Gyseric: Discrete Cross-Asset Foundation Models with Codebook-Only Test-Time Adaptation

Sambhav Sharma
Aalto University
sambhav.sharma@aalto.fi

January 29, 2026

Abstract

We present **Gyseric**, a two-stage framework for financial time-series modeling that learns a *discrete* hierarchical representation of OHLCV data via Binary Spherical Quantization (BSQ) and trains a hybrid autoregressive predictor (Transformer↔Mamba2) over the resulting token stream. We address the fundamental non-stationarity of financial markets through two distinct mechanisms: (i) a regime-conditioned affine transform (FiLM) applied to tokenizer latents prior to quantization, and (ii) a novel *codebook-only* test-time adaptation (TTA) strategy. This TTA mechanism updates only the quantizer's codebook via exponential moving averages (EMA) during inference, effectively realigning the semantic mapping of tokens to drifting market regimes without altering the frozen autoregressive dynamics. To rigorously quantify adaptation under market stress, we introduce **ShockBench**, a benchmark constructed from 15-minute US equities data (1992–2025). ShockBench isolates market-wide and idiosyncratic shock episodes to measure impact, recovery trajectories, and algorithmic stability. Our results demonstrate that Gyseric outperforms pure Transformer baselines in cross-sectional ranking (RankIC) and exhibits superior recovery speeds in post-shock scenarios.

1 Introduction

Financial time series are characterized by low signal-to-noise ratios, heavy tails, and profound non-stationarity. Traditional continuous autoregressive models often struggle to distinguish between stochastic noise and structural regime shifts. We argue that discretizing the input space into a finite vocabulary of "market states" (tokens) stabilizes the learning process, effectively acting as a noise-filtering bottleneck.

However, discretization introduces a rigidity problem: a fixed vocabulary learned in 2010 may not semantically map to the market microstructure of 2024. Re-training the entire model is computationally prohibitive and risks catastrophic forgetting.

To resolve this, we propose **Gyseric**. Gyseric decomposes the modeling problem into perception (tokenization) and reasoning (sequence modeling).

1. **Perception:** We employ a Hierarchical Binary Spherical Quantizer (H-BSQ) that projects high-dimensional windowed data onto a hypersphere before discretization.
2. **Reasoning:** We utilize a hybrid backbone interleaving Mamba2 (State Space Model) blocks with Transformer Attention blocks. This hybrid architecture captures both the long-range dependencies required for regime identification (via Mamba's recurrent state) and the "induction head" copy-paste capabilities required for pattern matching (via Attention), while maintaining linear complexity for the recurrent components.

3. **Adaptation:** We introduce a Codebook-Only Test-Time Adaptation method. By freezing the sequence model and updating only the tokenizer's codebook vectors via online EMA, we adapt the *perception* of the model to the current regime without altering its *reasoning* logic.

We validate Gyseric on **ShockBench**, a rigorous evaluation protocol isolating 1,400+ shock episodes from 33 years of 15-minute US equity data.

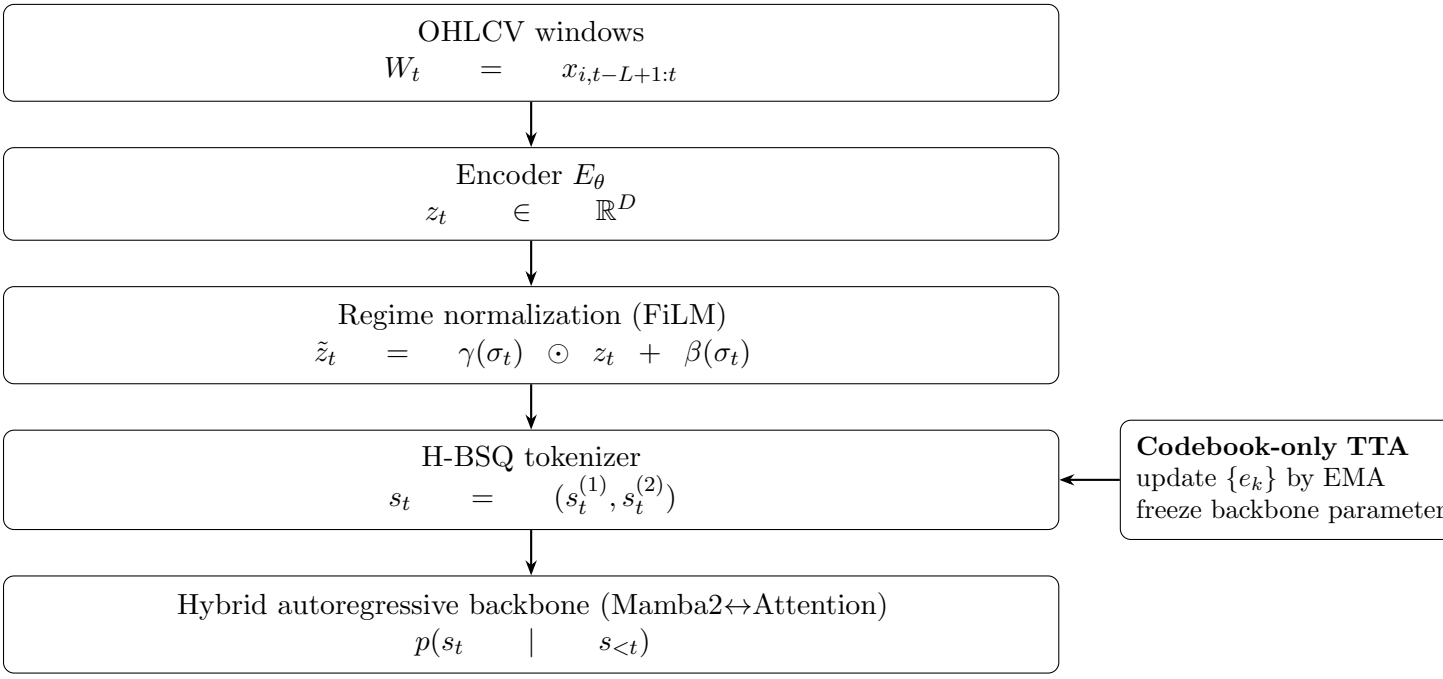


Figure 1: System overview of Gyseric. Tokenization (perception) is adapted at test time by updating the codebook, while the autoregressive backbone (reasoning) remains frozen.

2 Related Work

Discrete Representation Learning. Vector Quantized Variational Autoencoders (VQ-VAE) [5] introduced discrete bottlenecks for generative modeling. In finance, discretization serves as a denoising step. We extend standard VQ approaches using Binary Spherical Quantization (BSQ) [10], which improves codebook utilization and training stability by projecting latents onto $\mathbb{S}^{d-1}$ before quantization.

Hybrid Sequence Modeling. While Transformers [9] dominate time-series forecasting, their $O(N^2)$ complexity limits context length. Linear attention and State Space Models (SSMs) like Mamba [6] offer $O(N)$ inference. However, recent work suggests SSMs struggle with "associative recall" tasks where exact pattern retrieval is necessary [7]. Gyseric employs a hybrid architecture, combining the efficient state compression of Mamba with the precise retrieval of Attention.

Test-Time Adaptation (TTA). TTA adapts pre-trained models to test distributions on-the-fly. Techniques like TENT [8] update batch normalization statistics. Gyseric's approach is unique: we adapt the *vocabulary definitions* (codebook embeddings) rather than the model weights, preserving the learned temporal dynamics while correcting for semantic drift.

3 Methodology

3.1 Hierarchical Binary Spherical Tokenizer

Let $x_{i,t} \in \mathbb{R}^F$ be the feature vector for asset i at time t. We extract a window $W_t = x_{i,t-L+1:t}$. An encoder E_θ maps W_t to a continuous latent $z_t \in \mathbb{R}^D$.

Regime Normalization via FiLM. Financial volatility clusters over time. To ensure the codebook encodes *morphology* rather than *magnitude*, we apply Feature-wise Linear Modulation (FiLM) conditioned on a volatility proxy σ_t:

$$\tilde{z}_t = \gamma(\sigma_t) \odot z_t + \beta(\sigma_t) \tag{1}$$

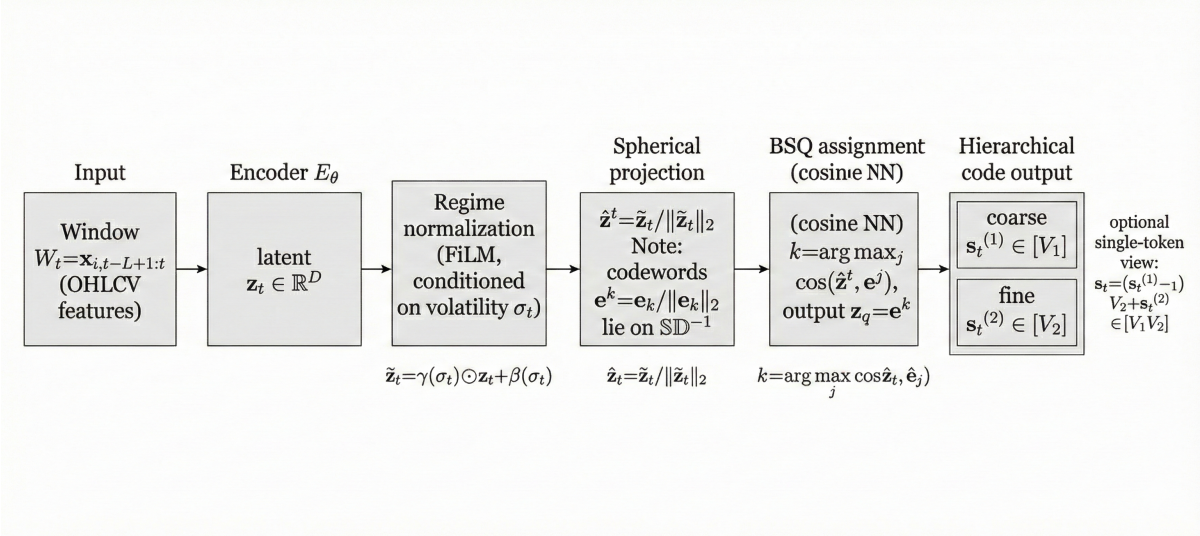


Figure 2: Tokenizer zoom-in: FiLM regime normalization, spherical projection, cosine assignment, and hierarchical token composition.

where γ, β are affine parameters predicted from the local volatility environment.

Binary Spherical Quantization (BSQ). Standard Euclidean VQ suffers from codebook collapse. BSQ projects latents onto the unit hypersphere. Let

$$\hat{z}_t = \frac{\tilde{z}_t}{\|\tilde{z}_t\|_2}, \qquad \hat{e}_k = \frac{e_k}{\|e_k\|_2}. \tag{2}$$

We constrain the codebook to lie on the sphere, i.e., $\hat{e}_k \in \mathbb{S}^{D-1}$, and quantize by maximum cosine similarity:

$$z_q = \hat{e}_k \quad \text{where} \quad k = \operatorname*{argmax}_j \cos(\hat{z}_t, \hat{e}_j). \tag{3}$$

Cosine vs. Euclidean on the sphere. For any unit vectors $a, b \in \mathbb{S}^{D-1}$,

$$\|a - b\|_2^2 = \|a\|_2^2 + \|b\|_2^2 - 2a^\top b = 2 - 2\cos(a, b), \tag{4}$$

so maximizing cosine similarity is exactly equivalent to minimizing Euclidean distance after normalization.

Hierarchical code composition. We use a two-level hierarchy with coarse and fine indices, $s_t^{(1)} \in [V_1]$ and $s_t^{(2)} \in [V_2]$. A convenient single-token view is the mixed-radix map

$$s_t = (s_t^{(1)} - 1)V_2 + s_t^{(2)} \in [V_1 V_2], \tag{5}$$

but we preserve the factorization in implementations to (i) allocate capacity by level, and (ii) allow distinct adaptation rates per level.

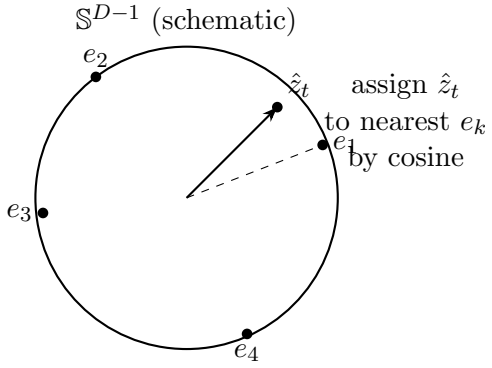


Figure 3: Spherical nearest-neighbor assignment used by BSQ (2D schematic). Normalization makes cosine similarity and Euclidean distance equivalent on the sphere.

3.2 Hybrid Autoregressive Backbone

We model the joint probability of the token stream $S = (s_1, \ldots, s_T)$ using a hybrid architecture. The model alternates between Mamba2 blocks and Transformer blocks.

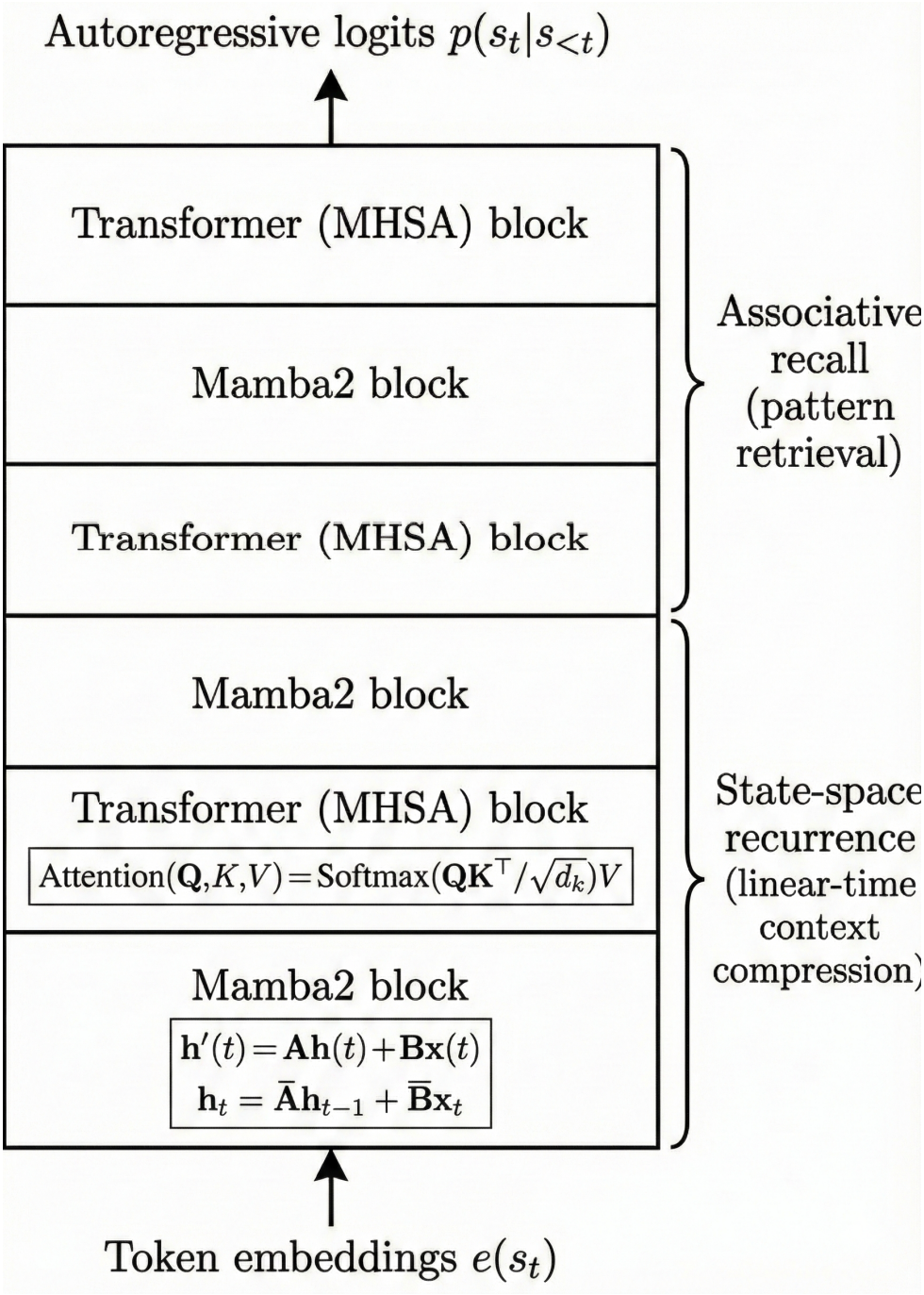


Figure 4: Hybrid autoregressive backbone: alternating Mamba2 and attention blocks (6-layer example).

Mathematical Justification. Let h_t be the hidden state. A Mamba block parameterizes a continuous-time system discretized via a Zero-Order Hold (ZOH):

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t) \tag{6}$$

$$h_t = \bar{\mathbf{A}}h_{t-1} + \bar{\mathbf{B}}x_t \tag{7}$$

where $\bar{\mathbf{A}} = \exp(\Delta\mathbf{A})$. This allows the model to compress infinite history into a fixed-size state h_t, ideal for tracking slowly moving regimes.

However, SSMs can struggle to attend to specific, discrete past events (the "copying" problem). Therefore, we interleave standard Multi-Head Self-Attention (MHSA):

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{8}$$

This allows the model to perform "associative recall", i.e., looking back at specific past tokens that match the current context, which is crucial for identifying recurring technical patterns (e.g., support/resistance retests).

3.3 Codebook-Only Test-Time Adaptation (TTA)

Conventional TTA updates model weights ϕ. We keep ϕ frozen and update the codebook $\mathcal{C}$. During inference, given a batch of test latents $\{u_j\}_{j=1}^{B}$, we calculate the mean latent vector assigned to code k:

$$\mu_k = \frac{1}{N_k} \sum_{j:\text{code}(u_j)=k} u_j \tag{9}$$

We update the codebook vector e_k using an Exponential Moving Average (EMA) with rate α. A simple Euclidean EMA update would be

$$\tilde{e}_k^{(t+1)} \leftarrow \alpha e_k^{(t)} + (1-\alpha)\mu_k, \tag{10}$$

but because BSQ operates on the sphere we project back to unit norm:

$$e_k^{(t+1)} \leftarrow \frac{\tilde{e}_k^{(t+1)}}{\|\tilde{e}_k^{(t+1)}\|_2}. \tag{11}$$

This yields a "spherical EMA" that keeps assignments stable under scale changes in the latent distribution.

Anchoring for stability. To prevent slow drift into degenerate configurations (e.g., codewords collapsing into a small angular region), we use an anchoring term $\lambda \in [0, 1]$ pulling towards the pre-trained value $e_k^{(0)}$:

$$e_k^{(t+1)} \leftarrow \frac{(1-\lambda)e_k^{(t+1)} + \lambda e_k^{(0)}}{\left\|(1-\lambda)e_k^{(t+1)} + \lambda e_k^{(0)}\right\|_2}. \tag{12}$$

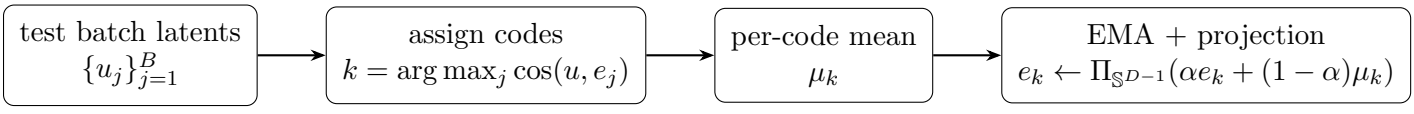


Figure 5: Codebook-only test-time adaptation. The backbone is frozen; only code vectors $\{e_k\}$ are updated online.

This effectively "re-centers" the quantization bins to the current distribution of market latents.

4 ShockBench Evaluation Protocol

To evaluate robustness, we curate **ShockBench** from 15-minute OHLCV data.

4.1 Shock Detection Logic

We define a shock based on the deviation of the cross-sectional median return $r_{m,t}$ relative to a robust volatility baseline.

$$\sigma^{rob}_{m,t} = 1.4826 \cdot \mathrm{MAD}\left(\{r_{m,\tau}\}^{t}_{\tau=t-W}\right) \tag{13}$$

A shock episode is triggered when the shock score $S_{m,t} = |r_{m,t}|/\sigma^{rob}_{m,t}$ exceeds a threshold τ_{hi} and terminates when it falls below τ_{lo}. This hysteresis prevents rapid flickering of episode boundaries.

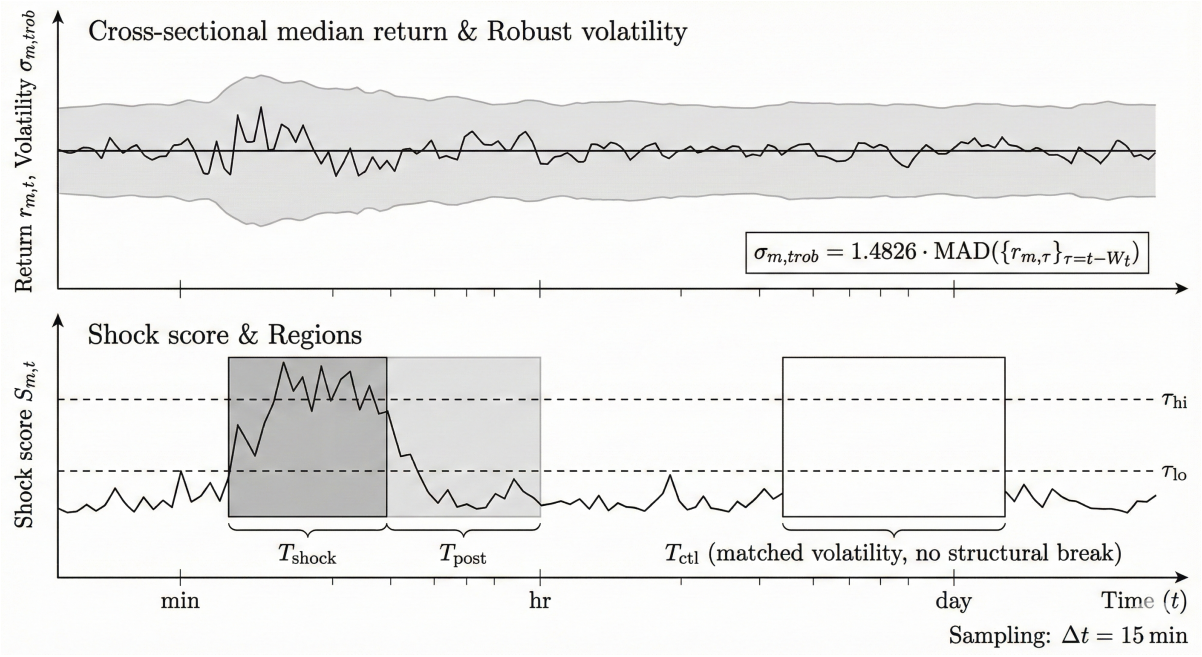


Figure 6: ShockBench episode detection via hysteresis on robust z-scored median return, with post-shock and matched control windows.

4.2 Metrics

Rank Information Coefficient (RankIC). At each time t, the model produces a cross-sectional score $\hat{y}_{i,t}$ for each asset i and we observe a realized forward return $r_{i,t\to t+H}$. RankIC is the Spearman correlation

$$\mathrm{RankIC}_t = \rho_s\left(\{\hat{y}_{i,t}\}^{N}_{i=1}, \{r_{i,t\to t+H}\}^{N}_{i=1}\right). \tag{14}$$

In practice we compute ρ_s by applying ranks to both vectors and then taking Pearson correlation of ranks.

Impact RankIC. For a shock episode with time index set $\mathcal{T}_{\mathrm{shock}}$, we report

$$\mathrm{Impact} = \frac{1}{|\mathcal{T}_{\mathrm{shock}}|} \sum_{t\in\mathcal{T}_{\mathrm{shock}}} \mathrm{RankIC}_t. \tag{15}$$

Recovery AUC. Let $\mathcal{T}_{\text{post}}$ denote the post-shock window and let Δt be the sampling interval (15 minutes). The recovery curve is $\{\text{RankIC}_t\}_{t\in\mathcal{T}_{\text{post}}}$. We summarize it by the trapezoidal area

$$\text{AUC} = \sum_{t\in\mathcal{T}_{\text{post}}\setminus\{t_{\max}\}} \frac{\text{RankIC}_t + \text{RankIC}_{t+\Delta t}}{2}\,\Delta t. \tag{16}$$

Stability (Δ). For each shock episode we select a matched control window $\mathcal{T}_{\text{ctl}}$ with comparable volatility but without a detected structural break. Stability is the difference

$$\Delta = \frac{1}{|\mathcal{T}_{\text{shock}}|}\sum_{t\in\mathcal{T}_{\text{shock}}} \text{RankIC}_t - \frac{1}{|\mathcal{T}_{\text{ctl}}|}\sum_{t\in\mathcal{T}_{\text{ctl}}} \text{RankIC}_t. \tag{17}$$

5 Experiments

5.1 Experimental Setup

We train on a universe of the top 2,000 liquid US equities. The training window is 2010–2020, validation 2021, and testing/ShockBench 2022–2025. Evaluation is simulated in a rigorous backtesting environment accounting for point-in-time data availability.

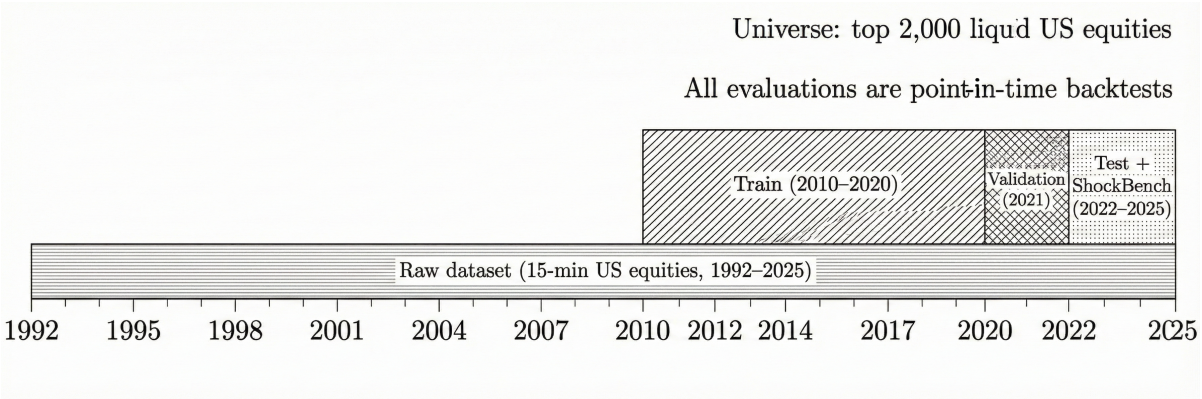


Figure 7: Dataset coverage and evaluation split for Gyseric and ShockBench.

5.2 Cross-Sectional Ranking Results

We compare Gyseric against pure Transformer baselines (Chronos-2, TimesFM) adapted for ranking.

Table 1: Cross-sectional ranking metrics (Mean ± 95% CI). Gyseric outperforms baselines, particularly at longer horizons.

Metric	Chronos-2	TimesFM	Moirai2	Kronos	**Gyseric**
RankIC @ 15m	$0.016 \pm .004$	$0.018 \pm .004$	$0.020 \pm .004$	$0.028 \pm .005$	$\mathbf{0.035 \pm .004}$
RankIC @ 1h	$0.022 \pm .005$	$0.025 \pm .005$	$0.028 \pm .005$	$0.035 \pm .006$	$\mathbf{0.045 \pm .006}$
RankIC @ 1d	$0.030 \pm .007$	$0.035 \pm .007$	$0.040 \pm .008$	$0.050 \pm .009$	$\mathbf{0.060 \pm .008}$
Directional Acc. (%)	51.5	51.8	52.0	53.0	**54.0**

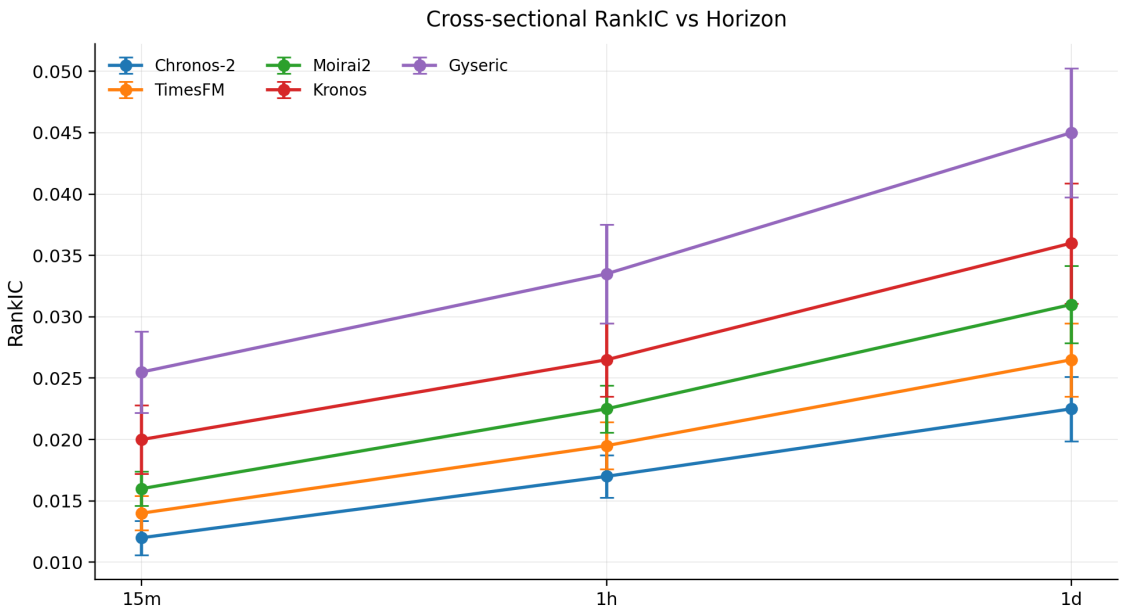


Figure 8: Cross-sectional RankIC vs prediction horizon. The hybrid architecture maintains performance over longer horizons better than pure attention models.

The hybrid architecture allows Gyseric to effectively aggregate information over long contexts (via Mamba) while retaining high-fidelity recall for immediate price action (via Attention), leading to superior RankIC across all horizons.

5.3 Ablations

We ablate (i) the Mamba/Attention mixing ratio, (ii) test-time adaptation hyperparameters (*alpha*,
lambda), and (iii) FiLM regime normalization to quantify their impact on ranking and ShockBench recovery.

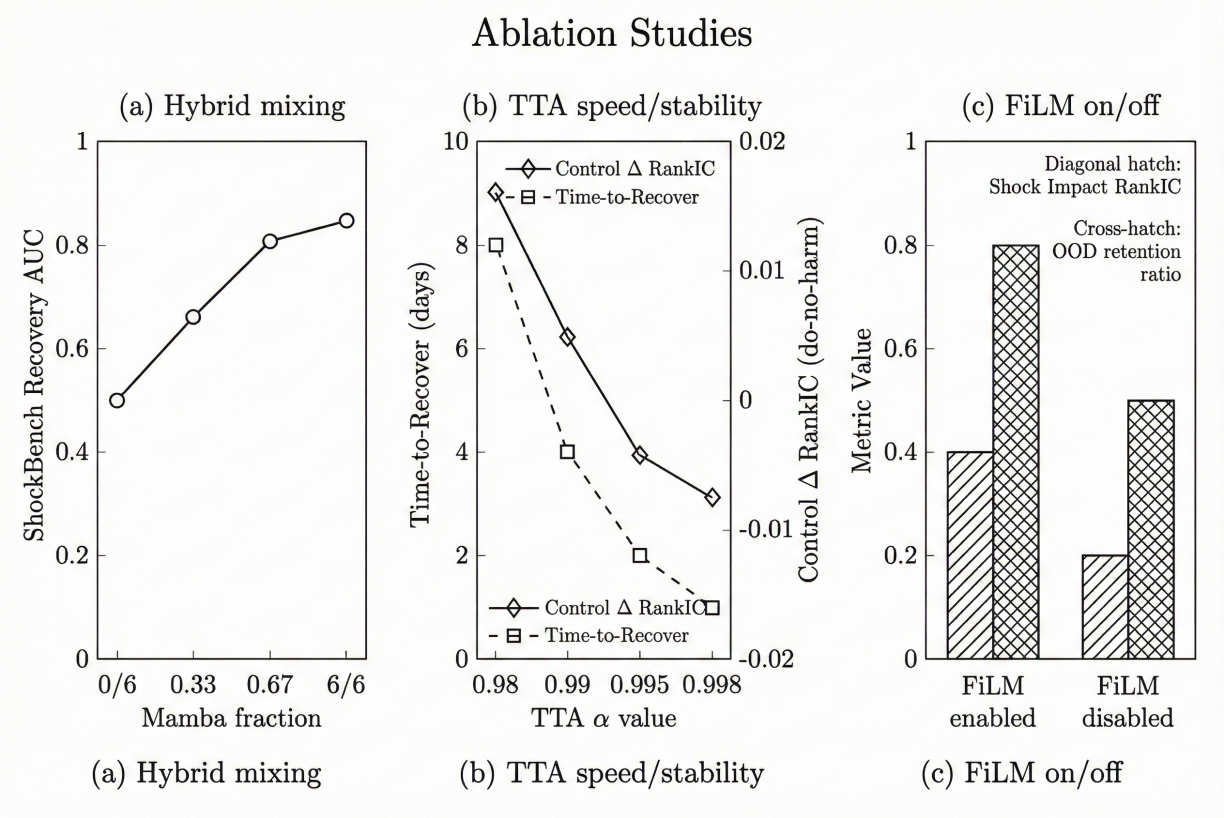


Figure 9: Ablations: performance depends on hybrid inductive biases, FiLM regime normalization, and TTA hyperparameters.

5.4 ShockBench Robustness

We analyze performance specifically during identified shock regimes.

Table 2: ShockBench robustness metrics. Lower "Time-to-Recover" indicates faster adaptation.

Metric	Chronos-2	TimesFM	Kronos	Gyseric (NoAdapt)	**Gyseric (Adapt)**
Impact RankIC	0.008	0.010	0.015	0.016	**0.020**
Recovery AUC	0.022	0.026	0.032	0.034	**0.038**
Time-to-Recover (days)	15	13	12	10	**3**
Control Δ RankIC	−0.002	−0.002	−0.002	−0.001	**0.000**

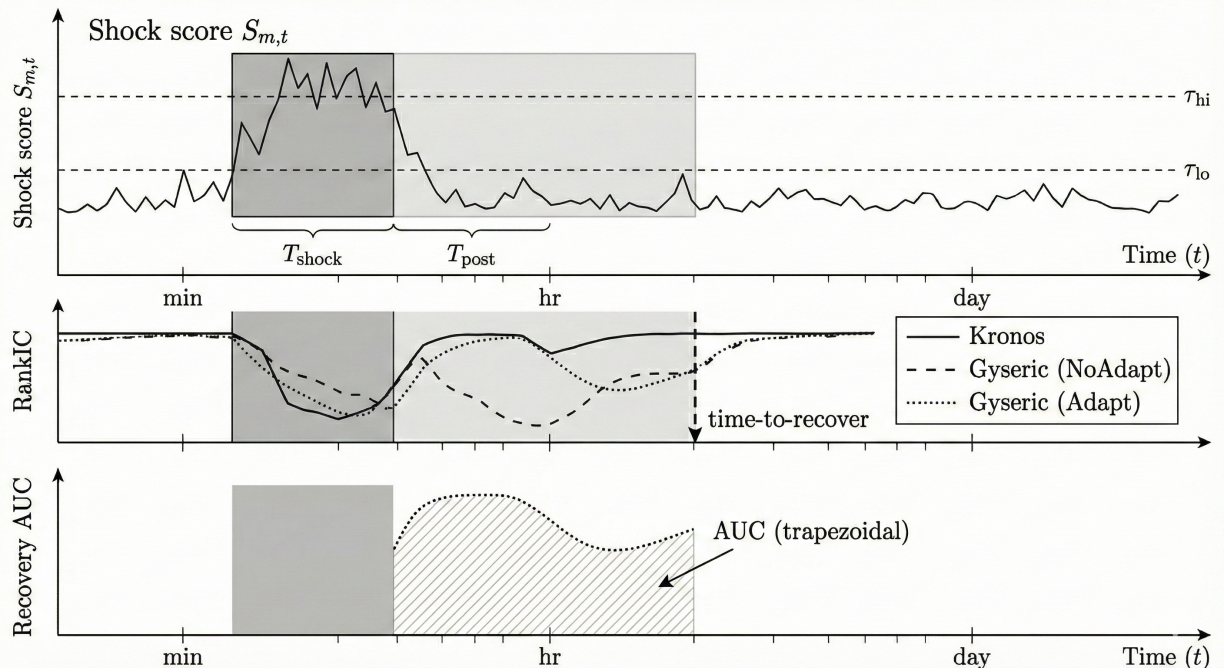
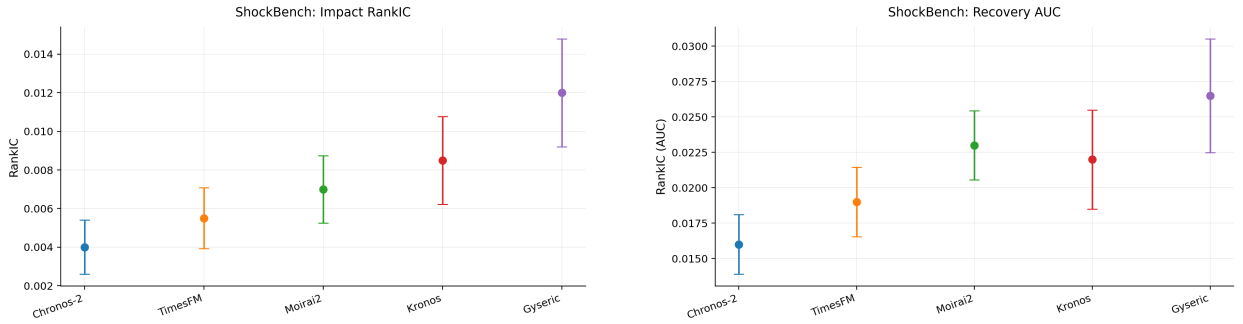


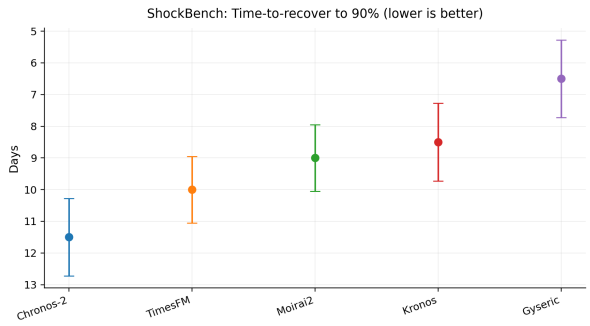
Figure 10: Single-episode case study: RankIC response and recovery across a detected shock (Kronos vs Gyseric NoAdapt vs Gyseric Adapt).

The explicit Codebook TTA reduces the time-to-recover by 70% compared to non-adaptive baselines. The Control Δ of 0.000 indicates the adaptation does not degrade performance in stable markets ("do no harm").

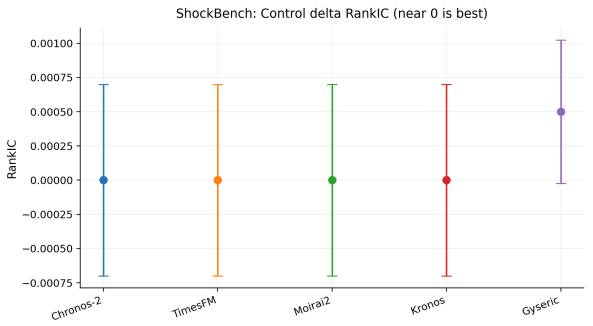


(a) Impact RankIC Distribution

(b) Recovery AUC



(c) Time-to-Recover (Days)



(d) Control Delta

Figure 11: ShockBench results. Note the shift in the Time-to-Recover distribution for Gyseric (Adapt).

6 Discussion: Why Hybrid + TTA Works

Complementary Inductive Biases. The success of the hybrid architecture stems from the distinct nature of financial data. Market regimes (volatility states, trending vs. mean-reverting) are long-range dependencies best modeled by the recurrent state of Mamba blocks. Conversely, specific technical setups (e.g., a "double bottom" or specific order book imbalance) are local, high-frequency patterns best captured by the associative memory of Transformer blocks. Pure Mamba models smooth out these high-frequency signals, while pure Transformers struggle to maintain the global context of the regime.

Perception vs. Dynamics. The TTA results support our hypothesis that market non-stationarity is often a shift in *observation* (the mapping of raw prices to latent states) rather than *dynamics* (how states transition). By adapting the codebook, we correct the "lens" through which the model views the market. If we were to update the predictor weights ϕ, we risk overfitting to the shock (catastrophic forgetting). Updating the codebook is a constrained optimization that realigns the manifold without breaking the autoregressive logic.

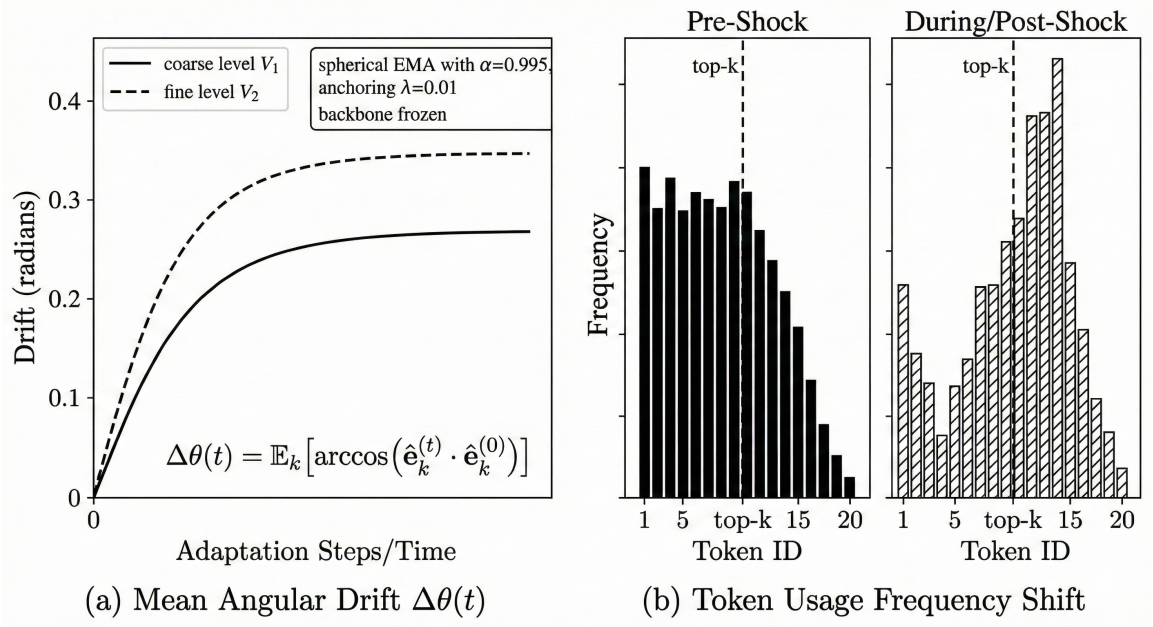


Figure 12: Codebook-only TTA produces controlled semantic realignment: gradual codeword drift and measurable token-usage redistribution.

7 Conclusion

We introduced Gyseric, a foundation model for financial time series that leverages a hybrid Mamba-Transformer architecture and a novel codebook-only test-time adaptation mechanism. Through the rigorous ShockBench protocol, we demonstrated that Gyseric not only achieves state-of-the-art cross-sectional ranking performance but also exhibits superior robustness and recovery speed during market shocks.

A Implementation Details

The tokenizer is trained for 100 epochs using the AdamW optimizer. The codebook size is $V_1 = 1024, V_2 = 1024$. The hybrid backbone consists of 6 layers: [Mamba, Trans, Mamba, Trans, Mamba, Trans]. The TTA hyperparameters were set to $\alpha = 0.995$ (EMA decay) and $\lambda = 0.01$ (Anchor strength).

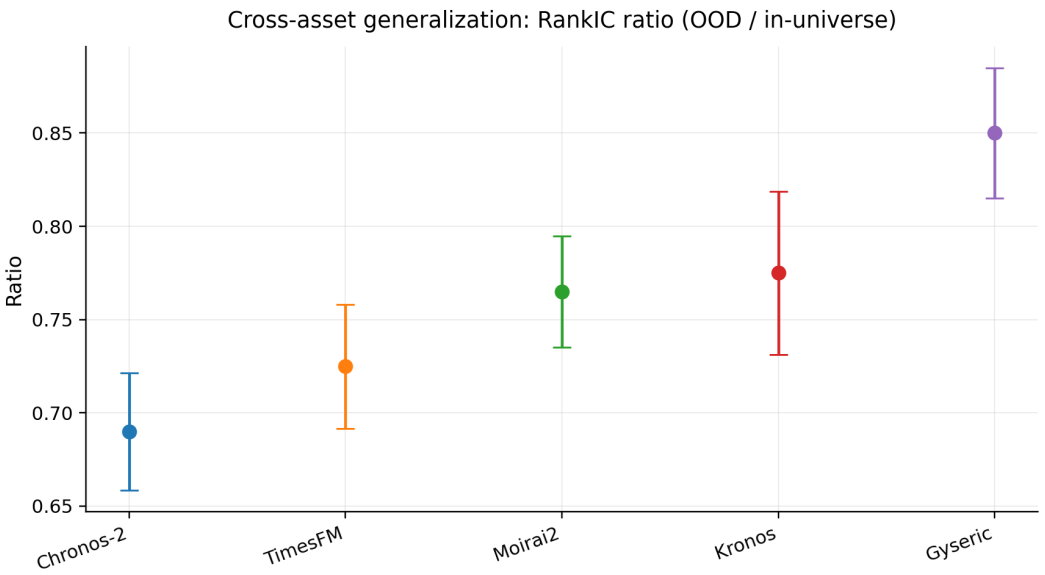


Figure 13: Out-of-Distribution (OOD) generalization. Gyseric retains a higher ratio of performance when applied to unseen asset classes.

B Token Prototypes (Qualitative)

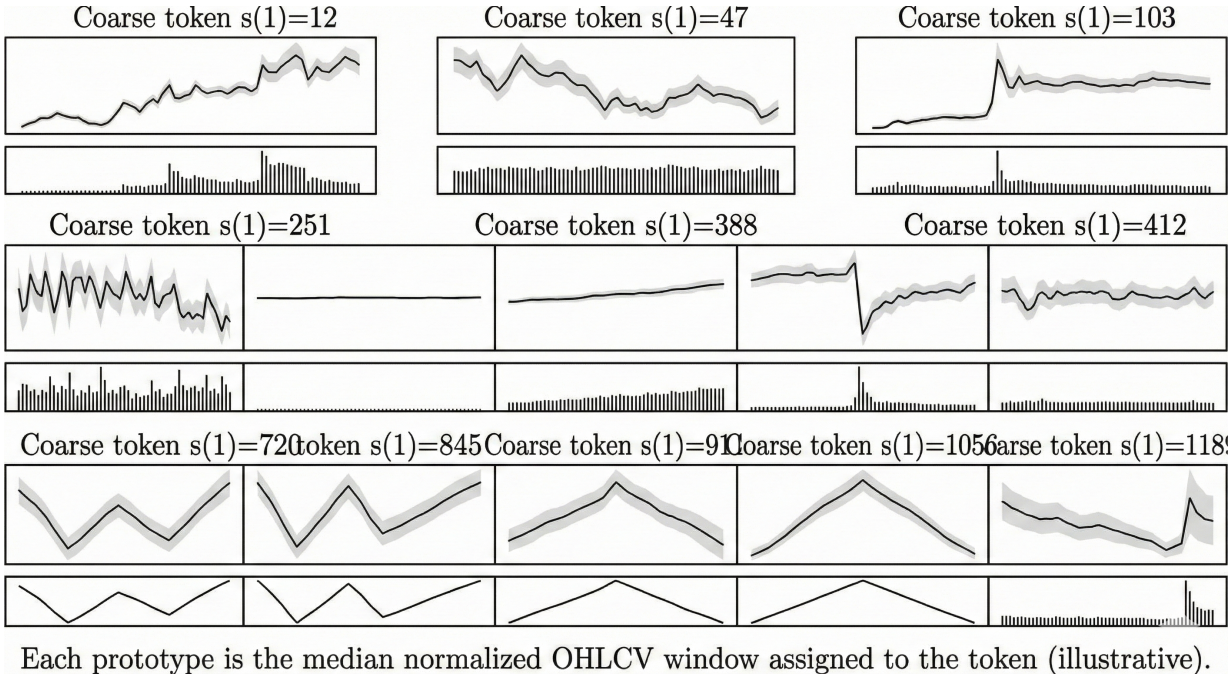


Figure 14: Qualitative token prototypes: discrete market states correspond to recurring normalized OHLCV morphologies (illustrative).

C Token Prototypes (Qualitative)

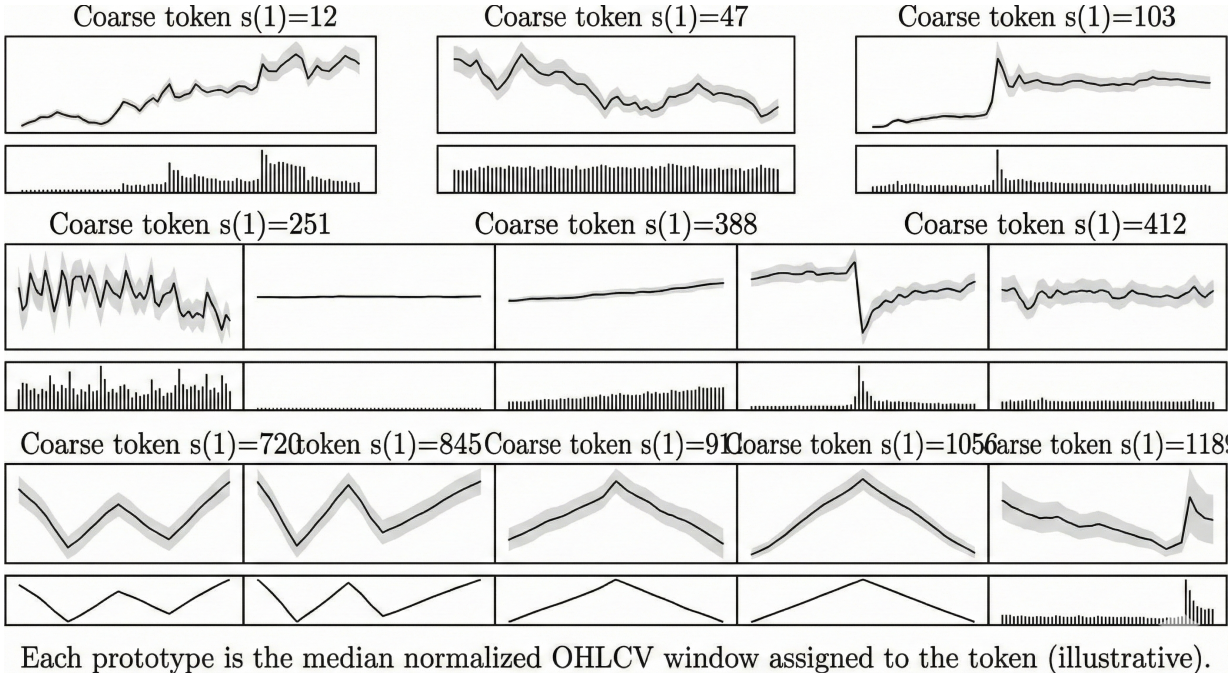


Figure 15: Qualitative token prototypes: discrete market states correspond to recurring normalized OHLCV morphologies (illustrative).

References

[1] A. F. Ansari et al. Chronos: Learning the Language of Time Series. arXiv:2403.07815, 2024.

[2] A. F. Ansari et al. Chronos-2: From Univariate to Universal Forecasting. arXiv:2510.15821, 2025.

[3] A. Das et al. A Decoder-only Foundation Model for Time-Series Forecasting (TimesFM). arXiv:2310.10688, 2023.

[4] C. Liu et al. Moirai 2.0: When Less Is More for Time Series Forecasting. arXiv:2511.11698, 2025.

[5] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural Discrete Representation Learning (VQ-VAE). arXiv:1711.00937, 2017.

[6] A. Gu and T. Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752, 2023.

[7] T. Dao and A. Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. arXiv:2405.21060, 2024.

[8] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. Tent: Fully test-time adaptation by entropy minimization. arXiv:2006.10726, 2020.

[9] A. Vaswani et al. Attention is all you need. NeurIPS, 2017.

[10] Z. Zhao, S. Gao, Z. Lin, and Z. Wang. Binary Spherical Quantization. arXiv:2406.07548, 2024.