# PERCEIVE FAST, THINK SLOW: ADAPTIVE PERIODIC TOKENIZATION FOR TIME SERIES ANALYSIS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Patch-based transformers encode raw observations into fixed patches and then rely on inter-patch attention for both dependency modeling and prediction; the same token grid must serve both purposes. PeCo-TS separates these two purposes into coexisting pathways. The Fast Path produces full-resolution timestep features with linear attention, while a spectral segmenter pools them into period-guided segment tokens for scale-adaptive dependency modeling. Temporal reprojection fuses both representations before the output head, so the prediction head never operates on compressed tokens alone. The complete design achieves best or tied-best forecasting MSE on five of seven datasets and the highest displayed PA-F1 in the MSL pathway control. The pathway controls reveal an asymmetric pattern: retaining only the full-resolution pathway trails the full model by a small margin, while retaining only the segment pathway falls far behind. This indicates that aggregation and fine-grained detail are complements, not substitutes. The results show that separating prediction-scale perception from segment-scale conceptualization—rather than forcing both purposes into the same token grid—is a viable design principle for time-series transformers.

## 1 INTRODUCTION

Tokenization determines how continuous observations are organized into the units processed by a time-series model. Patch-based Transformers encode consecutive timesteps into subsequence-level tokens, capturing local patterns while reducing sequence length and enabling longer historical contexts. PatchTST established this design and, with it, a central representational choice: what temporal span should a token cover? (Nie et al., 2023)

Fixed-length tokenization uses the same token span for every window and channel. One token can then bundle several short cycles in one channel while covering only part of a longer cycle in another, pushing the model to absorb scale differences through interactions among predefined tokens. Organizing features at a characteristic scale captures recurring structure, while amplitude, shape, and local dynamics within those temporal units carry prediction signal. The design problem is to adapt token spans to input structure while retaining fine-grained representations for the prediction head. Existing work addresses these objectives through multiscale modeling and adaptive tokenization. TimeMixer decomposes and mixes representations at different sampling scales; Pathformer uses multiple patch sizes and adaptive processing pathways (Wang et al., 2024a; Chen et al., 2024). LightGTS uses FFT-derived periods with per-variable processing; PATK constructs elastic tokens from temporal and spectral variation; Kairos selects local granularities with dynamic patching (Wang et al., 2025b; Liao et al., 2026; Feng et al., 2025). These methods establish adaptive spans as a design variable; PeCo-TS also preserves a separate timestep representation for the task head. We therefore evaluate the temporal units and the two retained representations through configuration-level controls.

Cognitive research on complementary processing modes and temporal event organization motivates the division of roles (Kahneman, 2011; Booch et al., 2021; Zacks et al., 2007). The architectural principle is to construct fine-grained contextual features, organize them into temporal abstractions, and combine both representations for prediction (Figure 1). "Perceive Fast, Think Slow" supplies the organizational analogy for the two resolutions; technical claims are evaluated by the reported controls.

**PeCo-TS** (Perception–Concept Transformer for Time Series; Figure 2) implements this principle through adaptive periodic tokenization: input-dependent, period-guided aggregation of contextualized timestep features into segment tokens. Its **Fast Path** uses linear attention to produce contextualized



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

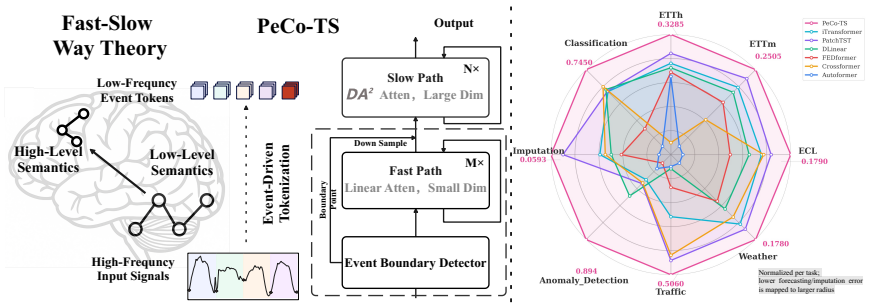



Figure 1: **PeCo-TS overview.** Left: the Fast/Slow architectural analogy and the PeCo-TS processing pathways. Right: a normalized visual summary across eight comparison axes; each axis uses its own task or dataset metric, is normalized within that axis, and a larger radius indicates better performance (lower forecasting and imputation errors map to a larger radius). Exact task protocols and values are reported in the corresponding tables.

features at timestep resolution. A spectral segmenter estimates an effective period per window and channel and pools these features into variable-span segment tokens, each encoding context gathered before pooling. The **Slow Path** then applies **dual-axis adaptive attention (DA$^2$)** to model segment-order and cross-channel dependencies among the tokens. Temporal reprojection returns their context to timestep coordinates for fusion with unpooled Fast Path features before the output head, so the Fast Path supplies both the content used to construct tokens and the fine-grained representation used for prediction.

We test this design through periodic-versus-fixed tokenization, pathway controls, and forecasting, classification, imputation, and anomaly-detection benchmarks. PeCo-TS achieves the best or tied-best mean forecasting MSE on five of seven datasets against ten baselines (Table 1). The evaluation separates three design consequences:

- **Input-dependent periodic spans replace a shared fixed temporal resolution with a per-window, per-channel one.** Replacing the spectral segmenter with fixed patches of length $\{8, 16, 32, 64\}$ raises mean forecasting MSE on ETTh1, ETTm1, Electricity, Exchange, and Traffic while the task head, split, loss, and backbone are held fixed.
- **Contextualizing before aggregation is paired with a retained full-resolution path.** The Fast Path supplies both the features pooled into segment tokens and an unpooled representation for the prediction head; in the pathway controls, Fast-only remains closer to Full than Slow-only, whose gap is much larger.
- **Temporal reprojection makes the two resolutions a fused predictor rather than competing alternatives.** Segment context is returned to timestep coordinates and combined with the unpooled features; the complete configuration attains best or tied-best mean forecasting MSE on five of seven datasets and the highest displayed PA-F1 on four of five anomaly datasets.

These controls compare complete configurations. The span control varies token length, token count, capacity, and reprojection weights together, while pathway removal varies token construction and final fusion. The results describe complete operating points and the combined effect of each design choice.

The advantage concentrates in series with channel-varying periodic structure. Weather and ETTm2 define the single-scale limit: diffuse or multi-frequency spectra reduce the periodic-versus-fixed gain, and classification or imputation accuracy tracks how closely the dominant frequency matches the learned span.

## 2 Related Work

**Fixed-length and adaptive tokenizers.** PatchTST encodes fixed-length subsequences within a channel-independent framework (Nie et al., 2023). TimeMixer decomposes multiscale series; Pathformer uses multiple patch sizes and adaptive pathways (Wang et al., 2024a; Chen et al., 2024);

LightGTS, PATK, and Kairos vary token construction with the input (Wang et al., 2025b; Liao et al., 2026; Feng et al., 2025). These methods change the token grid; PeCo-TS adds a head-level timestep view, constructing segment tokens *after* timestep-level contextualization and fusing their reprojected context with the unpooled representation.

**Multiscale representations.** MTST, TimeMixer++, and Pathformer construct representations at different granularities (Zhang et al., 2024; Wang et al., 2025a; Chen et al., 2024). PeCo-TS connects its pathways through a direct feature dependency: Slow Path tokens are built from Fast Path features, and their context is reprojected to timestep coordinates for fusion. TimeMixer++ applies dual-axis attention within time images; $DA^2$ operates along segment ordinals and channels on the variable-span token grid.

**Periodicity-aware models.** TimesNet and FEDformer select temporal scales from spectral structure (Wu et al., 2023; Zhou et al., 2022). MSGNet, PHAT, and CAMP model cycle-aware dependencies (Cai et al., 2024; Ma et al., 2026; Choi et al., 2026). iTransformer, CARD, and Crossformer organize dependencies on transformed representations (Liu et al., 2024; Wang et al., 2024b; Zhang & Yan, 2023). PeCo-TS uses periodicity to set token spans and reprojection weights, then applies $DA^2$ on the resulting segment grid. Paragraphs on cognitively inspired processing and general-purpose models appear in Appendix E.

# 3 Methodology

## 3.1 Overview and Design Principles

PeCo-TS separates aggregation span from retained timestep representation: the Fast Path contextualizes each timestep before pooling, a spectral branch selects input-dependent spans, and the Slow Path models dependencies among the resulting tokens; reprojection returns segment context to timestep coordinates for fusion with the unpooled features (Figure 2).

This separation is motivated by a tension in the standard patch-based pipeline. A patch token serves two purposes: it carries information that feeds inter-token attention (dependency modeling), and it carries information that the prediction head uses to reconstruct or classify individual timesteps. A fixed patch balances compact long-range attention against fine-grained prediction. PeCo-TS gives each purpose its own representation: segment tokens for scale-adaptive dependency modeling, and unpooled timestep features for the prediction head.

Let $x \in \mathbb{R}^{B\times L\times C}$ denote $B$ samples, $L$ timesteps, and $C$ channels, with coordinates $n$, $c$, and $t \in \{0,\dots,L-1\}$. The Fast Path extracts timestep features with linear attention at $O(L)$ cost, while a parallel spectral branch selects spans for pooling. Segment tokens enter the Slow Path for dependency modeling; reprojection returns their context to timestep coordinates for fusion with the unpooled Fast Path representation.

## 3.2 Adaptive Periodic Tokenization

PeCo-TS treats the aggregation span as input-dependent and evaluates it against fixed spans. It estimates one effective period per sample and channel via a softmax-weighted spectral centroid, using that period to pool features. The resulting spans specify an aggregation scale, while the unpooled timestep representation preserves within-span detail.

**Full-Resolution Perception.** The Fast Path produces a contextualized feature at every timestep before aggregation, so the same representation serves both segment construction and prediction. Scalar observations are embedded at width $d_f$ and processed per channel with positional encodings and linear attention. For one head with projections $Q$, $K$, $V$,

$$h_{\text{attn}} = \frac{\phi(Q)\big(\phi(K)^\top V\big)}{\phi(Q)\big(\phi(K)^\top \mathbf{1}_L\big)}, \qquad \phi(u) = \text{elu}(u) + 1. \tag{1}$$

Reassociating costs $O(Ld_f^2)$. The output $h_{\text{fast}} \in \mathbb{R}^{B\times C\times L\times d_f}$ is projected to $h_{\text{high}} \in \mathbb{R}^{B\times C\times L\times d_h}$. Because $h_{\text{high}}$ is formed before pooling, a segment token can summarize within-span positions using context gathered elsewhere in the window, while the same tensor remains available unpooled. Appendix F.2 gives the kernel identity.

**Input-Dependent Temporal Spans.** For each sample and channel, the spectral branch smooths the power spectrum and takes a softmax-weighted frequency centroid. The resulting period sets a cosine-comb proposal, and NMS converts it into hard boundaries (Algorithm 1). The period is

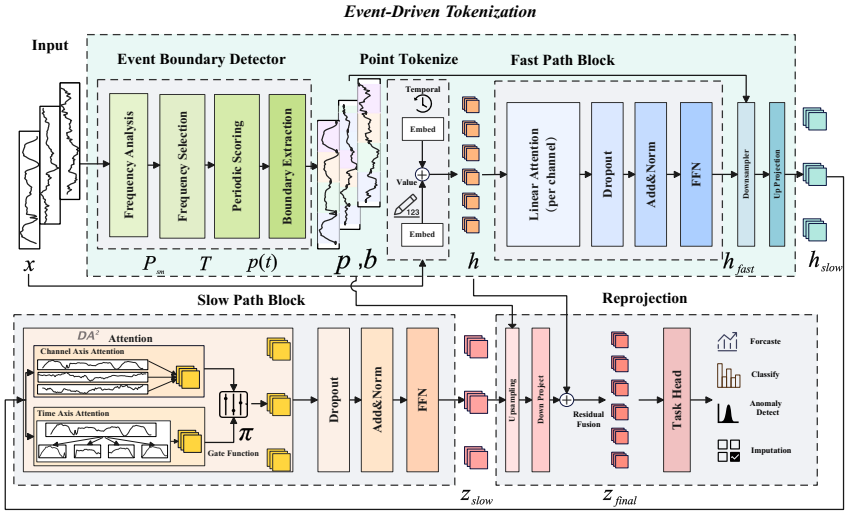



Figure 2: **PeCo-TS framework.** The spectral branch implements the periodic segmenter in Algorithm 1 and emits periodic aggregation proposals. The Fast Path produces full-resolution features, and the segmenter pools them into segment tokens. The Slow Path applies DA$^2$ to these tokens, and temporal reprojection returns segment context to timestep resolution for fusion with the unpooled features.

differentiable in the smoothed spectrum; this feeds the reprojection gradient while the partition stays fixed.

**Algorithm 1** Input-dependent Periodic Segmentation

**Input:** $x \in \mathbb{R}^{B\times L\times C}$; positive-frequency support $\mathcal{F}_L = \{1, \dots, \lfloor L/2 \rfloor\}$.
**Settings:** learnable spectral kernel $g_\theta$, temperature $\tau = 0.5$, sharpness $\gamma = 2$, NMS ratio $r_{\text{nms}} = 0.03$, threshold $\tau_{\text{seg}} = 0.5$.
**for** each sample $n$ and channel $c$ **do**
  $P(f) \leftarrow |\text{rFFT}(x_{n,:,c})(f)|^2$; $P_{\text{sm}} \leftarrow g_\theta * P$.
  $a_f \leftarrow \exp(P_{\text{sm}}(f)/\tau) / \sum_{u\in\mathcal{F}_L} \exp(P_{\text{sm}}(u)/\tau)$, $f \in \mathcal{F}_L$.
  $k_{\text{eff},n,c} \leftarrow \sum_{f\in\mathcal{F}_L} a_f f$; $T_{n,c} \leftarrow L/k_{\text{eff},n,c}$.
  $p_{n,c}(t) \leftarrow \big((1 + \cos(2\pi t/T_{n,c}))/2\big)^\gamma$, $t = 0, \dots, L-1$.
  $w_{\text{nms},n,c} \leftarrow \max(3, \lfloor r_{\text{nms}} T_{n,c} \rfloor)$.
  $b_{n,c} \leftarrow \text{NMS}(p_{n,c}, w_{\text{nms},n,c}, \tau_{\text{seg}})$.
**end for**
**Gradient path:** The spectral kernel $g_\theta$ reaches the task loss through the reprojection weights while the integer partition is held fixed.
**return** periodic weights $p_{n,c}$ and boundary indicators $b_{n,c}$.

Normalization excludes DC and conjugate negative frequencies, so $k_{\text{eff},n,c} \geq 1$ and $T_{n,c}$ is finite. Adaptation occurs across windows and channels, while proposals remain periodic within each window. A diffuse spectrum produces one effective scale (Appendix F.3) that summarizes its frequency distribution. Within-segment variation remains accessible through the unpooled Fast Path representation. The spectral branch reads the observed window directly, so span selection depends on the input spectrum and the spectral kernel.

**Token Construction and Feature Retention.** Hard boundaries turn the timestep features into segment tokens. Ordered, distinct interior boundaries plus the endpoints 0 and $L$ partition each sequence into nonempty half-open segments $S_{n,c,i} = [q_{i-1}, q_i)$, $i = 1, \dots, M_{n,c}$; if no interior boundary is proposed, the whole window forms one segment. With assignment indicator $A_{n,c}(t, i) =$

$\mathbf{1}[t \in S_{n,c,i}]$, mean pooling gives

$$h_{\text{slow}}(n,c,i) = \frac{\sum_{t=0}^{L-1} A_{n,c}(t,i)\, h_{\text{high}}(n,c,t)}{|S_{n,c,i}|}. \tag{2}$$

Each token aggregates contextualized features over its selected span, so its content already reflects the window context that entered the Fast Path. Because $M_{n,c}$ varies across samples and channels, we pad to $M = \max_{n,c} M_{n,c}$, giving shape $B \times C \times M \times d_h$, and define $m(n,c,i) = 1$ for an existing segment and 0 for padding. The unpooled $h_{\text{high}}$ bypasses this compression and remains available in Equation 8.

**Tokenization Gradients under a Fixed Partition.** NMS produces a discrete partition; backpropagation holds it fixed. The segment-mean weight $\bar{p}_{n,c,i}$ enters the reprojection weights in Equation 7, giving a differentiable route from spectral kernel $g_\theta$ to the task loss (Appendix F.3). Per-channel segmentation gives different boundaries and counts. We use ordinal $i$ as a grouping key: temporal attention follows segment order within a channel, and channel attention connects tokens with the same ordinal across channels.

### 3.3 Slow Path: Dependencies over Adaptive Tokens

DA$^2$ combines two attention axes on the segment grid: temporal attention follows segment order within a channel, and channel attention connects tokens with the same ordinal across channels.

Let $H$ be the number of heads and $d_a = d_h/H$ the head width, with $H$ dividing $d_h$. For one head,

$$\text{Attn}(Q,K,V;m) = \text{softmax}_m\left(\frac{QK^\top}{\sqrt{d_a}}\right)V. \tag{3}$$

The masked softmax normalizes over valid keys; empty key sets and padded queries have zero output (Appendix F.4). Let $\mathcal{A}_{\text{time}}$ and $\mathcal{A}_{\text{chan}}$ denote the corresponding multi-head operators with feature projections:

$$\tilde{z}_{\text{time}}(n,c,:,:) = \mathcal{A}_{\text{time}}\big(h_{\text{slow}}(n,c,:,:); m(n,c,:)\big), \tag{4}$$

$$\tilde{z}_{\text{chan}}(n,:,i,:) = \mathcal{A}_{\text{chan}}\big(h_{\text{slow}}(n,:,i,:); m(n,:,i)\big). \tag{5}$$

A trainable layerwise scalar $\eta$ sets $\pi = \text{sigmoid}(\eta)$ to combine the two contexts:

$$Y = \pi\tilde{z}_{\text{chan}} + (1-\pi)\tilde{z}_{\text{time}} \in \mathbb{R}^{B\times C\times M\times d_h}. \tag{6}$$

The gate is shared across inputs within a layer. Residual and feedforward transformations of $Y$, repeated over the Slow Path layers, produce $z_{\text{slow}}$. Appendix C.2 compares learned and fixed mixtures with single-axis attention.

### 3.4 Temporal Reprojection and Detail Fusion

A single interpolation bandwidth treats unequal segments uniformly, over-smoothing short ones; we instead use a Gaussian whose width scales with segment length.

**Segment-scaled kernel.** Let $\mu_{n,c,i}$, $\bar{p}_{n,c,i}$, and $\sigma_{n,c,i} = \alpha|S_{n,c,i}|$ (with $\alpha = 0.5$) denote the mean coordinate, mean periodic weight, and width of $S_{n,c,i}$. Suppressing $n, c$,

$$\tilde{w}_{t,i} = \frac{\bar{p}_i \exp\left(-\frac{(t-\mu_i)^2}{2\sigma_i^2}\right)}{\sum_{j=1}^{M_{n,c}} \bar{p}_j \exp\left(-\frac{(t-\mu_j)^2}{2\sigma_j^2}\right)}, \qquad h_{\text{full}}(n,c,t) = \sum_i \tilde{w}_{t,i} z_{\text{slow}}(n,c,i). \tag{7}$$

The reprojected $h_{\text{full}}$ is a convex mixture of segment representations at each timestep; $\alpha$ controls the mixing (Appendix F.6).

**Detail fusion and task heads.**

$$h_{\text{final}}(n,c,t) = W_{\text{out}} h_{\text{full}}(n,c,t) + W_{\text{res}} h_{\text{high}}(n,c,t). \tag{8}$$

Both projections map $d_h$ to $d_f$. Forecasting, classification, imputation, and anomaly-detection heads use this fused representation at timestep resolution. The task loss trains all continuous parameters; the tokenization gradient path follows Section 3.2.

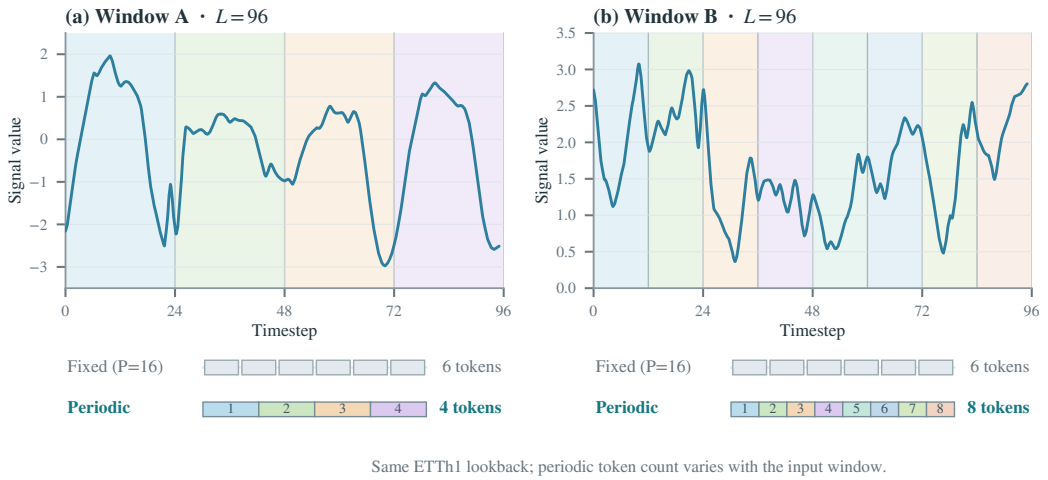



Figure 3: **Input-dependent periodic spans and a fixed-16 reference.** Two ETTh1 windows with the same lookback receive four and eight periodic segments, while fixed-16 patching gives six tokens in both. The lower rows show the resulting token counts; colors index displayed segments.

**Computational cost.** At padded segment count $M$, the Slow Path costs $O\big((CM^2 + MC^2)d_h\big)$, the Fast Path costs $O(CLd_f^2)$, and reprojection costs $O(CLMd_h)$. Appendix F.1 gives the full accounting, and Appendix D reports measured latency profiles.

## 4 Experiments

**Evaluation setting.** Forecasting follows established long-term forecasting protocols (Lai et al., 2018; Zhou et al., 2021; Wu et al., 2021) on ETTh1, ETTm1, ETTm2, Electricity (Trindade, 2015), Exchange, Traffic, and Weather (for Biogeochemistry, 2017). We use lookback $L = 96$ and prediction lengths $S \in \{96, 192, 336, 720\}$. The forecasting comparison contains AMD, PathFormer, CARD, UniTS, iTransformer, PatchTST, TSMixer, TimesNet, Mamba, and DLinear (Hu et al., 2025; Chen et al., 2024; Wang et al., 2024b; Gao et al., 2024; Liu et al., 2024; Nie et al., 2023; Chen et al., 2023; Wu et al., 2023; Gu & Dao, 2023; Zeng et al., 2023). We also evaluate classification on seven UCR/UEA datasets (Dau et al., 2019; Bagnall et al., 2018), imputation under random missingness, and point-adjusted anomaly detection on MSL, SMAP, PSM, SMD, and SWaT (Xu et al., 2022; Hundman et al., 2018; Abdulaal et al., 2021; Su et al., 2019; Goh et al., 2017). Complete protocols and tables are in Appendix A. The tables report aggregate values within each task protocol, and comparisons use their displayed ordering.

We ask three questions: how do periodic and fixed spans organize temporal units (RQ1); how do pathway variants and internal Slow Path choices affect the reported metrics and computation (RQ2); and how does the complete model compare with baselines under the separate task protocols (RQ3)?

### 4.1 Periodic spans versus fixed grids

We test temporal units by replacing periodic spans with fixed patches of length $\{8, 16, 32, 64\}$, holding head, split, loss, and backbone fixed. Figure 3 shows the mechanism on two ETTh1 windows with the same lookback: periodic segmentation produces four and eight tokens, while the fixed-16 reference always produces six. Appendix B.1 presents graphical forecasting comparisons; we summarize their pattern here. Span length, token count, and segment capacity vary together. The periodic weights also enter reprojection, so the comparison tests the full tokenization path.

Boundary perturbations on ETTh1 raise MSE by $0.22$–$2.27\%$ under $20\%$ jitter and $0.47$–$1.32\%$ under complete randomization (Appendix B.2).

Fixed spans apply one uniform grid across channels, a mismatch most pronounced on ETTh1 and Electricity where sensor readings follow distinct seasonal patterns. Visual illustrations appear in Appendix B.4.

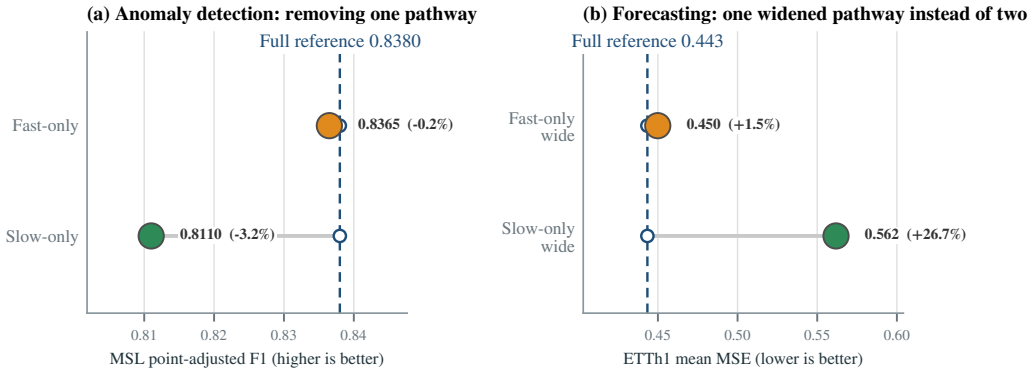



Figure 4: **Pathway controls.** (a) MSL anomaly detection: Full, Fast-only, and Slow-only, measured by PA-F1 (higher is better). (b) ETTh1 forecasting: Full and the two widened single-pathway variants, measured by mean MSE (lower is better). Points and connecting lines show change relative to Full; the panels use separate protocols and metrics.

Table 1: **PeCo-TS across forecasting and additional time-series tasks.** Each cell pairs PeCo-TS with the best displayed baseline on that dataset under the task metric in Appendices A.2, A.5, A.4, and A.3; baseline methods are named in those tables. Forecasting averages MSE over prediction lengths $S \in \{96, 192, 336, 720\}$ at lookback $L = 96$; imputation averages MSE over the four random-missingness ratios; anomaly detection reports PA-F1 under point adjustment; classification reports accuracy. **Bold** marks the better of the two displayed values. Superscripts give PeCo-TS's rank in the corresponding appendix table (eleven forecasting methods; seven methods in each additional-task comparison), with = marking a tie.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Forecast.** | | ETTh1 | ETTm1 | ETTm2 | Electricity | Exchange | Traffic | Weather |
| | PeCo-TS | **0.4145**$^{(1)}$ | **0.3695**$^{(1)}$ | 0.2873$^{(2)}$ | **0.2010**$^{(1)}$ | **0.3193**$^{(1)}$ | **0.5258**$^{(=1)}$ | 0.2600$^{(3)}$ |
| | Best baseline | 0.4350 | 0.3855 | **0.2853** | 0.2090 | 0.3390 | **0.5258** | **0.2430** |
| **Imput.** | | ETTh1 | ETTh2 | ETTm1 | ETTm2 | Electricity | Weather | |
| | PeCo-TS | 0.1366$^{(4)}$ | 0.0721$^{(2)}$ | **0.0426**$^{(1)}$ | 0.0293$^{(2)}$ | **0.0625**$^{(1)}$ | **0.0321**$^{(1)}$ | |
| | Best baseline | **0.1146** | **0.0650** | 0.0451 | **0.0291** | 0.0687 | 0.0331 | |
| **Anomaly** | | MSL | PSM | SMAP | SMD | SWaT | | |
| | PeCo-TS | **0.8380**$^{(1)}$ | 0.9626$^{(=2)}$ | **0.8615**$^{(1)}$ | **0.8482**$^{(1)}$ | **0.9635**$^{(1)}$ | | |
| | Best baseline | 0.8230 | **0.9661** | 0.7411 | 0.8104 | 0.9266 | | |
| **Classif.** | | Ethanol-Conc. | Face Det. | Japanese Vowels | SelfReg. SCP1 | SelfReg. SCP2 | Spoken Arab. | UWave Gesture |
| | PeCo-TS | **0.3270**$^{(1)}$ | 0.6831$^{(2)}$ | 0.9676$^{(3)}$ | 0.9144$^{(4)}$ | **0.5560**$^{(1)}$ | **0.9818**$^{(1)}$ | 0.7919$^{(5)}$ |
| | Best baseline | 0.3030 | **0.6864** | **0.9757** | **0.9215** | 0.5467 | 0.9804 | **0.8625** |

### 4.2 Pathway and Internal Representation Controls

Pathway removal compares variants retaining segment context, unpooled detail, or both. On MSL, Full reaches PA-F1 0.8380, Fast-only 0.8365, and Slow-only 0.8110; on ETTh1, mean MSE is 0.4434 for Full, 0.4499 for wide Fast-only, and 0.5618 for wide Slow-only (Figure 4). Full leads in each comparison, and Fast-only stays closer to Full than Slow-only. The full-resolution pathway carries the primary representational power, while segment-level dependencies provide complementary context.

Slow Path ablations include learned and fixed temporal–channel mixtures and single-axis attention (Appendix C.2).

### 4.3 FORECASTING AND ADDITIONAL TIME-SERIES TASKS

The benchmarks test the complete design across four tasks (Table 1). PeCo-TS attains the best or tied-best mean forecasting MSE on five of seven datasets and the highest displayed PA-F1 on four of five anomaly datasets. Forecasting wins combine consistent leads at short and long horizons, indicating that the retained timestep detail supports the head across scales. Anomaly-detection margins are largest on MSL, SMAP, SMD, and SWaT—datasets where periodic episodes are visually salient—suggesting that adaptive segmentation aligns the Slow Path with the characteristic temporal scale. Classification and imputation show a more dataset-dependent pattern, consistent with the single-scale segmenter assumption. Complete comparisons appear in Appendix A.

Horizon-level patterns confirm the retained timestep representation supports prediction across scales. PeCo-TS leads all four prediction lengths on ETTh1, ETTm1, and Exchange, and three of four on Electricity. The absolute PA-F1 anomaly gains range from $0.015$ to $0.120$, and forecasting MSE reductions range from $0.008$ to $0.021$. These magnitudes are consistent with the pathway controls: the Fast Path supplies most of the improvement, and the Slow Path adds a smaller but measurable gain. Forecasting and anomaly detection, where periodic episodes are salient, benefit from both pathways; classification, which requires one global decision, relies more heavily on segment-level structure; and imputation, where missing entries fragment periodic structure, shows mixed results.

**Operating regime.** PeCo-TS targets series with channel-specific periodic structure that fixed spans misalign. Its advantage is largest when the characteristic scale changes across channels or windows, which is where adaptive tokenization changes the organization of the representation.

**Internal representation controls.** Within the segment pathway, the learned temporal–channel mixture matches the performance of fixed mixtures and single-axis attention in the recorded settings. The segment-proportional reprojection weights attain their lowest displayed MSE at $\alpha = 0.5$ on ETTh1, Weather, and Traffic ($0.365$, $0.177$, and $0.502$, respectively). On ETTh1, the fixed-width endpoints at $\sigma = 0.3$ and $2.0$ give MSE $0.389$ and $0.394$. These comparisons validate the selected internal operating points within the recorded variants.

**Retained-detail cost.** Periodic pooling reduces the mean token count to $4.3$–$6.5\%$ of the lookback in the ETTh1 profile, while the Fast Path dominates measured cost. At $L = 1536$, width $8$ uses $14.7$ ms and reaches MSE $0.427$, while width $16$ uses $23.0$ ms and reaches $0.405$; width $4$ uses $16.6$ ms and reaches $0.400$. The width sweep therefore defines model-side latency–accuracy operating points; compression and Fast Path width together determine the end-to-end trade-off.

## 5 CONCLUSION

PeCo-TS tests whether input-dependent periodic tokenization combined with a retained full-resolution pathway is a useful alternative to regular-grid patching across time-series tasks. The complete design achieves the best or tied-best mean forecasting MSE on five of seven datasets and first-ranked PA-F1 on four of five anomaly benchmarks.

Three lines of evidence define the result. First, periodic spans give the segment pathway an adaptive vocabulary: replacing them with fixed patches of length $\{8, 16, 32, 64\}$ raises MSE on ETTh1, ETTm1, Electricity, Exchange, and Traffic. Second, aggregation and full-resolution features are complements, not substitutes: pathway controls show that Fast-only—retaining only the full-resolution pathway—trails Full by a small margin, while Slow-only—using only segment tokens—falls far behind. Third, the same representation design performs across four task families, with its strongest results on series whose periodic structure is channel-varying and predictive.

More broadly, the results position tokenization in time-series Transformers as an interface between abstraction and evidence. Segment tokens provide the units for long-range dependency modeling at an input-derived scale; the unpooled timestep stream keeps the evidence needed for prediction. This division is the transferable design principle: learn the temporal units used for organization while the output draws on both compressed and uncompressed views. The current model realizes that interface with one effective period per window. Extending it to multiple competing scales would turn scale selection into a learned temporal hierarchy. Such a model would adapt abstraction to both the dominant scale and the interactions among several scales within one channel, making periodicity a learned representation structure rather than a preprocessing choice. The evidence-preserving interface would remain the organizing principle as each added scale expands the model's temporal vocabulary while keeping timestep-level evidence available for prediction, turning the number and relation of temporal resolutions into part of the learned representation.

## AI USE STATEMENT

Generative AI tools, including Kimi K2.6, assisted with language editing, narrative revision, mathematical and bibliographic review, and figure reformatting from existing data and image panels. AI assistance was also used in earlier figure preparation and experimental planning. The ETTh1 segmentation visualization reuses existing colored panels and adds token-count comparisons with a fixed-16 reference. The authors are responsible for the manuscript and its experimental claims.

## ETHICS STATEMENT

The study uses public time-series benchmarks. The proposed architecture is evaluated for forecasting, classification, imputation, and anomaly detection; deployment decisions require evaluation in the intended application setting.

## REPRODUCIBILITY STATEMENT

Algorithm 1 specifies the periodic proposal; the method section defines pooling, dual-axis attention, and reprojection; the appendix provides per-task result tables and experimental settings. An anonymized implementation repository is available at https://anonymous.4open.science/r/PeCO-TS-Code-102C. The artifact records source panels, displayed-value transcriptions, recovered aggregate CSV files, and rendering procedures. The reported tables and figures use these displayed aggregates.

## REFERENCES

Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 2485–2494, 2021. doi: 10.1145/3447548.3467174. URL https://dl.acm.org/doi/10.1145/3447548.3467174.

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. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=gerNCVqqtR.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Grady Booch, Francesco Fabiano, Lior Horesh, Kiran Kate, Jon Lenchner, Nick Linck, Andrea Loreggia, Keerthiram Murugesan, Nicholas Mattei, Francesca Rossi, and Biplav Srivastava. Thinking fast and slow in ai. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 15042–15046, 2021. doi: 10.1609/aaai.v35i17.17765. URL https://ojs.aaai.org/index.php/AAAI/article/view/17765.

Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. MSGNet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pp. 11141–11149, 2024. doi: 10.1609/aaai.v38i10.28991. URL https://ojs.aaai.org/index.php/AAAI/article/view/28991.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2402.05956.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6571–6583, 2018.

Si-An Chen, Chun-Liang Li, Sercan Ö. Arik, Nathanael C. Yoder, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=wbpxTuXgm0.

Jung Min Choi, Vijaya Krishna Yalavarthi, and Lars Schmidt-Thieme. CAMP: A cycle-aware multi-scale patch mixer for time series forecasting. *arXiv preprint arXiv:2608.04051*, 2026. doi: 10.48550/arXiv.2608.04051. URL https://arxiv.org/abs/2608.04051.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019. doi: 10.1109/JAS.2019.1911747. URL https://doi.org/10.1109/JAS.2019.1911747.

Kun Feng, Shaocheng Lan, Yuchen Fang, Wenchao He, Sihan Lu, Shuqi Gu, Lintao Ma, Xingyu Lu, and Kan Ren. Kairos: Toward adaptive and parameter-efficient time series foundation models. *arXiv preprint arXiv:2509.25826*, 2025. URL https://arxiv.org/abs/2509.25826.

Max Planck Institute for Biogeochemistry. Mpi-bgc jena weather data. Dataset, 2017. URL https://www.bgc-jena.mpg.de/wetter/.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: A unified multi-task time series model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2403.00131.

Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems. In *Critical Information Infrastructures Security*, volume 10242 of *Lecture Notes in Computer Science*, pp. 88–99. Springer, 2017. doi: 10.1007/978-3-319-71368-7_8. URL https://doi.org/10.1007/978-3-319-71368-7_8.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Yifan Hu, Peiyuan Liu, Peng Zhu, Dawei Cheng, and Tao Dai. Adaptive multi-scale decomposition framework for time series forecasting. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 17359–17367, 2025. doi: 10.1609/aaai.v39i16.33908. URL https://ojs.aaai.org/index.php/AAAI/article/view/33908.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 387–395, 2018. doi: 10.1145/3219819.3219845. URL https://doi.org/10.1145/3219819.3219845.

Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning (ICML)*, pp. 5156–5165. PMLR, 2020.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104, 2018. doi: 10.1145/3209978.3210006. URL https://doi.org/10.1145/3209978.3210006.

Huaizhang Liao, Zhixiong Yang, Jingyuan Xia, Yuheng Sun, Yue Zhang, Shengxi Li, and Yongxiang Liu. Dynamic semantic tokenization for time series via elastic sampling on physics-aware perception. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 23460–23468, 2026. doi: 10.1609/aaai.v40i28.39517. URL https://ojs.aaai.org/index.php/AAAI/article/view/39517.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Nicholas R Lomb. Least-squares frequency analysis of unequally spaced data. *Astrophysics and Space Science*, 39(2):447–462, 1976.

Jiaming Ma, Qihe Huang, Haofeng Ma, Guanjun Wang, Sheng Huang, Zhengyang Zhou, Pengkun Wang, Xu Wang, Binwu Wang, and Yang Wang. PHAT: Modeling period heterogeneity for multivariate time series forecasting. In *Proceedings of the 14th International Conference on Learning Representations (ICLR)*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/dbb5180957513805ebeea787b8c66ac9-Abstract-Conference.html.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*, Kigali, Rwanda, 2023. OpenReview.net. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ODEs for irregularly-sampled time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5320–5330, 2019.

Jeffrey D Scargle. Studies in astronomical time series analysis. ii. statistical aspects of spectral analysis of unevenly spaced data. *The Astrophysical Journal*, 263:835–853, 1982.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 2828–2837, 2019. doi: 10.1145/3292500.3330672. URL https://doi.org/10.1145/3292500.3330672.

Xiaoyu Tao, Shilong Zhang, Mingyue Cheng, Daoyu Wang, Tingyue Pan, Bokai Pan, Changqing Zhang, and Shijin Wang. From values to tokens: An llm-driven framework for context-aware time series forecasting via symbolic discretization. *arXiv preprint arXiv:2508.09191*, 2025.

A. Trindade. Electricityloaddiagrams20112014 data set. UCI Machine Learning Repository, 2015. URL https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014. Donated 2015-03-12.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024a. URL https://openreview.net/forum?id=7oLshfEIC2.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. TimeMixer++: A general time series pattern machine for universal predictive analysis. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=1CLzLXSFNn.

Xue Wang, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. Card: Channel aligned robust blend transformer for time series forecasting. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024b. URL https://arxiv.org/abs/2305.12095.

Yihang Wang, Yuying Qiu, Peng Chen, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. LightGTS: A lightweight general time series forecasting model. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 64109–64126, 2025b. URL https://proceedings.mlr.press/v267/wang25ch.html.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 22419–22430, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*, Kigali, Rwanda, 2023. OpenReview.net. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LzQQ89U1qm_.

Janghoon Yang. TOKON: Tokenization-optimized normalization for time series analysis with a large language model. *arXiv preprint arXiv:2502.05701*, 2025. URL https://arxiv.org/abs/2502.05701.

Jeffrey M. Zacks, Nicole K. Speer, Khena M. Swallow, Todd S. Braver, and Jeremy R. Reynolds. Event perception: A mind/brain perspective. *Psychological Bulletin*, 133(2):273–293, 2007. doi: 10.1037/0033-2909.133.2.273.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI)*, pp. 11121–11128. AAAI Press, 2023. doi: 10.1609/aaai.v37i9.26317. URL https://doi.org/10.1609/aaai.v37i9.26317.

Yitian Zhang, Liheng Ma, Soumyasundar Pal, Yingxue Zhang, and Mark Coates. Multi-resolution time-series transformer for long-term forecasting. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4222–4230, 2024. URL https://proceedings.mlr.press/v238/zhang24l.html.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325. URL https://ojs.aaai.org/index.php/AAAI/article/view/17325.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 27268–27286. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

# A Experimental Protocols and Complete Benchmark Results

The appendices follow the evidence chain of the main text. Appendix A.1 gives evaluation conventions, and the complete task results for RQ3 appear in the tables of Appendix A; Appendix B examines the periodic-span comparisons for RQ1; Appendix C examines the pathway, dependency, and reprojection controls for RQ2; Appendix D reports the corresponding computation and missing-observation studies; and Appendix F collects the operator identities and derivations used by the method.

## A.1 Evaluation Settings and Metric Conventions

Forecasting uses lookback $L = 96$ and prediction lengths $S \in \{96, 192, 336, 720\}$. Tables 2 and 3 report each prediction length; Table 1 averages these four values within each dataset. Forecasting and imputation use MSE and MAE, classification uses accuracy, and anomaly detection reports precision and point-adjusted (PA) F1. Accuracy, precision, and F1 are proportions on $[0, 1]$.

All anomaly results use point adjustment, so PA-F1 characterizes the recorded detection protocol.

The reported task-training setup uses PyTorch and Adam, learning rate $10^{-4}$, and batch size 32. DA$^2$ uses eight attention heads and a layerwise scalar gate. Algorithm 1 defines the manuscript operator; study-specific settings below delimit the corresponding comparisons. The task setup and short-context summaries specify NVIDIA RTX 3090 GPUs, the detector sensitivity grid specifies an RTX 4090, and the long-context figure specifies an H200. We report hardware-specific results within their own measurement settings.

The additional-task comparisons include iTransformer, PatchTST, DLinear, FEDformer, Crossformer, and Autoformer (Liu et al., 2024; Nie et al., 2023; Zeng et al., 2023; Zhou et al., 2022; Zhang & Yan, 2023; Wu et al., 2021). Bold and underlined values mark the best and second-best entries within a comparison, and the forecasting, anomaly-detection, and imputation tables additionally use red and blue. Lower MSE/MAE and higher accuracy, precision, and F1 are better.

Displayed aggregates and configuration sweeps are reported as point estimates; numerical ordering is read within each comparison.

## A.2 Complete Forecasting Results

Tables 2 and 3 report per-prediction-length MSE and MAE for every dataset and baseline in the forecasting benchmark. They preserve every dataset row, including rows led by other methods.

## A.3 Classification

Table 5 reports classification accuracy on seven datasets using benchmark splits and z-score normalization. PeCo-TS attains the highest displayed accuracy on EthanolConcentration, SelfRegulationSCP2, and SpokenArabicDigits. The remaining datasets show a different ordering, yielding a dataset-dependent pattern across additional tasks.

## A.4 Anomaly Detection

Tables 6 and 7 report precision and PA-F1 on five datasets. PeCo-TS attains the highest displayed PA-F1 on MSL, SMAP, SMD, and SWaT, and the highest displayed precision on SMAP and SMD. The PSM and SWaT rows assign different leaders to precision and PA-F1. All F1 values use point adjustment.

## A.5 Imputation

Tables 8 and 9 report MSE and MAE across datasets and random missingness ratios. The leading method changes across datasets, ratios, and metrics. The complete rows show settings led by PeCo-TS and settings led by other methods.

Figures 5–7 illustrate ETTh1, Weather, and Electricity forecasts at $L = S = 96$. The numerical tables provide the dataset-level comparisons, while these panels show individual trajectories.

# B Periodic Tokenization Analyses

These analyses test three properties of periodic tokenization: whether aggregation spans change with the input window, how forecasting error responds to changed boundaries, and how detector settings relate to realized token count.

Table 2: **Complete forecasting results (part 1).** MSE and MAE for all prediction lengths, showing PeCo-TS through iTransformer at lookback $L = 96$. Best and second-best marks are computed across all eleven methods in both parts; bold entries mark top-two ranks.

| Dataset | Horizon | PeCo-TS | | AMD | | PathFormer | | CARD | | UniTS | | iTransformer | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | 96 | **0.365** | 0.396 | 0.375 | 0.398 | 0.385 | **0.391** | 0.383 | **0.391** | 0.393 | 0.417 | 0.395 | 0.409 |
| | 192 | **0.409** | 0.422 | 0.430 | 0.427 | 0.443 | 0.421 | 0.435 | **0.420** | 0.440 | 0.448 | 0.449 | 0.441 |
| | 336 | **0.440** | 0.438 | 0.462 | **0.424** | 0.459 | 0.430 | 0.480 | 0.443 | 0.471 | 0.465 | 0.492 | 0.465 |
| | 720 | **0.444** | **0.454** | 0.473 | 0.462 | 0.493 | 0.463 | 0.469 | 0.460 | 0.510 | 0.504 | 0.522 | 0.504 |
| ETTm1 | 96 | **0.320** | **0.360** | 0.334 | 0.379 | 0.348 | 0.377 | 0.337 | 0.377 | 0.351 | 0.379 | 0.341 | 0.376 |
| | 192 | **0.329** | **0.366** | 0.352 | 0.382 | 0.367 | 0.390 | 0.395 | 0.400 | 0.394 | 0.403 | 0.381 | 0.395 |
| | 336 | **0.372** | **0.389** | 0.410 | 0.405 | 0.417 | 0.409 | 0.415 | 0.420 | 0.413 | 0.419 | 0.419 | 0.419 |
| | 720 | **0.457** | 0.438 | 0.462 | **0.434** | 0.469 | 0.438 | 0.458 | 0.455 | 0.471 | 0.451 | 0.486 | 0.456 |
| ETTm2 | 96 | 0.181 | 0.264 | 0.197 | 0.279 | 0.200 | 0.289 | 0.188 | 0.277 | 0.279 | 0.315 | 0.184 | 0.267 |
| | 192 | **0.246** | 0.308 | 0.251 | 0.312 | 0.263 | 0.321 | 0.263 | 0.321 | 0.405 | 0.381 | 0.253 | 0.312 |
| | 336 | 0.311 | 0.350 | 0.315 | 0.377 | 0.335 | 0.361 | 0.310 | 0.358 | 0.480 | 0.429 | 0.315 | 0.352 |
| | 720 | 0.411 | 0.409 | 0.416 | 0.422 | 0.415 | 0.426 | 0.410 | 0.416 | 0.606 | 0.504 | 0.412 | 0.406 |
| Electricity | 96 | 0.179 | **0.261** | 0.195 | 0.282 | 0.189 | 0.279 | 0.184 | 0.272 | **0.176** | 0.285 | 0.196 | 0.281 |
| | 192 | **0.184** | **0.267** | 0.207 | 0.295 | 0.204 | 0.282 | 0.194 | 0.280 | 0.197 | 0.304 | 0.206 | 0.293 |
| | 336 | **0.200** | **0.284** | 0.232 | 0.321 | 0.216 | 0.339 | 0.211 | 0.301 | 0.219 | 0.325 | 0.226 | 0.313 |
| | 720 | 0.241 | **0.317** | 0.265 | 0.343 | 0.263 | 0.360 | 0.275 | 0.348 | 0.277 | 0.353 | 0.270 | 0.347 |
| Exchange | 96 | **0.082** | **0.203** | 0.083 | 0.204 | 0.115 | 0.237 | 0.087 | 0.207 | 0.112 | 0.233 | 0.087 | 0.207 |
| | 192 | **0.175** | **0.299** | 0.201 | 0.323 | 0.247 | 0.352 | 0.182 | 0.306 | 0.249 | 0.357 | 0.180 | 0.303 |
| | 336 | **0.322** | **0.416** | 0.342 | 0.432 | 0.469 | 0.489 | 0.333 | 0.432 | 0.474 | 0.495 | 0.333 | 0.419 |
| | 720 | **0.698** | **0.658** | 1.005 | 0.752 | 1.396 | 0.832 | 0.866 | 0.710 | 1.434 | 0.878 | 0.856 | 0.700 |
| Traffic | 96 | **0.502** | **0.327** | 0.536 | 0.359 | 0.521 | 0.343 | 0.512 | 0.334 | 0.508 | 0.333 | 0.574 | 0.386 |
| | 192 | 0.517 | **0.332** | 0.582 | 0.365 | 0.534 | 0.342 | 0.520 | 0.342 | 0.531 | 0.348 | 0.584 | 0.390 |
| | 336 | **0.521** | 0.338 | 0.596 | 0.369 | 0.547 | 0.352 | 0.540 | 0.352 | 0.550 | 0.354 | 0.613 | 0.405 |
| | 720 | 0.563 | **0.352** | 0.629 | 0.392 | 0.574 | 0.368 | 0.567 | 0.360 | 0.584 | 0.373 | 0.676 | 0.434 |
| Weather | 96 | 0.177 | 0.220 | 0.167 | **0.214** | 0.203 | 0.239 | 0.204 | 0.241 | 0.193 | 0.239 | 0.183 | 0.225 |
| | 192 | 0.227 | 0.263 | 0.216 | **0.260** | 0.261 | 0.285 | 0.264 | 0.285 | 0.252 | 0.279 | 0.234 | 0.266 |
| | 336 | 0.281 | 0.299 | 0.271 | 0.299 | 0.340 | 0.337 | 0.342 | 0.337 | 0.333 | 0.330 | 0.287 | 0.304 |
| | 720 | 0.355 | 0.349 | 0.343 | **0.344** | 0.383 | 0.379 | 0.392 | 0.381 | 0.373 | 0.359 | 0.362 | 0.352 |

Table 3: **Complete forecasting results (part 2).** MSE and MAE for all prediction lengths, showing PatchTST through DLinear at lookback $L = 96$. Best and second-best marks are computed across all eleven methods in both parts; bold entries mark top-two ranks.

| Dataset | Horizon | PatchTST | | TSMixer | | TimesNet | | Mamba | | DLinear | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | 96 | 0.377 | 0.397 | 0.494 | 0.502 | 0.389 | 0.412 | 0.486 | 0.452 | 0.396 | 0.411 |
| | 192 | 0.425 | 0.428 | 0.581 | 0.557 | 0.439 | 0.442 | 0.555 | 0.506 | 0.445 | 0.440 |
| | 336 | 0.473 | 0.458 | 0.677 | 0.618 | 0.494 | 0.471 | 0.537 | 0.500 | 0.487 | 0.465 |
| | 720 | 0.518 | 0.501 | 0.752 | 0.674 | 0.518 | 0.494 | 0.624 | 0.577 | 0.513 | 0.510 |
| ETTm1 | 96 | 0.324 | 0.365 | 0.479 | 0.470 | 0.336 | 0.376 | 0.372 | 0.391 | 0.346 | 0.374 |
| | 192 | 0.365 | 0.386 | 0.480 | 0.482 | 0.387 | 0.402 | 0.436 | 0.421 | 0.382 | 0.391 |
| | 336 | 0.393 | 0.408 | 0.541 | 0.525 | 0.414 | 0.422 | 0.558 | 0.511 | 0.415 | 0.415 |
| | 720 | 0.460 | 0.443 | 0.616 | 0.574 | 0.513 | 0.472 | 0.625 | 0.548 | 0.473 | 0.451 |
| ETTm2 | 96 | **0.178** | **0.260** | 0.195 | 0.275 | 0.188 | 0.268 | 0.196 | 0.275 | 0.193 | 0.293 |
| | 192 | 0.247 | 0.308 | 0.263 | 0.320 | 0.252 | **0.307** | 0.302 | 0.342 | 0.285 | 0.361 |
| | 336 | **0.309** | 0.347 | 0.318 | 0.355 | 0.317 | **0.346** | 0.372 | 0.393 | 0.385 | 0.429 |
| | 720 | **0.407** | **0.403** | 0.416 | 0.430 | 0.421 | 0.406 | 0.637 | 0.510 | 0.556 | 0.523 |
| Electricity | 96 | 0.189 | 0.277 | 0.200 | 0.305 | 0.276 | 0.358 | 0.188 | 0.290 | 0.210 | 0.301 |
| | 192 | 0.193 | 0.283 | 0.220 | 0.331 | 0.285 | 0.367 | 0.204 | 0.308 | 0.210 | 0.305 |
| | 336 | 0.209 | 0.298 | 0.242 | 0.353 | 0.296 | 0.378 | 0.207 | 0.313 | 0.223 | 0.319 |
| | 720 | 0.251 | 0.331 | 0.271 | 0.372 | 0.333 | 0.402 | **0.237** | 0.333 | 0.258 | 0.350 |
| Exchange | 96 | 0.084 | **0.203** | 0.232 | 0.388 | 0.111 | 0.238 | 0.127 | 0.258 | 0.094 | 0.227 |
| | 192 | 0.181 | 0.302 | 0.464 | 0.549 | 0.209 | 0.333 | 0.287 | 0.391 | 0.186 | 0.325 |
| | 336 | 0.337 | 0.421 | 0.754 | 0.720 | 0.374 | 0.448 | 0.651 | 0.603 | 0.327 | 0.435 |
| | 720 | 0.875 | 0.703 | 0.705 | 0.701 | 0.931 | 0.735 | 1.706 | 0.970 | 0.749 | 0.664 |
| Traffic | 96 | 0.509 | 0.331 | 0.578 | 0.388 | 0.868 | 0.499 | 0.679 | 0.383 | 0.696 | 0.429 |
| | 192 | **0.514** | 0.337 | 0.579 | 0.394 | 0.919 | 0.537 | 0.645 | 0.367 | 0.646 | 0.407 |
| | 336 | 0.522 | **0.334** | 0.604 | 0.409 | 0.898 | 0.514 | 0.636 | 0.360 | 0.653 | 0.410 |
| | 720 | **0.558** | 0.353 | 0.664 | 0.439 | 0.927 | 0.542 | 0.755 | 0.414 | 0.694 | 0.429 |
| Weather | 96 | 0.179 | 0.220 | **0.163** | 0.241 | 0.180 | 0.225 | 0.195 | 0.243 | 0.196 | 0.257 |
| | 192 | 0.228 | **0.260** | **0.210** | 0.285 | 0.227 | 0.266 | 0.252 | 0.291 | 0.236 | 0.294 |
| | 336 | 0.279 | **0.297** | **0.267** | 0.333 | 0.281 | 0.304 | 0.327 | 0.342 | 0.283 | 0.333 |
| | 720 | 0.355 | 0.347 | **0.332** | 0.379 | 0.360 | 0.355 | 0.406 | 0.385 | 0.347 | 0.384 |

Table 4: **Forecasting MSE ratios versus representative baselines.** Each entry is baseline MSE divided by PeCo-TS MSE at lookback $L = 96$; values above 1.0 favor PeCo-TS. The table reports the $S$=96 and $S$=720 endpoints for PatchTST, iTransformer, DLinear, and AMD; all prediction lengths and baselines appear in Tables 2 and 3.

| Dataset | PatchTST | | iTransformer | | DLinear | | AMD | |
|---|---|---|---|---|---|---|---|---|
| | $S$=96 | $S$=720 | $S$=96 | $S$=720 | $S$=96 | $S$=720 | $S$=96 | $S$=720 |
| ETTh1 | 1.033 | 1.167 | 1.082 | 1.176 | 1.085 | 1.155 | 1.027 | 1.065 |
| ETTm1 | 1.012 | 1.007 | 1.066 | 1.063 | 1.081 | 1.035 | 1.044 | 1.011 |
| ETTm2 | 0.983 | 0.990 | 1.017 | 1.002 | 1.066 | 1.353 | 1.088 | 1.012 |
| Electricity | 1.056 | 1.041 | 1.095 | 1.120 | 1.173 | 1.071 | 1.089 | 1.100 |
| Exchange | 1.024 | 1.254 | 1.061 | 1.226 | 1.146 | 1.073 | 1.012 | 1.440 |
| Traffic | 1.014 | 0.991 | 1.143 | 1.201 | 1.386 | 1.233 | 1.068 | 1.117 |
| Weather | 1.011 | 1.000 | 1.034 | 1.020 | 1.107 | 0.977 | 0.944 | 0.966 |

Table 5: **Classification results on UCR/UEA.** Accuracy is reported as a proportion on $[0, 1]$. Methods use the benchmark splits and z-score normalization.

| Dataset | PeCo-TS | iTransformer | PatchTST | DLinear | FEDformer | Crossformer | Autoformer |
|---|---|---|---|---|---|---|---|
| EthanolConcentration | **0.3270** | 0.2852 | 0.2814 | 0.2928 | 0.2776 | 0.3030 | 0.2433 |
| FaceDetection | 0.6831 | 0.6654 | **0.6864** | 0.6822 | 0.6751 | 0.6512 | 0.5951 |
| JapaneseVowels | 0.9676 | **0.9757** | 0.9595 | 0.9649 | 0.9674 | **0.9757** | 0.9649 |
| SelfRegulationSCP1 | 0.9144 | **0.9215** | 0.8737 | 0.9147 | 0.5802 | 0.9147 | 0.5631 |
| SelfRegulationSCP2 | **0.5560** | 0.5444 | 0.5278 | 0.5444 | 0.5278 | 0.5467 | 0.5333 |
| SpokenArabicDigits | **0.9818** | 0.9804 | 0.9741 | 0.9650 | 0.9782 | 0.9627 | 0.9759 |
| UWaveGestureLibrary | 0.7919 | 0.8594 | **0.8625** | 0.8219 | 0.5656 | 0.8531 | 0.5000 |

Table 6: **Anomaly detection precision.** Precision for the five datasets under the recorded point-adjustment protocol.

| Dataset | PeCo-TS | iTransformer | PatchTST | DLinear | FEDformer | Crossformer | Autoformer |
|---|---|---|---|---|---|---|---|
| MSL | 0.9036 | 0.8615 | 0.8860 | 0.8969 | **0.9068** | 0.9031 | 0.9054 |
| PSM | 0.9864 | 0.9797 | 0.9910 | 0.9864 | **0.9999** | 0.9729 | **0.9999** |
| SMAP | **0.9330** | 0.9069 | 0.8988 | 0.8987 | 0.9015 | 0.8998 | 0.9127 |
| SMD | **0.7891** | 0.7627 | 0.7648 | 0.7612 | 0.7278 | 0.7204 | 0.7298 |
| SWaT | 0.9810 | 0.9221 | 0.9124 | 0.9227 | 0.9995 | 0.9782 | **0.9996** |

Table 7: **Anomaly detection PA-F1.** Point-adjusted F1 for the five datasets under the recorded protocol.

| Dataset | PeCo-TS | iTransformer | PatchTST | DLinear | FEDformer | Crossformer | Autoformer |
|---|---|---|---|---|---|---|---|
| MSL | **0.8380** | 0.7253 | 0.7913 | 0.8187 | 0.8230 | 0.8060 | 0.8187 |
| PSM | 0.9626 | 0.9532 | 0.9626 | **0.9661** | 0.9007 | 0.9239 | 0.8823 |
| SMAP | **0.8615** | 0.6675 | 0.6726 | 0.6729 | 0.6863 | 0.6874 | 0.7411 |
| SMD | **0.8482** | 0.8057 | 0.8104 | 0.8007 | 0.7688 | 0.7758 | 0.7723 |
| SWaT | **0.9635** | 0.9265 | 0.8673 | 0.9266 | 0.7918 | 0.9063 | 0.7918 |

Table 8: **Imputation MSE.** MSE across datasets and random missingness ratios.

| Dataset | Mask Ratio | PeCo-TS | iTransformer | PatchTST | DLinear | FEDformer | Crossformer | Autoformer |
|---|---|---|---|---|---|---|---|---|
| ETTh1 | 12.5% | 0.1010 | 0.0987 | 0.0928 | 0.1117 | **0.0735** | 0.1064 | 0.0978 |
| | 25.0% | 0.1230 | 0.1250 | 0.1063 | 0.1496 | **0.1055** | 0.1172 | 0.1208 |
| | 37.5% | 0.1470 | 0.1573 | **0.1188** | 0.1874 | 0.1411 | 0.1293 | 0.1550 |
| | 50.0% | 0.1755 | 0.2177 | **0.1403** | 0.2316 | 0.2006 | 0.1478 | 0.2119 |
| ETTh2 | 12.5% | 0.0635 | 0.0932 | **0.0570** | 0.1091 | 0.1296 | 0.1215 | 0.1702 |
| | 25.0% | 0.0682 | 0.1209 | **0.0620** | 0.1449 | 0.1788 | 0.1335 | 0.2229 |
| | 37.5% | 0.0719 | 0.1485 | **0.0674** | 0.1794 | 0.2335 | 0.1451 | 0.2781 |
| | 50.0% | 0.0846 | 0.1931 | **0.0736** | 0.2161 | 0.3462 | 0.1614 | 0.3747 |
| ETTm1 | 12.5% | **0.0338** | 0.0456 | 0.0396 | 0.0556 | 0.0448 | 0.0436 | 2.010 |
| | 25.0% | **0.0397** | 0.0605 | 0.0420 | 0.0766 | 0.0531 | 0.0466 | 1.109 |
| | 37.5% | **0.0450** | 0.0774 | 0.0466 | 0.0998 | 0.0809 | 0.0506 | 0.3463 |
| | 50.0% | **0.0517** | 0.1067 | 0.0523 | 0.1286 | 0.1278 | 0.0567 | 0.3391 |
| ETTm2 | 12.5% | **0.0253** | 0.0518 | 0.0254 | 0.0662 | 0.0601 | 0.0557 | 2.788 |
| | 25.0% | **0.0277** | 0.0707 | **0.0277** | 0.0893 | 0.0921 | 0.0741 | 0.9562 |
| | 37.5% | **0.0300** | 0.0915 | 0.0301 | 0.1117 | 0.1328 | 0.0796 | 1.463 |
| | 50.0% | 0.0340 | 0.1176 | **0.0332** | 0.1382 | 0.2415 | 0.0877 | 0.6442 |
| Electricity | 12.5% | **0.0492** | 0.0724 | 0.0526 | 0.0844 | 0.1808 | 0.0640 | 0.1875 |
| | 25.0% | **0.0559** | 0.0898 | 0.0623 | 0.1131 | 0.2020 | 0.0716 | 0.2123 |
| | 37.5% | **0.0651** | 0.1068 | 0.0726 | 0.1412 | 0.2205 | 0.0804 | 0.2289 |
| | 50.0% | **0.0796** | 0.1259 | 0.0874 | 0.1726 | 0.2425 | 0.0901 | 0.2600 |
| Weather | 12.5% | **0.0285** | 0.0376 | 0.0287 | 0.0380 | 0.0425 | 0.2314 | 0.0387 |
| | 25.0% | **0.0310** | 0.0460 | **0.0310** | 0.0471 | 0.0568 | 0.1888 | 0.0398 |
| | 37.5% | **0.0330** | 0.0549 | 0.0350 | 0.0558 | 0.0732 | 0.1156 | 0.0399 |
| | 50.0% | **0.0360** | 0.0671 | 0.0378 | 0.0663 | 0.1134 | 0.1655 | 0.0432 |

Table 9: **Imputation MAE.** MAE across datasets and random missingness ratios.

| Dataset | Mask Ratio | PeCo-TS | iTransformer | PatchTST | DLinear | FEDformer | Crossformer | Autoformer |
|---|---|---|---|---|---|---|---|---|
| ETTh1 | 12.5% | 0.2089 | 0.2218 | 0.2010 | 0.2319 | **0.1958** | 0.2283 | 0.2266 |
| | 25.0% | 0.2315 | 0.2504 | **0.2165** | 0.2690 | 0.2365 | 0.2427 | 0.2516 |
| | 37.5% | 0.2540 | 0.2818 | **0.2306** | 0.3004 | 0.2748 | 0.2572 | 0.2851 |
| | 50.0% | 0.2755 | 0.3325 | **0.2486** | 0.3326 | 0.3293 | 0.2764 | 0.3348 |
| ETTh2 | 12.5% | 0.1615 | 0.2080 | **0.1510** | 0.2229 | 0.2437 | 0.2310 | 0.2887 |
| | 25.0% | 0.1692 | 0.2395 | **0.1594** | 0.2593 | 0.2911 | 0.2435 | 0.3302 |
| | 37.5% | 0.1727 | 0.2650 | **0.1670** | 0.2895 | 0.3323 | 0.2565 | 0.3632 |
| | 50.0% | 0.1914 | 0.3026 | **0.1753** | 0.3186 | 0.3988 | 0.2710 | 0.4198 |
| ETTm1 | 12.5% | 0.1294 | 0.1474 | **0.1280** | 0.1612 | 0.1594 | 0.1487 | 1.204 |
| | 25.0% | **0.1260** | 0.1723 | 0.1318 | 0.1906 | 0.1633 | 0.1524 | 0.8591 |
| | 37.5% | **0.1380** | 0.1959 | 0.1390 | 0.2175 | 0.2013 | 0.1580 | 0.4382 |
| | 50.0% | **0.1470** | 0.2316 | **0.1470** | 0.2463 | 0.2545 | 0.1677 | 0.4195 |
| ETTm2 | 12.5% | **0.0911** | 0.1514 | 0.0931 | 0.1707 | 0.1681 | 0.1576 | 1.326 |
| | 25.0% | 0.0999 | 0.1789 | **0.0982** | 0.2007 | 0.2089 | 0.1802 | 0.7293 |
| | 37.5% | **0.1010** | 0.2043 | 0.1028 | 0.2256 | 0.2464 | 0.1779 | 0.8603 |
| | 50.0% | 0.1150 | 0.2327 | **0.1079** | 0.2514 | 0.3297 | 0.1861 | 0.5610 |
| Electricity | 12.5% | **0.1413** | 0.1895 | 0.1550 | 0.2063 | 0.3204 | 0.1792 | 0.3259 |
| | 25.0% | **0.1521** | 0.2134 | 0.1692 | 0.2427 | 0.3367 | 0.1899 | 0.3442 |
| | 37.5% | **0.1654** | 0.2344 | 0.1826 | 0.2731 | 0.3512 | 0.2025 | 0.3557 |
| | 50.0% | **0.1853** | 0.2553 | 0.2022 | 0.3034 | 0.3670 | 0.2155 | 0.3768 |
| Weather | 12.5% | 0.0555 | 0.0858 | **0.0485** | 0.0885 | 0.1033 | 0.3437 | 0.0947 |
| | 25.0% | 0.0556 | 0.1054 | **0.0531** | 0.1074 | 0.1305 | 0.2963 | 0.0973 |
| | 37.5% | **0.0560** | 0.1209 | 0.0588 | 0.1216 | 0.1575 | 0.2205 | 0.0967 |
| | 50.0% | **0.0600** | 0.1407 | 0.0626 | 0.1368 | 0.2095 | 0.2691 | 0.1017 |

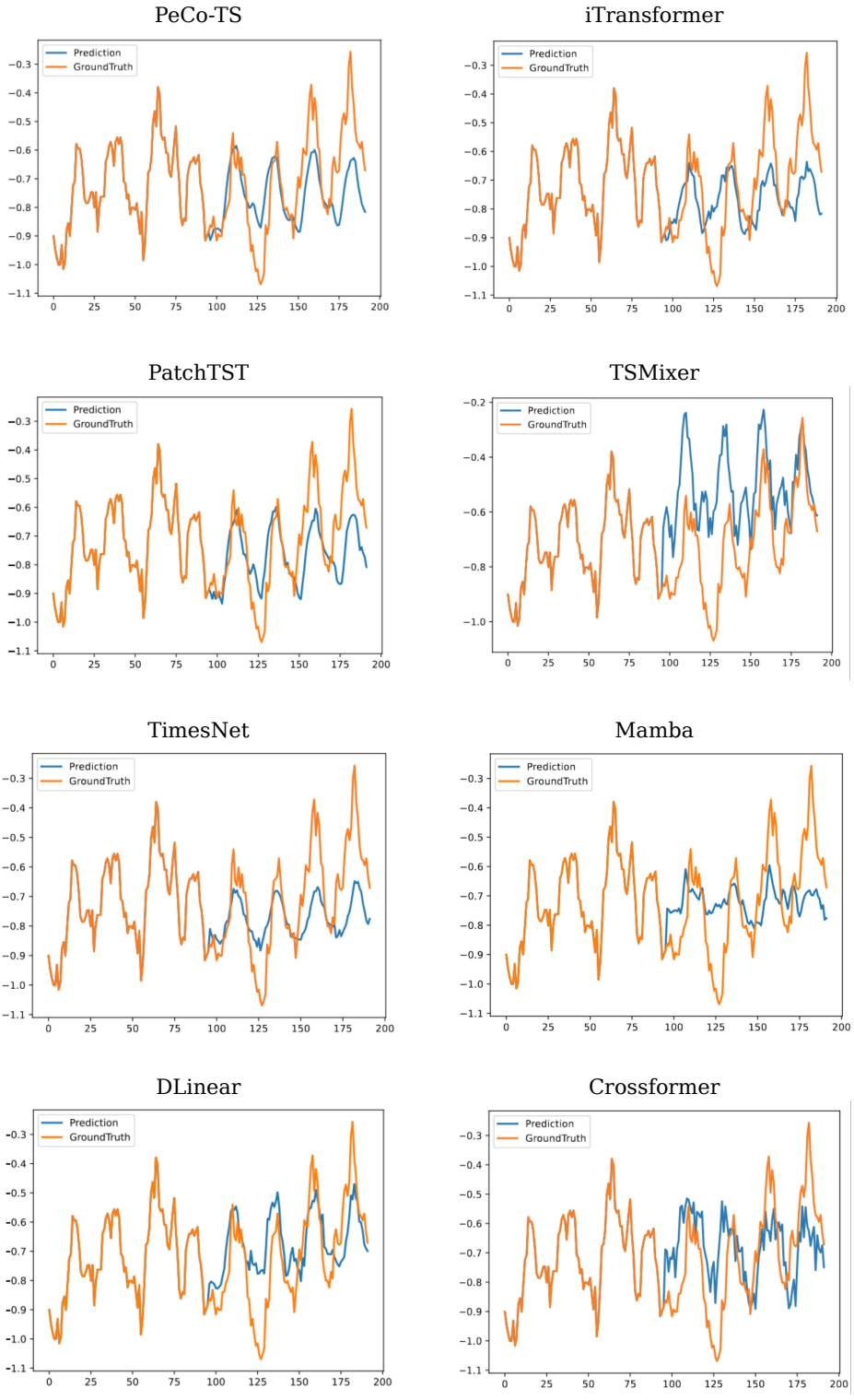



Figure 5: ETTh1 forecast examples with lookback $L = 96$ and prediction length $S = 96$.

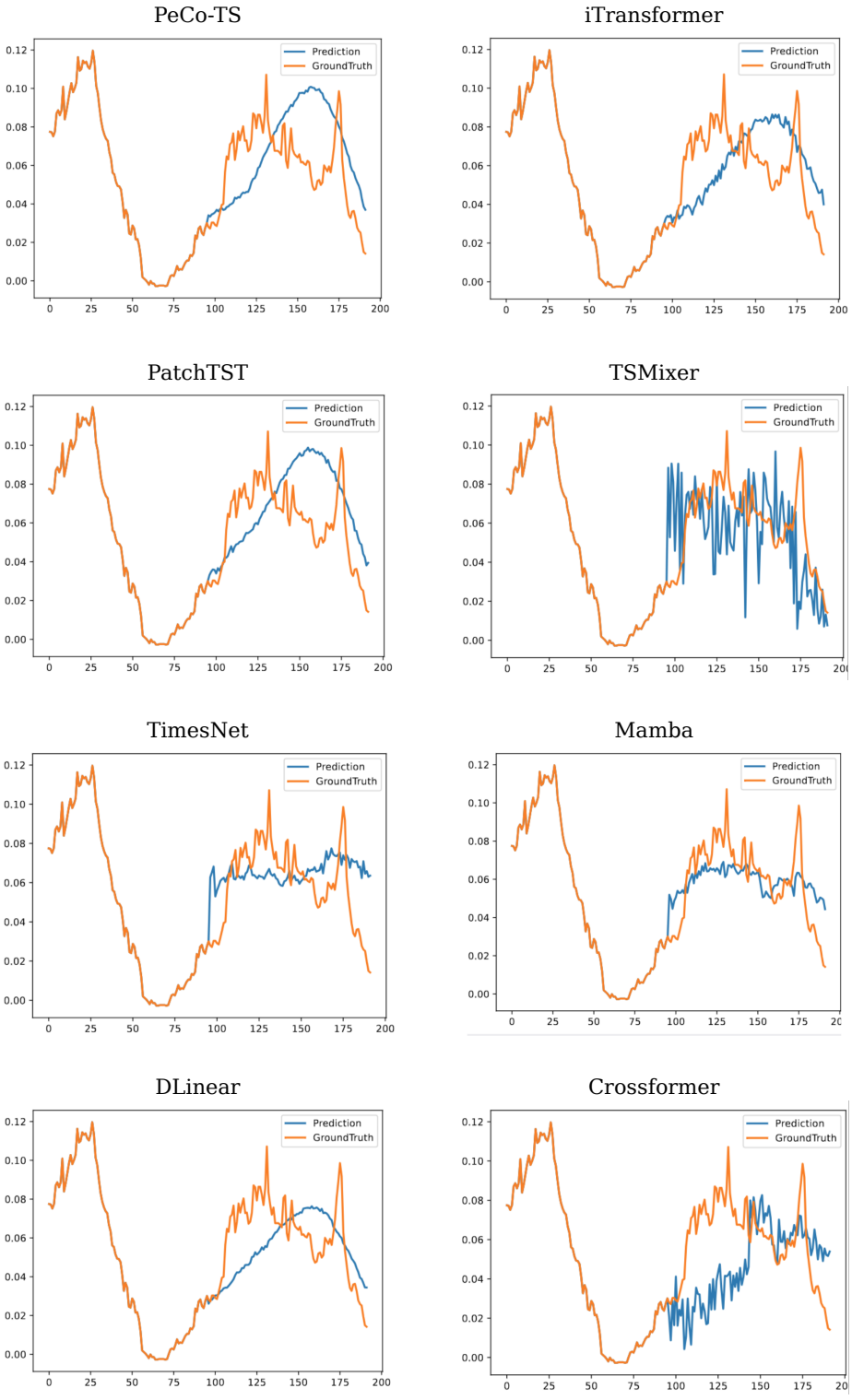



Figure 6: Weather forecast examples with lookback $L = 96$ and prediction length $S = 96$.

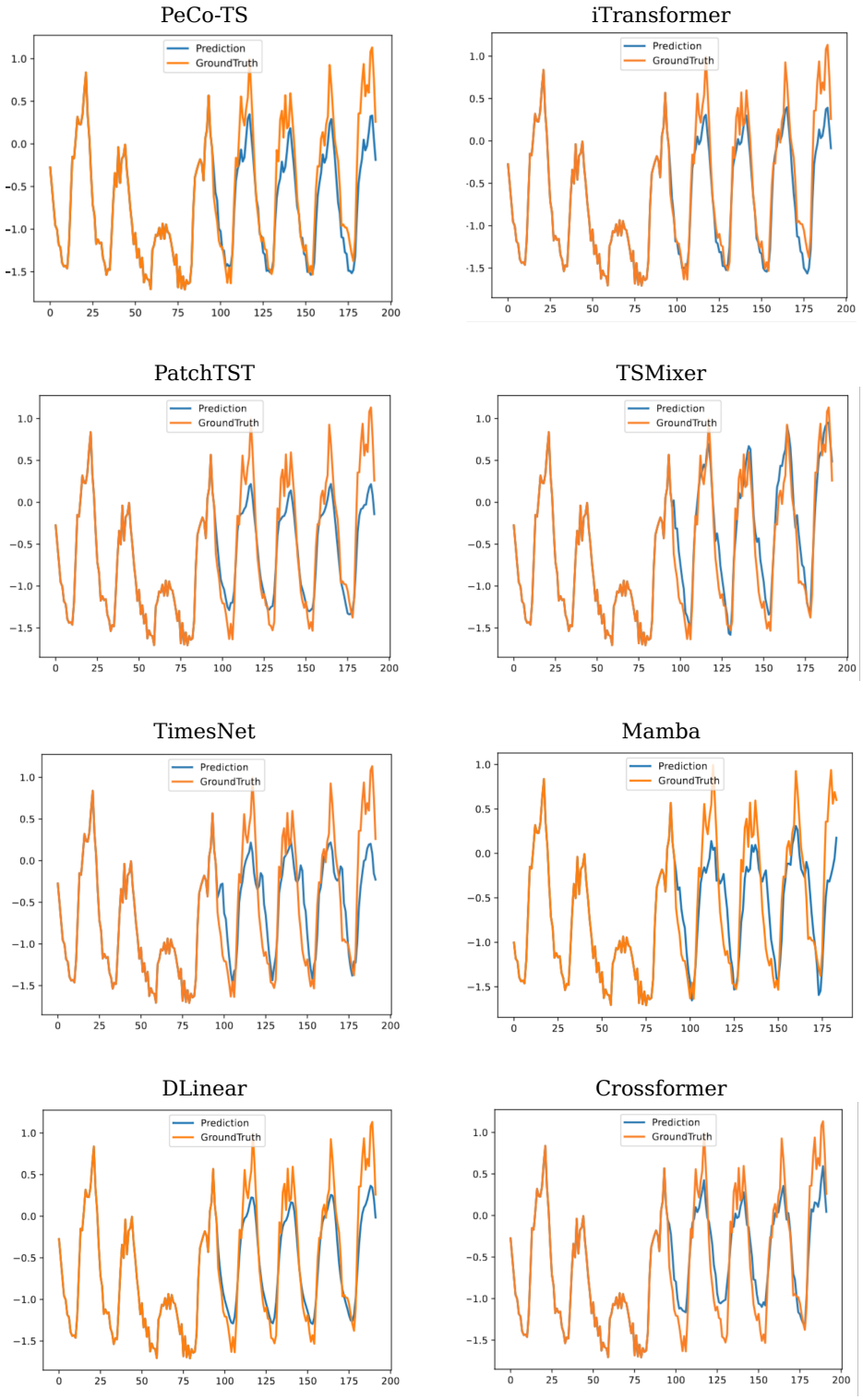



Figure 7: Electricity forecast examples with lookback $L = 96$ and prediction length $S = 96$.

### B.1 Segmentation Method Comparison

We compare input-dependent periodic spans with fixed patch lengths on ETTh1, ETTm1, and Weather. The main figure illustrates two equal-lookback ETTh1 windows with four and eight periodic segments, documenting the temporal units passed to pooling. The displayed per-horizon values appear in Figure 22(a) and Figures 8–9. Because span length, token count, and segment-block capacity vary together across these configurations, we read those curves qualitatively at the displayed operating points.

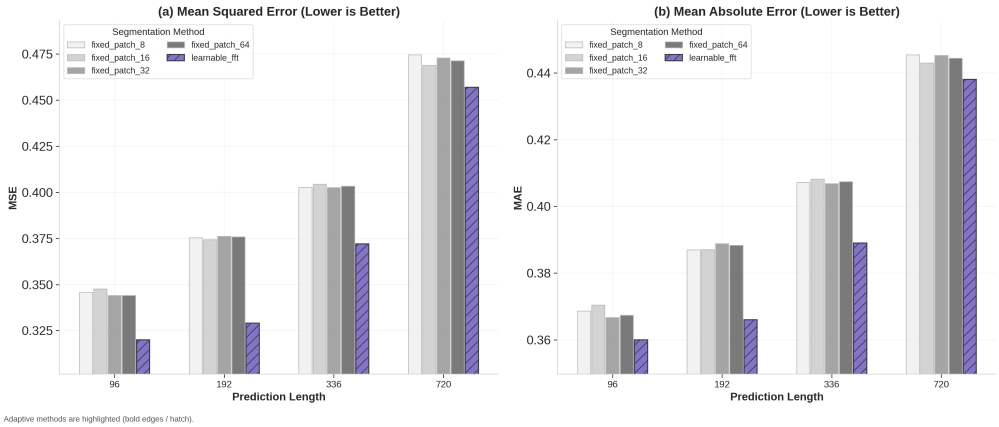



Figure 8: **Tokenization comparison on ETTm1.** Forecasting MSE and MAE for fixed patch lengths $\{8, 16, 32, 64\}$ and periodic tokenization across prediction lengths $\{96, 192, 336, 720\}$. Lower values are better.

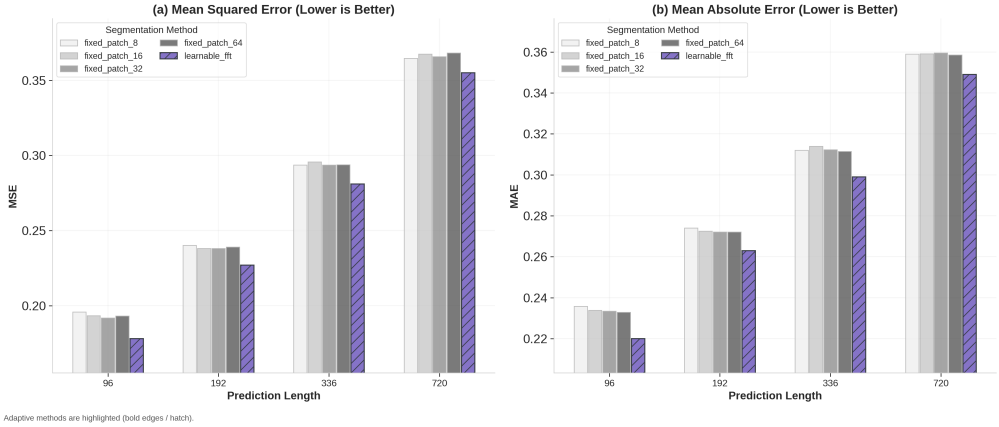



Figure 9: **Tokenization comparison on Weather.** Forecasting MSE and MAE for fixed patch lengths $\{8, 16, 32, 64\}$ and periodic tokenization across prediction lengths $\{96, 192, 336, 720\}$. Lower values are better.

The compared tokenizers differ in both span length and token count, because those choices define the tokenizers being compared. Fixed-length tokenization imposes one temporal resolution on every sequence, whereas the periodic variant allocates tokens according to within-window spectral structure. This comparison evaluates the full input-dependent resolution mechanism at its recorded token counts.

### B.2 Boundary Perturbations and Shared-Boundary Control

The boundary-perturbation comparison applies jitter at 5%, 10%, and 20% of average segment length, boundary replacement at strengths 0.1, 0.3, and 0.5, and complete randomization. Figure 10 reports each operating point relative to its unperturbed MSE at the same prediction length and mode, and Table 10 reports the underlying MSE values. Complete randomization raises MSE by

0.47–1.32% across the eight prediction-length–mode combinations; 20% jitter raises MSE by 0.22–2.27%. Replacement reaches 7.88% at strength 0.3 and prediction length 720 in normal mode, so the measured sensitivity varies with the perturbation and operating point.

These results measure the response of the complete architecture to the specified boundary changes while the full-resolution stream remains active. The normal and fallback modes are separate conditions, and their MSE values are reported separately.

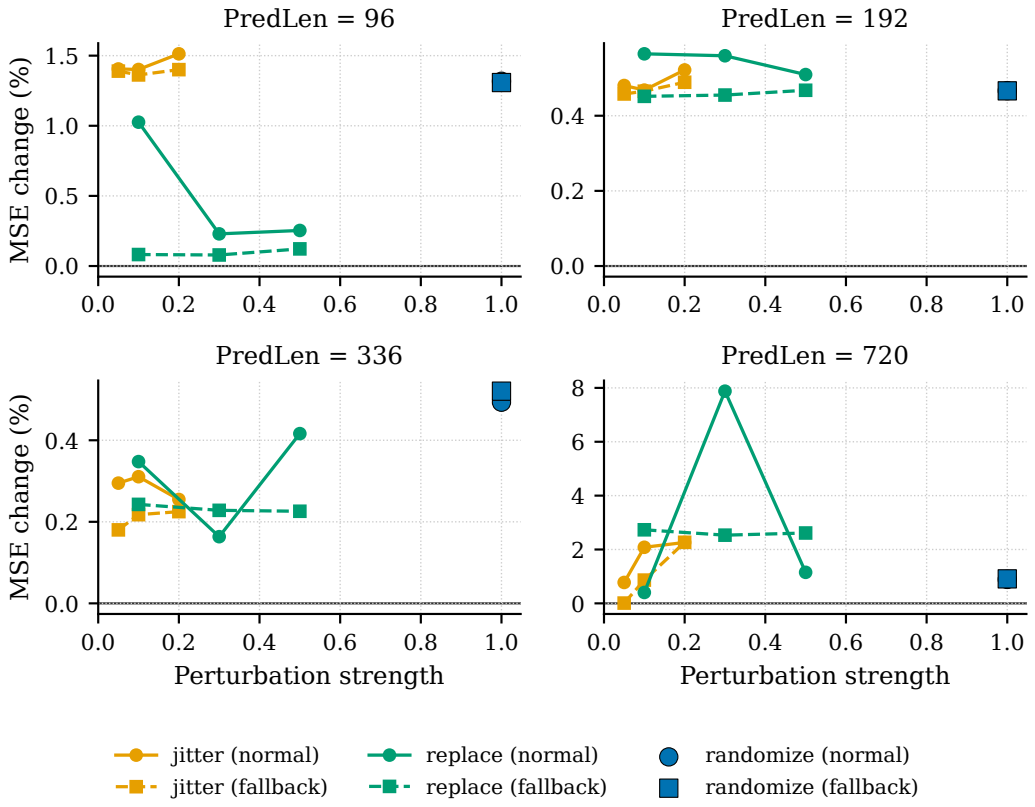



Figure 10: **Boundary perturbation on ETTh1.** MSE change is $100(\mathrm{MSE}_{\mathrm{perturbed}}/\mathrm{MSE}_{\mathrm{none}} - 1)$ at the same prediction length and mode (normal or fallback). Positive values indicate higher error. All jitter, replacement, and randomization points are retained; panels use independent vertical scales. Mode labels follow the recorded normal and fallback conditions.

Table 10: **Complete boundary-perturbation MSE on ETTh1.** Each row fixes prediction length $S$ and mode; the unperturbed (None) column is the reference for Figure 10. Jitter strengths are fractions of average segment length. Replacement strengths use the recorded values, and randomization uses fully randomized boundaries.

| $S$ | Mode | None | Jit. .05 | Jit. .10 | Jit. .20 | Repl. .10 | Repl. .30 | Repl. .50 | Random |
|---|---|---|---|---|---|---|---|---|---|
| 96 | normal | **0.36923** | 0.37442 | 0.37441 | 0.37482 | 0.37302 | 0.37008 | 0.37017 | 0.37410 |
| 96 | fallback | **0.36927** | 0.37440 | 0.37430 | 0.37444 | 0.36957 | 0.36956 | 0.36972 | 0.37410 |
| 192 | normal | **0.40947** | 0.41144 | 0.41139 | 0.41161 | 0.41178 | 0.41176 | 0.41156 | 0.41138 |
| 192 | fallback | **0.40947** | 0.41135 | 0.41138 | 0.41147 | 0.41132 | 0.41133 | 0.41139 | 0.41138 |
| 336 | normal | **0.44015** | 0.44145 | 0.44152 | 0.44127 | 0.44168 | 0.44087 | 0.44198 | 0.44232 |
| 336 | fallback | **0.44003** | 0.44083 | 0.44099 | 0.44102 | 0.44110 | 0.44104 | 0.44103 | 0.44232 |
| 720 | normal | **0.44411** | 0.44755 | 0.45335 | 0.45415 | 0.44590 | 0.47909 | 0.44922 | 0.44805 |
| 720 | fallback | **0.44401** | 0.44406 | 0.44785 | 0.45408 | 0.45614 | 0.45526 | 0.45561 | 0.44805 |

We also compare per-channel boundaries with a shared centroid on Weather. Across 12 detector configurations, per-channel segmentation has lower MSE in 10 settings and the shared-centroid variant in two; the best displayed values are 0.17939 and 0.17968, respectively. This sweep varies temporal spans and channel grouping together.

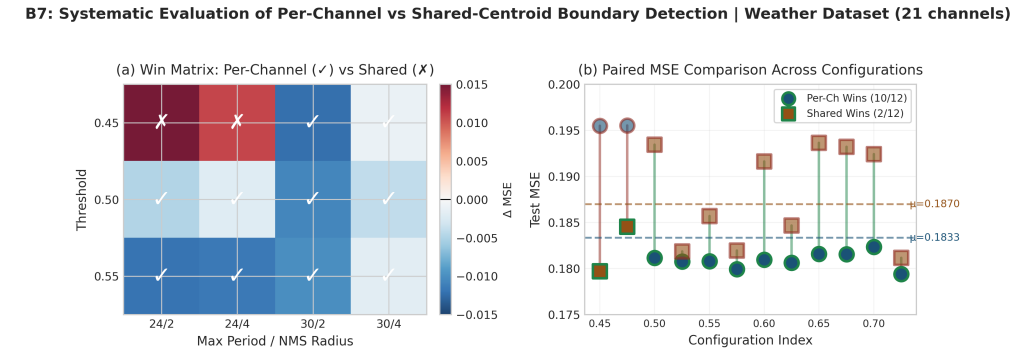



Figure 11: **Per-channel and shared-centroid segmentation on Weather.** MSE across 12 detector configurations. Per-channel segmentation has lower error in 10 configurations and higher error in two.

### B.3 Detector Settings and Token Counts

**Detector settings.** The sensitivity study evaluates temperature $\tau \in \{0.5, 1, 2\}$, sharpness $\gamma \in \{2, 4, 8\}$, NMS-radius settings $\{0, 3, 7\}$, and threshold $\tau_{\text{seg}} \in \{0.3, 0.5\}$ on ETTh1, ETTm1, and Weather. The Cartesian grid contains 54 configurations per dataset at prediction length 96, and each temperature–sharpness cell averages six NMS-radius and threshold combinations. This grid uses fixed NMS radii on an RTX 4090, so it characterizes a different detector configuration space from the period-scaled NMS window in Algorithm 1.

**Configuration-level interaction summaries.** Figure 12 reports mean MSE over the temperature–sharpness grid, and Table 11 reports the corresponding interaction statistics. The ETTh1 grid has a nominal interaction p-value below 0.05 ($p = 0.0062$); the corresponding values are 0.8098 on ETTm1, 0.2275 on Weather, 0.99998 on the pooled grid, and 0.0543 on the normalized pooled grid.

These are configuration-level statistics: the six observations per cell vary NMS radius and threshold. The dataset-wise grids provide the direct error comparisons.

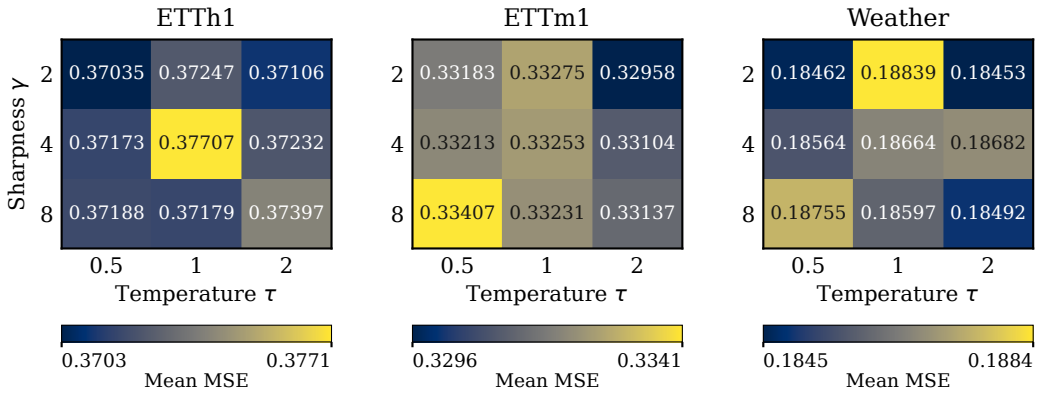



Figure 12: **Temperature–sharpness interaction.** Each cell reports mean forecasting MSE over six NMS-radius and threshold combinations at prediction length 96. All 27 cells are shown. Each dataset uses its own color scale, and lower MSE is better.

Table 11: **Detector interaction ANOVA.** Two-way ANOVA for the $\tau$–$\gamma$ interaction. $N$ is the number of configuration rows (6 NMS$\times$threshold combinations per $\tau$–$\gamma$ cell); the rows represent configuration settings.

| Dataset | $N$ | $F_{\text{interaction}}$ | $p_{\text{interaction}}$ |
|---|---|---|---|
| ETTh1 | 54 | 4.1288 | 0.006206 |
| ETTm1 | 54 | 0.3968 | 0.809834 |
| Weather | 54 | 1.4689 | 0.227469 |
| Pooled | 162 | 0.0029 | 0.999983 |
| Normalized pooled | 162 | 2.3777 | 0.054317 |

**Marginal trends and token count.** Figure 13 reports mean MSE trends for temperature and sharpness over the same grid. Figure 14 plots all 54 configuration-level mean token counts per dataset and their associated forecasting MSE. Mean token count measures realized compression, whereas the padded maximum $M$ in the complexity expressions is a batch-dependent tensor width. The grid reports fixed-configuration trends; validation-tuned selection is a separate protocol.

### B.4 Periodic Segmentation Visualization

**Enhanced spectral flux (ESF).** Enhanced spectral flux (ESF) provides a post-hoc description of local spectral changes in the segmentation visualizations. The detector in Algorithm 1 assigns periodic boundary weights with a cosine comb derived from the window-level spectrum. Let $S_t(f)$ be the magnitude spectrum from a short-time FFT of $x$ using a Hann window. We whiten each frequency band using its local temporal median $M(f)$ and smooth along the frequency axis with kernel $h$:

$$\hat{S}_t(f) = \frac{S_t(f)}{M(f) + \varepsilon}, \quad \tilde{S}_t(f) = (h * \hat{S}_t)(f). \tag{9}$$

The unnormalized flux is the sum of positive frame-to-frame spectral increments, optionally weighted by $w(f)$:

$$\mathrm{ESF}(t) = \sum_f w(f) \left[ \tilde{S}_t(f) - \tilde{S}_{t-1}(f) \right]_+, \quad [x]_+ = \max(x, 0). \tag{10}$$

The visualization rescales this statistic to $[0, 1]$ over the displayed window; the detector uses the cosine-comb proposal derived from the window spectrum.

Across the seven examples, each row shows boundaries on the signal, the periodic weight separated from the ESF trace, and the segmented signal. The repeated signal panels provide two views of the same segmentation: explicit cuts and the spans between them. The first two ETTh1 windows yield four and eight segments over the same lookback, illustrating how the same lookback can yield different segment counts.

The ETTm2 example contains an isolated trough in an otherwise nearly constant trace, yet its proposals repeat throughout the window. The first Weather example places many short proposed segments across a gradual rise and fall. These cases show how a window-level periodic proposal responds to the displayed inputs. Aggregate comparisons provide the predictive results.

## C Representation, Dependency, and Reprojection Analyses

The following studies assess complete pathway variants, alternatives for attention-axis mixing, and the distribution of segment context back to timestep resolution.

### C.1 Fast Path and Slow Path Comparison

Figure 23 compares Full, Fast-only, and Slow-only forecasting variants. The Full model uses Fast Path features both for token construction and for the final unpooled residual.

Removing a pathway changes more than one route through the model. These controls compare complete variants that change contextualization before pooling and detail retention at fusion together.

A wider single pathway offers an alternative to maintaining both resolutions. In the ETTh1 width-control comparison, Full, wide Fast-only, and wide Slow-only obtain mean MSE $0.4434$, $0.4499$, and $0.5618$, respectively. Widening the Fast-only variant brings its error close to Full; wide Slow-only retains a larger gap.

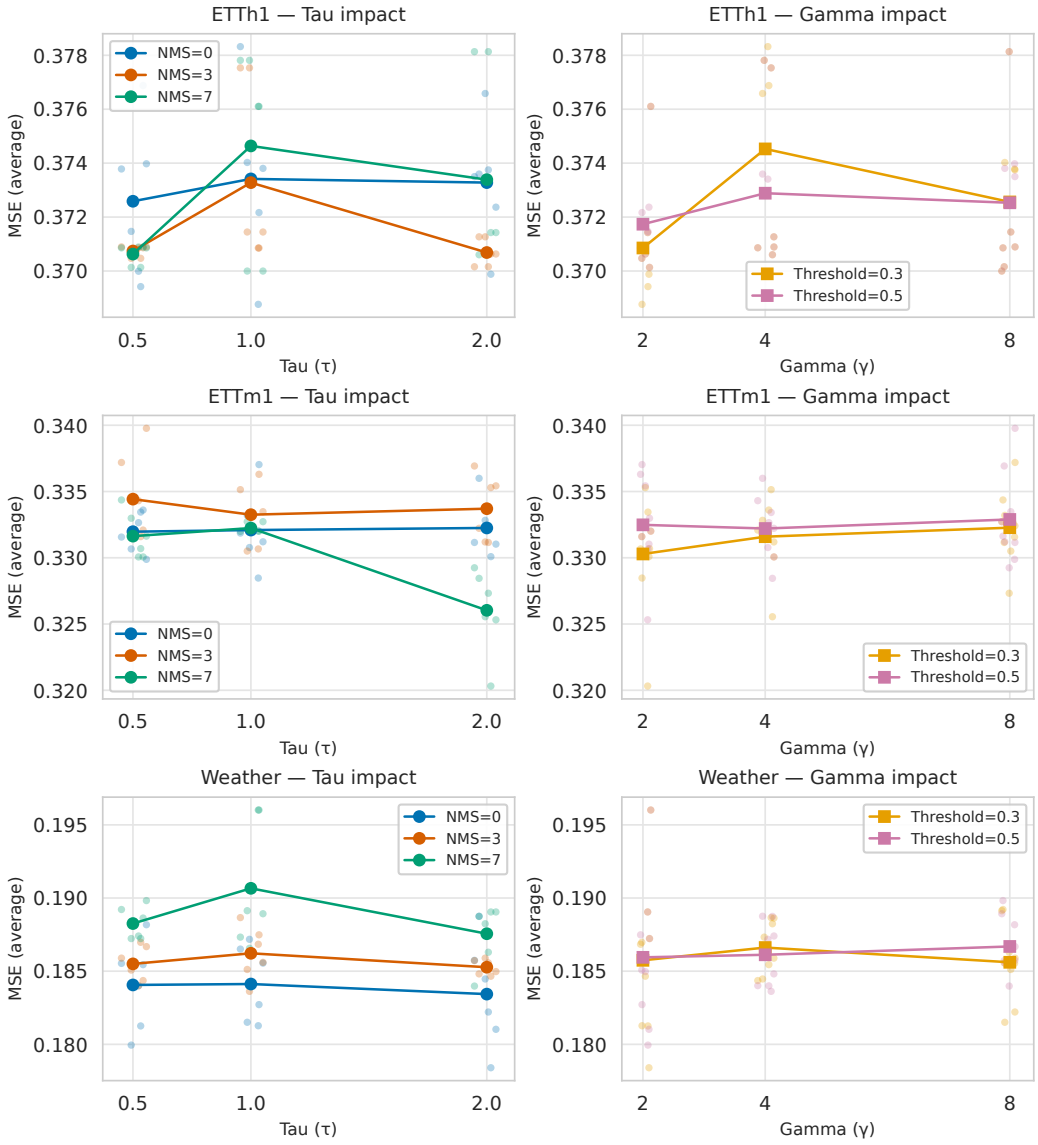



Figure 13: **Detector sensitivity by dataset.** Left: mean MSE versus temperature $\tau$, grouped by NMS radius and averaged over sharpness and threshold. Right: mean MSE versus sharpness $\gamma$, grouped by threshold and averaged over temperature and NMS radius. Faint dots show all configuration values with small horizontal offsets for visibility. Each row corresponds to one dataset.

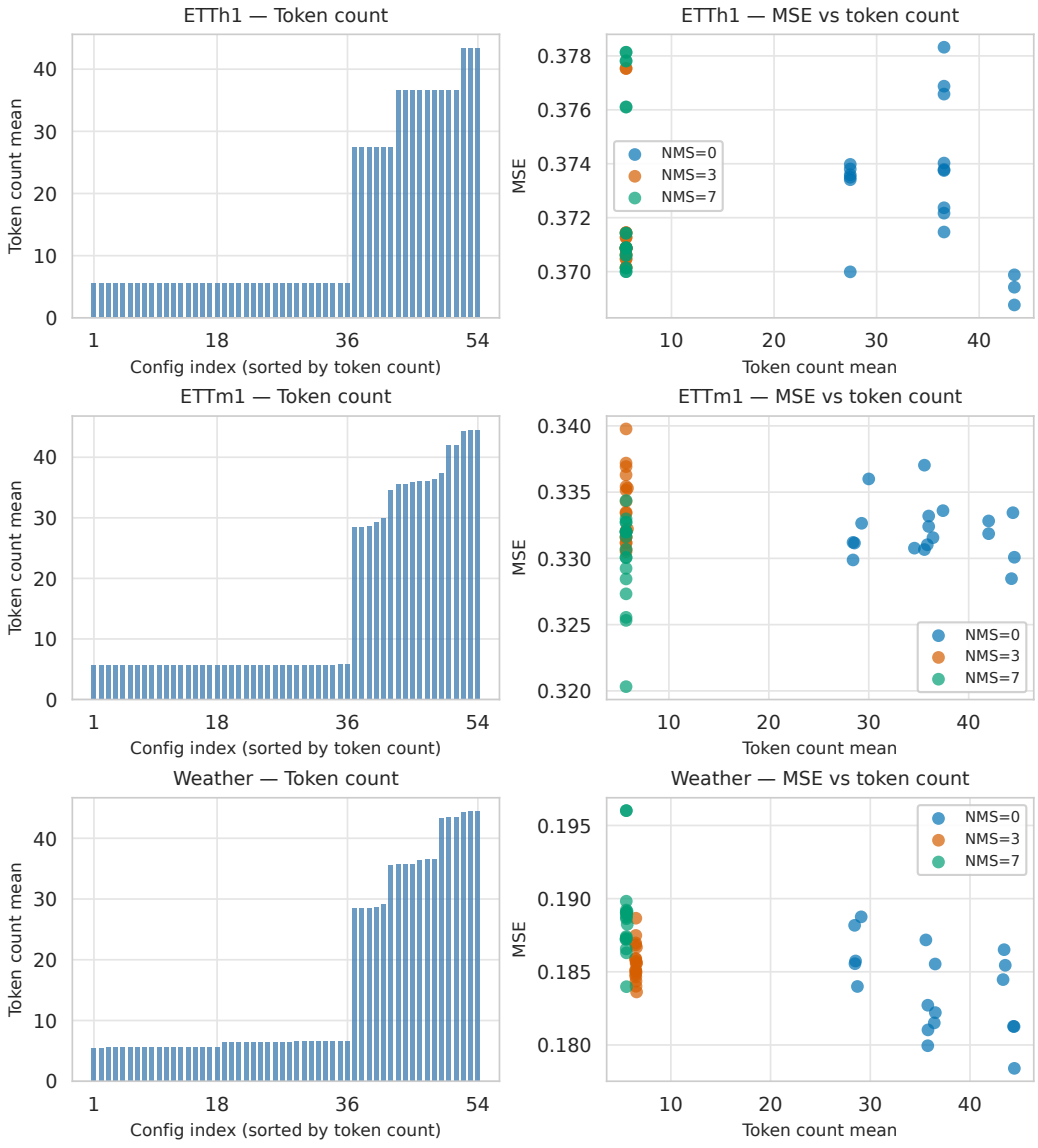



Figure 14: **Token count and prediction error.** Left: all 54 configuration-level mean token counts per dataset, sorted by count. Right: forecasting MSE versus mean token count for the same configurations, with colors indicating NMS-radius settings. Each row corresponds to one dataset.

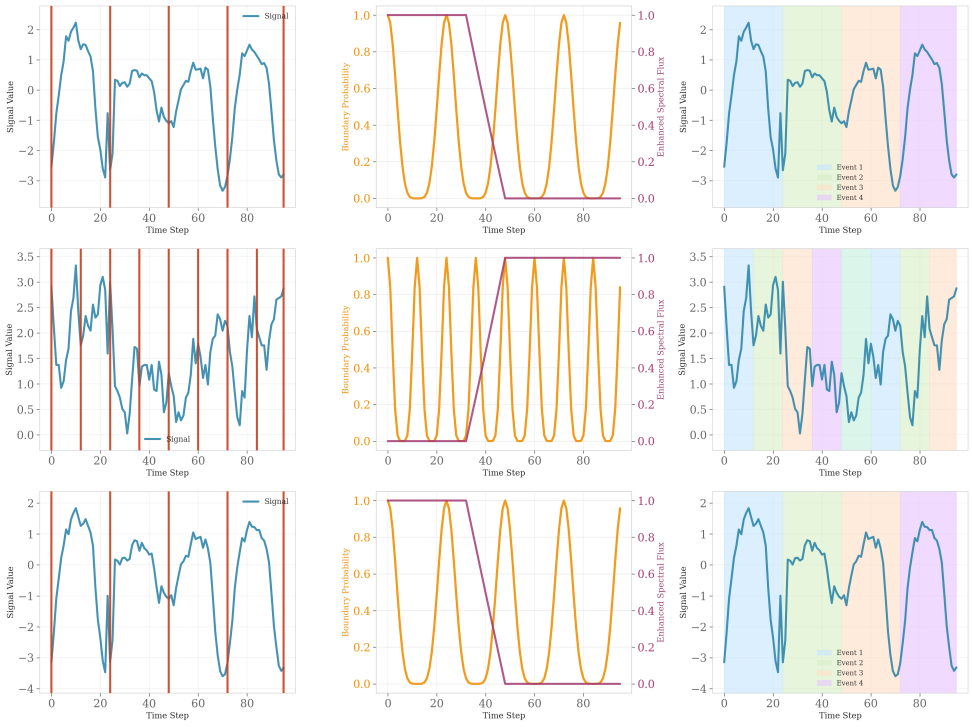



Figure 15: **ETTh1 periodic segmentation.** Input signal, spectral-change statistic, cosine-comb weights, and resulting segments.

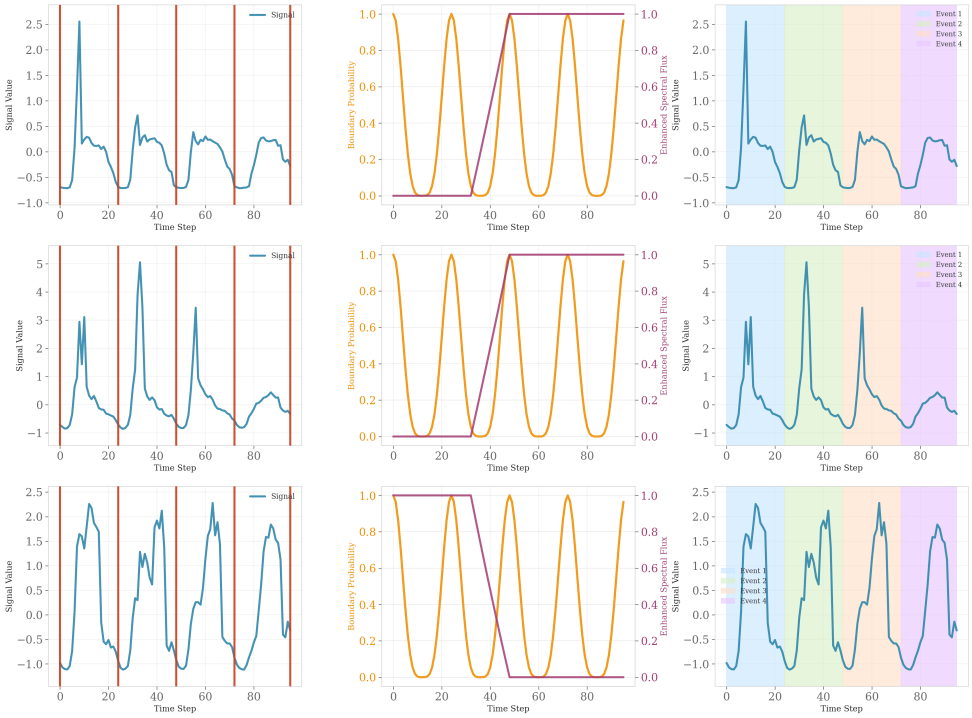



Figure 16: **Traffic periodic segmentation.** The cosine comb sets periodic proposal spacing; local spectral changes are displayed separately.

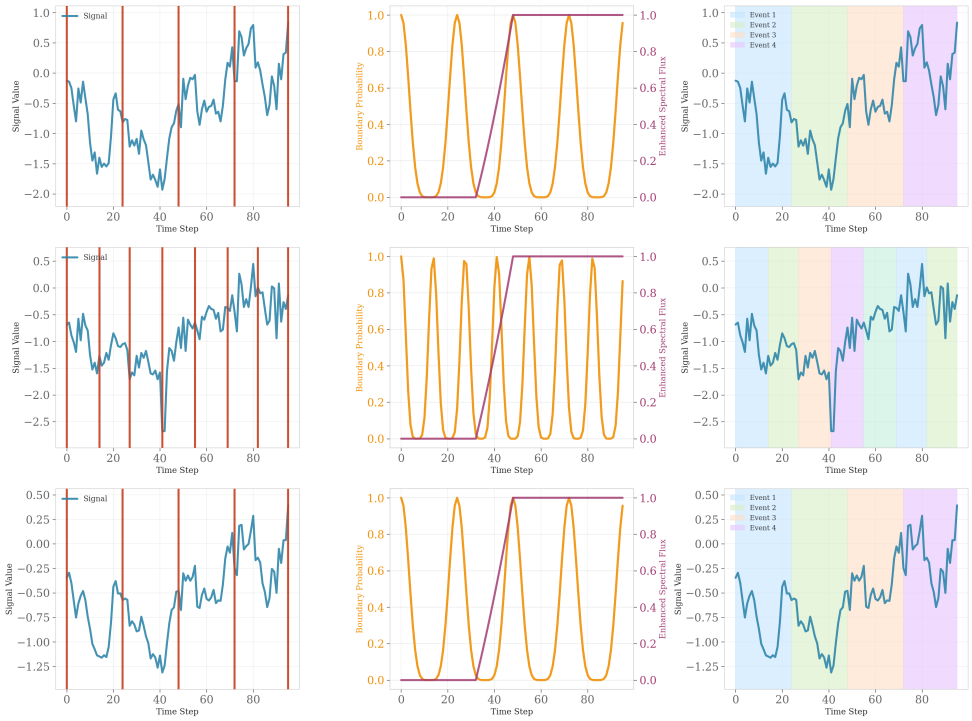



Figure 17: **ETTh2 periodic segmentation.** Segment spacing follows the effective period of each displayed input window.

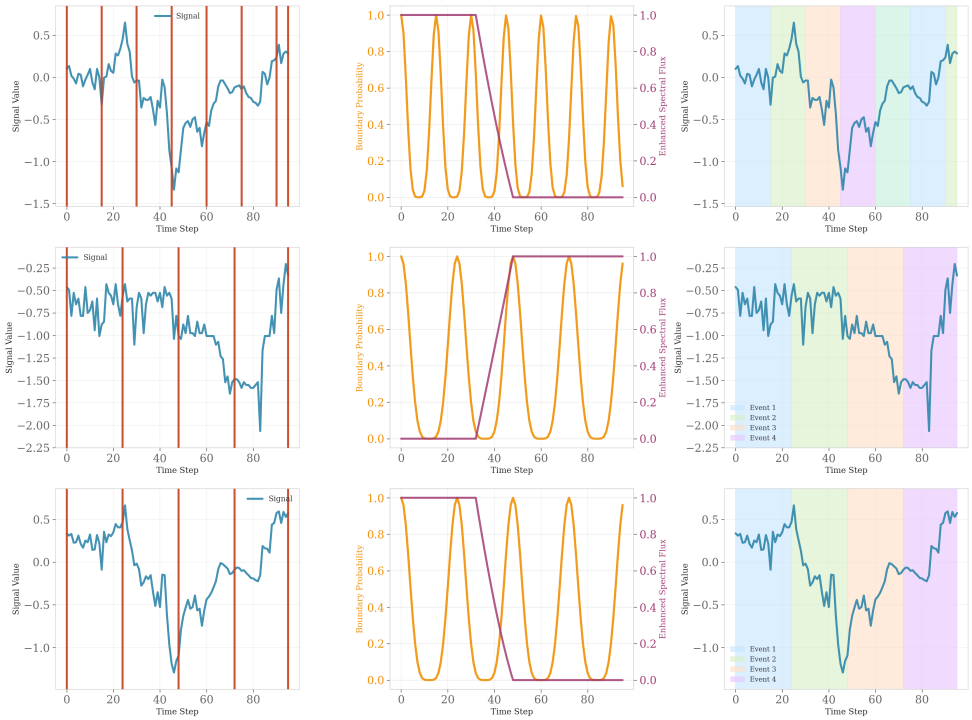



Figure 18: **ETTm1 periodic segmentation.** The proposal is shown in timestep coordinates for the minute-level series.

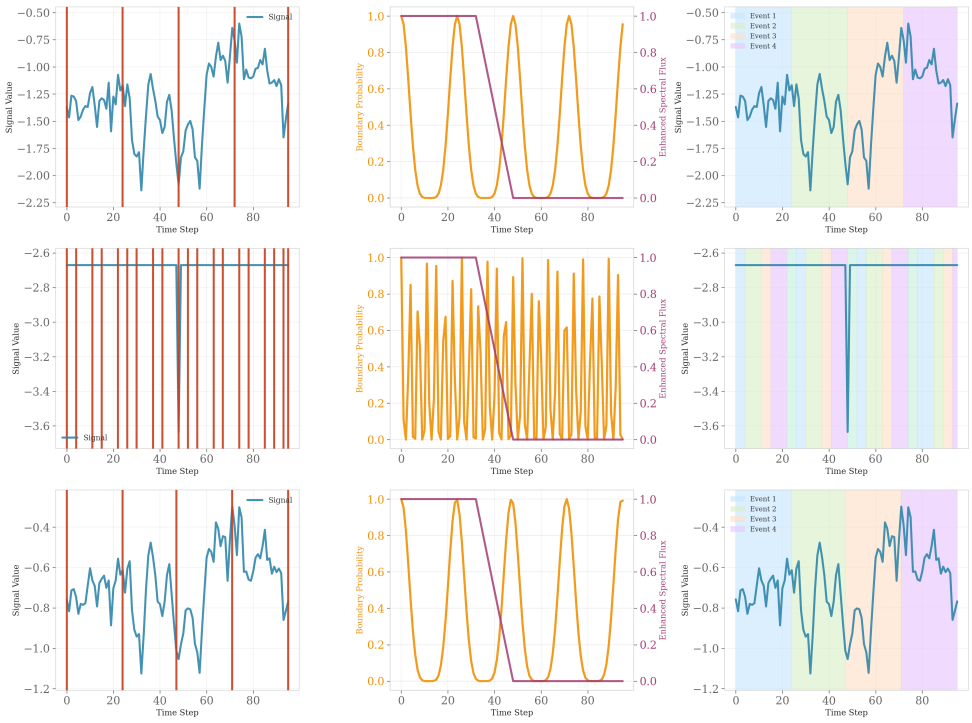



Figure 19: **ETTm2 periodic segmentation.** The periodic-weight peaks and segment boundaries illustrate the selected periodic scale.

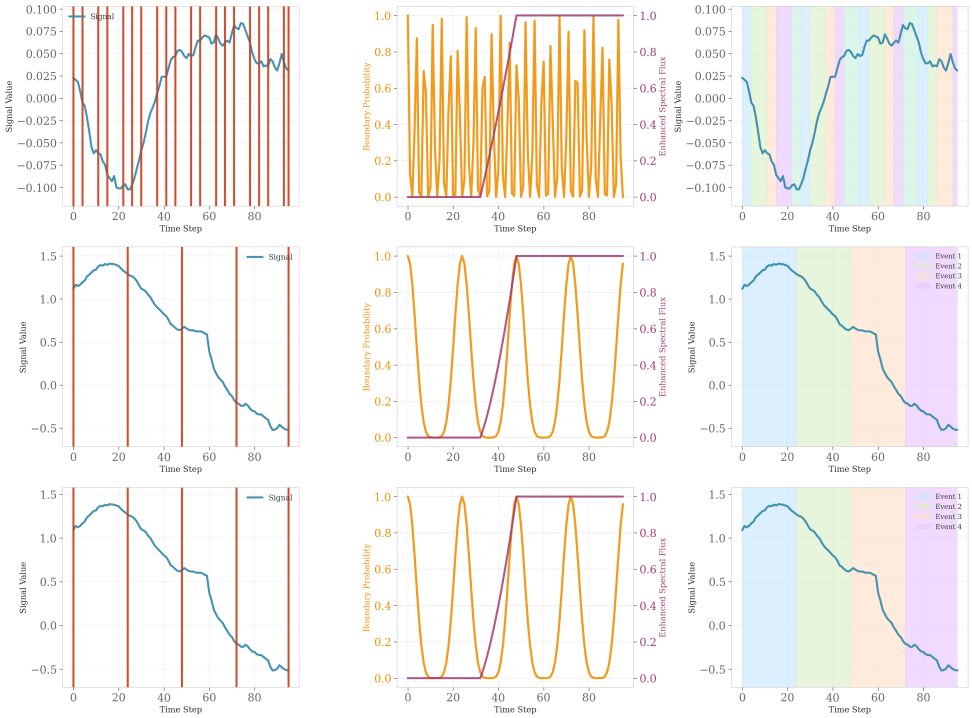



Figure 20: **Weather periodic segmentation.** The spectral-change statistic describes local variation alongside the periodic proposal.

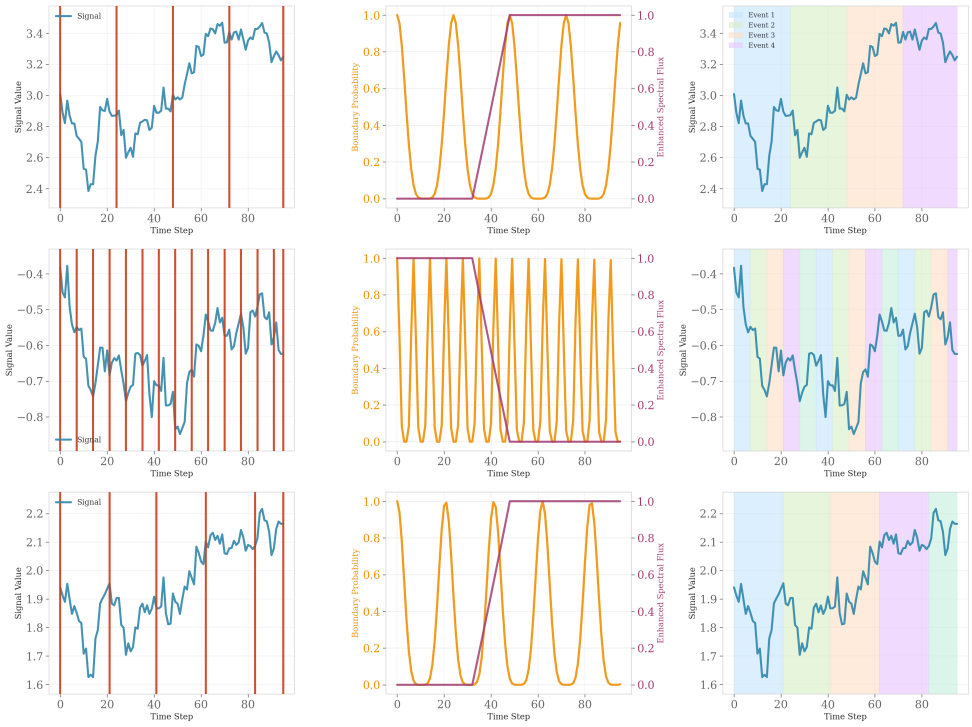



Figure 21: **Exchange periodic segmentation.** The proposed segments follow the cosine-comb spacing even when local variation differs within the window.

The displayed means correspond to the recorded width configurations and form a separate forecasting protocol from the MSL anomaly result in Figure 4(a).

### C.2 DA$^2$ Ablations and Alternative Mixing

We compare the learned mixing of temporal and channel attention with fixed mixtures and single-axis attention. Figure 24 compares the alternatives on forecasting.

In the displayed comparisons, the learned gate has lower MSE at many recorded settings, while differences among alternatives are small on some datasets. These comparisons evaluate the mixing rule applied to the compressed representations within the recorded variants.

**Layerwise gate trajectories.** Figure 25 shows the learned scalar $\pi$ during training. Each layer shares its mixture weight across inputs, and interior values give nonzero weight to both attention outputs.

The trajectory describes the learned gate; Figure 24 provides the corresponding mixing-rule comparisons.

### C.3 Reprojection Kernel Sensitivity Analysis

The temporal reprojection layer uses widths $\sigma_{n,c,i} = \alpha|S_{n,c,i}|$, where $|S_{n,c,i}|$ is the segment length and $\alpha$ controls how widely its representation is distributed over time. This proportional width makes the normalized Gaussian weights invariant to uniform time rescaling, as shown in Appendix F.6.

**Empirical sensitivity analysis.** Table 12 compares $\alpha \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$ on ETTh1, Weather, and Traffic at lookback $L = 96$ and prediction length $S = 96$.

The lowest displayed MSE occurs at $\alpha = 0.5$ on each dataset, and the reported standard deviations summarize the five width settings. The comparison evaluates the width rule; Appendix F.6 analyzes normalized mass crossing segment boundaries.

**Comparison with fixed $\sigma$.** On ETTh1, the fixed-width endpoints give MSE 0.389 at $\sigma = 0.3$ and 0.394 at $\sigma = 2.0$, with $\sigma$ measured in timesteps, compared with 0.365 for $\sigma_{n,c,i} = 0.5|S_{n,c,i}|$. These endpoints provide two fixed-width references for the segment-proportional rule.

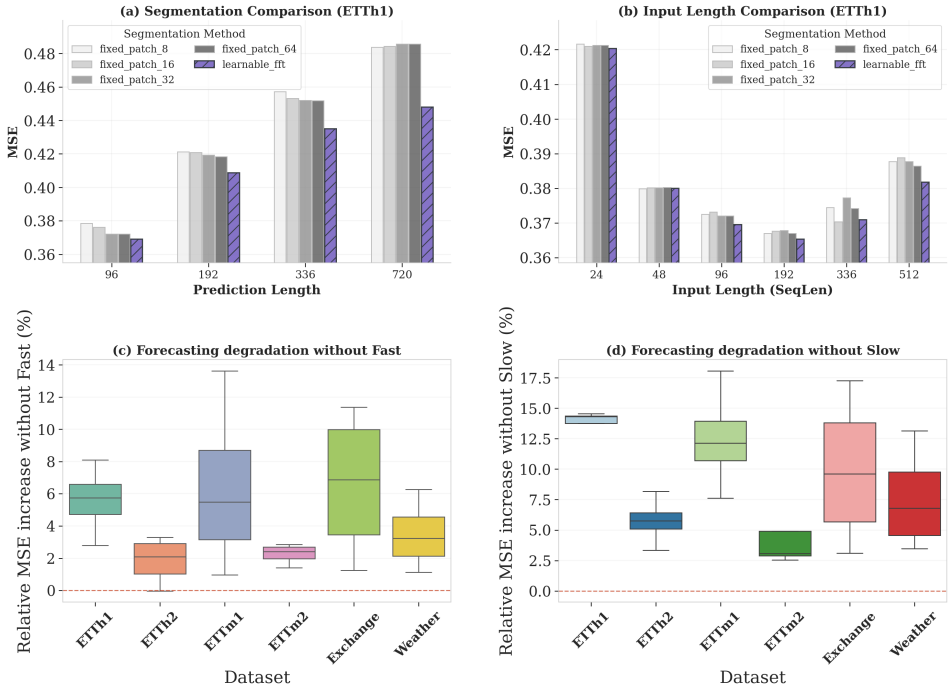



Figure 22: **Aggregation scale and pathway comparisons.** (a) ETTh1 forecasting MSE for periodic tokenization and fixed patch lengths. (b) Lookback sweep on ETTh1 at prediction length 96. (c,d) Relative MSE increases after removing the Fast Path or Slow Path, respectively, for six datasets (ETTh1, ETTh2, ETTm1, ETTm2, Exchange, Weather) and prediction lengths $\{96, 192, 336, 720\}$; ETTh2 is an auxiliary comparison outside the seven-dataset forecasting benchmark. Positive values indicate higher error after removal. The panels answer different questions and are interpreted within their stated comparisons.

Table 12: **Reprojection-width sensitivity.** Forecasting MSE across kernel scaling factors $\alpha$. The lowest displayed MSE occurs at $\alpha = 0.5$ on each dataset.

| **Dataset** | $\alpha = 0.3$ | $\alpha = 0.4$ | $\alpha = 0.5$ | $\alpha = 0.6$ | $\alpha = 0.7$ | **Std Dev** |
|---|---|---|---|---|---|---|
| ETTh1 | 0.371 | 0.368 | **0.365** | <u>0.367</u> | 0.372 | 0.003 |
| Weather | 0.182 | 0.179 | **0.177** | <u>0.178</u> | 0.183 | 0.002 |
| Traffic | 0.509 | <u>0.505</u> | **0.502** | 0.506 | 0.511 | 0.004 |
| **Average** | 0.354 | 0.351 | **0.348** | <u>0.350</u> | 0.355 | 0.003 |

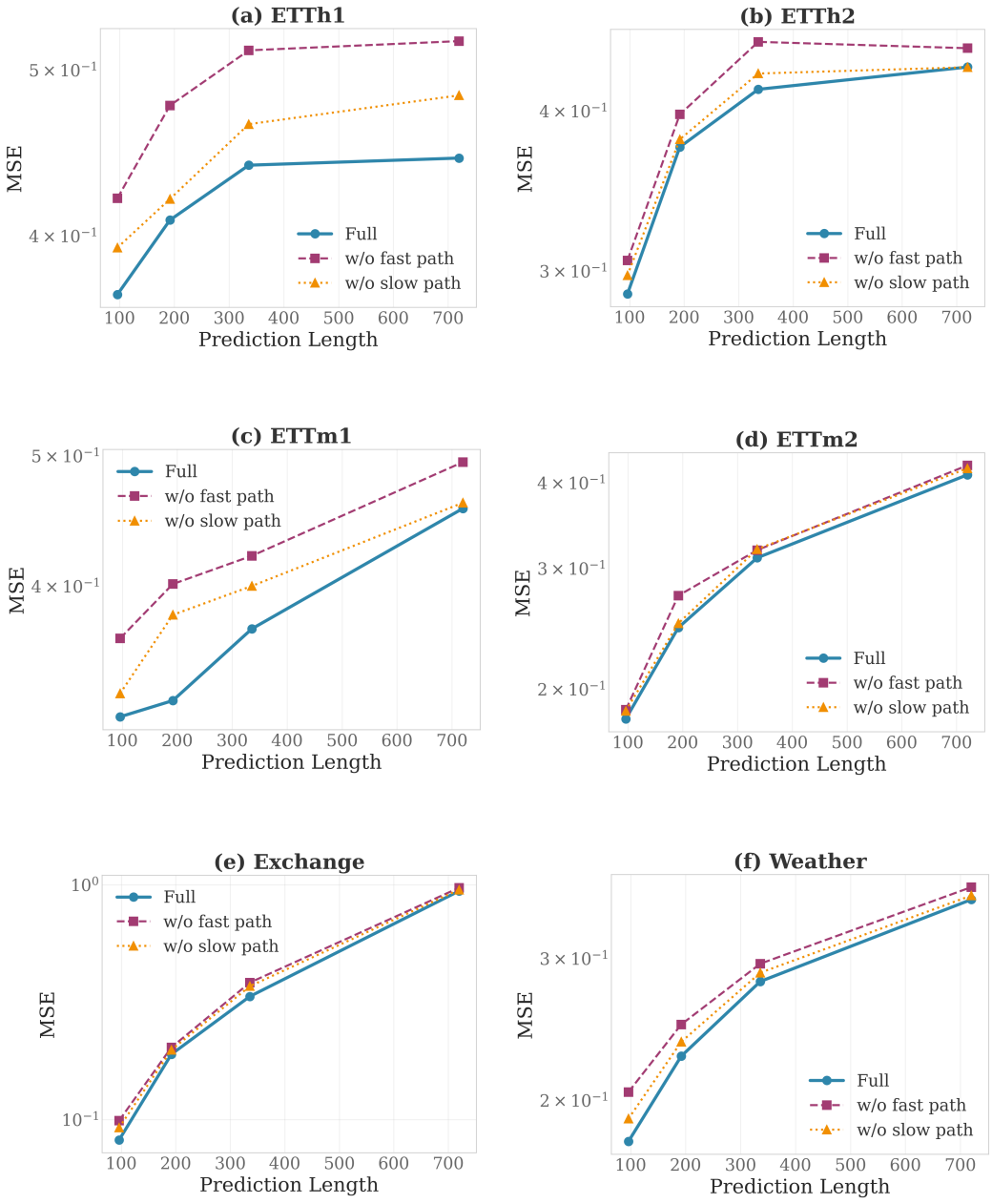



Figure 23: **Pathway ablations in forecasting.** MSE for the full model, Fast-only, and Slow-only variants. Lower values are better.

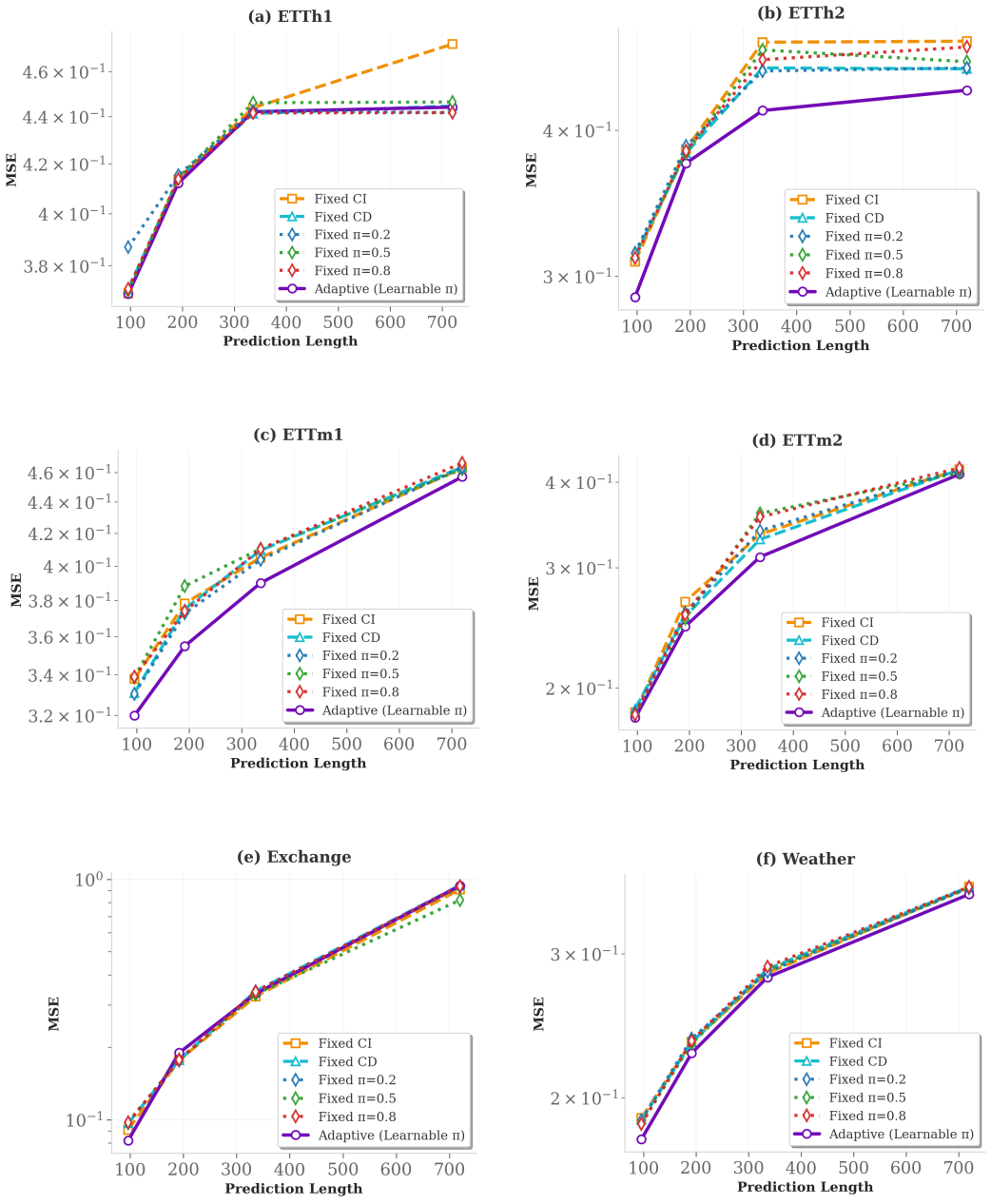



Figure 24: **Attention-axis mixing comparisons.** Forecasting MSE for temporal-only and channel-only attention, fixed mixtures, and a learned scalar mixture across datasets and prediction lengths.

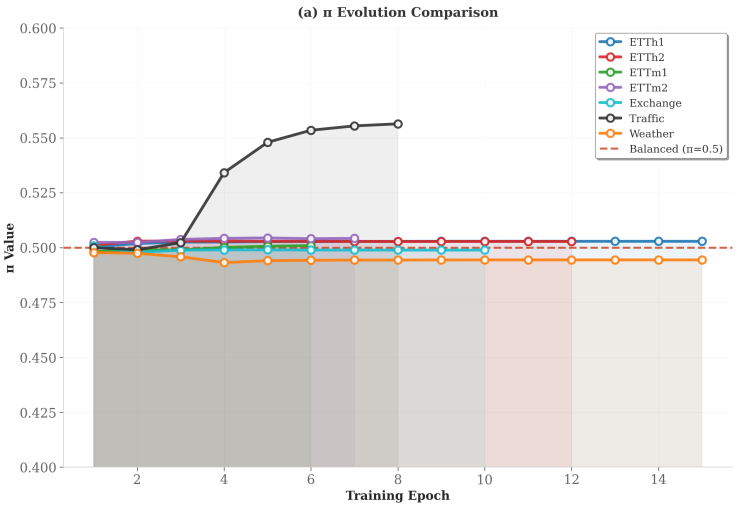



Figure 25: **Gate trajectories during training.** The layerwise scalar $\pi$ starts near an equal mixture and changes by different amounts across datasets.

The proportional rule interpolates at the scale of each segment, whereas fixed widths smooth uniformly regardless of segment duration.

### C.4 Dual-Pathway Visualization

The visualizations show the signal at both representation resolutions, its temporal reprojection, and the attention maps used to combine temporal and channel information.

## D Computational Trade-offs and Observation Scope

Hardware results are reported within each measurement setting. Missing-observation experiments test the regular-grid assumption of the spectral branch.

### D.1 Profiled Latency and Component Profiles

#### D.1.1 Short-Context Profile on RTX 3090

The ETTh1 profile reports per-sample latency, GPU memory, and effective token counts. Mean token count measures compression, whereas the padded maximum sets the attention tensor dimensions.

Table 13: **ETTh1 hardware and token-count profile.** Profiled per-sample latency, GPU memory, and mean effective token count at batch size 32 on an RTX 3090. $\bar{M}$ denotes the mean count; $M$ is the padded maximum.

| Lookback | Latency (ms) | Memory (MB) | Tokens ($\bar{M}$) | $\bar{M}/L$ |
|---|---|---|---|---|
| 96 | 31.30 | 139 | 5.57 | 0.058 |
| 192 | 62.59 | 215 | 12.43 | 0.065 |
| 384 | 70.80 | 406 | 19.29 | 0.050 |
| 768 | 76.80 | 640 | 33.00 | 0.043 |
| 1536 | 85.75 | 1539 | 74.14 | 0.048 |

#### D.1.2 Long-Context Inference on H200

The H200 comparison reports latency and memory as lookback increases at fixed model width. Figure 28 compares model-only forward-pass latency and peak GPU memory on an H200 at batch size 32; preprocessing and postprocessing are excluded. PatchTST uses patch length 8 and stride 4, and both neural models use $d_{\text{model}} = 128$ with six layers. Panels (a) and (b) show absolute measurements through $L = 8192$. Panel (c) reports PatchTST/PeCo-TS ratios separately for latency and memory, where a ratio above one favors PeCo-TS for that metric.

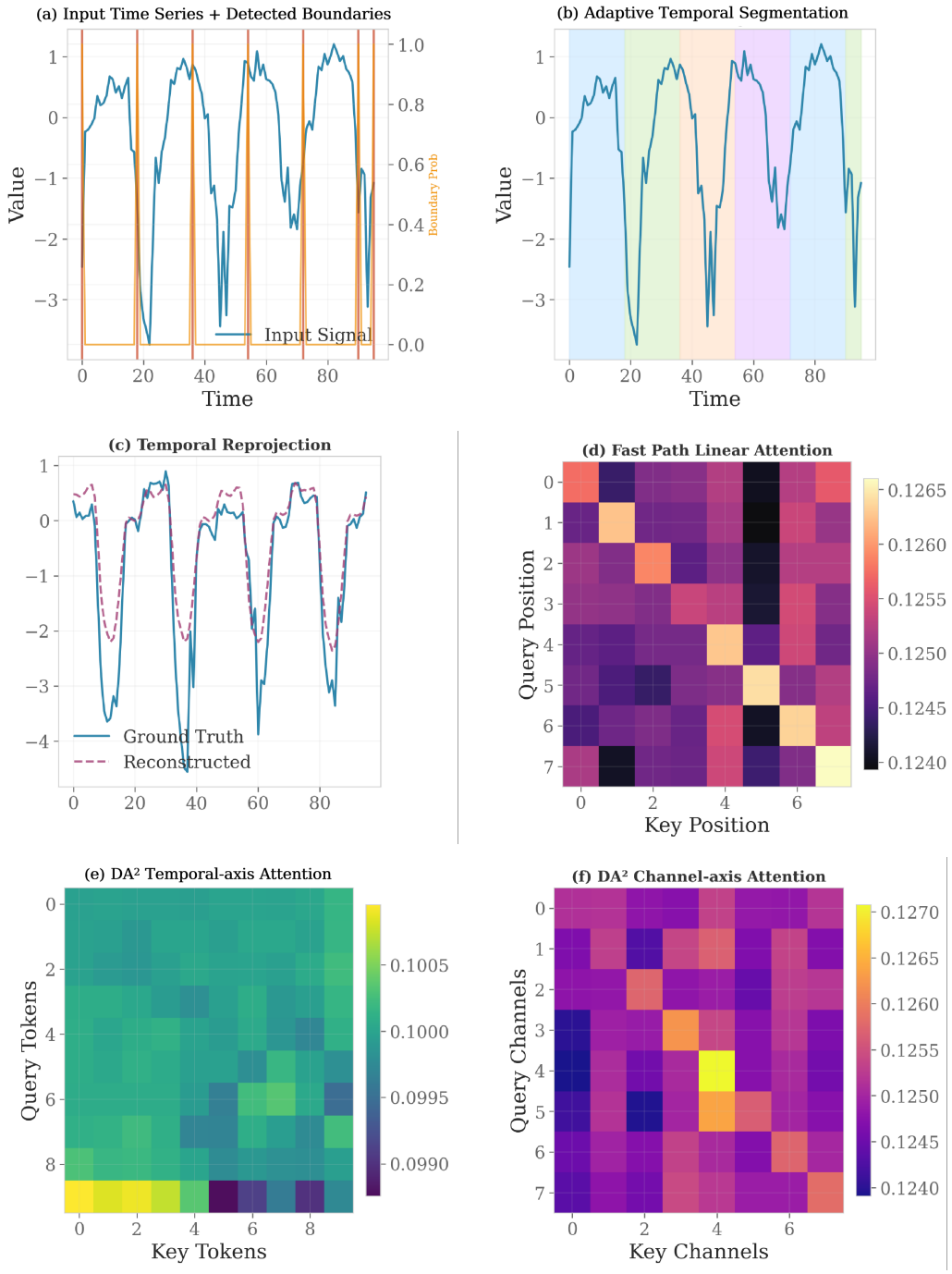



Figure 26: **Dual-pathway visualization.** (a,b) Input signal with proposed boundaries and segments. (c) Temporal reprojection to full resolution. (d) Fast Path attention map. (e,f) Slow Path token-axis and channel-axis attention maps.

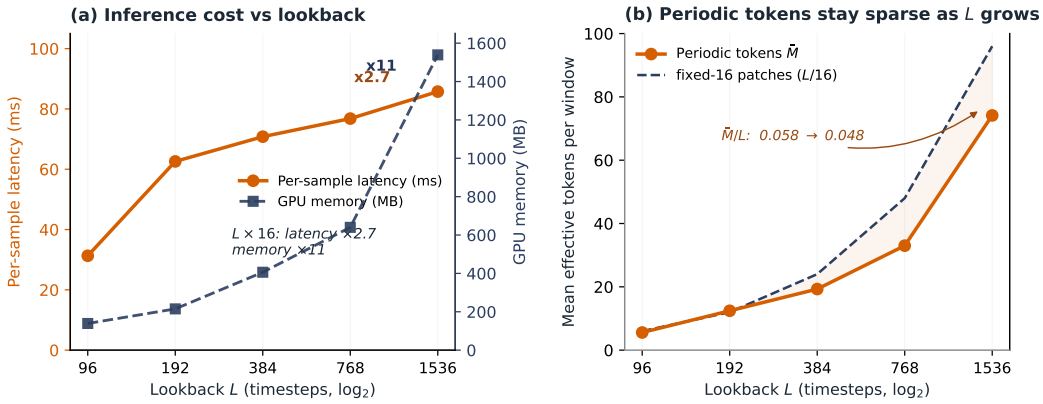



Figure 27: **Inference cost and effective compression versus lookback (ETTh1 profile, RTX 3090, batch 32).** (a) Per-sample latency and GPU memory as the lookback grows $96 \rightarrow 1536$; (b) mean effective tokens per window compared with the fixed-16 reference. The profile shows a small token grid but a dominant full-resolution component.

### D.1.3 Component Profiles

The component profiles separate the work performed by segmentation, the Fast Path, the Slow Path, and reprojection with the task head. They describe within-profile allocations; their totals are kept separate from the per-sample latency summary in Table 13.

## D.2 FastLite Efficiency Variants

We characterize the latency–accuracy operating points of narrower Fast Path widths. Figure 30 compares Fast Path widths 16, 8, and 4 at four lookbacks. Widths 8 and 4 have lower latency than width 16 at all four lookbacks; the latency ordering changes at $L = 1536$ and the MSE ordering changes with context length. At $L = 1536$, narrowing to width 8 raises MSE from 0.405 to 0.427. At $L = 3072$, width 4 reports 21.0 ms latency and 0.498 MSE, compared with 30.6 ms and 0.511 for width 16.

These measurements identify latency–accuracy operating points within this variant family. MSE changes with lookback as the Fast Path narrows.

Table 14: **Fast Path width and lookback.** Latency (ms) / forecasting MSE for the same operating points as Figure 30. The narrower variants reduce latency relative to width 16; their error ordering varies with lookback.

| Lookback | Width 16 | Width 8 | Width 4 |
|---|---|---|---|
| 384 | 9.5 / **0.388** | 8.4 / **0.388** | **7.8** / 0.411 |
| 768 | 19.0 / 0.392 | 16.9 / **0.388** | **14.4** / 0.414 |
| 1536 | 23.0 / 0.405 | **14.7** / 0.427 | 16.6 / **0.400** |
| 3072 | 30.6 / 0.511 | 25.8 / 0.522 | **21.0** / **0.498** |

## D.3 Effect of Lookback Length

Table 15 compares lookbacks $L = 96$ and $L = 720$ on ETTh1 and Traffic. The displayed MSE decreases by 3.4% and 4.1%, respectively.

Increasing lookback changes the available historical context and the number of observations, so the comparison varies lookback and tokenization together.

## D.4 Missing Observations on a Regular Grid

**Sampling assumptions.** The FFT segmenter operates on a regular time grid. For missing observations on this grid, linear interpolation provides values before spectral analysis. Irregular timestamps require spectral estimation, segment duration, and reprojection defined in elapsed-time coordinates.

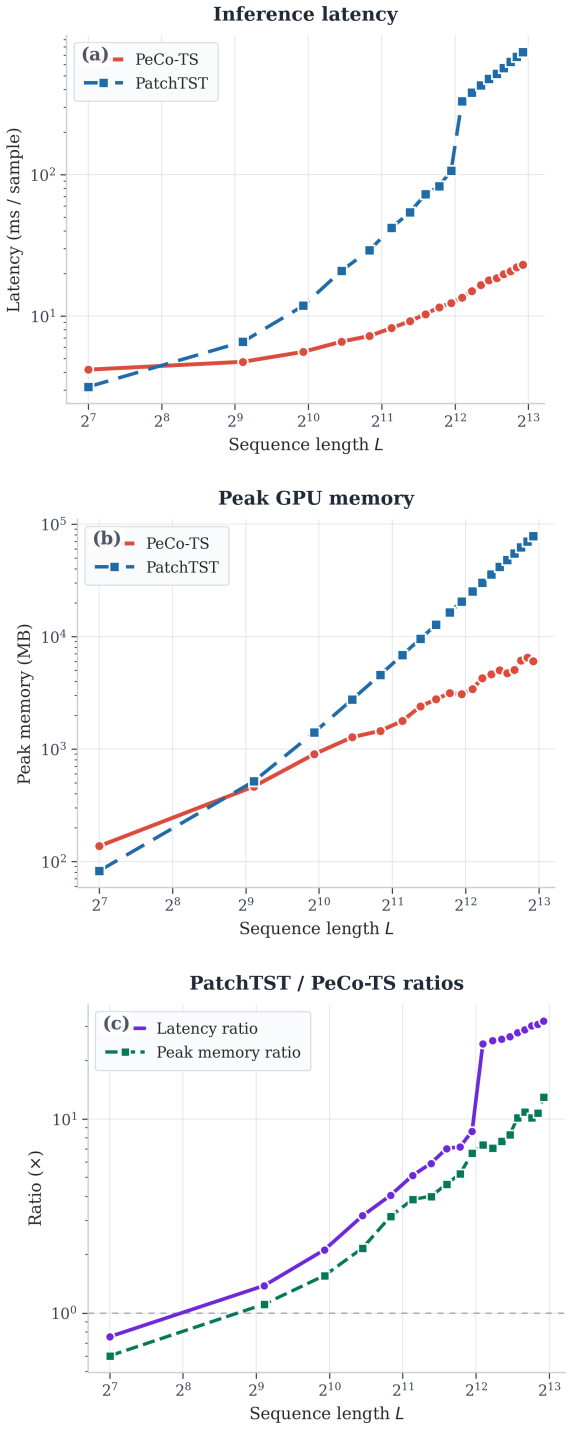



Figure 28: **Long-context forward-pass inference comparison (H200, batch size** 32**).** (a,b) Model-only forward-pass latency and peak GPU memory as lookback $L$ increases; preprocessing and postprocessing are excluded. (c) PatchTST/PeCo-TS ratios for each metric; values above one favor PeCo-TS. Width is 128 with six layers, and PatchTST uses patch length 8 and stride 4.

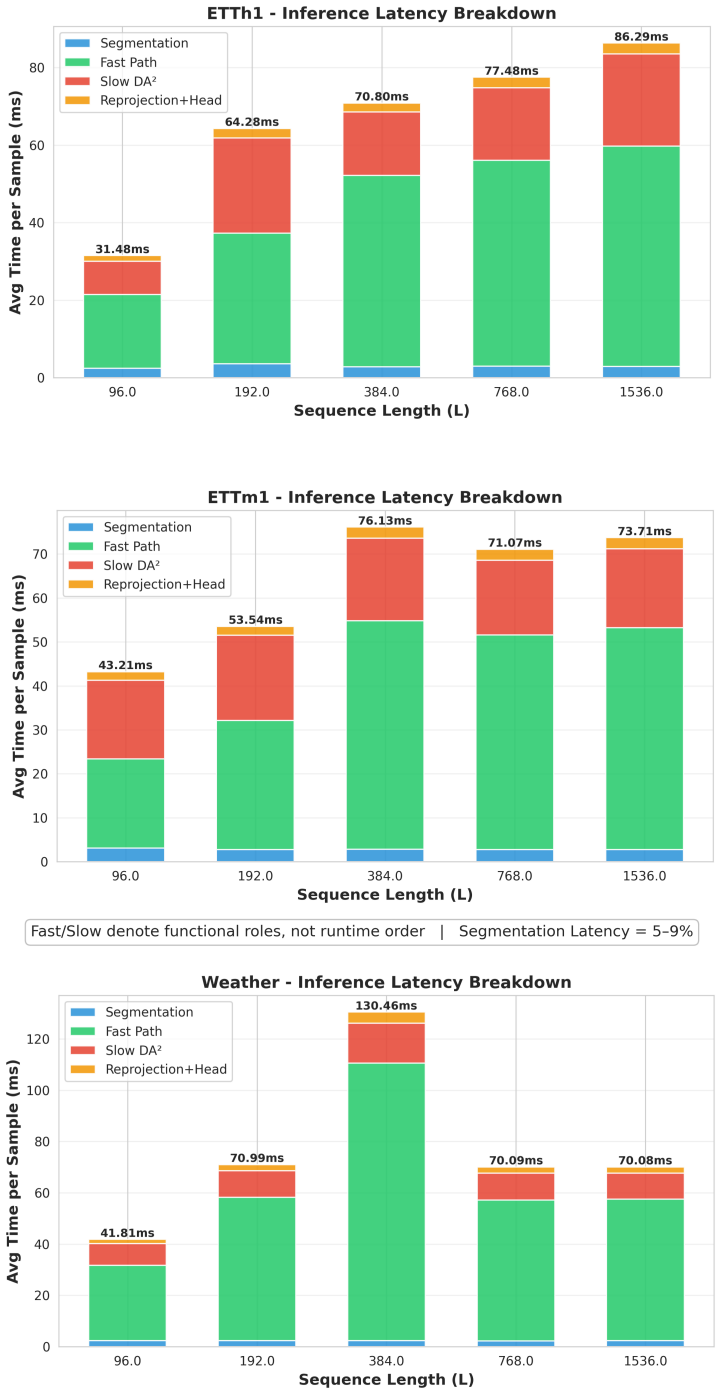
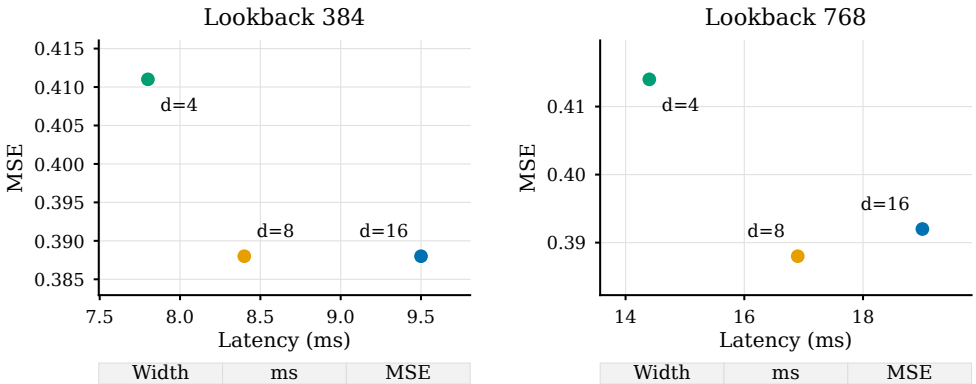
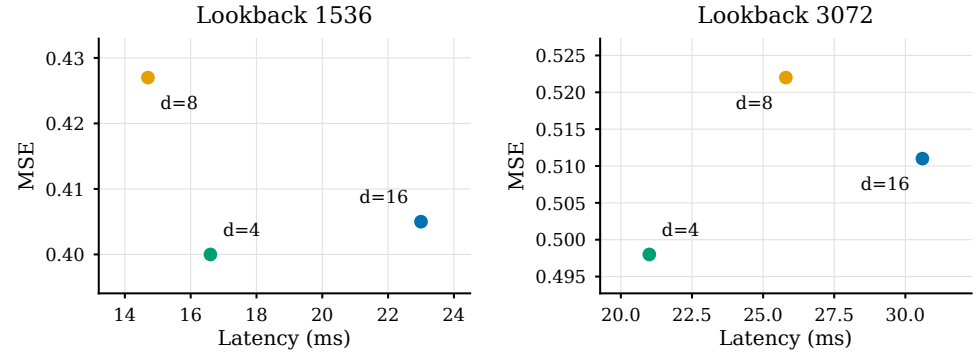



Figure 29: **Component profiles across three datasets.** The ETTh1, ETTm1, and Weather panels separate segmentation, Fast Path, Slow Path, and reprojection with the task head. Profiles are measured at batch size 32 on an RTX 3090, and the ETTh1 panel sweeps lookback 96–1536. Table 13 reports per-sample latency separately from these component totals.




| Width | ms | MSE |
|---|---|---|
| 16 | 9.5 | 0.388 |
| 8 | 8.4 | 0.388 |
| 4 | 7.8 | 0.411 |

| Width | ms | MSE |
|---|---|---|
| 16 | 19.0 | 0.392 |
| 8 | 16.9 | 0.388 |
| 4 | 14.4 | 0.414 |




| Width | ms | MSE |
|---|---|---|
| 16 | 23.0 | 0.405 |
| 8 | 14.7 | 0.427 |
| 4 | 16.6 | 0.400 |

| Width | ms | MSE |
|---|---|---|
| 16 | 30.6 | 0.511 |
| 8 | 25.8 | 0.522 |
| 4 | 21.0 | 0.498 |

Figure 30: **Fast Path width comparisons.** Latency and forecasting MSE for widths 16, 8, and 4 at each lookback. Each panel plots latency against MSE and lists all three operating points below. Lower values are better on both axes. Prediction error varies with width and lookback.

Table 15: **Lookback comparison on ETTh1 and Traffic.** MSE at $L = 96$ and $L = 720$ at prediction length $S = 96$, with relative reduction computed against $L = 96$.

| **Dataset** | **L=96 (MSE)** | **L=720 (MSE)** | **Reduction (%)** |
|---|---|---|---|
| ETTh1 | 0.3695 | **0.3569** | 3.4% |
| Traffic | 0.5061 | **0.4853** | 4.1% |

### D.4.1 Random Deletion and Interpolation

Table 16 evaluates ETTh1 after randomly removing 10%, 20%, 30%, or 50% of observations and filling the gaps by linear interpolation. The table reports forecasting MSE at four prediction lengths. This setup measures tolerance to missing entries on an underlying regular grid.

Table 16: **ETTh1 forecasting with missing observations.** Randomly removed entries are filled by linear interpolation. Parentheses report relative MSE increases against the complete-input result at the same prediction length.

| PredLen | Regular | 10% Miss | 20% Miss | 30% Miss | 50% Miss |
|---|---|---|---|---|---|
| 96 | **0.3695** | 0.3750 (+1.5%) | 0.3787 (+2.5%) | 0.3753 (+1.6%) | 0.3976 (+7.6%) |
| 192 | **0.4095** | 0.4129 (+0.8%) | 0.4170 (+1.8%) | 0.4137 (+1.0%) | 0.4390 (+7.2%) |
| 336 | **0.4402** | 0.4443 (+0.9%) | 0.4461 (+1.3%) | 0.4455 (+1.2%) | 0.4608 (+4.7%) |
| 720 | **0.4448** | 0.4513 (+1.5%) | 0.4532 (+1.9%) | 0.4607 (+3.6%) | 0.4692 (+5.5%) |

At 30% missingness, MSE increases by 1.0–3.6% across the four prediction lengths. At 50% missingness, the increases are 4.7–7.6%.

The larger errors at 50% missingness quantify the accuracy cost of discarding half the observations and reconstructing them by interpolation on this grid.

### D.4.2 FFT and Lomb–Scargle Comparison

Lomb–Scargle estimates periodic structure from unevenly spaced observations (Lomb, 1976; Scargle, 1982). Table 17 compares an FFT segmenter and a Lomb–Scargle variant after removing 10% or 30% of ETTh1 observations. Lomb–Scargle has higher MSE at prediction length 96 and lower MSE at 192 and 336, with 11.59–13.06 ms inference time per sample versus 2.09–5.80 ms for Learnable FFT. The two tables use the same Learnable FFT reference values at the overlapping settings.

Continuous-time neural ODEs offer representations evaluated at arbitrary timestamps (Rubanova et al., 2019; Chen et al., 2018). The present comparison evaluates random deletion on an underlying regular grid with interpolation and separately measures a Lomb–Scargle segmenter for uneven observations.

Table 17: **FFT and Lomb–Scargle segmentation with missing ETTh1 observations.** MSE and inference time per sample at two missingness levels and three prediction lengths. The Learnable FFT entries match the corresponding Table 16 entries. Lomb–Scargle is slower in each within-table comparison.

| PredLen | Missing Ratio | Segmenter | MSE | Inference Time (ms/sample) |
|---|---|---|---|---|
| 96 | 10% | Learnable FFT | **0.3750** | **2.09** |
| 96 | 10% | Lomb-Scargle | 0.3814 | 11.81 |
| 96 | 30% | Learnable FFT | **0.3753** | **2.31** |
| 96 | 30% | Lomb-Scargle | 0.3820 | 11.96 |
| 192 | 10% | Learnable FFT | 0.4129 | **2.92** |
| 192 | 10% | Lomb-Scargle | **0.4119** | 13.06 |
| 192 | 30% | Learnable FFT | 0.4137 | **4.62** |
| 192 | 30% | Lomb-Scargle | **0.4118** | 12.35 |
| 336 | 10% | Learnable FFT | 0.4443 | **5.80** |
| 336 | 10% | Lomb-Scargle | **0.4427** | 11.59 |
| 336 | 30% | Learnable FFT | 0.4455 | **2.12** |
| 336 | 30% | Lomb-Scargle | **0.4414** | 12.50 |

## E Extended Related Work

**Cognitive inspiration for temporal abstraction.** Fast–slow processing motivates a division between a rapid, parallel mode and a more deliberative mode (Kahneman, 2011; Booch et al., 2021), and event perception examines how continuous activity is organized into temporal units that support perception and memory (Zacks et al., 2007). PeCo-TS uses this organizational analogy for its architecture: the Fast Path constructs contextualized features at timestep resolution, segment tokens form

the temporal units, and the Slow Path performs relational processing over them. Section 1 states the corresponding technical design principles.

**General-purpose and cross-dataset models.** Chronos trains a single forecasting model across many datasets through tokenization and scaling (Ansari et al., 2024), while UniTS unifies multiple time-series tasks through task tokenization (Gao et al., 2024). Tokenization and normalization also shape interfaces between time series and language models (Tao et al., 2025; Yang, 2025). PeCo-TS studies temporal resolution within a task-trained architecture, and its forecasting, classification, imputation, and anomaly-detection evaluations examine how one representation design serves different prediction objectives.

# F Technical Derivations

The identities below characterize the specified operators under their stated assumptions; predictive behavior is assessed in the trained-model experiments.

## F.1 Computation on Compressed Representations

Let $L$ denote the lookback, $M_{n,c}$ the realized segment count for sample $n$ and channel $c$, $C$ the number of channels, and $d_h$ the Slow Path width. Pooling pads each batch to $M = \max_{n,c} M_{n,c}$ for tensor computation.

At this padded width, token-axis attention costs $O(CM^2 d_h)$, whereas timestep attention at the same width costs $O(CL^2 d_h)$. The ratio of temporal pair counts is $(M/L)^2$. Channel-axis attention adds $O(MC^2 d_h)$, giving a DA$^2$ attention cost of

$$O\big((CM^2 + MC^2)d_h\big). \tag{11}$$

The remaining computation includes the FFT ($O(CL \log L)$), Fast Path linear attention ($O(CLd_f^2)$), feature projections and feedforward blocks, and Gaussian reprojection. Evaluating the dense reprojection in Equation 7 and aggregating all segment features costs $O(CLMd_h)$.

These expressions count tensor operations at the padded segment width; batch size multiplies each cost, and repeated layers contribute their corresponding layer counts. Compression changes the allocation of computation, while end-to-end latency also depends on the full-resolution stream and the implementation (Appendix D.1).

**What pooling changes.** Pooling retains a segment mean but changes the unit over which attention normalizes. For a fixed query, suppose every key and value is constant within segment $j$, with logit $a_j$, value $v_j$, and length $n_j$. Full-resolution softmax attention then equals

$$\frac{\sum_j n_j \exp(a_j) v_j}{\sum_j n_j \exp(a_j)} = \sum_j \mathrm{softmax}_j(a_j + \log n_j) v_j. \tag{12}$$

This identity follows by grouping identical timestep terms within each segment. Ordinary segment attention normalizes over one representation per segment. The two coincide for equal-length segments under these assumptions; for unequal lengths, the segment representation changes the weighting from timestep multiplicity to segment membership. The main-text model uses the stated segment-level normalization.

**Segment multiplicity.** Consider two segments of lengths $n_1 = 3$ and $n_2 = 1$, with constant key logits $\ell_1 = \ell_2 = 0$ and constant values $v_1, v_2$ within each segment. Full-resolution softmax attention gives $(3v_1 + v_2)/4$, whereas attention over the two pooled tokens gives $(v_1 + v_2)/2$. The first expression weights segments by their timestep multiplicities; the second assigns equal mass to the two segment tokens. With equal segment lengths, both expressions reduce to $(v_1 + v_2)/2$. Thus pooling changes the measure over temporal units even when each segment is internally constant.

## F.2 Theoretical Properties of the Fast Path Linear Attention

The Fast Path employs linear attention with $O(Ld_f^2)$ cost per sample and channel, linear in $L$ at fixed head count, while retaining timestep-level features.

**Feature map.** Following Katharopoulos et al. (2020), we use $\phi(x) = \mathrm{elu}(x) + 1$, which produces a nonnegative attention kernel. Matrix multiplication can be reassociated as

$$\big(\phi(Q)\phi(K)^\top\big)V = \phi(Q)\big(\phi(K)^\top V\big), \tag{13}$$

allowing PeCo-TS to compute $\phi(K)^\top V$ first (complexity $O(Ld_f^2)$) and then multiply by $\phi(Q)$, avoiding the $O(L^2)$ cost of the full attention matrix.

**Full-resolution representation.** The Fast Path produces a feature at every input timestep. Segment pooling supplies a compressed representation to the Slow Path, while fusion retains access to the unpooled Fast Path features. Maintaining both resolutions gives the task head access to information before and after pooling.

### F.3 Periodic Scale and Tokenization Gradients

The detector combines spectral smoothing with soft frequency selection and cosine-comb weighting. The temperature $\tau$ controls spectral concentration, and the sharpness $\gamma$ controls periodic-weight concentration around periodic peaks. Algorithm 1 gives the stated settings.

**Spectral smoothing.** The kernel $g_\theta$ convolves the power spectrum along the frequency axis before soft frequency selection. Its parameters shape the spectrum from which the effective period is computed.

**Soft frequency selection.** The softmax distribution

$$a_f = \frac{\exp(P_{\mathrm{sm}}(f)/\tau)}{\sum_{f' \in \mathcal{F}_L} \exp(P_{\mathrm{sm}}(f')/\tau)} \tag{14}$$

is differentiable for $\tau > 0$. Lower temperature concentrates mass on larger smoothed spectral values, while higher temperature distributes it more broadly. The centroid $k_{\mathrm{eff}} = \sum_{f \in \mathcal{F}_L} a_f f$ converts these weights into a single segmentation scale.

**Spectral concentration and effective period.** For a fixed smoothed spectrum with a unique maximum at $m \in \mathcal{F}_L$, the ratio of any competing softmax weight to $a_m$ satisfies

$$\frac{a_f}{a_m} = \exp\left(\frac{P_{\mathrm{sm}}(f) - P_{\mathrm{sm}}(m)}{\tau}\right) \longrightarrow 0 \quad (f \neq m,\ \tau \to 0^+).$$

Consequently $k_{\mathrm{eff}} \to m$ and $T \to L/m$. This limit relates spectral concentration to the spacing of the continuous periodic proposal; integer peak selection still follows Algorithm 1.

For a flat smoothed spectrum, all $K = \lfloor L/2 \rfloor$ positive frequencies receive equal weight. The effective frequency and period are then

$$k_{\mathrm{eff}} = \frac{1}{K}\sum_{f=1}^{K} f = \frac{K+1}{2}, \qquad T = \frac{2L}{K+1}.$$

At $L = 96$, this gives $T = 192/49 \approx 3.92$ timesteps. Even a diffuse spectrum can yield a short effective period; the spectrum distribution supplies the distinction.

**Cosine-comb weighting.** The exponent $\gamma$ changes the concentration of the periodic weights around their peaks. The effective period determines peak spacing. Thus periodic-weight sharpness and segmentation scale have distinct roles in the detector.

**Visualization.** The visualization panels in Figures 15–21 show the signal, a post-hoc spectral-change statistic, cosine-comb weights, and segment boundaries. The periodic weights illustrate the period selected from the lookback spectrum; the spectral-change statistic describes local signal variation separately.

**Tokenization gradients under a fixed partition.** The forward pass applies NMS to $p(t)$ to produce binary boundaries; backpropagation treats this discrete operation as fixed. For the default $\gamma = 2$, the periodic weight is $p(t) = \cos^4(\pi t/T)$, which is smooth for $T > 0$, and

$$\frac{\partial p(t)}{\partial T} = \frac{4\pi t}{T^2}\cos^3\left(\frac{\pi t}{T}\right)\sin\left(\frac{\pi t}{T}\right). \tag{15}$$

For bounded $t$ and $T$ bounded away from zero, this derivative is bounded. The periodic weights provide a differentiable path from the effective period to the reprojection layer. As in Section 3.2, this path updates $g_\theta$ with fixed segment assignments.

### F.4 Masked softmax with padding

Let $G \in \mathbb{R}^{N \times N}$ be attention logits and $m \in \{0,1\}^N$ indicate valid tokens. Write $\mathcal{V} = \{k : m_k = 1\}$ for the valid-key set. For $\mathcal{V} \neq \emptyset$, let $a_i = \max_{k \in \mathcal{V}} G_{ik}$. Define

$$W_{ij} = \begin{cases} \dfrac{\exp(G_{ij} - a_i)}{\sum_{k \in \mathcal{V}} \exp(G_{ik} - a_i)}, & m_i = m_j = 1,\ \mathcal{V} \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases}$$

Valid query rows sum to one over valid keys. Padded keys receive zero mass, and padded queries have zero output. The empty-key branch returns zero without evaluating a softmax. Subtracting $a_i$ leaves the normalized weights unchanged and bounds each evaluated exponential by one. The token-axis mask ranges over segments within a channel; the channel-axis mask ranges over channels at one segment ordinal.

### F.5 Properties of DA$^2$ Mixing

For temporal-axis output $\tilde{z}_{\text{time}}$ and channel-axis output $\tilde{z}_{\text{chan}}$, the scalar mixture is

$$Y(\pi) = \pi \tilde{z}_{\text{chan}} + (1-\pi)\tilde{z}_{\text{time}}. \tag{16}$$

At the endpoints of the closed interval $[0,1]$, $Y(0) = \tilde{z}_{\text{time}}$ and $Y(1) = \tilde{z}_{\text{chan}}$. These are limiting cases of the interior gate used in the main text. The endpoints select the attention contribution of one axis; the input representations and residual connections remain part of the model.

For any norm and fixed $\pi \in [0,1]$, the triangle inequality gives

$$\|Y(\pi)\| \le (1-\pi)\|\tilde{z}_{\text{time}}\| + \pi\|\tilde{z}_{\text{chan}}\|. \tag{17}$$

If the two attention maps are $\mathcal{L}_c$- and $\mathcal{L}_m$-Lipschitz on a specified input domain, their fixed scalar mixture is at most $((1-\pi)\mathcal{L}_c + \pi\mathcal{L}_m)$-Lipschitz on that domain. This follows by applying the same inequality to the output difference for two inputs.

The scalar gate interpolates between the two branch outputs, and its output norm never exceeds the convex combination of the branch norms. Its predictive value is assessed by the mixing comparisons in Appendix C.2.

### F.6 Gaussian Reprojection and Cross-Segment Mixing

For fixed segment centers, positive widths, nonnegative periodic weights $\bar{p}_{n,c,i}$, and at least one positive weight, the Gaussian weights in Equation 7 are nonnegative and sum to one. Consequently, for any norm,

$$\|h_{\text{full}}(n,c,t)\| \le \sum_i \tilde{w}_{t,i,c}\|z_{\text{slow}}(n,c,i)\| \le \max_i \|z_{\text{slow}}(n,c,i)\|. \tag{18}$$

Reprojection produces a convex mixture of the Slow Path representations at each timestep.

**Interpolation error.** Suppress the batch and channel indices, let $s$ be the segment containing $t$, and write $z_i = z_{\text{slow}}(n,c,i)$ and $h_t = h_{\text{high}}(n,c,t)$. Since the weights sum to one,

$$h_{\text{full}}(t) - h_t = (z_s - h_t) + \sum_{j \ne s} \tilde{w}_{t,j}(z_j - z_s). \tag{19}$$

The triangle inequality gives

$$\|h_{\text{full}}(t) - h_t\| \le \|z_s - h_t\| + \sum_{j \ne s} \tilde{w}_{t,j}\|z_j - z_s\|. \tag{20}$$

The first term measures how the containing segment's representation differs from the timestep feature; the second measures mixing with other segments. A global Gaussian assigns positive weight to every segment with a positive periodic weight. Its width controls a smoothing trade-off, including near boundaries. The Fast Path supplies an unpooled representation alongside this interpolated context in the final fusion.

**Uniform time rescaling.** Let $D_i$ denote segment duration in the chosen time coordinates; on the unit-spaced grid, $D_i = |S_i|$. Under a change of time units $t' = at$, $\mu_i' = a\mu_i$, and $D_i' = aD_i$ for $a > 0$, the width rule $\sigma_i = \alpha D_i$ gives

$$\frac{(t' - \mu_i')^2}{(\alpha D_i')^2} = \frac{(t-\mu_i)^2}{(\alpha D_i)^2}. \tag{21}$$

With fixed periodic weights, the normalized Gaussian weights are unchanged. Changing units scales duration while leaving the integer number of samples in a segment unchanged. Resampling the observed signal can change the detected segments and their periodic weights.

**Unnormalized kernel decay.** For the Gaussian kernel $G_i(t) = \exp(-(t-\mu_i)^2/(2\sigma_i^2))$ with $\sigma_i = \alpha|S_i|$, its value at distance $\delta|S_i|$ relative to its own center is

$$\frac{G_i(\mu_i + \delta|S_i|)}{G_i(\mu_i)} = \exp\left(-\frac{\delta^2}{2\alpha^2}\right).$$

At $\alpha = 0.5$, distances of one and two segment lengths give $e^{-2} \approx 0.1353$ and $e^{-8} \approx 3.35 \times 10^{-4}$, respectively. These ratios describe one unnormalized kernel; the normalized fraction of context received from another segment also depends on the other segment centers, widths, and periodic weights. The global Gaussian has no finite cutoff.