horizons with Enhanced In-Context Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Anthony Bao, Venkata Hasith Vattikuti, Jeffrey Lai, and William Gilpin. Universal redundancies in time series foundation models. *arXiv preprint arXiv:2602.01605*, 2026.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning (ICML)*, 2024.

Felix Divo, Maurice Kraus, Anh Q. Nguyen, Hao Xue, Imran Razzak, Flora D. Salim, Kristian Kersting, and Devendra Singh Dhami. Quants: Question answering on time series. *arXiv preprint arXiv:2511.05124*, 2025.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Lars Graf, Thomas Ortner, Stanisław Woźniak, and Angeliki Pantazi. Flowstate: Sampling-rate invariant time series foundation model with dynamic forecasting horizons. In *Workshop on Recent Advances in Time Series Foundation Models*, 2025.

Ruben Härle, Felix Friedrich, Manuel Brack, Stephan Wäldchen, Björn Deiseroth, Patrick Schramowski, and Kristian Kersting. Measuring and guiding monosemanticity. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Matīss Kalnāre, Sofoklis Kitharidis, Thomas Bäck, and Niki van Stein. Mechanistic interpretability for transformer-based time series classification. In *International Joint Conference on Computational Intelligence (IJCCI)*, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Maurice Kraus, Felix Divo, Devendra Singh Dhami, and Kristian Kersting. xLSTM-mixer: Multivariate time series forecasting by mixing via scalar memories. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting. *arXiv preprint arXiv:2511.11698*, 2025.

Khalid Oublal, Quentin Bouniot, Qi Gan, Stephan Clémençon, and Zeynep Akata. TimeSAE: Sparse decoding for faithful explanations of black-box time series models. *arXiv preprint arXiv:2601.09776*, 2026.

Mateusz Pach, Shyamgopal Karthik, Quentin Bouniot, Serge Belongie, and Zeynep Akata. Sparse autoencoders learn monosemantic features in vision-language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Atharva Pandey, Abhilash Neog, and Gautam Jajoo. On the internal semantics of time-series foundation models. In *NeurIPS Workshop on Recent Advances on Time Series Foundation Models (BERT^{2}S)*, 2025.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning (ICML)*, 2024.

Youngjin Park, Anh Tong, Sehyun Lee, Jihyeon Seong, Qin Xie, and Jaesik Choi. Towards transparent time series analysis: Exploring methods and enhancing interpretability. *ACM Computing Surveys*, 2026.

Gonçalo Santos Paulo, Alex Troy Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In *International Conference on Machine Learning (ICML)*, 2025.

Owen Queen, Thomas Hartvigsen, Teddy Koker, Huan He, Theodoros Tsiligkaridis, and Marinka Zitnik. Encoding time-series explanations through self-supervised model behavior consistency. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Lee Sharkey, Dan Braun, and Beren Millidge. Taking features out of superposition with sparse autoencoders. *arXiv preprint arXiv:2211.11279*, 2022.

Elana Simon and James Zou. InterPLM: Discovering interpretable features in protein language models via sparse autoencoders. *Nature Methods*, pp. 2107–2117, 2025.

David Steinmann, Felix Divo, Maurice Kraus, Antonia Wüst, Lukas Struppek, Felix Friedrich, and Kristian Kersting. Navigating shortcuts, spurious correlations, and confounders: From origins via detection to mitigation. *arXiv preprint arXiv:2412.05152*, 2024.

Samuel Stevens, Wei-Lun Chao, Tanya Berger-Wolf, and Yu Su. Interpretable and testable vision features via sparse autoencoders. *arXiv preprint arXiv:2502.06755*, 2025.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L. Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024.

Michał Wiliński, Mononito Goswami, Nina Żukowska, Willa Potosnak, and Artur Dubrawski. Exploring representations and interventions in time series foundation models. In *International Conference on Machine Learning (ICML)*, 2025.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *Proceedings of the VLDB Endowment*, pp. 2385–2398, 2025.

Zeyu Yun, Yubei Chen, Bruno A. Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: Contextualized embedding as a linear superposition of transformer factors. In *Workshop on Deep Learning Inside Out (DeeLIO), Association for Computational Linguistics (ACL)*, 2021.

Yufeng Zou, Zijian Wang, Diego Klabjan, and Han Liu. Investigating hallucinations of time series foundation models through signal subspace analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

A Experimental Details

This appendix provides implementation details for the linear probes and sparse autoencoder training used in Section 4. We base all results on hidden states extracted from layers 2 and 9 of a frozen Chronos-2 model (Ansari et al., 2025). All experiments involving GPUs were run on a single *Nvidia Tesla V100* with 32GB of VRAM. Performance is reported using precision, recall, and F1 score.

A.1 Configuration of Linear Probes

For the linear probing experiments, we used the logistic regression implementation of *scikit-learn*[1] with the `liblinear` solver, regularization strength $C = 1.0$, balanced class weights, and learning the intercept term. Optimization is run with a maximum of 1000 iterations and convergence tolerance 1×10^{-4}. Each temporal concept is treated as a separate binary classification task.

A.2 Configuration of Sparse Autoencoders

We ran a simple grid search over the hyperparameters shown in Table 3. Beyond the ablated settings above, we kept the training setup fixed: A BatchTopK SAE (Bussmann et al., 2024) with expansion factor 8× ($d_{\text{model}} = 768$, $d_{\text{sae}} = 6144$) and $k = 100$, trained for 66×10^6 tokens with a batch size of 4096 tokens using Adam ($\beta_1 = 0.0,\ \beta_2 = 0.999$) (Kingma & Ba, 2015). We used a constant learning-rate scheduler (with configured warmup/decay horizons of 805/3222 steps and $\eta_{\text{end}} = \eta/10$), top-k threshold learning rate 0.01, decoder init norm 0.1, and b_{dec} initialization statistics computed from 50×10^3 tokens when using mean/geometric-median init. Dead-feature handling used windows of 1000 (dead-feature) and 2000 (feature-sampling) steps, and training was run in full 32-bit floating point arithmetic. The 6× configuration differed only in using $k = 32$ for additional sparsity. We used the *SAE Lens*[2] library implemented in *PyTorch*[3]. We do not shuffle the training tokens to implicitly group similar representations.

Table 3: Hyperparameter grid used in SAE training. The chosen combination is underlined.

Hyperparameter	Explored Values
Decoder bias init	{zero, mean, geometric median}
Activation normalization	{none, expected average only in}
Decoder normalization	{enabled, disabled}
Learning rate η	$\{2 \times 10^{-5}, 5 \times 10^{-5}, 8 \times 10^{-5}\}$
Auxiliary loss weight λ	$\{0.5, 1.0\}$

[1] `https://scikit-learn.org`
[2] `https://github.com/decoderesearch/SAELens`
[3] `https://pytorch.org`

B SYNTHETIC DATASET SAMPLES

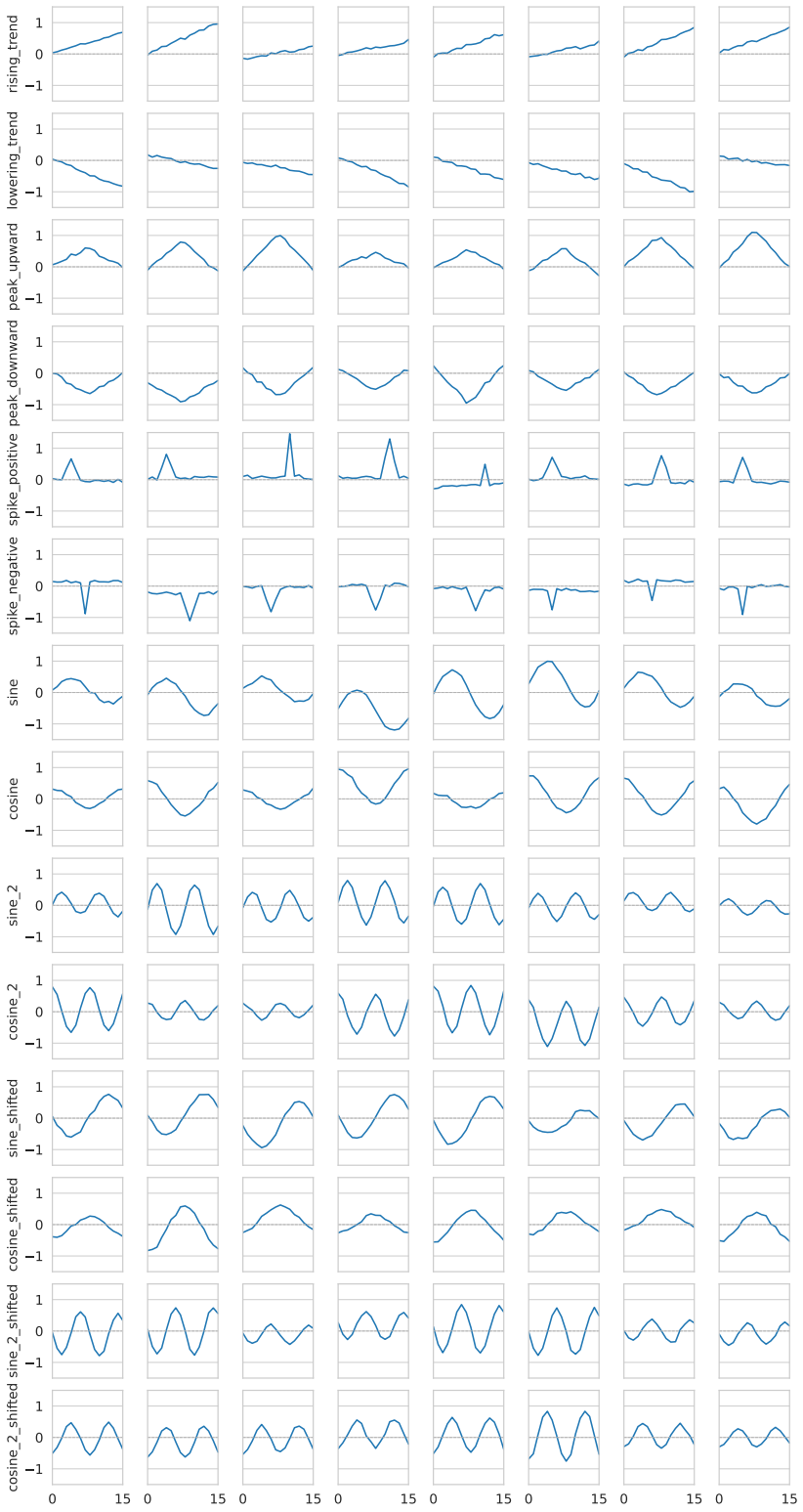


Figure 3: Probing patches used to assess which concepts are encoded in hidden and latent layers.

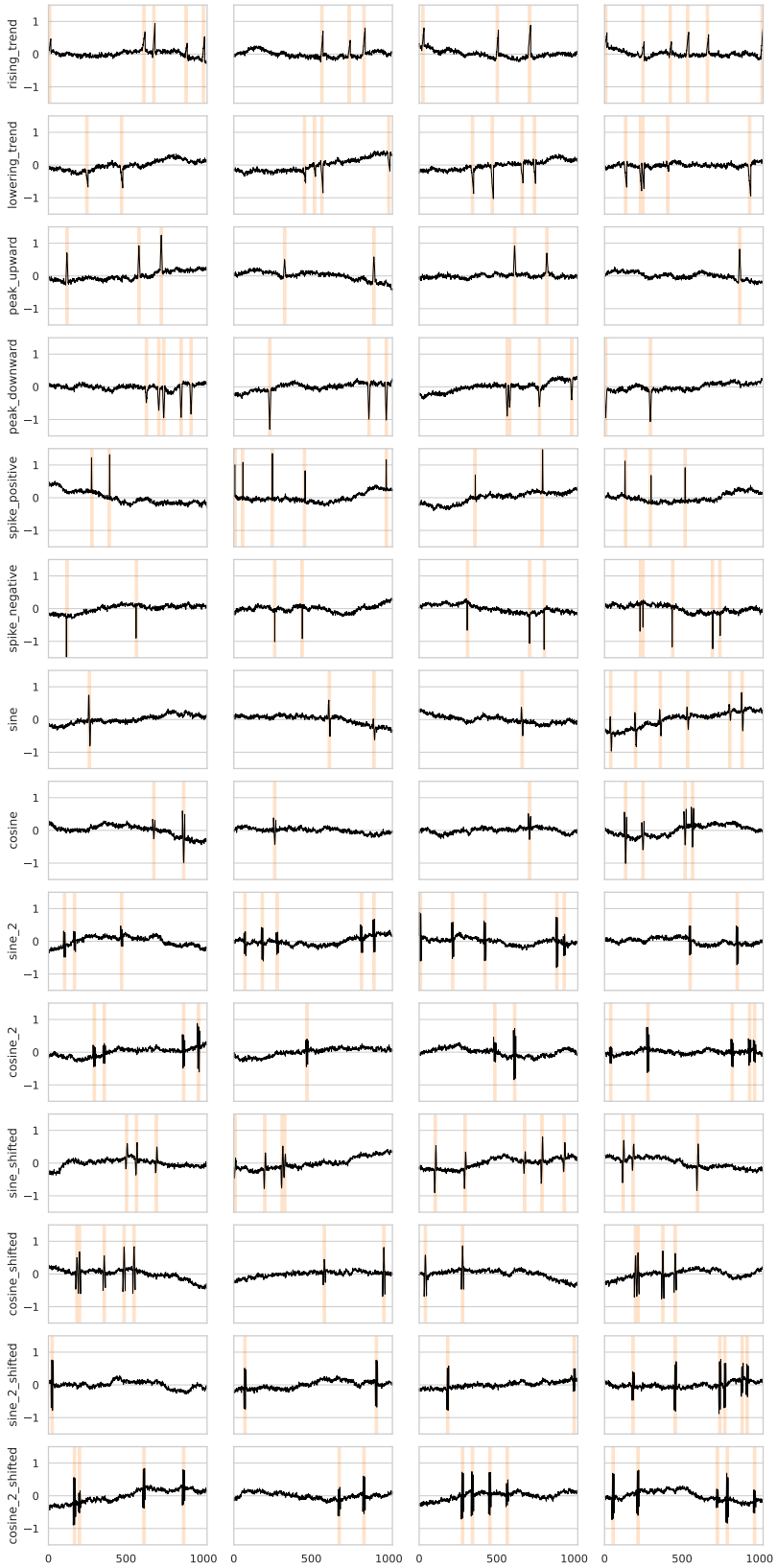


Figure 4: The patches from the synthetic dataset (shaded regions) embedded in the full time series.

C Linear Probe Detailed Results

Table 4: Linear probe metrics by concept for layers 2 and 9.

	Layer 2			Layer 9		
Concept	Precision	Recall	F1	Precision	Recall	F1
rising_trend	0.996	1.000	0.998	0.995	1.000	0.998
lowering_trend	0.994	1.000	0.997	0.994	1.000	0.997
peak_upward	0.991	0.999	0.995	0.990	0.999	0.995
peak_downward	0.993	1.000	0.997	0.992	1.000	0.996
spike_positive	0.984	1.000	0.992	0.984	1.000	0.992
spike_negative	0.986	1.000	0.993	0.986	1.000	0.993
sine	0.996	1.000	0.998	0.995	1.000	0.997
cosine	0.996	1.000	0.998	0.993	1.000	0.997
sine_2	0.996	1.000	0.998	0.996	0.999	0.998
cosine_2	0.989	1.000	0.995	0.988	1.000	0.994
sine_shifted	0.993	1.000	0.996	0.993	1.000	0.997
cosine_shifted	0.993	1.000	0.996	0.995	1.000	0.997
sine_2_shifted	0.993	1.000	0.996	0.992	1.000	0.996
cosine_2_shifted	0.997	1.000	0.999	0.998	1.000	0.999
Geometric mean	0.993	1.000	0.996	0.992	1.000	0.996

D SAE Detailed Results

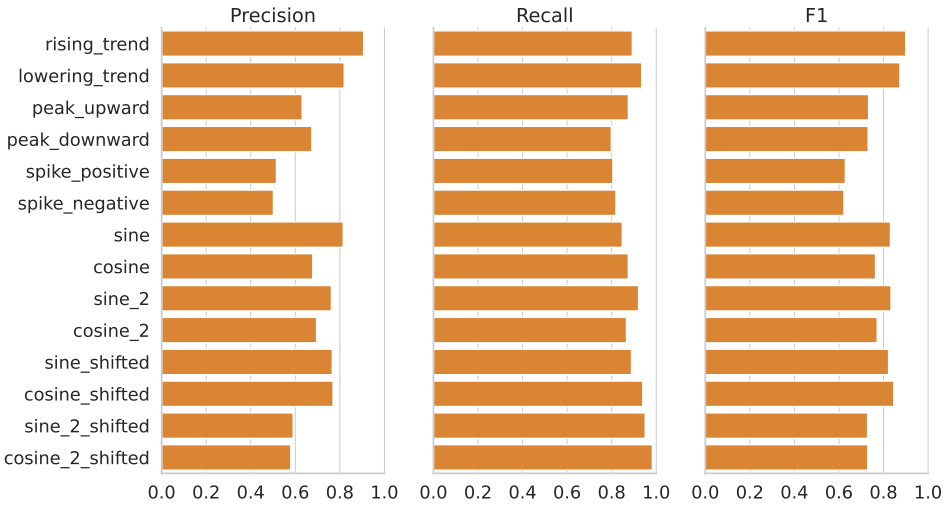


Figure 5: Precision, recall, and F1 scores for the best-performing SAE configuration (8× expansion, layer 9, depth 2), corresponding to the top result in Table 2.

E Additional Qualitative SAE Features

We include further examples of sparsely firing SAE neurons that exhibit coherent temporal behavior, including features that respond to phase transitions and exhibit anticipatory patterns.

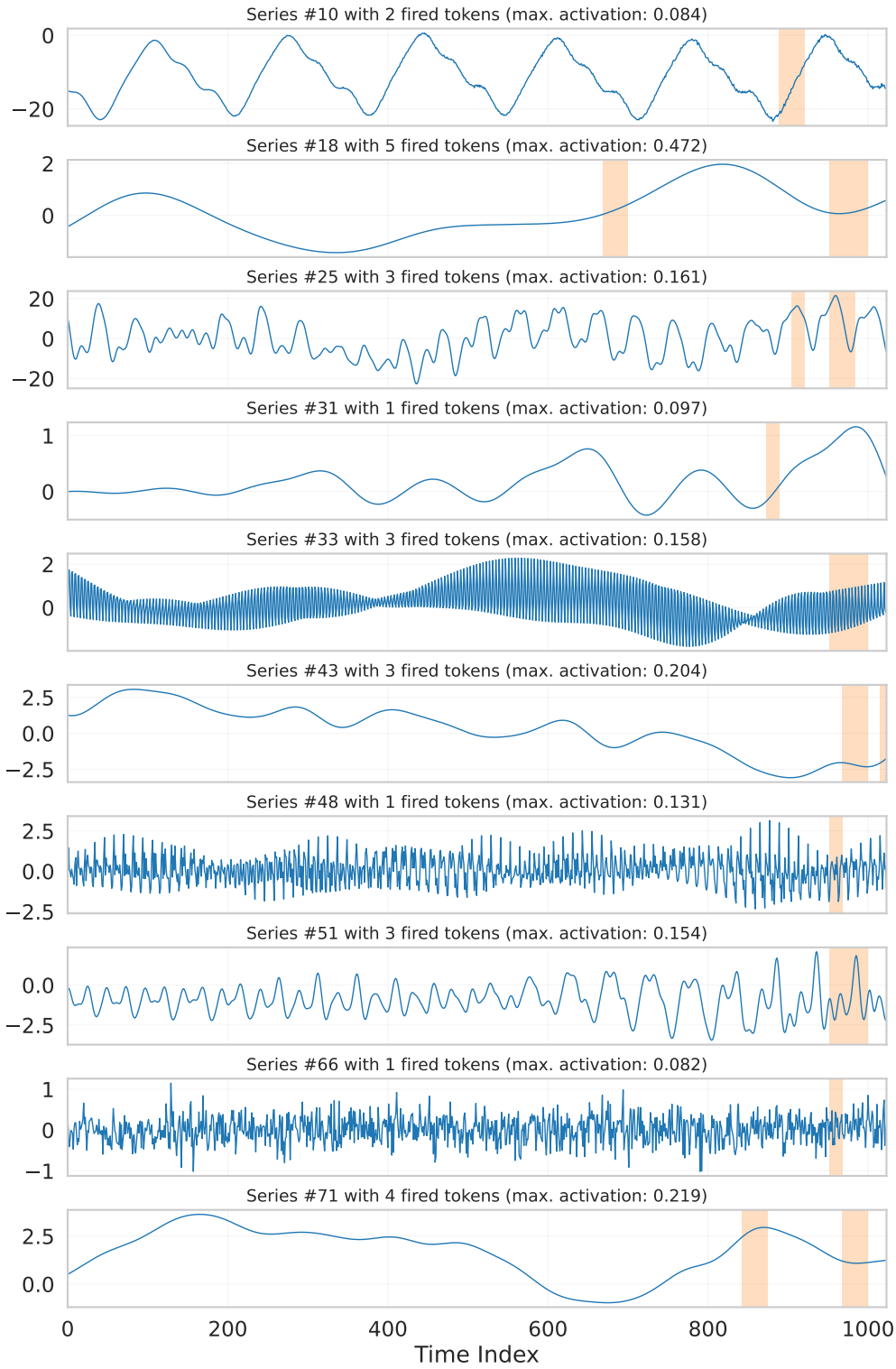


Figure 6: Visualization of SAE neuron 5478 across multiple sampled time series. Shaded regions indicate activating patches. Activations occur sparsely near the terminal occurrence of low-frequency oscillatory structure, often close to the end of the sequence. The neuron does not respond to low-frequency content in general, but appears selective to the end of such patterns.

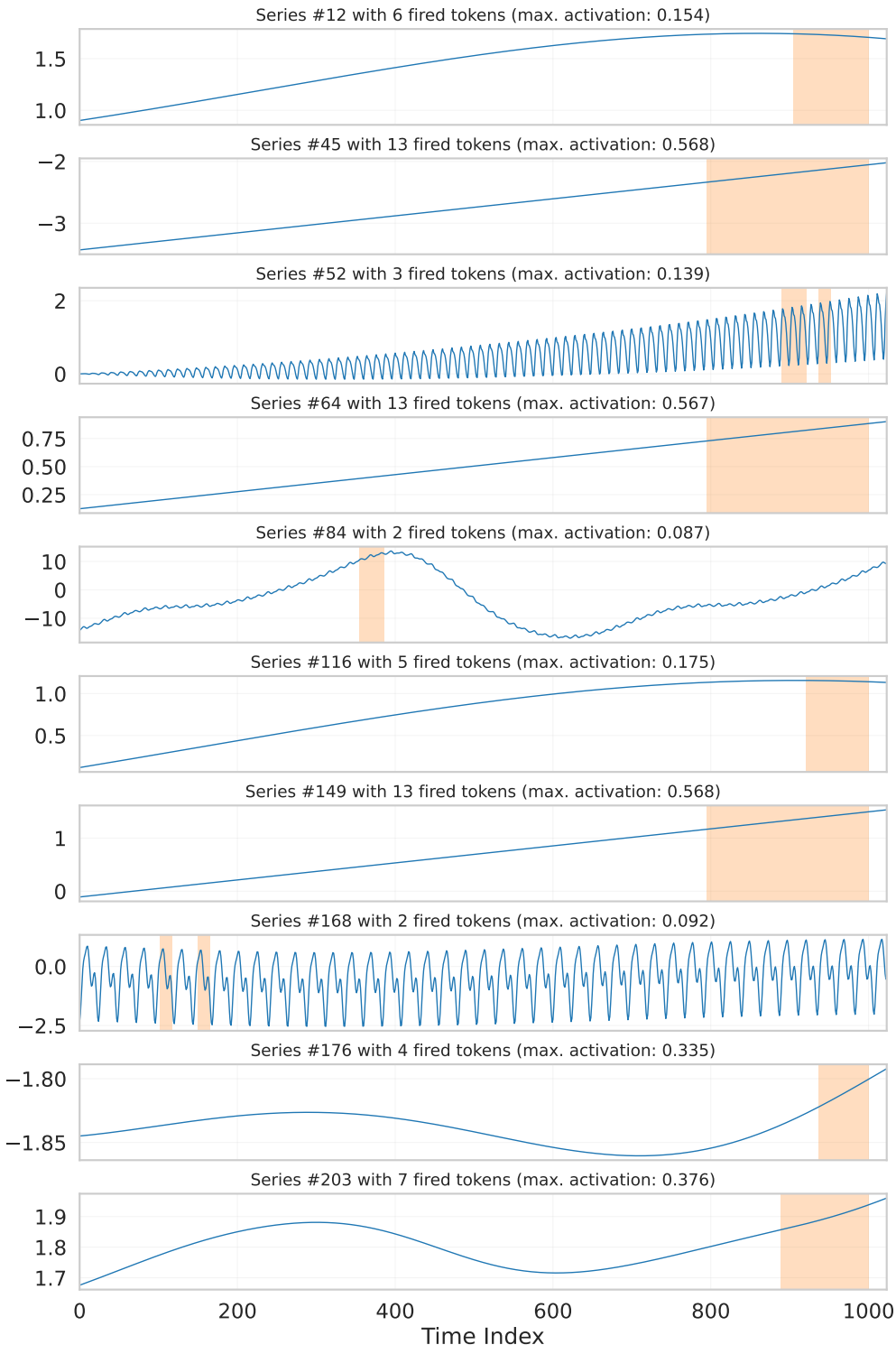


Figure 7: Visualization of SAE neuron 4981 across multiple time series. Shaded regions indicate activations occurring near the termination of extended flat or gently rising segments. The neuron does not respond to generic upward trends, but instead fires at the end of prolonged increases, suggesting sensitivity to phase boundaries that depend on longer temporal context.

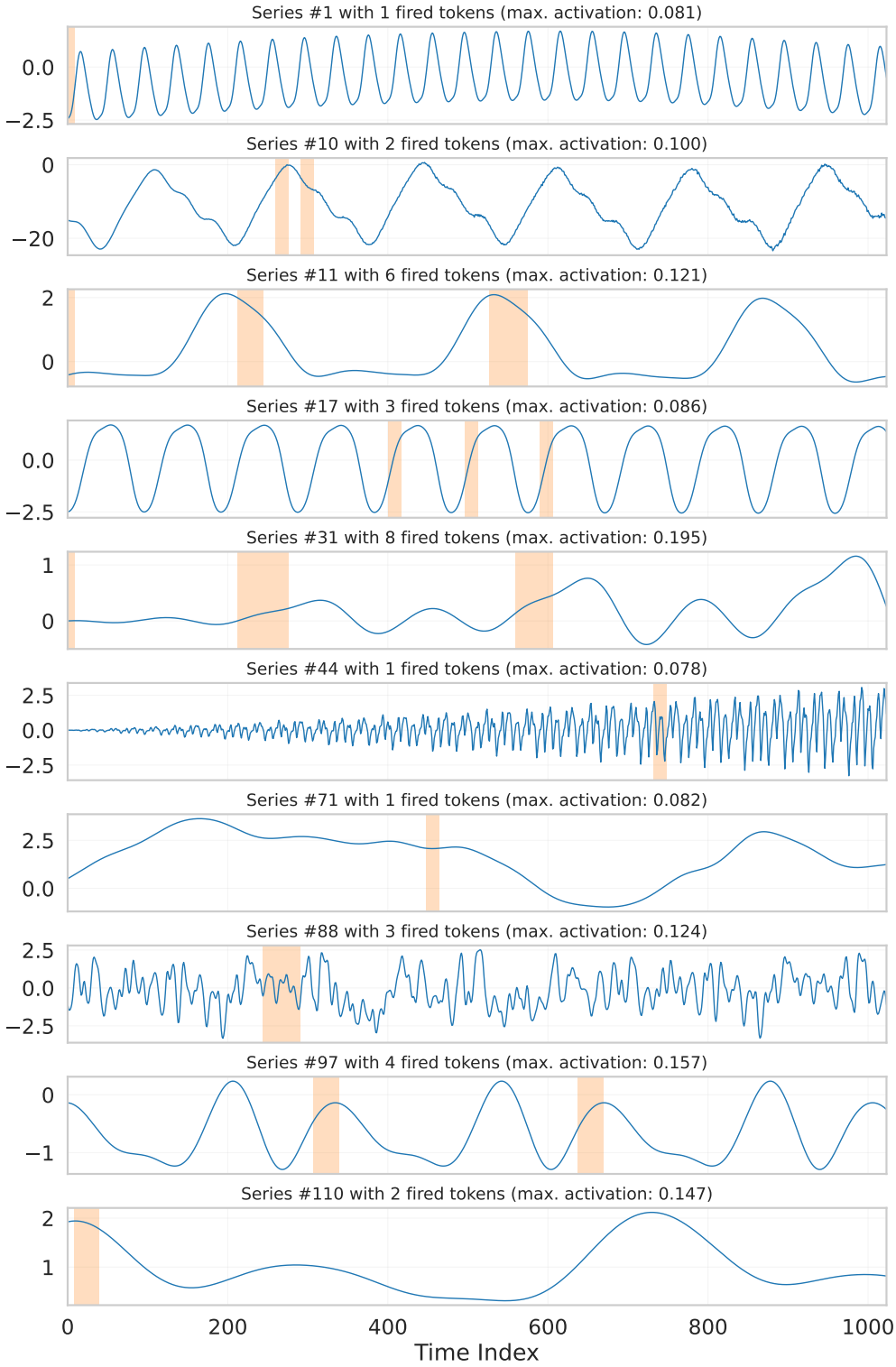


Figure 8: Visualization of SAE neuron 2660 across multiple time series. Activations occur sparsely at a consistent offset shortly before the onset of declining behavior, typically one to two patches ahead of a downward transition. This pattern suggests responsiveness to precursor structure preceding trend changes rather than to declines themselves.

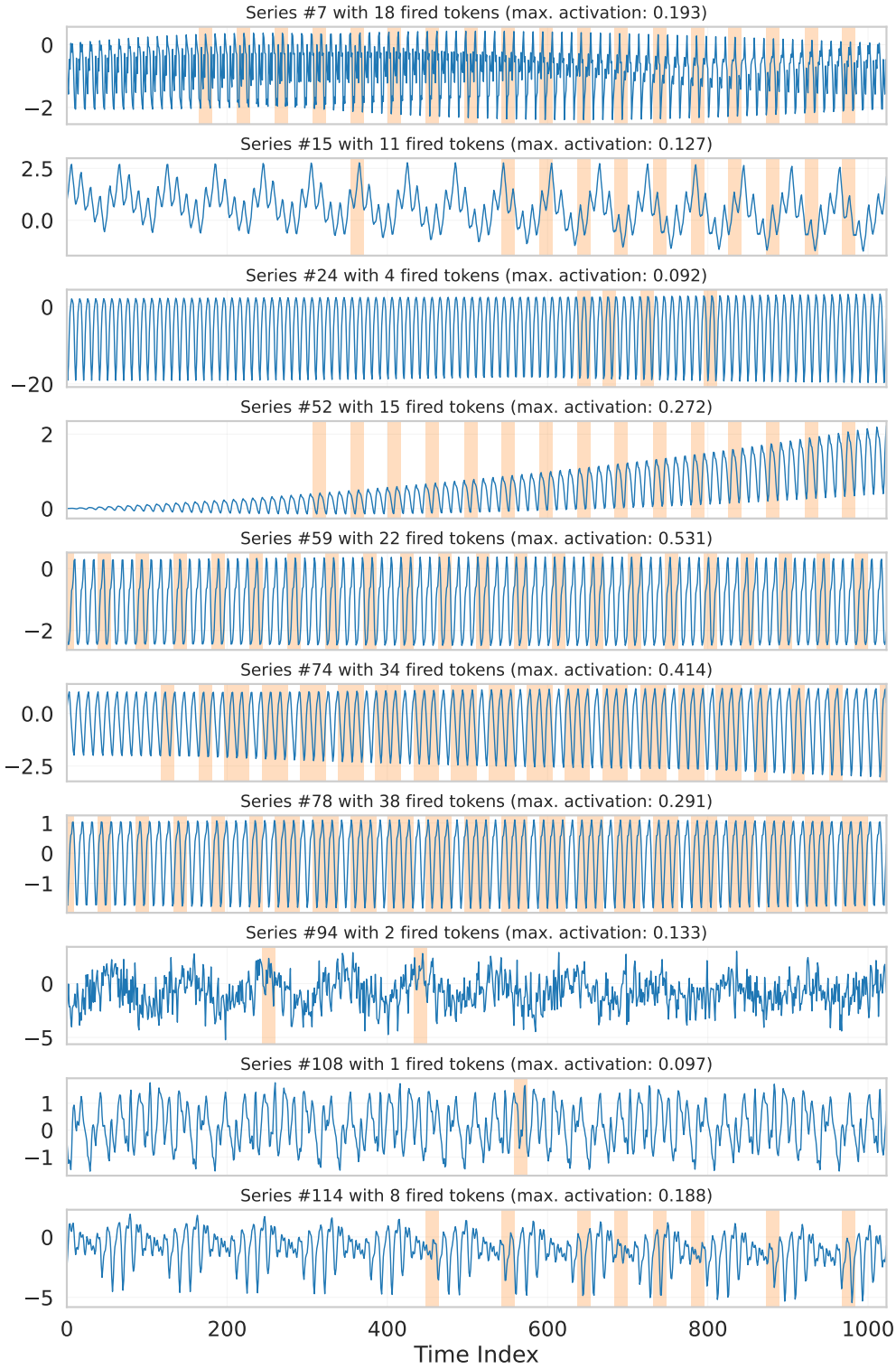


Figure 9: Visualization of SAE neuron 5801 across multiple time series. The neuron activates at local maxima in signals exhibiting intermediate-frequency oscillations, while remaining inactive for very low- or high-frequency patterns. This indicates selectivity for a specific frequency band combined with peak structure.

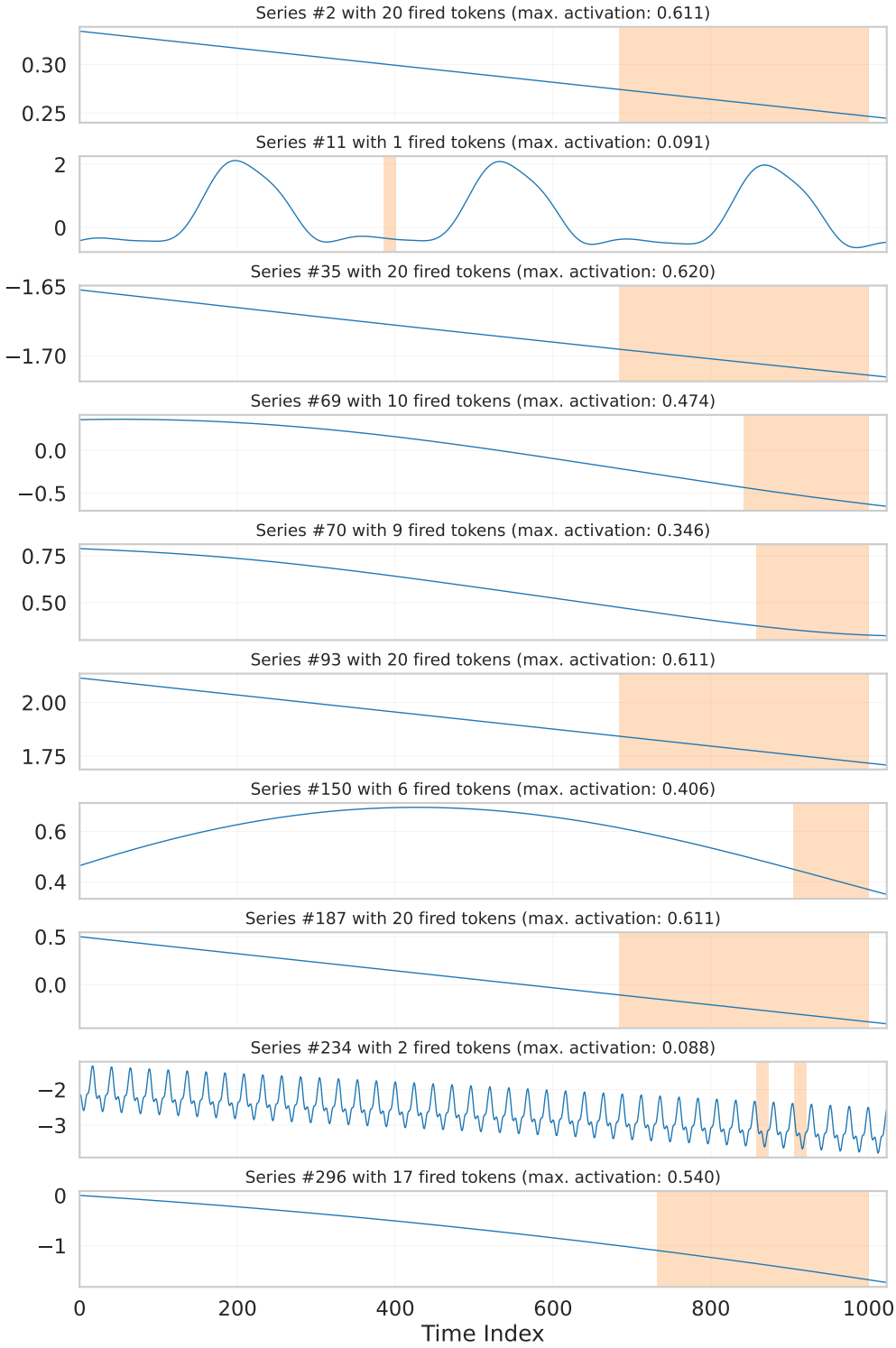


Figure 10: Visualization of SAE neuron 139 across multiple time series. Shaded regions highlight activations near the end of long declining segments, often close to the end of the sequence. The neuron appears sensitive to the termination of extended decreases rather than to negative slope alone.

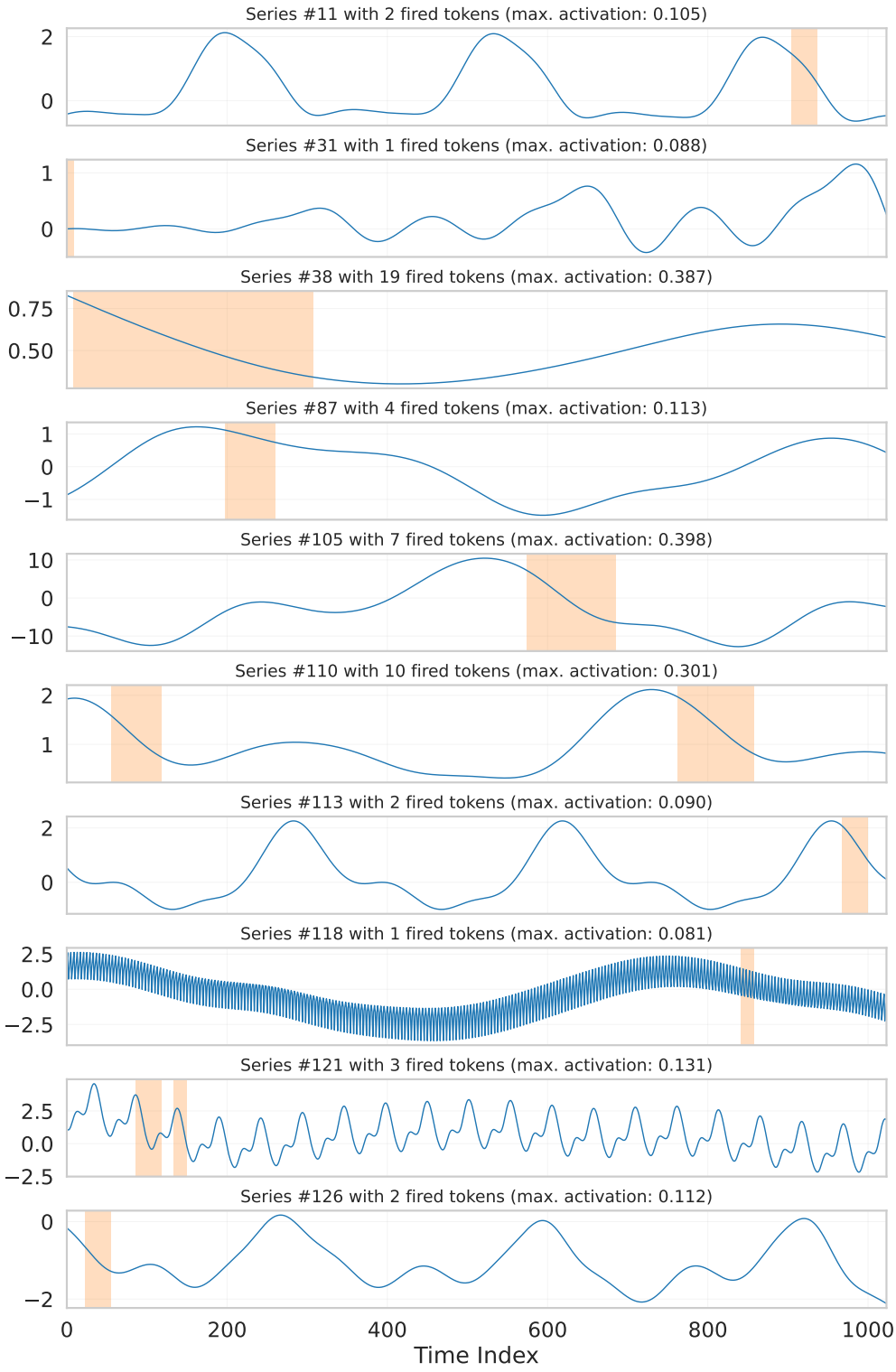


Figure 11: Full visualization of SAE neuron 827 across multiple sampled time series. Shaded regions indicate activating patches. Activations occur immediately after the onset of declines, suggesting sensitivity to post-transition or stabilization structure rather than to negative slope alone.