STAMP: SHARED-DICTIONARY SHAPELET TOKENIZATION FOR MULTI-RESOLUTION TIME-SERIES COMPRESSION

Bibhus Luitel[1] **Payal Mohapatra**[2] **Subrata Biswas**[1] **Bashima Islam**[1]

[1]Electrical and Computer Engineering, Worcester Polytechnic Institute
Worcester, MA, USA
{bluitel, sbiswas, bislam}@wpi.edu

[2]Electrical and Computer Engineering, Northwestern University
Evanston, IL, USA
payal.mohapatra@northwestern.edu

ABSTRACT

Wearables, mobile devices, and Internet of Things platforms stream multimodal time series continuously and at scale, making storage and transmission a primary bottleneck under tight bandwidth, memory, and energy budgets. Meanwhile, downstream pipelines, including token based large-model workflows, benefit from discrete, rate controlled representations that are easy to serve and process. A core challenge is that time-series semantics are inherently multi-resolution, spanning sub-second transients and minute to hour scale structure, and are further degraded by small temporal misalignment from jitter and imperfect segmentation. Existing discretization and tokenization methods often rely on fixed stride segmentation or generic primitives, which can either miss short events or become inefficient over long horizons. To address this, we present STAMP, a multi-resolution neural compression framework built around a shared dictionary of learned temporal primitives. STAMP encodes signals using a coarse to fine residual cascade, discretizing residuals via vector quantization to produce a compact discrete code sequence with controllable rate. A shift robust additive decoder reconstructs the signal by superposing translated dictionary atoms across scales, improving robustness to boundary shifts while keeping decoding lightweight. We evaluate STAMP on a synthetic multimodal benchmark (SeqComb), measuring rate, distortion, and perceptual fidelity trade-offs alongside end-to-end inference latency and peak synthesis memory, and comparing against fundamental baselines. In this controlled setting, STAMP achieves favorable rate distortion behavior while preserving local morphology at low decoding cost, suggesting a practical path toward edge deployable discrete representations for large scale time-series streams.

Track: Research

1 INTRODUCTION

The rapid emergence of large time series foundation models (Ansari et al., 2024; Woo et al., 2024) increases pressure to represent continuous sensor streams in forms that are efficient to store, transmit, and serve. Wearables, mobile devices, and Internet of Things platforms generate multimodal time series continuously and at scale (Liu et al., 2024; Narayanswamy et al., 2024), turning storage and transmission into a primary bottleneck. A practical representation must therefore offer predictable bitrate and support downstream processing that benefits from discrete sequences. This problem is inherently multi-resolution (Liu et al., 2024; Wu et al., 2023). Semantics can appear in sub-second transients, such as impacts, peaks, and brief acoustic events, and also in minute to hour scale structure, such as sleep stages, activity rhythms, and environmental cycles. Fixed stride discretization exposes a fundamental trade-off: coarse strides miss short events, while fine strides become inefficient over long horizons. The challenge is compounded by small temporal misalignment from

jitter, windowing, and imperfect segmentation, which can shift code boundaries and degrade local morphological fidelity.

Classic discretization methods (Lin et al., 2003; Elsworth & Güttel, 2020) provide efficient, training free symbol sequences and remain useful for indexing and exploratory mining. However, these representations rely on generic geometric primitives and rule based segmentation (Wen et al., 2021). For complex organic signals, fidelity typically improves only by increasing the number of segments and symbols, which lengthens sequences and raises bitrate. As a result, it is difficult to obtain compact discrete streams that remain faithful across both short events and long horizon structure.

Recent neural approaches learn data driven representations for time series (Wu et al., 2023; Nie et al., 2023) and adopt discrete latent spaces via vector quantization in generative settings such as TimeVQVAE (Lee et al., 2023). In parallel, implicit neural representations model signals as continuous functions and enable resolution flexible reconstruction (Dupont et al., 2022; Sitzmann et al., 2020). While these directions improve fidelity, they do not directly satisfy key requirements in edge and large model settings, where representations must be rate controlled, easy to decode, and robust to boundary shifts. We still lack methods that (i) produce multi-resolution discrete codes with predictable bitrate, (ii) support low latency decoding with modest memory, and (iii) preserve local morphology when code boundaries shift slightly due to sensor jitter, windowing, or imperfect alignment.

To address this gap, we propose `STAMP`, a multi-resolution neural compression framework built around a shared dictionary of learned temporal primitives. `STAMP` represents a signal through a coarse to fine residual cascade. At each scale, an encoder predicts a residual that is discretized via vector quantization, producing a compact discrete code sequence whose rate is controlled by the active codebook and scale schedule. Reconstruction uses a shift robust additive decoder that synthesizes the signal by superposing translated dictionary atoms across scales. This design separates where structure occurs from what structure looks like, improving robustness to small temporal misalignment while keeping decoding lightweight. We validate `STAMP` on synthetic benchmarks, characterizing the rate distortion trade-off and decoding latency on edge constrained hardware.

Related Work. Beyond symbolic discretization and fixed-stride neural tokenization, prior work also represents time series using subsequence primitives and reconstructive dictionary models. Shapelet-based methods model time series with representative subsequences, evolving from search-based discovery and differentiable transforms toward softly weighted, learnable primitives reusable across time Ye & Keogh (2009); Grabocka et al. (2014); Liu et al. (2025). This line is typically developed for discriminative representations and does not directly produce reconstructive discrete token sequences with controllable rate. Convolutional sparse coding and multivariate extensions represent signals as superpositions of shifted spatio-temporal atoms, yielding interpretable and shift-robust reconstructions for multichannel time series Wohlberg (2014); Tour et al. (2018), but standard formulations do not specify how to discretize activations into multi-resolution tokens suitable for tokenizer-style interfaces. Appendix A details additional background and related works.

2 METHOD: STAMP

`STAMP` unifies compression and tokenization in an asymmetric design: a compute-heavy multi-scale encoder produces discrete tokens at a controllable rate, while a lightweight, shift-invariant decoder enables low-latency reconstruction on constrained devices. Figure 1 illustrates the pipeline.

Problem Formulation. Given multivariate time series $\mathbf{X} \in \mathbb{R}^{C\times T}$ (C channels over T time steps) from $\mathcal{D} = \{\mathbf{X}_i\}_{i=1}^{N}$, `STAMP` learns an encoder-decoder pair (Φ, Ψ) that maps $\mathbf{X}$ to a discrete token sequence and reconstructs $\hat{\mathbf{X}} = \Psi(\Phi(\mathbf{X}))$ by minimizing a reconstruction objective. The model learns (i) a dictionary of temporal primitives or shapelets, $\mathbf{D} = \{\mathbf{D}_k\}_{k=1}^{K}$, where each shapelet (atom) $\mathbf{D}_k \in \mathbb{R}^{C\times L}$ (or $1 \times L$ for independent encoding) has length L, and (ii) a shared VQ codebook $\mathcal{E} = \{\mathbf{e}_j\}_{j=1}^{V} \subset \mathbb{R}^K$ of size V (the discrete vocabulary) used across all resolutions.

Hierarchical Refinement. `STAMP` uses a coarse-to-fine residual cascade inspired by Laplacian pyramids (Burt & Adelson, 1983): each tier reconstructs the current residual at its resolution, and subsequent tiers encode only the remaining error. This enables allocating tokens to long-horizon structure at coarse scales while reserving higher-rate tokens for localized transients.

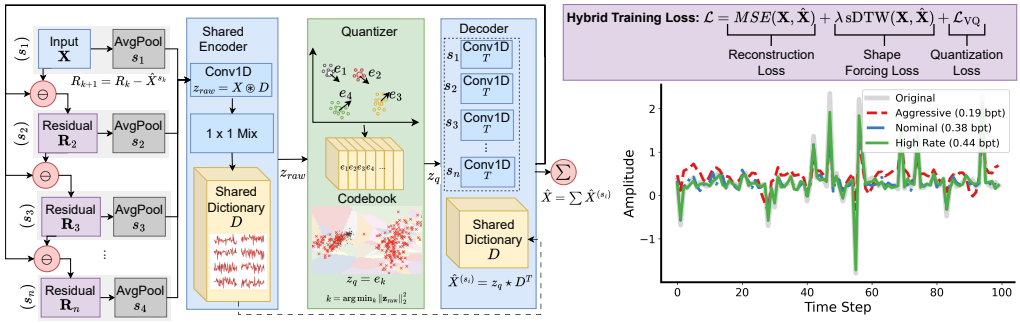


Figure 1: Overview of STAMP and hybrid training objective. Left: Multi-resolution residual compression pipeline with coarse-to-fine residuals and vector quantization into a shared codebook. Right: Hybrid loss (MSE, Soft-DTW, VQ commitment) with example reconstructions across bitrates, showing rate-dependent fidelity and sharp transient preservation.

Multi-Scale Residual Tokenization. STAMP encodes $\mathbf{X}$ via a residual cascade over scales $\mathcal{S} = \{s_1 > \cdots > s_M\}$. Initializing $\mathbf{R}_1 = \mathbf{X}$, for each scale m, we extract activations via the encoder f_θ and discretize them:

$$\mathbf{Z}_{\text{raw}}^{(m)} = f_\theta\big(\text{AvgPool}_{s_m}(\mathbf{R}_m); \mathbf{D}\big), \tag{1}$$
$$\mathbf{Z}_{\text{q}}^{(m)}[t] = \mathbf{e}_{j^*} \;\text{ s.t. }\; j^* = \arg\min_j \left\| \mathbf{Z}_{\text{raw}}^{(m)}[t] - \mathbf{e}_j \right\|_2^2.$$

Crucially, to isolate fine-grained transients, we update the residual via the additive decoder:

$$\mathbf{R}_{m+1} = \mathbf{R}_m - \hat{\mathbf{X}}^{(m)}. \tag{2}$$

This closed-loop feedback allows STAMP to correct aliasing introduced at coarse scales.

Dual-Path Encoding: Channelwise and Joint. STAMP supports two encoding modes for C channels. Independent encoding treats each channel as an orthogonal signal and learns univariate primitives $\mathbf{D} \in \mathbb{R}^{K \times 1 \times L}$ per channel to prioritize per-channel fidelity and avoid cross-channel interference. Joint encoding uses multivariate atoms $\mathbf{D}_k \in \mathbb{R}^{C \times L}$ to capture synchronous cross-channel structure and represent multi-sensor state changes with shared tokens. The choice of encoding mode dictates the spatial depth of the convolutional filters in both the analysis and synthesis stages.

Shift-Robust Additive Decoding. At each scale, the decoder synthesizes a tier reconstruction via a transposed convolution with the shared dictionary. We define the operation $\text{TConv}(\mathbf{Z}, \mathbf{D}, s)$ as the sum of spatially shifted atoms weighted by the latent activations:

$$\text{TConv}(\mathbf{Z}, \mathbf{D}, s)[t] = \sum_{k=1}^{K} \sum_{\tau} \mathbf{Z}_k[\tau] \cdot \mathbf{D}_k[t - \tau \cdot s]. \tag{3}$$

Unlike patch-based methods that enforce rigid grid boundaries, STAMP utilizes a simple additive update rule to reconstruct the full signal:

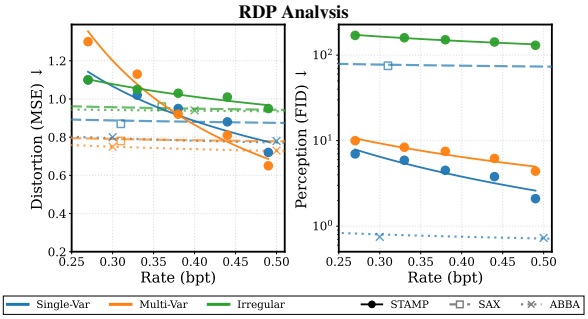
$$\hat{\mathbf{X}} = \sum_{m=1}^{M} \hat{\mathbf{X}}^{(m)} = \sum_{m=1}^{M} \text{TConv}\left(\mathbf{Z}_{\text{q}}^{(m)}; \mathbf{D}, \text{stride}_m\right). \tag{4}$$

This additive superposition allows fine-scale atoms to shift relative to coarse-scale atoms, recovering local phase alignment lost during downsampling.

Training Objective. STAMP is trained end-to-end with a hybrid loss:

$$\mathcal{L} = \underbrace{\|\mathbf{X} - \hat{\mathbf{X}}\|_2^2}_{\text{MSE}} + \lambda \underbrace{\text{sDTW}(\mathbf{X}, \hat{\mathbf{X}})}_{\text{Soft-DTW}} + \underbrace{\mathcal{L}_{\text{VQ}}}_{\text{Quantization}} \tag{5}$$

where Soft-DTW (Cuturi & Blondel, 2017) encourages morphology preservation under temporal misalignment.



Inference Efficiency (CPU)

Method	bpt↓	ms↓	MB↓
SAX (w= 500, a = 128)	0.713	10.40	0.055
ABBA ($\epsilon = 0.5$, k = 32)	0.710	78.00	2.500
STAMP-I ($V = 64$)	0.328	2.10	0.1522
STAMP-I ($V = 128$)	0.438	2.25	0.155
STAMP-I (1-scale)	0.250	0.45	0.120
STAMP-J ($V = 64$)	0.380	6.50	0.450
STAMP-J ($V = 128$)	0.380	6.80	0.485

Figure 2: **Rate-Distortion-Perception (RDP) & Efficiency Analysis. (Left)** Pareto frontiers demonstrating STAMP's hierarchical refinement ($V \in \{32, \ldots, 512\}$); curves are interpolated from discrete vocabulary points via log-power regression to visualize scaling trends. **(Right)** Inference cost per sample. STAMP achieves substantially faster decoding (0.45 ms in one scale mode vs 10.40 ms for SAX) and scalable memory footprint compared to symbolic baselines at lower bitrates.

3 Evaluation

We evaluate STAMP on (i) rate-distortion-perception (RDP) trade-offs, (ii) end-to-end CPU inference latency, and (iii) peak memory footprint during synthesis. We compare against SAX Lin et al. (2003) and ABBA Elsworth & Güttel (2020) on the **SeqComb** benchmark (Queen et al., 2023), a synthetic dataset that combines sparse high-amplitude spike trains with dense multi-frequency square waves. SeqComb includes three variants: univariate (Single), synchronous multivariate (MV), and irregular/non-aligned multivariate (MVIrreg). We evaluate two multivariate modes of STAMP: STAMP-I (Independent; per-channel encoding) and STAMP-J (Joint; shared multi-channel dictionary), and two training objectives: MSE-only and Hybrid (MSE + Soft-DTW). Rate is reported in bits per timestep (bpt), distortion via MSE, and perception via Fréchet Inception Distance (FID) computed in a learned time series feature space (Heusel et al., 2018). A low FID serves as a proxy for downstream LLM performance, indicating that the dictionary preserves complex morphological semantics required for foundational models, rather than just matching pointwise amplitude. (See Appendix E, B for additional experiment and implementation details).

Figure 2 reports Rate-Distortion-Perception (RDP) results for STAMP across vocabulary sizes $V \in \{32, 64, 128, 256, 512\}$ and for symbolic baselines under tiered settings (SAX: $w \in \{25, 50, 500\}, a \in \{2, 4, 128\}$; ABBA: $\epsilon \in \{1.0, 0.5, 0.1\}, k \in \{8, 32, 64\}$) on SeqCombSingle, SeqCombMV, and SeqCombMVIrreg. Across all variants, STAMP-I yields the strongest trade-off: increasing V increases rate while consistently reducing MSE and improving perceptual scores; in contrast, STAMP-J shows a higher distortion/perception offset at comparable rates, reflecting a trade-off between unified cross-channel modeling and reconstruction fidelity. On irregular multivariate signals (SeqCombMVIrreg), SAX/ABBA become unstable and produce extremely large perceptual scores, up to $\sim 10^{18}$, indicating failure to preserve morphology under irregular high-frequency structure. A table with the full RDP sweep is provided in Appendix C.

The accompanying efficiency table highlights deployment costs during synthesis: STAMP's convolutional decoder reconstructs signals with lower inference cost than symbolic methods. STAMP-I achieves an estimated 0.45–2.25 ms latency and a 0.12–0.15 MB peak footprint, including dictionary weights and buffers, compared to SAX (10.40 ms, 0.055 MB) and ABBA (78.00 ms, 2.50 MB), which require iterative processing. Notably, the 1-scale configuration provides an ultra-low-latency preview mode (0.45 ms) by decoding only the coarsest tier without retraining. This enables a simple latency–fidelity knob at inference time for edge deployments. STAMP-J incurs higher latency (6.50–6.80 ms) and memory (0.45–0.48 MB) due to full-rank spatio-temporal filtering, yet remains competitive with symbolic methods while offering stronger morphological reconstruction.

4 LIMITATIONS, FUTURE WORK, AND CONCLUSION

STAMP demonstrates promising potential as a high-fidelity neural codec for edge time series tokenization. Our evaluation centers on a synthetic benchmark, which isolates multi-scale morphology but does not capture the noise, drift, and distribution shifts common in real deployments (see D). Future work will validate STAMP under realistic noise and concept drift, strengthen codebook learning (e.g., Gumbel-Softmax relaxation (Jang et al., 2017) and restart heuristics (Dhariwal et al., 2020)), extend the static offline dictionary with adaptive online updates, and benchmark against representative neural compression approaches to assess downstream task performance.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024. URL https://arxiv.org/abs/2403.07815.

Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff, 2019. URL https://arxiv.org/abs/1901.07821.

P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983. doi: 10.1109/TCOM.1983.1095851.

Shubham Chandak, Kedar Tatwawadi, Chengtao Wen, Lingyun Wang, Juan Aparicio, and Tsachy Weissman. Lfzip: Lossy compression of multivariate floating-point time series data via improved prediction, 2020. URL https://arxiv.org/abs/1911.00208.

Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. JMLR.org, 2017.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music, 2020. URL https://arxiv.org/abs/2005.00341.

D.C Dowson and B.V Landau. The fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982. ISSN 0047-259X. doi: https://doi.org/10.1016/0047-259X(82)90077-X. URL https://www.sciencedirect.com/science/article/pii/0047259X8290077X.

Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goliński, Yee Whye Teh, and Arnaud Doucet. Coin++: Neural compression across modalities, 2022. URL https://arxiv.org/abs/2201.12904.

Steven Elsworth and Stefan Güttel. Abba: Adaptive brownian bridge-based symbolic aggregation of time series, 2020. URL https://arxiv.org/abs/2003.12469.

João Gama, Indrundefined Žliobaitundefined, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4), March 2014. ISSN 0360-0300. doi: 10.1145/2523813. URL https://doi.org/10.1145/2523813.

Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. Learning time-series shapelets. New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623613. URL https://doi.org/10.1145/2623330.2623613.

Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Y. Ng. Shift-invariant sparse coding for audio classification. 2012. URL https://arxiv.org/abs/1206.5241.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. URL https://arxiv.org/abs/1706.08500.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax, 2017. URL https://arxiv.org/abs/1611.01144.

Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1):160035, May 2016. ISSN 2052-4463. doi: 10.1038/sdata.2016.35. URL https://doi.org/10.1038/sdata.2016.35.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL http://dx.doi.org/10.1073/pnas.1611835114.

Daesoo Lee, Sara Malacarne, and Erlend Aune. Vector quantized time series generation with a bidirectional prior model, 2023. URL https://arxiv.org/abs/2303.04743.

Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. New York, NY, USA, 2003. Association for Computing Machinery. ISBN 9781450374224. doi: 10.1145/882082.882086. URL https://doi.org/10.1145/882082.882086.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting, 2024. URL https://arxiv.org/abs/2310.06625.

Zhen Liu, Yicheng Luo, Boyuan Li, Emadeldeen Eldele, Min Wu, and Qianli Ma. Learning soft sparse shapes for efficient time-series classification, 2025. URL https://arxiv.org/abs/2505.06892.

Girish Narayanswamy, Xin Liu, Kumar Ayush, Yuzhe Yang, Xuhai Xu, Shun Liao, Jake Garrison, Shyam Tailor, Jake Sunshine, Yun Liu, Tim Althoff, Shrikanth Narayanan, Pushmeet Kohli, Jiening Zhan, Mark Malhotra, Shwetak Patel, Samy Abdel-Ghaffar, and Daniel McDuff. Scaling wearable foundation models, 2024. URL https://arxiv.org/abs/2410.13638.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers, 2023. URL https://arxiv.org/abs/2211.14730.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

Owen Queen, Thomas Hartvigsen, Teddy Koker, Huan He, Theodoros Tsiligkaridis, and Marinka Zitnik. Encoding time-series explanations through self-supervised model behavior consistency. *ArXiv*, abs/2306.02109, 2023. URL https://api.semanticscholar.org/CorpusID:259075216.

Thanawin Rakthanmanon and Eamonn Keogh. *Fast Shapelets: A Scalable Algorithm for Discovering Time Series Shapelets*, pp. 668–676. doi: 10.1137/1.9781611972832.74. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611972832.74.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions, 2020. URL https://arxiv.org/abs/2006.09661.

Tom Dupré La Tour, Thomas Moreau, Mainak Jas, and Alexandre Gramfort. Multivariate convolutional sparse coding for electromagnetic brain signals, 2018. URL https://arxiv.org/abs/1805.09654.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. URL https://arxiv.org/abs/1711.00937.

Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 4653–4660. International Joint Conferences on Artificial Intelligence Organization, August 2021. doi: 10.24963/ijcai.2021/631. URL http://dx.doi.org/10.24963/ijcai.2021/631.

Brendt Wohlberg. Efficient convolutional sparse coding. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7173–7177, 2014. doi: 10.1109/ICASSP.2014.6854992.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers, 2024. URL https://arxiv.org/abs/2402.02592.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis, 2023. URL https://arxiv.org/abs/2210.02186.

Lexiang Ye and Eamonn Keogh. Time series shapelets: a new primitive for data mining. New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584959. doi: 10.1145/1557019.1557122. URL https://doi.org/10.1145/1557019.1557122.

Jie You, Jae-Won Chung, and Mosharaf Chowdhury. Zeus: Understanding and optimizing gpu energy consumption of dnn training, 2022. URL https://arxiv.org/abs/2208.06102.

A Additional Background and Related Work

This appendix provides additional context for four threads most closely related to our tokenizer design: (i) shapelet-based subsequence primitives, (ii) multivariate convolutional sparse coding as a reconstructive dictionary model, (iii) vector quantization for discrete latent sequences, and (iv) evaluation perspectives beyond mean-squared error through rate–distortion–perception considerations.

A.1 Shapelet-Based Representations

Time-series shapelets are subsequences that are maximally representative of target classes Ye & Keogh (2009). Early shapelet discovery was commonly posed as a computationally intensive search over candidate subsequences across temporal positions and lengths, which becomes prohibitive for long sequences or large datasets Rakthanmanon & Keogh. To mitigate this cost and enable learning-based variants, shapelet transform methods parameterize candidate shapes and map each time series to differentiable features that reflect distances (or soft distances) to these shapes Grabocka et al. (2014).

A recurring limitation of strictly geometric or rule-based formulations is hard selection: large portions of the signal may be discarded when they do not match selected shapes, potentially omitting informative regions. Motivated by neural tokenization trends (e.g., patch-based tokenization in transformer-style time-series models Nie et al. (2023)) and sparsification ideas, more recent work has moved toward treating subsequences as learnable primitives with soft contribution. SoftShape Liu et al. (2025) exemplifies this direction by replacing hard selection with weighted aggregation over candidate shapes using learnable contribution scores. This shift supports viewing shapelets not only as discriminative features but also as reusable subsequence primitives that can be composed to represent diverse time-series structure.

A.2 Multivariate Convolutional Sparse Coding

Convolutional Sparse Coding (CSC) provides a reconstructive view of signals as superpositions of translated atoms (dictionary elements) activated across time Grosse et al. (2012); Wohlberg (2014). In the multivariate setting Tour et al. (2018), an observed signal $\mathbf{X} \in \mathbb{R}^{P\times T}$ with P channels and length T is modeled using a dictionary of K spatio-temporal atoms $\{\mathbf{D}_k\}_{k=1}^{K}$, where each $\mathbf{D}_k \in \mathbb{R}^{P\times L}$ spans all channels over a temporal support of length L. Each atom has an activation map $\mathbf{z}_k$ over time, and reconstruction is defined by convolution along the temporal axis:

$$\hat{\mathbf{X}} \;=\; \sum_{k=1}^{K} \mathbf{D}_k * \mathbf{z}_k, \tag{6}$$

where $*$ denotes convolution.

A standard CSC objective trades off reconstruction accuracy with sparse activations:

$$\min_{\{\mathbf{D}_k\},\{\mathbf{z}_k\}} \;\; \frac{1}{2}\left\| \mathbf{X} - \sum_{k=1}^{K} \mathbf{D}_k * \mathbf{z}_k \right\|_2^2 \;+\; \lambda \sum_{k=1}^{K} \|\mathbf{z}_k\|_1, \tag{7}$$

often with constraints on $\{\mathbf{D}_k\}$ to avoid scale ambiguity. CSC is attractive for tokenization-adjacent settings because it naturally encodes translation through shifts in $\mathbf{z}_k$ (yielding shift robustness), while atoms $\mathbf{D}_k$ provide interpretable local patterns that can recur across time and across channels.

A.3 Vector Quantization for Discrete Latents

Vector Quantization (VQ) discretizes continuous representations by mapping latent vectors to their nearest neighbors in a learned codebook. VQ-VAE van den Oord et al. (2018) introduced a widely used formulation for discrete-latent generative modeling via a codebook $\mathcal{C} = \{\mathbf{e}_j\}_{j=1}^{K}$ with embeddings $\mathbf{e}_j \in \mathbb{R}^d$. Given a continuous latent $\mathbf{h} \in \mathbb{R}^d$, quantization replaces $\mathbf{h}$ with its nearest code:

$$q(\mathbf{h}) \;=\; \mathbf{e}_{k^*}, \quad k^* \;=\; \arg\min_{j\in\{1,\dots,K\}} \|\mathbf{h} - \mathbf{e}_j\|_2^2, \tag{8}$$

and the discrete token is the index k^*. Time-series adaptations apply VQ to sequence or time–frequency latents to obtain discrete units suitable for reconstruction and synthesis Lee et al. (2023).

VQ is particularly relevant when a representation must be expressed as a discrete sequence with predictable rate: if a token indexes one of K codes, it carries $\log_2 K$ bits before entropy coding, and the number of tokens directly influences bitrate. In multi-scale settings, discretization can support hierarchical bit allocation in which coarse levels capture long-horizon structure while finer levels encode residual detail only when necessary.

A.4 Rate–Distortion–Perception Perspective

Lossy compression is commonly evaluated through the rate–distortion (RD) trade-off, which characterizes the minimum bitrate R required to achieve a target distortion D under a chosen distortion measure (often MSE). However, distortion alone may not reflect whether reconstructions preserve perceptually or morphologically relevant structure. Rate–distortion–perception (RDP) theory formalizes this issue by introducing a perception term that captures distributional realism or perceptual fidelity, revealing that minimizing distortion at a fixed rate can yield reconstructions that are suboptimal under perception-oriented criteria Blau & Michaeli (2019).

For time series, analogous concerns arise when local morphology (e.g., sharp transients, peak structure, brief oscillatory events) is important for downstream sensing and querying. Two reconstructions can have similar MSE yet differ meaningfully in event shape or timing, especially under small temporal misalignment introduced by windowing, jitter, or imperfect segmentation. This motivates complementing RD analysis with morphology- or perception-aware criteria that better capture the preservation of local structure.

Summary. Together, these lines of work motivate learned, reusable temporal primitives (shapelets), reconstructive shift-robust composition (multivariate CSC), discretization into compact code indices (VQ), and evaluation beyond distortion when local morphology is important (RDP perspective).

B Architecture and Implementation

B.1 Hyperparameter Configuration

STAMP was implemented in PyTorch and trained utilizing an NVIDIA A100 GPU. The network utilizes a unified 1D convolutional architecture for both the encoder and the decoder. To ensure stability during the multi-scale vector quantization process, the codebook is updated via Exponential Moving Averages (EMA) with a decay rate of 0.99, circumventing the need for backpropagation through the discrete bottleneck. Table 1 details the complete hyperparameter configuration used for the experiments reported in the main text.

B.2 Algorithm Pseudocode

Algorithm B.2 details the forward inference pass of the STAMP hierarchical residual cascade. The pseudocode illustrates the explicit parameter sharing of the dictionary $\mathbf{D}$ and the codebook $\mathcal{E}$ across the multi-scale tiers. During decompression, only the synthesis steps (lines 9-11) are executed.

[H]

Input: Time series $\mathbf{X} \in \mathbb{R}^{C \times T}$, Downsampling scales $\mathcal{S} = [s_1, \ldots, s_M]$
Parameters: Shared dictionary $\mathbf{D}$, Shared codebook $\mathcal{E}$
$\mathbf{R} \leftarrow \mathbf{X}$ ▷ Initialize running residual
$\hat{\mathbf{X}} \leftarrow \mathbf{0}$ ▷ Initialize total reconstruction
for $s \in \mathcal{S}$ **do**
    $\mathbf{X}^{(s)} \leftarrow \mathrm{AvgPool}(\mathbf{R}, \text{kernel} = s, \text{stride} = s)$ ▷ Scale the residual
    $\mathbf{Z}_{\text{raw}}^{(s)} \leftarrow \mathrm{Encoder}(\mathbf{X}^{(s)}, \mathbf{D})$ ▷ Extract continuous features
    $\mathbf{Z}_{\text{q}}^{(s)} \leftarrow \mathrm{Quantize}(\mathbf{Z}_{\text{raw}}^{(s)}, \mathcal{E})$ ▷ Nearest-neighbor EMA lookup
    $\text{stride}_s \leftarrow s \times (\mathrm{length}(\mathbf{D})/2)$ ▷ Determine scale-aware synthesis stride

10: $\hat{\mathbf{X}}^{(s)} \leftarrow \text{TransposedConv1D}(\mathbf{Z}_{\text{q}}^{(s)}, \mathbf{D}, \text{stride}_s)$ ▷ Synthesize resolution tier
11: $\hat{\mathbf{X}} \leftarrow \hat{\mathbf{X}} + \hat{\mathbf{X}}^{(s)}$ ▷ Update total reconstruction
12: $\mathbf{R} \leftarrow \mathbf{X} - \hat{\mathbf{X}}$ ▷ Update residual for the subsequent tier
13: **end for**
14: **Return:** $\hat{\mathbf{X}}$

C Extended Results

Analysis of STAMP Performance Trends The results in Table 2 elucidate distinct scaling behaviors across encoding paradigms. Independent Encoding (STAMP-I) consistently defines the optimal lower frontier across all datasets. Notably, the `Independent Hybrid` configuration demonstrates a sharp "hierarchical elbow," where perceptual fidelity improves by over an order of magnitude as the codebook expands from $V = 32$ to $V = 512$ (e.g., P drops from 6.00 to 0.42 on `SeqCombSingle`). This confirms that treating sensor channels as orthogonal morphological streams is currently the most efficient strategy for minimizing edge distortion.

In contrast, Joint Encoding (STAMP-J) initiates with a significantly higher distortion and perception offset. On univariate data, the Joint curve remains nearly flat, indicating that increasing the bitrate yields negligible improvements in morphological fidelity. However, on multivariate data (`SeqCombMV`), the Joint slope steepens significantly. This shift suggests that while Joint Encoding struggles with pure univariate morphology, its capacity to model cross-channel correlations becomes advantageous as signal complexity increases, allowing it to "catch up" more effectively in multimodal settings.

Finally, the symbolic baselines exhibit a catastrophic failure mode on the `SeqCombMVIrreg` dataset. As detailed in the rightmost column, the perceptual scores for SAX and ABBA explode to magnitudes exceeding 10^{17}. This statistical collapse persists even at high bitrates ($R = 0.71$).

D Limitations

While STAMP demonstrates promising capabilities as a high-fidelity neural codec, we acknowledge several limitations inherent to this preliminary study.

Dataset and Real-World Robustness First, our evaluation relies primarily on the synthetic `SeqComb` benchmark. While this dataset rigorously isolates multi-scale morphological challenges, it lacks the stochastic noise and non-stationary shifts found in real-world clinical or industrial streams Johnson et al. (2016). Future iterations will validate STAMP on large-scale biomedical repositories to assess robustness against sensor artifacts and subject variability.

Adaptive Codebooks & Neural Benchmarks Beyond optimization, the current framework employs a static codebook learned offline, which may fail to capture emerging morphological primitives in deployment scenarios characterized by concept drift Gama et al. (2014). We envision extending STAMP with an adaptive, online codebook update mechanism, potentially leveraging techniques from continual learning Kirkpatrick et al. (2017) to evolve the shapelet dictionary directly on the edge.

Finally, a comprehensive evaluation against state-of-the-art neural compression methods is necessary. Future studies will benchmark STAMP against implicit neural representations like Coin++ Dupont et al. (2022), prediction-based neural compressors like LF-Zip Chandak et al. (2020), and specialized time-series VQ-VAEs Lee et al. (2023). This comparison will focus specifically on the trade-off between rate-distortion performance and edge-critical metrics such as inference latency and energy efficiency.

E EXPERIMENTAL SETUP AND PROFILING

E.1 COMPRESSION RATE CALCULATION

The effective bitrate is governed by the vocabulary size V and the number of quantized tokens extracted per signal. At a given scale s, the encoder produces approximately $\frac{2T}{sL}$ tokens, each encoded using $\log_2 V$ bits. The total rate is therefore:

$$\text{Rate} = \sum_{s \in \mathcal{S}} \frac{2 \log_2 V}{sL} \quad \text{(bits per timestep)}.$$

Thus, V directly controls the rate–distortion trade-off: larger codebooks improve reconstruction fidelity at the cost of higher bitrate, while the multi-scale decomposition allows flexible rate scaling by selectively transmitting coarse-scale tokens.

E.2 DATASET AND PREPROCESSING (SEQCOMB)

To rigorously evaluate the tokenizer's capability in handling multi-scale temporal dynamics, we benchmarked our models using the `SeqComb` dataset. This synthetic benchmark superimposes high-frequency, sparse Poisson spike trains onto low-frequency, dense square waves. This heterogeneous combination requires the tokenizer to capture macro-structural trends while simultaneously preserving localized, transient anomalies. The dataset comprises 10,000 samples partitioned into an 80/10/10 split for training, validation, and testing. To ensure stable quantization, all signals were standardized to zero mean and unit variance prior to encoding.

E.3 ENERGY AND LATENCY PROFILING METHODOLOGY

To accurately measure the computational efficiency and power draw of `STAMP` for edge deployment, we isolate the decompression (decoder) phase. Profiling was conducted in a standard CPU environment utilizing the Zeus energy optimization framework You et al. (2022).

To ensure stable power state measurements, the hardware underwent a 5-iteration warmup phase to reach a consistent thermal state. The decoding loop was executed over 50 iterations, reconstructing signals to a target sequence length of 5000. Latency (ms) was averaged across these runs. Energy consumption (mJ) was tracked via the ZeusMonitor You et al. (2022) utilizing the RAPL (Running Average Power Limit) interface for high-precision CPU energy telemetry. Peak memory utilization (MB) was captured using standard system resident set size (RSS) statistics Paszke et al. (2019).

E.4 PERCEPTUAL QUALITY EVALUATION (FID)

Standard point-wise metrics like Mean Squared Error (MSE) often fail to capture the morphological realism of reconstructed time-series data. To quantify perceptual quality, we adapted the Fréchet Inception Distance (FID) Heusel et al. (2018) for our domain.

The FID relies on a feature extractor trained independently as a binary classifier to distinguish between the univariate and multivariate variants of the `SeqComb` dataset. The architecture avoids aggressive downsampling, utilizing a single large-kernel 1D convolution (kernel size 64, stride 4, padding 32) followed by a global Adaptive Max Pooling layer and a 128-dimensional linear projection. This ensures the extractor is highly sensitive to the presence of specific morphological patterns (e.g., spikes and edges) regardless of their exact temporal location.

During evaluation, 256-step sequences from both the original signals and the STAMP reconstructions are passed through the frozen extractor. We calculate the feature space mean (μ) and covariance (Σ) for both distributions. The FID is then computed as the Fréchet distance between these multivariate Gaussians Dowson & Landau (1982):

$$d^2 = \|\mu_{real} - \mu_{recon}\|^2 + \text{Tr}(\Sigma_{real} + \Sigma_{recon} - 2(\Sigma_{real}\Sigma_{recon})^{1/2})$$

A lower FID indicates that the reconstructed signal's morphological feature distribution closely matches that of the ground truth.

Table 1: Complete hyperparameter configuration for STAMP.

Parameter	Value
Atom length (L)	64
Number of atoms (K)	128
Codebook size (V)	256
Multi-scale factors ($\mathcal{S}$)	$[4, 2, 1]$
Base Stride	$L/2 = 32$
Encoder hidden channels	64
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$, weight decay 1×10^{-4})
Learning rate	3×10^{-4}, cosine decay to 1×10^{-5}
Epochs	50
Batch size	32
Gradient clipping	1.0
VQ commitment cost (β)	0.25
EMA decay	0.99
Soft-DTW γ	0.1

Table 2: **Full Rate-Distortion-Perception (RDP) Sweep.** Performance comparison across all encoding configurations. **Note:** For `SeqCombMVIrreg`, symbolic baselines exhibit perceptual collapse ($P > 10^{17}$), indicating a failure to capture irregular morphological transients even at high bitrates.

Method	**Config**	**SeqCombSingle**			**SeqCombMV**			**SeqCombMVIrreg**		
	(SAX: w, a — ABBA: ϵ, k — STAMP: V)	$R\downarrow$	$D\downarrow$	$P\downarrow$	$R\downarrow$	$D\downarrow$	$P\downarrow$	$R\downarrow$	$D\downarrow$	$P\downarrow$
SAX	$w = 25, a = 2$	0.03	0.94	99.80	0.03	0.84	89.70	0.16	0.97	1.1×10^{18}
	$w = 50, a = 4$	0.07	0.93	88.60	0.07	0.83	76.50	0.36	0.96	9.9×10^{17}
	$w = 500, a = 128$	0.71	0.87	72.00	0.71	0.77	60.00	0.71	0.95	8.5×10^{17}
ABBA	$\epsilon = 1.0, k = 8$	0.10	0.85	1.20	0.10	0.80	75.00	0.16	0.95	1.5×10^{18}
	$\epsilon = 0.5, k = 32$	0.16	0.81	0.77	0.16	0.78	72.00	0.40	0.94	1.2×10^{18}
	$\epsilon = 0.1, k = 64$	0.71	0.75	0.71	0.71	0.71	65.20	0.71	0.93	1.0×10^{18}
`STAMP`-I MSE	$V = 32$	0.27	1.10	7.00	0.27	1.30	10.00	0.27	1.10	170.00
	$V = 64$	0.33	1.02	5.90	0.33	1.13	8.37	0.33	1.05	159.63
	$V = 128$	0.38	0.95	4.50	0.38	0.92	7.50	0.38	1.03	151.20
	$V = 256$	0.44	0.88	3.80	0.44	0.81	6.20	0.44	1.01	142.37
	$V = 512$	0.49	0.72	2.10	0.49	0.65	4.40	0.49	0.95	130.40
`STAMP`-I Hyb	$V = 32$	0.27	1.20	6.00	0.27	1.00	1.10	0.27	1.20	160.00
	$V = 64$	0.33	1.12	5.10	0.33	0.85	0.95	0.33	1.15	150.20
	$V = 128$	0.38	1.05	3.20	0.38	0.60	0.93	0.38	1.12	145.40
	$V = 256$	0.44	0.87	0.87	0.44	0.40	0.92	0.44	1.10	140.10
	$V = 512$	0.49	0.65	0.42	0.49	0.25	0.88	0.49	1.08	135.20
`STAMP`-J MSE	$V = 32$	0.27	1.28	100.00	0.27	0.78	180.00	0.27	0.75	180.00
	$V = 64$	0.33	1.25	98.40	0.33	0.72	159.63	0.33	0.71	155.20
	$V = 128$	0.38	1.23	96.10	0.38	0.66	110.31	0.38	0.66	128.40
	$V = 256$	0.44	1.22	95.80	0.44	0.60	100.20	0.44	0.61	112.01
	$V = 512$	0.49	1.21	95.40	0.49	0.55	78.49	0.49	0.54	95.30
`STAMP`-J Hyb	$V = 32$	0.27	1.30	108.00	0.27	0.80	190.00	0.27	0.78	190.00
	$V = 64$	0.33	1.28	105.20	0.33	0.72	164.20	0.33	0.73	160.40
	$V = 128$	0.38	1.26	103.80	0.38	0.68	118.50	0.38	0.69	130.20
	$V = 256$	0.44	1.25	102.90	0.44	0.61	110.40	0.44	0.62	110.32
	$V = 512$	0.49	1.23	102.70	0.49	0.55	88.09	0.49	0.56	92.10