IMTS-Tokenizer: Time-Aware Tokenization for Irregular Multivariate Time Series Forecasting

Bin Xu*
School of Software
Shandong University
Jinan, China
binxu@link.cuhk.edu.cn

Yinghua Li*
School of Software
Shandong University
Jinan, China
liyinghua@mail.sdu.edu.cn

Linqi Han
School of Software
Shandong University
Jinan, China
linqihan@tju.edu.cn

Xiaoyu Li
The Hong Kong Polytechnic University
Hong Kong SAR
xiao-y.li@connect.polyu.hk

Xinghao Yang
China University of Petroleum (East China)
Qingdao, China
yangxh@upc.edu.cn

Wei Liu
University of Technology Sydney
Sydney, Australia
wei.liu@uts.edu.au

Yongshun Gong†
Shandong University
Jinan, China
ysgong@sdu.edu.cn

Abstract

Modeling Irregular Multivariate Time Series (IMTS) poses significant challenges due to asynchronous sampling and data sparsity. While existing methods focus on handling temporal irregularity after encoding, the tokenization stage itself remains underexplored. We propose IMTS-Tokenizer, a time aware tokenization framework built on the Irregular-time-aware Module (IRAM), which employs learnable temporal anchors and Gaussian-kernel attention to aggregate observations while preserving temporal information. To improve robustness under sparse and asynchronous sampling, the framework further incorporates Relative Time Feature (RTF) for pre-aggregation temporal encoding, Variable-Specific Anchor Initialization (VSA) for data-aligned anchor placement, anchor regularization for stable training, and multi-scale modeling. Experiments on four benchmarks demonstrate state-of-the-art (SOTA) performance, with the best results on seven out of eight metrics and statistically significant aggregate gains. Our code is publicly available at https://github.com/baimao00001/IMTS-Tokenizer.

1 Introduction

Time series analysis underpins numerous applications, from clinical monitoring to financial forecasting [1, 2]. In practice, observations rarely arrive at uniform intervals: medical sensors record vital signs as clinically needed, financial instruments trade at variable frequencies, and IoT devices transmit opportunistically under resource constraints [3, 4]. This Irregular Multivariate Time Series (IMTS) paradigm, characterized by asynchronous cross-channel sampling and non-uniform temporal density, poses fundamental challenges to conventional methods built on uniform-sampling assumptions.

*These authors contributed equally to this work.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

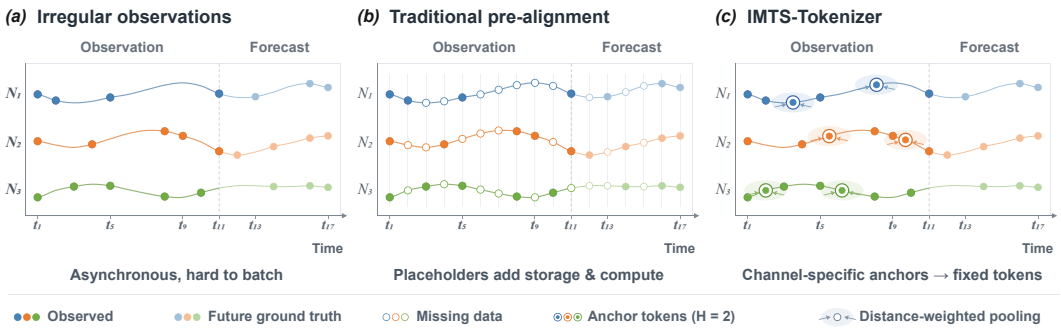

Figure 1: Illustration of irregular multivariate time series representations. (a) Three variables are observed asynchronously. (b) Traditional pre-alignment inserts aligned placeholders, incurring storage and computational overhead. (c) IMTS-Tokenizer maps each variable to fixed anchor tokens via local aggregation, preserving temporal distribution without dense padding.

Existing IMTS forecasting approaches follow two primary directions. First, neural ordinary differential equation methods [5, 6, 7] model temporal dynamics in continuous time, treating observations as boundary conditions for learned differential equations. Though theoretically principled, these methods incur substantial overhead from numerical integration. Second, attention-based methods [8, 9, 10] improve efficiency by directly modeling irregular temporal dependencies. However, both paradigms focus on downstream sequence modeling, implicitly assuming that input representations are readily available. The tokenization stage, which transforms raw observations into fixed-dimensional embeddings, remains underexplored for irregular data. Existing tokenizers [11, 12], while successful for regular time series, assume uniform sampling and fixed observation counts per patch, rendering them inapplicable to IMTS without preprocessing that sacrifices temporal precision.

We identify four key problems that render standard approaches inadequate for IMTS forecasting:

P1 **Temporal Irregularity.** Observations occur at arbitrary, non-uniform time points, violating the equispaced input assumption underlying most sequence models.

P2 **Asynchronous Variables.** Different channels are observed at different timestamps, resulting in sparse observation matrices where complete feature vectors are unavailable at any single time point.

P3 **Extreme Sparsity.** The density of observations varies considerably across time and channels, limiting the information available for modeling [13].

P4 **Temporal Dependency under Irregularity.** Mechanisms that capture temporal locality in regular data rely on uniform intervals and fail under irregular sampling.

We propose IMTS-Tokenizer to address these challenges through the following contributions:

C1 **Tokenization Paradigm.** We introduce a paradigm that transforms variable-length, irregularly-sampled observations into fixed-dimensional token representations, directly addressing P1 P2 P3 without interpolation or discretization.

C2 **Irregular-time Aware Module (IRAM).** We propose a mechanism employing learnable temporal anchors and Gaussian distance-aware aggregation, which explicitly models how temporal proximity influences observation contributions, addressing P4.

C3 **Relative Time Feature (RTF).** We incorporate anchor-relative temporal encoding prior to aggregation, ensuring distinct representations across anchors and enhancing temporal awareness.

C4 **Multi-scale Processing.** We develop a hierarchical architecture applying IRAM at multiple temporal resolutions, capturing both fine-grained local patterns and coarse-grained global trends.

C5 **Efficiency and Compatibility.** Our framework avoids expensive ODE solvers and complex graph constructions, achieving significant speedup while producing outputs directly compatible with standard Transformer architectures.

Through systematic analysis under extreme sparsity, we further identify attention collapse, a phenomenon where attention distributions across anchors converge to identical observation subsets. We trace this to anchor drift, temporal information loss, and softmax amplification, and address these issues through RTF (C3), Variable-Specific Anchor Initialization (VSA) for data-aligned anchor placement, and bounded regularization for training stability.

Comprehensive experiments across four benchmark datasets spanning healthcare (PhysioNet, MIMIC), human activity, and climate (USHCN) domains demonstrate that IMTS-Tokenizer achieves state-of-the-art performance on the majority of evaluation metrics. Ablation studies confirm that each proposed component contributes meaningfully to the overall improvement.

2 Related work

2.1 Irregular multivariate time series forecasting

Within this line of work, two primary paradigms handle temporal irregularity. Neural ordinary differential equation approaches model observations as boundary conditions for continuous dynamics. Neural ODE [5] introduced this paradigm, with extensions including Latent-ODE [6], CRU [7], and Neural Flows [14]. Neural Flows directly parameterize the solution map, whereas the cost of numerical integration during training and inference can limit the scalability of Neural ODEs and latent ODE models. Recent advances such as ACSSM [15] construct conditional stochastic differential equations with factorized encoders mapping observations independently, while ProFITi [16] introduces normalizing flow-based models for learning joint conditional distributions.

Attention-based methods have emerged as computationally efficient alternatives. GRU-D [9] augments recurrent units with temporal decay mechanisms. SeFT [17] leverages set functions for permutation-invariant processing of variable-length sequences. mTAND [10] employs multi-time attention networks to learn continuous-time representations. RainDrop [18] introduces graph-guided attention for capturing sensor dependencies in clinical time series. Warpformer [8] extends attention mechanisms with multi-scale temporal modeling.

Foundation model approaches introduce new mechanisms for temporal irregularity. MIRA [19] proposes Continuous-Time Rotary Positional Encoding (CT-RoPE), generalizing RoPE to real-valued timestamps so that attention scores depend solely on timestamp differences. Different from our RTF approach, CT-RoPE modulates query-key interactions during attention computation, whereas RTF augments observations prior to aggregation. ISTS-PLM [20] introduces continuous-time embeddings replacing fixed positional embeddings for adapting language models to irregular time series. Additionally, Hi-Patch [21] introduces hierarchical intra- and inter-patch graph attention layers for multi-scale modeling. Hi-Patch captures inter-channel dependencies through message passing, but typically requires explicit graph construction.

IP-Net uses temporal kernels to interpolate irregular observations at reference times [22]. APN learns soft temporal intervals, pools observations into patches according to their interval membership weights, and combines the patches with a channel-specific query for forecasting [23]. Despite these advances in model architecture, many IMTS forecasting methods still focus on backbone design and rely on simple point-wise tokenization, leaving IMTS-specific tokenization underexplored.

2.2 Tokenizer designed for multivariate time series

Early studies [24, 25] encode each time point into a token, incurring quadratic computational complexity and limited semantic representation. Inspired by ViT [26] which divides images into patches and encodes them into tokens, PatchTST [11] splits time series into multiple patches and transforms them into tokens. This significantly reduces the computational footprint and enhances model performance, establishing the dominant tokenizer paradigm. Building on this foundation, iTransformer [12] encodes an entire channel into one token, achieving better performance. TOTEM [27] employs vector quantization to map features to a discrete codebook, while wavelet-based tokenization [28] decomposes data into time-localized frequencies to capture complex patterns.

However, existing methods assume regular sampling within each patch or channel, an assumption violated by IMTS where observation counts and temporal positions vary unpredictably. Our anchor-based tokenization addresses this limitation directly: learnable temporal positions query the observation

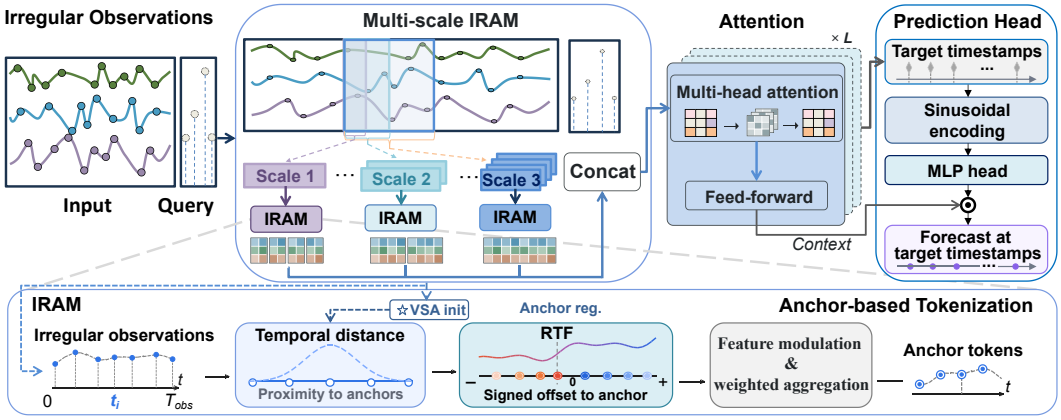

Figure 2: Overview of IMTS-Tokenizer. Multi-scale IRAM tokenizes each channel, and the concatenated representations are processed by cross-channel attention and fused with sinusoidal target-time queries for forecasting. RTF adds signed anchor-relative time features before aggregation to encode temporal direction alongside distance-based weighting. Appendix B.6 illustrates this construction and analyzes its effect on the resulting tokens.

sequence through distance-weighted aggregation, producing fixed-dimensional embeddings that explicitly encode both values and temporal positions while preserving the efficiency benefits of tokenization without sacrificing temporal precision to preprocessing artifacts. This formulation preserves the efficiency benefits of tokenization while accommodating irregular real-world time series without sacrificing temporal precision to preprocessing artifacts.

3 Preliminaries

We formalize the irregular multivariate time series forecasting problem. Let an IMTS set be denoted by $S = (\mathbf{X}, \mathbf{M}, \mathbf{T_x}, \mathbf{Y}, \mathbf{T_y})$. Here $\mathbf{X} \in \mathbb{R}^{B\times C\times L}$ represents the time series data matrix. $\mathbf{M} \in \{0,1\}^{B\times C\times L}$ is a matrix indicating missingness in $\mathbf{X}$, with values 0 and 1 denoting missing and observed data points respectively. $\mathbf{T_x} \in \mathbb{R}^{B\times C\times L}$ is the timestamp matrix. Given historical observations ($\mathbf{X}$, $\mathbf{M}$, $\mathbf{T_x}$), the objective is to predict future values $\mathbf{Y} \in \mathbb{R}^{B\times C\times L_y}$ at specified timestamps $\mathbf{T_y}$. In these notations, B, C, and L represent the batch size, the number of channels (variates), and the length of time series, respectively.

Since different samples have varying sampling times, padding is applied to align matrices within the same batch. For different batches, the lengths L and L_y may differ, and the time intervals between any two consecutive sampling points are not fixed. Standard tokenizers assume uniform sampling and fixed lengths. Modeling IMTS thus requires aggregation mechanisms aware of temporal distances.

4 Method

4.1 IMTS-Tokenizer with IRAM

IRAM converts irregular observations into fixed-dimensional tokens through learnable temporal anchors and distance-aware aggregation, addressing P1–P4. Figure 2 summarizes the full architecture.

4.1.1 Learnable temporal anchors and temporal gaussian weights

We introduce a learnable anchor matrix $\mathbf{A} \in \mathbb{R}^{C\times H}$, where H is the base number of anchors per channel and $A_{c,h}$ is the temporal query position of anchor h for channel c. For each anchor, attention weights over all observations are computed from temporal proximity through a Gaussian kernel:

$$\mathbf{K} = \text{Softmax}\left(-\gamma(\overline{\mathbf{T}_x} - \overline{\mathbf{A}})^2 + (\mathbf{M}_x - 1)\infty\right), \tag{1}$$

where $\gamma > 0$ is a learnable distance-sensitivity parameter, $\overline{\mathbf{T}_x}$ denotes broadcast timestamps, and $(\mathbf{M}_x - 1)\infty$ masks invalid positions before softmax. The softmax is taken over the observation

dimension L for each batch item, channel, and anchor. The negative quadratic form enforces temporal locality: observations closer to the anchor receive larger weights, while masked positions are excluded from normalization. This Gaussian weighting is chosen to preserve local temporal relevance under irregular sampling without forcing observations into fixed windows or interpolation bins, so each anchor can form a temporally meaningful token directly from the original observation set. At the same nominal half-maximum radius, the Gaussian has a smooth center and decays faster beyond that radius than the Laplacian and Cauchy profiles (Appendix B.1). Appendix H.7 reports the kernel comparison.

4.1.2 Anchor regularization and initialization

To keep anchors data-aligned and stable, we combine variable-specific initialization with boundary and diversity regularization terms. VSA centers anchors near representative observation regions for each channel, while the two regularizers keep the learned anchors within valid temporal ranges and discourage degenerate overlap, thereby stabilizing token formation under highly uneven sampling patterns. We first initialize anchors through variable-specific anchor initialization (VSA). For channel c, let $t_{c,i}$ and $m_{c,i}$ denote the i-th timestamp and mask value. The mean observation time is

$$\overline{t_c} = \frac{\sum_i t_{c,i} m_{c,i}}{\sum_i m_{c,i}}. \tag{2}$$

For channel c, anchor h is initialized as

$$a_{c,h} = \overline{t_c} + \delta_h + \varepsilon_{c,h}, \tag{3}$$

where $\delta_h = \left(h - \frac{H+1}{2}\right) \Delta_c$ provides uniform spacing across the temporal range, $\Delta_c = \frac{t_{\max} - t_{\min}}{\max(H-1,1)}$, and $\varepsilon_{c,h} \sim \mathcal{N}(0,1)$ breaks symmetry. To keep these initialized anchors within the observed temporal range $[t_{\min}, t_{\max}]$, we add a boundary regularization term:

$$\mathcal{L}_{\text{bound}} = \lambda_b \cdot \frac{1}{CH} \sum_{c=1}^{C} \sum_{h=1}^{H} \Big[\text{ReLU}\left(t_{\min} - a_{c,h}\right) + \text{ReLU}\left(a_{c,h} - t_{\max}\right) \Big]. \tag{4}$$

To exploit the available temporal queries more fully, we additionally encourage distinct temporal coverage through a diversity regularization term:

$$\mathcal{L}_{\text{div}} = \lambda_r \cdot \frac{1}{CH(H-1)} \sum_{c=1}^{C} \sum_{h \neq h'} \frac{1}{|a_{c,h} - a_{c,h'}| + \varepsilon_{\text{div}}}. \tag{5}$$

Together, these regularization terms stabilize anchor learning and preserve adaptive temporal coverage.

4.1.3 Embedding generation with Relative Time Feature (RTF)

The aggregated value embedding is

$$z_{c,h} = \sum_i k_{c,h,i} \cdot x_{c,i}. \tag{6}$$

However, Eq. (1) depends on temporal distance rather than temporal direction, so two anchors attending to similar values at opposite signed offsets can still produce indistinguishable representations. Relative Time Feature (RTF, C3) addresses this issue through pre-aggregation temporal encoding, allowing the tokenizer to preserve signed offset information before it is washed out by weighted averaging. Specifically, for anchor h and observation i, we augment values with anchor-relative temporal information before aggregation:

$$x'_{c,h,i} = x_{c,i} + f\left(t_{c,i} - a_{c,h}; \theta_f\right), \tag{7}$$

where $f : \mathbb{R} \to \mathbb{R}$ is a learnable encoding function. We instantiate it as

$$f\left(d; \boldsymbol{\theta}\right) = \theta_1 \cdot \tanh\left(\theta_2 \cdot d\right). \tag{8}$$

The $\tanh$ saturates at $\pm\theta_1$ for large offsets, preventing unbounded growth, while providing near-linear response for small offsets where fine-grained temporal distinctions matter. The learnable parameters θ_1 and θ_2 control output magnitude and input sensitivity, respectively. RTF uses signed temporal

offsets to distinguish observations at equal distances on opposite sides of an anchor. Appendix B.6 illustrates this pre-aggregation encoding and analyzes its contribution to the resulting token. The resulting token is

$$z_{c,h} = \sum_i k_{c,h,i} \cdot x_{c,i} + \sum_i k_{c,h,i} \cdot f\left(t_{c,i} - a_{c,h}\right), \tag{9}$$

which preserves both value and relative-time information by combining the aggregated observation magnitudes with their signed anchor-relative offsets. With timestamps, masks, and model parameters fixed, changing an observed value leaves the kernel weights and the relative-time term in Eq. (9) unchanged. If only $x_{c,i}$ changes by $\Delta x_{c,i}$, the change in the token at each anchor h is $\Delta z_{c,h} = k_{c,h,i}\, \Delta x_{c,i}$ (Appendix B.3).

4.2 Multi-scale modeling

To capture both local and long-range temporal patterns, we apply IRAM at progressively coarser resolutions (C4). The number of scales is

$$n_{\text{scales}} = \lfloor \log_2 \frac{t_h}{p} \rfloor + 1, \tag{10}$$

where t_h is the history length and p the patch size. At scale s, the anchor count is $H_s = H \times 2^s$, providing finer temporal granularity to compensate for the coarser input structure at higher scales. For $s < n_{scales} - 1$, the next-scale representation is obtained by interleaved down-sampling:

$$\mathbf{X}' = \text{Concat}\left(\mathbf{X}_{:2}, \mathbf{X}_{1:2}, \text{dim} = \text{time}\right). \tag{11}$$

This operation merges adjacent temporal patches, doubling the temporal extent per patch while halving the number of patches, thereby coarsening the effective temporal resolution for subsequent scales. Embeddings from all scales are then concatenated:

$$\mathbf{Z}_{\text{multi}} = \text{Concat}\left(\mathbf{Z}_0, \mathbf{Z}_1, ..., \mathbf{Z}_{n_{scales}-1}\right). \tag{12}$$

Let P denote the number of patches at the finest scale and m the number of scales. At scale s, each channel has $P/2^s$ patches with $2^s H$ anchor features per patch. Each scale therefore contributes PH features per channel, and concatenating the scales yields a representation of width $D = mPH$. This fusion preserves multiple temporal granularities in one representation, enabling downstream modules to use local and long-range temporal cues within a shared embedding space.

4.3 Prediction generation

IRAM tokenizes each channel independently, after which Transformer self-attention layers [29] model cross-channel dependencies. Let $\mathbf{Z}_{\text{ctx}}$ denote the output of cross-channel attention applied to $\mathbf{Z}_{\text{multi}}$. To generate forecasts at arbitrary target timestamps $\mathbf{T}_y$, we use a learnable sinusoidal query encoding. Time series data often exhibit periodic patterns at multiple frequencies, such as diurnal cycles, weekly rhythms, and seasonal variations. A fixed set of frequency components cannot capture the diverse periodic patterns inherent in different datasets. We therefore employ a learnable frequency matrix $\mathbf{W} \in \mathbb{R}^{C \times D/2}$ that discovers relevant frequencies from data:

$$\mathbf{E} = \text{Concat}\left(\sin\left(\mathbf{T}_y \cdot \mathbf{W}\right), \cos\left(\mathbf{T}_y \cdot \mathbf{W}\right)\right), \tag{13}$$

where $\mathbf{E} \in \mathbb{R}^{B \times C \times L_y \times D}$ is refined by a two-layer MLP to obtain $\mathbf{Q}_{\text{pred}}$. The prediction must integrate two distinct information sources: the learned temporal representation $\mathbf{Z}_{\text{ctx}}$ encoding historical patterns, and the query encoding $\mathbf{Q}_{\text{pred}}$ specifying the target timestamp. We combine them through multiplicative fusion:

$$\hat{\mathbf{Y}} = \text{Linear}\left(\mathbf{Q}_{\text{pred}} \odot \text{Broadcast}\left(\mathbf{Z}_{\text{ctx}}\right)\right), \tag{14}$$

where $\odot$ denotes element-wise multiplication and Broadcast expands $\mathbf{Z}_{\text{ctx}}$ along the prediction horizon. Multiplicative fusion enables the query encoding to modulate the representation by selectively scaling feature dimensions according to the target timestamp. We first train the tokenizer-based forecasting model, then freeze its validation-selected checkpoint and train the residual correction module on its predictions using training-set masked MSE. We use validation MSE for checkpoint selection and early stopping in both stages. The model is trained with

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{bound}} + \mathcal{L}_{\text{div}}. \tag{15}$$

$\mathcal{L}_{\text{pred}}$ is MSE on valid positions, $\mathcal{L}_{\text{bound}}$ regularizes anchor validity, and $\mathcal{L}_{\text{div}}$ promotes anchor diversity. Appendix B provides fuller rationale for Gaussian weighting, anchor regularization, RTF design, multi-scale resolution, and the prediction head, while Appendix E lists the main training hyperparameters.

Table 1: Dataset statistics. Avg. Obs.: average non-missing entries per series. Sparsity: percentage of missing observations.

Datasets	PhysioNet	MIMIC	Human Activity	USHCN
Samples	12 000	23 457	5400	26 736
Channels	41	96	12	5
Max. len.	216	643	130	214
Avg. Obs.	75.15	55.96	1.64	0.018
Sparsity	85.74%	96.72%	75.00%	77.98%
Obs. Window	24h	24h	3 000ms	24months
Pred. Horizon	next 24h	next 24h	next 1 000ms	next month

Table 2: Ablation study results. MSE reported as mean ± std over 5 seeds. **Red**: the best results. **Blue**: the second-best results.

	PhysioNet ($\times 10^{-3}$)	**MIMIC-III** ($\times 10^{-2}$)	**Activity** ($\times 10^{-3}$)	**USHCN** ($\times 10^{-1}$)
Full Model	$\mathbf{4.82 \pm 0.05}$	$\mathbf{1.59 \pm 0.02}$	$\mathbf{2.52 \pm 0.00}$	$\mathbf{4.98 \pm 0.04}$
w/o VSA	$\mathbf{4.99 \pm 0.11}$	$\mathbf{1.61 \pm 0.04}$	2.87 ± 0.02	5.02 ± 0.04
w/o RTF	5.08 ± 0.19	1.66 ± 0.01	2.73 ± 0.01	5.12 ± 0.09
w/o AC	5.00 ± 0.15	1.62 ± 0.03	2.87 ± 0.03	$\mathbf{5.01 \pm 0.05}$
w/o CA	5.57 ± 0.03	1.72 ± 0.01	3.20 ± 0.01	5.15 ± 0.02
w/o multi-scale	5.39 ± 0.04	1.68 ± 0.03	$\mathbf{2.54 \pm 0.02}$	5.26 ± 0.04
w/o channel share	5.40 ± 0.10	1.68 ± 0.09	2.65 ± 0.01	5.11 ± 0.07
rp-MLP	5.40 ± 0.08	1.69 ± 0.01	2.73 ± 0.02	5.13 ± 0.06

5 Experiments

5.1 Datasets

Following prior IMTS work [30], we evaluate on PhysioNet, MIMIC, Human Activity, and USHCN using chronological 60%/20%/20% train/validation/test splits. Table 1 summarizes the core statistics, while Appendix C provides dataset-specific descriptions and preprocessing details.

5.2 Baseline

We organize existing SOTA methods applicable to IMTS into three paradigms: (1) **Regular MTS Forecasting**: DLinear [31], TimesNet [32], PatchTST [11], Crossformer [33], Graphwavenet [34], MTGNN [35], StemGNN [36], CrossGNN [37] and FourierGNN [38]; (2) **IMTS Classification**: GRU-D [9], SeFT [17], RainDrop [18], Warpformer [8]; and (3) **IMTS Forecasting**: mTAND [10], Latent-ODE [6], CRU [7], Neural Flows [14], tPatchGNN [30], ISTS-PLM [20], Hi-Patch [21]. Appendix D provides detailed model descriptions together with reproduction notes on minor deviations between rerun baseline results, including Hi-Patch, and the numbers reported in the corresponding original papers.

We do not include IMTS-Mixer in the main benchmark baseline set because its available results are not trained under the same fully unified reimplementation pipeline; the rationale and a supplementary contextual comparison are provided in Appendix A.

5.3 Main results

Table 3 reports the forecasting performance on all four benchmarks. IMTS-Tokenizer achieves the best result on seven of eight metrics, indicating that irregularity-aware tokenization improves performance across very different sparsity regimes and application domains.

On the sparse clinical benchmarks, our model improves over Latent-ODE by 17.2% on PhysioNet and 7.6% on MIMIC, supporting the value of IRAM (C2) under extreme sparsity. On Human Activity it improves MSE over tPatchGNN by 8.0% with negligible variance, and it also attains the best MSE on USHCN. The consistent gains over mTAND, GRU-D, Warpformer, ISTS-PLM, and Hi-Patch suggest that better tokenization yields benefits beyond backbone choice alone.

To avoid overstating individual overlaps in mean±std intervals, we also report aggregate paired analyses against ISTS-PLM and Hi-Patch across the eight dataset-metric tasks. The gains remain statistically significant under paired Wilcoxon tests for both comparisons ($p = 0.0313$), with strictly positive bootstrap confidence intervals of $[1.90, 2.82]$ and $[2.33, 4.47]$ for relative MSE improvement, respectively; detailed statistics and average ranks are provided in Appendix G.1.

Beyond the four standard irregular benchmarks, Appendix G.2 further evaluates the tokenizer under controlled random masking, synthetic generation, and structured block-missing regimes. These supplementary studies show that the advantage remains visible on regularly sampled data after masking and becomes more pronounced when missingness is temporally correlated, with additional setting-level significance tests reported in the appendix. We also reused the forecasting tokenizer settings for classification on P19 and PAM and for interpolation on PhysioNet and Human Activity (Appendix G.4).

Table 3: Overall performance evaluated by MSE and MAE (mean ± std). **Red**: the best results. **Blue**: the second-best results.

Method	PhysioNet		MIMIC		Human Activity		USHCN	
	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-1}	MAE×10^{-1}
DLinear	41.86 ± 0.05	15.52 ± 0.03	4.90 ± 0.00	16.29 ± 0.05	4.03 ± 0.01	4.21 ± 0.01	6.21 ± 0.00	3.88 ± 0.02
TimesNet	16.48 ± 0.11	6.14 ± 0.03	5.88 ± 0.08	13.62 ± 0.07	3.12 ± 0.01	3.56 ± 0.02	5.58 ± 0.05	3.60 ± 0.04
PatchTST	12.00 ± 0.23	6.02 ± 0.14	3.78 ± 0.03	12.43 ± 0.10	4.29 ± 0.14	4.80 ± 0.09	5.75 ± 0.01	3.57 ± 0.02
Crossformer	6.66 ± 0.11	4.81 ± 0.11	2.65 ± 0.10	9.56 ± 0.29	4.29 ± 0.20	4.89 ± 0.17	5.25 ± 0.04	3.27 ± 0.09
Graph Wavenet	6.04 ± 0.28	4.41 ± 0.11	2.93 ± 0.09	10.50 ± 0.15	2.89 ± 0.03	3.40 ± 0.05	5.29 ± 0.04	3.16 ± 0.09
MTGNN	6.26 ± 0.18	4.46 ± 0.07	2.71 ± 0.23	9.55 ± 0.65	3.03 ± 0.03	3.53 ± 0.03	5.39 ± 0.05	3.34 ± 0.02
StemGNN	6.86 ± 0.28	4.76 ± 0.19	1.73 ± 0.02	7.71 ± 0.11	8.81 ± 0.37	6.90 ± 0.02	5.75 ± 0.09	3.40 ± 0.09
CrossGNN	7.22 ± 0.36	4.96 ± 0.12	2.95 ± 0.16	10.82 ± 0.21	3.03 ± 0.10	3.48 ± 0.08	5.66 ± 0.04	3.53 ± 0.05
FourierGNN	6.84 ± 0.35	4.65 ± 0.12	2.55 ± 0.03	10.22 ± 0.08	2.99 ± 0.02	3.42 ± 0.02	5.82 ± 0.06	3.62 ± 0.07
GRU-D	5.59 ± 0.09	4.08 ± 0.05	1.76 ± 0.03	7.53 ± 0.09	2.94 ± 0.05	3.53 ± 0.06	5.54 ± 0.38	3.40 ± 0.28
SeFT	9.22 ± 0.18	5.40 ± 0.08	1.87 ± 0.01	7.84 ± 0.08	12.20 ± 0.17	8.43 ± 0.07	5.80 ± 0.19	3.70 ± 0.11
RainDrop	9.82 ± 0.08	5.57 ± 0.06	1.99 ± 0.03	8.27 ± 0.07	14.92 ± 0.14	9.45 ± 0.05	5.78 ± 0.22	3.67 ± 0.17
Warpformer	5.94 ± 0.35	4.21 ± 0.12	1.73 ± 0.04	7.58 ± 0.13	2.79 ± 0.04	3.39 ± 0.03	5.25 ± 0.05	3.23 ± 0.05
mTAND	6.23 ± 0.24	4.51 ± 0.17	1.85 ± 0.06	7.73 ± 0.13	3.22 ± 0.07	3.81 ± 0.07	5.33 ± 0.05	3.26 ± 0.10
Latent-ODE	5.82 ± 0.16	4.07 ± 0.03	1.72 ± 0.02	7.53 ± 0.04	2.72 ± 0.04	3.34 ± 0.06	5.03 ± 0.11	3.91 ± 0.15
CRU	8.40 ± 0.34	5.07 ± 0.13	1.94 ± 0.02	7.75 ± 0.08	5.06 ± 0.25	5.19 ± 0.16	6.09 ± 0.14	3.59 ± 0.12
Neural Flow	7.16 ± 0.08	4.64 ± 0.02	1.86 ± 0.04	8.01 ± 0.14	4.08 ± 0.13	4.48 ± 0.11	5.36 ± 0.04	3.24 ± 0.04
tPatchGNN	5.47 ± 0.08	3.87 ± 0.10	1.71 ± 0.03	7.40 ± 0.25	2.74 ± 0.05	3.20 ± 0.05	5.06 ± 0.03	3.15 ± 0.06
ISTS-PLM	**5.07 ± 0.10**	3.76 ± 0.05	**1.64 ± 0.02**	**7.02 ± 0.14**	2.58 ± 0.10	3.14 ± 0.13	5.24 ± 0.06	**2.99 ± 0.07**
Hi-Patch	5.26 ± 0.09	**3.75 ± 0.06**	1.75 ± 0.26	7.24 ± 0.18	**2.57 ± 0.01**	**3.13 ± 0.02**	**5.03 ± 0.04**	**3.07 ± 0.08**
IMTS-Tokenizer	**4.82 ± 0.05**	**3.65 ± 0.05**	**1.59 ± 0.02**	**6.93 ± 0.07**	**2.52 ± 0.00**	**3.08 ± 0.00**	**4.98 ± 0.04**	**3.07 ± 0.06**
Δ *vs. the best baseline*	+4.93%	+2.67%	+3.05%	+1.28%	+1.95%	+1.60%	+0.99%	-2.68%

5.4 Ablation studies

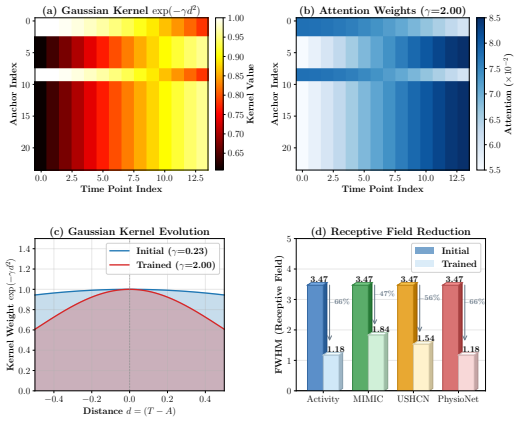

Figure 3: Visualization of IRAM learning dynamics on Activity dataset. (a) Gaussian kernel values showing distance-dependent decay. (b) Attention weights after softmax normalization. (c) Kernel bandwidth evolution from initialization ($\gamma = 0.23$) to trained state ($\gamma = 2.00$). (d) FWHM reduction across four benchmarks, with reduction magnitude correlating with data sparsity.

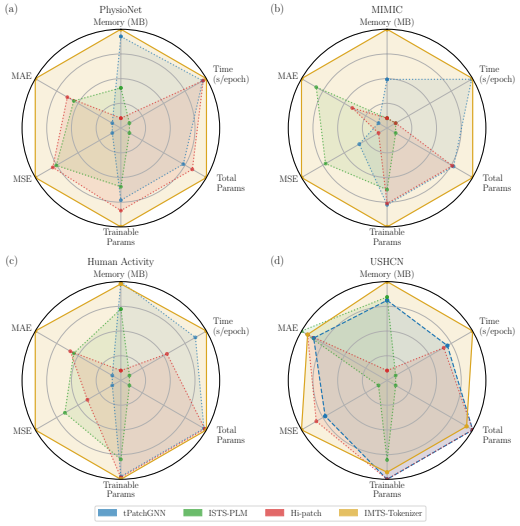

Figure 4: Multi-dimensional comparison of model efficiency and performance across four datasets. Each axis represents a normalized metric where larger area indicates better performance. IMTS-Tokenizer (gold) achieves the largest coverage on most datasets.

Table 2 summarizes the core ablation results, while Appendix H provides the complete MSE/MAE table and extended design variants. Removing cross-channel attention increases mean MSE on all four benchmarks. RTF (C3) contributes 4–6% gains on the clinical benchmarks, VSA is especially important on Activity, and the anchor regularization terms improve training stability. Removing multi-scale processing also leads to measurable degradation, most notably on PhysioNet, confirming

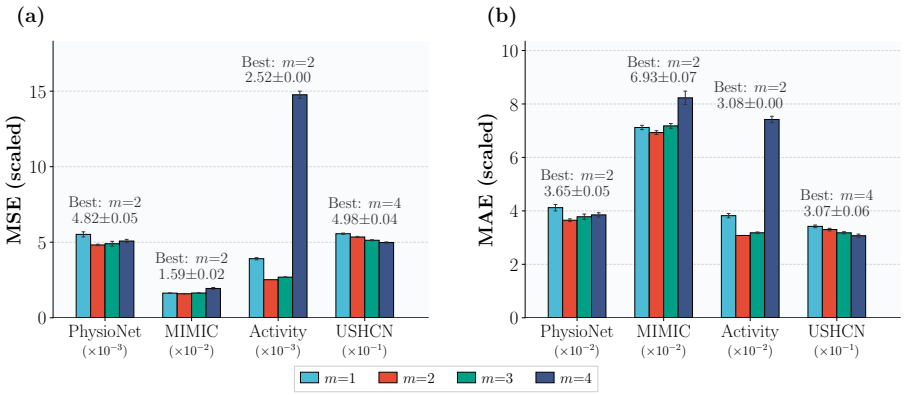

Figure 5: Sensitivity analysis of the number of scales m. Results are reported as mean $\pm$ std over 5 random seeds. For most datasets, $m = 2$ achieves optimal performance, while USHCN benefits from deeper multi-scale hierarchies ($m = 4$).

that hierarchical temporal resolution contributes materially to forecasting quality. Sharing a single tokenizer across channels further degrades performance, especially on Activity, indicating that channel-specific tokenization remains important under asynchronous observations. Finally, replacing the attention interaction module with an MLP consistently weakens performance when the tokenizer is retained. With the attention predictor held fixed, IRAM yields lower mean MSE than linear interpolation on all four benchmarks. The IRAM–MLP model also achieves lower mean MSE on each benchmark than the standalone MLP trained on interpolated inputs (Appendix H.3).

Additional analyses in Appendix H show that Tanh encoding performs best on the clinical datasets, while Sinusoidal encoding is competitive on USHCN. Progressive module-removal further shows that the tokenizer backbone itself contributes substantially: even the tokenizer-only variant remains markedly stronger than the interpolation front end with the attention predictor, indicating that the gains cannot be reduced to auxiliary modules alone. Appendix H.3 reports the full progression in detail. Removing AC and CA together increases mean MSE across the matched configurations in Appendix H.5.

Figure 3 visualizes learned attention patterns on Activity. The Gaussian kernel (a) and attention weights (b) confirm that anchors assign higher weights to temporally proximate observations, validating Eq. (1). The bandwidth γ increases from 0.23 to 2.00 during training (c), demonstrating learned receptive field narrowing. Across datasets (d), FWHM reduces by 47%–66%, with MIMIC retaining the widest field due to its extreme sparsity (96.7%), indicating adaptive locality-coverage trade-off. The Gaussian gives the lowest mean MSE of the three kernels on USHCN, Activity, and PhysioNet. On MIMIC-III, the heavier-tailed Cauchy and Laplacian profiles yield lower MSE (1.47 and 1.48, respectively) than the Gaussian (1.59), in units of 10^{-2} (Appendix H.7).

5.5 Sensitivity analysis

Sensitivity to the base anchor count H, network depth L, minimum kernel precision $\gamma_{\min}$, and the number of scales m is summarized in Figure 5 and Appendix F. Across the dataset-specific grids, the largest relative range in marginal mean MSE is 2.99% for H and 0.47% for $\gamma_{\min}$ (Appendix F.6). Marginal mean MSE also varies little across the tested depths. Mean MSE is lowest at $m = 2$ except on USHCN, where $m = 4$ performs best.

5.6 Computational efficiency

We compare against the strongest baselines from Table 3: tPatchGNN, Hi-Patch, and ISTS-PLM, while Appendix I expands the efficiency comparison to a broader set of competitive graph-based and continuous-time baselines. Figure 4 and Appendix I show that IMTS-Tokenizer consistently reduces training cost while preserving the strongest accuracy profile.

Beyond empirical profiling, the dominant temporal interaction in IRAM scales as $\mathbf{O}(CLH)$ because the anchor count H is fixed and small. By avoiding graph construction and ODE solving, IMTS-Tokenizer achieves substantial efficiency gains (C5). This linear aggregation cost avoids repeated ODE integration and the quadratic token mixing used by PLM-style baselines; Appendix I.1 summarizes the core-operation comparison.

On PhysioNet, our method is 2.4× faster than tPatchGNN and 30× faster than ISTS-PLM, while using substantially less memory. On MIMIC, ISTS-PLM exceeds 24GB GPU memory whereas IMTS-Tokenizer requires 5.6GB. On Activity and USHCN, IMTS-Tokenizer achieves 2.3× and 2.6× training speedups, respectively. These results validate anchor-based tokenization as an efficient alternative to graph-based and PLM-based approaches.

5.7 Implementation details

All experiments were conducted on NVIDIA RTX 4090 GPUs. We used Adam optimizer with learning rate decay and validation-based early stopping, and each experiment was repeated with five random seeds.

As all baselines were rerun under this unified protocol, minor deviations from the numbers reported in their original papers may remain. Appendix D discusses these reproduction details, and the authoritative dataset-specific training configuration is reported in Appendix E.

6 Conclusion

We have presented IMTS-Tokenizer, a framework addressing IMTS forecasting through anchor-based tokenization (C1). The core IRAM module (C2) transforms variable-length observations into fixed-dimensional embeddings via learnable temporal anchors and Gaussian distance-aware aggregation, addressing temporal irregularity (P1–P3) and dependency modeling under irregularity (P4). Relative Time Feature (RTF), Variable-Specific Anchor Initialization (VSA), and anchor regularization improve robustness under sparse and asynchronous observations, while multi-scale processing (C4) captures patterns at various temporal resolutions and the framework maintains computational efficiency (C5). Experiments across four datasets validate state-of-the-art (SOTA) performance on almost all metrics.

The framework currently targets forecasting and may face scalability constraints for high-dimensional data. We also evaluated the same tokenizer design on classification and interpolation by training a separate model for each task. Future directions include multi-task extension and hierarchical compression for scalability.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China (62476154), and Major Basic Research Project of Shandong Provincial Natural Science Foundation (ZR2024ZD03), and the Shandong University Qilu Young Scholars Program.

References

[1] Shilu Yuan, Xiaoyu Li, Wenqian Mu, Ji Zhong, Meng Chen, Haoliang Sun, and Yongshun Gong. Spatio-temporal prototype-based hierarchical learning for OD demand prediction. In *International Joint Conference on Artificial Intelligence*, 2025.

[2] Min Yang, Xiaoyu Li, Bin Xu, Xiushan Nie, Muming Zhao, Chengqi Zhang, Yu Zheng, and Yongshun Gong. STDA: Spatio-temporal deviation alignment learning for cross-city fine-grained urban flow inference. *IEEE Transactions on Knowledge and Data Engineering*, 37(8):4833–4845, 2025.

[3] Yongshun Gong, Zhibin Li, Jian Zhang, Wei Liu, and Yu Zheng. Online spatio-temporal crowd flow distribution prediction for complex metro system. *IEEE Transactions on Knowledge and Data Engineering*, 34(2):865–880, 2022.

[4] Yongshun Gong, Tiantian He, Meng Chen, Bin Wang, Liqiang Nie, and Yilong Yin. Spatio-temporal enhanced contrastive and contextual learning for weather forecasting. *IEEE Trans. on Knowl. and Data Eng.*, 36(8):4260–4274, 2024.

[5] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural Ordinary Differential Equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[6] Yulia Rubanova, Ricky T. Q. Chen, and David K Duvenaud. Latent Ordinary Differential Equations for irregularly-sampled time series. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[7] Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular time series with continuous recurrent units. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning*, volume 162, pages 19388–19405. PMLR, 2022.

[8] Jiawen Zhang, Shun Zheng, Wei Cao, Jiang Bian, and Jia Li. Warpformer: A multi-scale modeling approach for irregular clinical time series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 3273–3285, New York, NY, USA, 2023. Association for Computing Machinery.

[9] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent Neural Networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

[10] Satya Narayan Shukla and Benjamin Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021.

[11] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with Transformers. In *International Conference on Learning Representations*, 2023.

[12] 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 *International Conference on Learning Representations*, 2024.

[13] Yang An, Zhibin Li, Xiaoyu Li, Wei Liu, Xinghao Yang, Haoliang Sun, Meng Chen, Yu Zheng, and Yongshun Gong. Spatio-Temporal multivariate probabilistic modeling for traffic prediction. *IEEE Transactions on Knowledge and Data Engineering*, 37(5):2986–3000, 2025.

[14] Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural Flows: Efficient alternative to Neural ODEs. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21325–21337. Curran Associates, Inc., 2021.

[15] Byoungwoo Park, Hyungi Lee, and Juho Lee. Amortized control of continuous state space Feynman-Kac model for irregular time series. In *International Conference on Learning Representations*, 2025.

[16] Vijaya Krishna Yalavarthi, Randolf Scholz, Stefan Born, and Lars Schmidt-Thieme. Probabilistic forecasting of irregularly sampled time series with missing values via conditional Normalizing Flows. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(20):21877–21885, 2025.

[17] Max Horn, Michael Moor, Christian Bock, Bastian Rieck, and Karsten Borgwardt. Set functions for time series. In Hal Daumé III and Aarti Singh, editors, *International Conference on Machine Learning*, volume 119, pages 4353–4363. PMLR, 2020.

[18] Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. In *International Conference on Learning Representations*, 2022.

[19] Hao Li, Bowen Deng, Chang Xu, Zhiyuan Feng, Viktor Schlegel, Yu-Hao Huang, Yizheng Sun, Jingyuan Sun, Kailai Yang, Yiyao Yu, and Jiang Bian. MIRA: Medical time series foundation model for real-world health data, 2025.

[20] Weijia Zhang, Chenlong Yin, Hao Liu, and Hui Xiong. Unleashing the power of Pre-Trained Language Models for irregularly sampled time series. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, page 3831–3842, New York, NY, USA, 2025. Association for Computing Machinery.

[21] Yicheng Luo, Bowen Zhang, Zhen Liu, and Qianli Ma. Hi-Patch: Hierarchical patch GNN for irregular multivariate time series. In *International Conference on Machine Learning*, 2025.

[22] Satya Narayan Shukla and Benjamin Marlin. Interpolation-prediction networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2019.

[23] Xvyuan Liu, Xiangfei Qiu, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Jilin Hu, and Bin Yang. Rethinking irregular time series forecasting: A simple yet effective baseline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(28):23873–23881, 2026.

[24] 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, 2021.

[25] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022.

[26] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[27] Sabera J Talukder, Yisong Yue, and Georgia Gkioxari. TOTEM: TOkenized time series EMbeddings for general time series analysis. *Transactions on Machine Learning Research*, 2024.

[28] Luca Masserano, Abdul Fatir Ansari, Boran Han, Xiyuan Zhang, Christos Faloutsos, Michael W. Mahoney, Andrew Gordon Wilson, Youngsuk Park, Syama Sundar Rangapuram, Danielle C. Maddix, and Bernie Wang. Enhancing foundation models for time series forecasting via wavelet-based tokenization, 2025.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[30] Weijia Zhang, Chenlong Yin, Hao Liu, Xiaofang Zhou, and Hui Xiong. Irregular multivariate time series forecasting: A transformable patching Graph Neural Networks approach. In *International Conference on Machine Learning*, 2024.

[31] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are Transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, 2023.

[32] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

[33] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

[34] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph Wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, page 1907–1913. AAAI Press, 2019.

[35] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 753–763, New York, NY, USA, 2020. Association for Computing Machinery.

[36] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Spectral temporal Graph Neural Network for multivariate time-series forecasting. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17766–17778. Curran Associates, Inc., 2020.

[37] Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. CrossGNN: Confronting noisy multivariate time series via cross interaction refinement. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46885–46902. Curran Associates, Inc., 2023.

[38] Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 69638–69660. Curran Associates, Inc., 2023.

[39] Christian Klötergens, Tim Dernedde, Lars Schmidt-Thieme, and Vijaya Krishna Yalavarthi. Mixing it up: Exploring mixer networks for irregular multivariate time series forecasting, 2026.

[40] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753, 2019.

[41] Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of ICU patients: The PhysioNet/computing in cardiology challenge 2012. In *2012 computing in cardiology*, pages 245–248. IEEE, 2012.

[42] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[43] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[44] Matthew Reyna, Chris Josef, Russell Jeter, Supreeth Shashikumar, Benjamin Moody, M. Brandon Westover, Ashish Sharma, Shamim Nemati, and Gari D. Clifford. Early prediction of sepsis from clinical data: The PhysioNet/Computing in Cardiology Challenge 2019. *PhysioNet*, 2019.

[45] Attila Reiss. PAMAP2 physical activity monitoring. UCI Machine Learning Repository, 2012.

[46] Andrea Cini, Ivan Marisca, and Cesare Alippi. Filling the g_ap_s: Multivariate time series imputation by graph neural networks. In *International Conference on Learning Representations*, 2022.

[47] Min Yang, Yang An, Jinliang Deng, Xiaoyu Li, Bin Xu, Ji Zhong, Xiankai Lu, and Yongshun Gong. CAN-ST: Clustering adaptive normalization for spatio-temporal OOD learning. In James Kwok, editor, *International Joint Conference on Artificial Intelligence*, pages 3543–3551. International Joint Conferences on Artificial Intelligence Organization, 2025.

[48] Xiaoyu Li, Yongshun Gong, Wei Liu, Yilong Yin, Yu Zheng, and Liqiang Nie. Dual-track spatio-temporal learning for urban flow prediction with adaptive normalization. *Artificial Intelligence*, 328:104065, 2024.

[49] Bin Wang, Yongqi Han, Minbo Ma, Tianrui Li, Junbo Zhang, Feng Hong, and Yanwei Yu. Non-collective calibrating strategy for time series forecasting. In *International Joint Conference on Artificial Intelligence*, 2025.

A Supplementary related work

This appendix provides a contextual comparison with IMTS-Mixer (Section A.1) and a separate comparison with APN under its released forecasting protocol (Section A.2). The broader review of irregular forecasting backbones and tokenizer literature is already included in the main text to avoid unnecessary duplication here.

A.1 Comparison with IMTS-Mixer

IMTS-Mixer[39] is a closely related tokenizer-oriented method for irregular forecasting because both methods aim to derive fixed-size representations directly from irregular observations before downstream forecasting. The overlap, however, is primarily at the task level rather than at the mechanism level. IMTS-Mixer aggregates each channel into a fixed-size representation and then applies Mixer-style processing, whereas IMTS-Tokenizer performs anchor-based temporal querying, distance-aware aggregation, explicit relative-time augmentation, and multi-scale token construction before downstream interaction. The two methods therefore differ not only in architecture, but also in where and how temporal irregularity is resolved.

Table 4 provides a supplementary MSE comparison using additional matched-setting values. We report both the Adam configuration used in the main paper and an AdamW configuration because optimizer choice materially affects the final numbers. At the same time, this comparison should be interpreted cautiously rather than as a definitive head-to-head benchmark: the released IMTS-Mixer results and the present model are not trained under a fully unified experimental pipeline, and the methods differ not only in optimization details but also in architectural components beyond the tokenizer itself. In particular, IMTS-Mixer has not been released as an open-source training pipeline, so we cannot rerun it under the same implementation stack or replace its AdamW-based setting with Adam to align it with the optimization protocol used for the uniformly reimplemented baselines. Accordingly, we use Table 4 as contextual evidence about the scale of performance differences, while reserving the main benchmark claims for the uniformly reimplemented baselines in Table 3.

The supplementary numbers nevertheless remain informative in two respects. First, they show that IMTS-Tokenizer stays competitive with a closely related tokenizer-oriented alternative across all four datasets, with the AdamW variant matching or exceeding IMTS-Mixer on three datasets and tying on USHCN. Second, they clarify that any comparison should be framed around mechanistic differences rather than a simplistic claim of strict dominance: IMTS-Mixer emphasizes post-aggregation Mixer-style interaction, whereas IMTS-Tokenizer emphasizes temporally grounded token construction before downstream modeling. For this reason, we treat IMTS-Mixer primarily as a related tokenizer design that helps contextualize the contribution, rather than as the central basis of the paper's empirical claim.

A.2 Comparison with adaptive patching under the APN protocol

APN [23] is a forecasting framework. Its Time-Aware Patch Aggregation (TAPA) module learns temporal intervals for each channel. For each observation, APN multiplies two sigmoid functions of the signed distances from the interval boundaries to obtain a soft membership weight. It uses the normalized weights to average the observation values and their time embeddings within each patch. A channel-specific query then combines the patch representations, and an MLP produces the forecasts.

APN defines each patch by a learned temporal interval. In IMTS-Tokenizer, each input token is indexed by a learned point in the observed history of a channel (Section 4.1). Learned inducing points in set encoders [40] provide a related form of aggregation, but have no prescribed temporal coordinate.

We trained and evaluated IMTS-Tokenizer under the released APN task definition, including its data construction, fold-0 split, history and prediction windows, stride, and transformations of values and timestamps. Test MSE is pooled over all observed target entries. Because these settings differ from those of our main forecasting benchmark, Table 5 reports this comparison separately. We include the three datasets for which the released task could be reproduced. The controlled-access MIMIC-III snapshot used by APN differs from the one available for our main benchmark.

Table 4: Supplementary comparison with IMTS-Mixer. Values are MSE.

Method	PhysioNet	MIMIC	Activity	USHCN
IMTS-Mixer	4.88	1.61	2.50	4.91
IMTS-Tokenizer (Ours, Adam)	4.82	1.59	2.52	4.98
IMTS-Tokenizer (Ours, AdamW)	4.74	1.58	2.52	4.91

Table 5: Forecasting under the APN task protocol. Entries are pooled test MSE (mean $\pm$ standard deviation across runs). The APN values are from Table 2 of [23].

Dataset	APN	IMTS-Tokenizer
Human Activity	0.0421 ± 0.0001	0.0413 ± 0.0002
USHCN	0.1590 ± 0.0137	0.1565 ± 0.0241
PhysioNet	0.3093 ± 0.0011	0.3037 ± 0.0045

IMTS-Tokenizer has a lower reported mean MSE than APN on each task. Relative to APN, the reductions are 1.90% on Human Activity, 1.57% on USHCN, and 1.81% on PhysioNet. On PhysioNet, Warpformer has the lowest mean MSE among the methods in APN's published table (0.3056). Relative to this reference, the reduction is 0.62%.

B Supplementary method details

This appendix supplements Section 4 and records the derivation-oriented material compressed from the main text.

B.1 Gaussian weight intuition

This subsection supplements Section 4.1. The central challenge in processing irregular time series lies in mapping variable-length observation sequences to fixed-dimensional representations without losing temporal information. We introduce a learnable anchor matrix $\mathbf{A} \in \mathbb{R}^{C \times H}$, where each entry $A_{c,h}$ represents the temporal query position of anchor h for channel c, and H denotes the dimension of the hidden space. The VSA parameterization enables each anchor to adapt its temporal focus according to the characteristic sampling patterns of individual channels.

Given that anchors are designed to capture the characteristics of the sequence, and that observations in closer temporal proximity to an anchor exert greater influence on its representation, for each anchor h, we compute attention weights over all time positions based on temporal proximity through a Gaussian kernel. Here $\overline{\mathbf{A}}$ denotes the broadcast-expanded anchor positions, and the masking term assigns negative infinity to invalid positions so that they receive zero weight after softmax normalization. The negative coefficient before γ implements temporal locality: the Gaussian kernel $e^{-\gamma d^2}$ achieves its maximum at $d = 0$ and decays with increasing distance.

In practice, weight matrix $\mathbf{K} \in \mathbb{R}^{B \times C \times L \times H}$ captures temporal priors while ensuring numerical stability through masking of unsampled positions. Let $\delta = t_i - a_{c,h}$ be the temporal offset from an anchor. For the unnormalized Gaussian profile $\exp(-\gamma\delta^2)$, the half-maximum radius is $r = \sqrt{\ln 2/\gamma}$. Writing $u = |\delta|/r$ gives three profiles with the same nominal radius.

$$k_{\mathrm{G}}(u) = 2^{-u^2}, \qquad k_{\mathrm{L}}(u) = 2^{-u}, \qquad k_{\mathrm{C}}(u) = (1 + u^2)^{-1}.$$

All three equal one half at $u = 1$. Among these profiles, the Gaussian gives the greatest weight for $0 < u < 1$ and the smallest for $u > 1$. Its derivative at the center is zero, so small offsets near the center change the kernel value only to second order. The learnable precision γ sets the temporal width of the Gaussian, with larger values producing narrower profiles. Appendix H.7 compares the corresponding distance kernels in the forecasting model.

B.2 Anchor regularization and initialization rationale

This subsection supplements Section 4.1. Effective learning of anchor positions requires data-aligned initialization together with soft regularization terms. We therefore combine variable-specific initialization with boundary and diversity regularization.

Standard Gaussian initialization places anchors without regard to actual data distributions. We address this through data-driven initialization. For channel c, let $t_{c,i}$ and $m_{c,i}$ denote the i-th timestamp and mask value respectively. The mean observation time is used to center the initial anchor positions, so that each channel starts from representative temporal regions rather than arbitrary random points.

The temporal anchors are designed to query observations at specific time positions. For anchors to produce meaningful representations, they must reside within $[t_{\min}, t_{\max}]$ where observations exist. The ReLU boundary penalty therefore imposes zero cost within the valid range, while penalizing boundary violations proportionally to their magnitude.

With H anchors, the model can capture up to H distinct temporal perspectives. To fully exploit this capacity, anchors should maintain diverse positions across the temporal range. The inverse-distance formulation generates repulsive gradients between proximate anchors while exerting minimal influence on well-separated ones. This promotes distributed temporal coverage without imposing rigid position constraints, allowing anchors to adaptively concentrate in regions of high information density when beneficial.

B.3 RTF design rationale

This subsection supplements Section 4.1. The weighted aggregation produces embeddings that capture observed values weighted by temporal proximity. However, two anchors attending to identical values at different time offsets can still produce indistinguishable embeddings. RTF addresses this limitation by injecting anchor-relative temporal information before aggregation.

To retain temporal structure through aggregation, we augment observation values with anchor-relative positional information prior to the weighted sum. The quantity $t_{c,i} - a_{c,h}$ represents the temporal offset between an observation and the anchor's query position. By encoding this offset into the observation representation before aggregation, the resulting embedding inherently carries information about the temporal distribution of attended observations.

The encoding function f must satisfy three properties: boundedness to ensure numerical stability, sensitivity to small offsets where temporal distinctions are most informative, and learnability to adapt to dataset-specific temporal characteristics. We adopt the hyperbolic tangent because it saturates at large offsets while providing near-linear response for small offsets where fine-grained temporal distinctions matter. The learnable parameters θ_1 and θ_2 control output magnitude and input sensitivity, respectively.

The resulting embedding decomposes into a value component and a relative-time component. This decomposition allows downstream modules to leverage both observation magnitude and temporal offset information without changing the overall token dimension. Appendix B.6 develops the symmetric-sampling example and illustrates the observation features before aggregation. A short-lived extreme can still be attenuated when its temporal weights are small.

B.4 Multi-scale resolution design

This subsection supplements Section 4.2. Temporal patterns manifest at multiple scales, from rapid fluctuations to long-term trends. To capture this hierarchical structure, we employ parallel IRAM modules operating at progressively coarser temporal resolutions.

At scale s, the anchor count expands to $H_s = H \times 2^s$, providing finer temporal granularity to compensate for the coarser input structure at higher scales. The IRAM module then processes the current temporal representation to produce embeddings $\mathbf{Z}_s$. For scales $s < n_{scales} - 1$, we reorganize the temporal structure through interleaved down-sampling. This operation merges adjacent temporal patches, doubling the temporal extent per patch while halving the number of patches, thereby coarsening the effective temporal resolution for subsequent scales.

Subsequently, embeddings from all scales are concatenated to form a unified representation. This fusion preserves information across temporal granularities within a single embedding space, enabling downstream components to jointly leverage patterns from all scales.

B.5 Prediction head design

This subsection supplements Section 4.3. IRAM processes each channel independently, yet real-world multivariate time series exhibit rich cross-channel dependencies that must be captured for accurate forecasting. We therefore apply standard Transformer self-attention layers [29] across the channel dimension so that each channel's representation can incorporate information from correlated channels. Concretely, this module consists of N_{layers} stacked blocks with multi-head self-attention and feed-forward networks.

The prediction task requires generating forecasts at specified future timestamps $\mathbf{T}_y$. We design the prediction head to accept arbitrary query timestamps, enabling flexible prediction horizons without architectural modification.

Time series data often exhibit periodic patterns at multiple frequencies, such as diurnal cycles, weekly rhythms, and seasonal variations. A fixed set of frequency components cannot capture the diverse periodic patterns inherent in different datasets. The learnable sinusoidal frequency matrix $\mathbf{W}$ therefore allows the query encoder to discover dataset-specific periodicities rather than relying on a fixed frequency basis. The resulting sinusoidal timestamp encoding is refined through a two-layer MLP with GELU activations to obtain the final query representation $\mathbf{Q}_{\text{pred}}$.

The prediction must integrate the learned temporal representation $\mathbf{Z}_{\text{ctx}}$ encoding historical patterns and the query encoding $\mathbf{Q}_{\text{pred}}$ specifying the target timestamp. Multiplicative fusion allows the query encoding to modulate the representation element-wise, selectively scaling different feature dimensions based on the target timestamp. Broadcast expands $\mathbf{Z}_{\text{ctx}}$ along the prediction horizon to match $\mathbf{Q}_{\text{pred}}$. This yields a lightweight mechanism for conditional prediction without adding a separate decoder over the full observation sequence.

B.6 Signed temporal encoding under symmetric sampling

Figure 6 illustrates how RTF adds temporal direction to observations before aggregation. The Gaussian kernel depends on the magnitude of the offset from an anchor, while RTF encodes its sign through the odd function $f(d) = \theta_1 \tanh(\theta_2 d)$.

For the anchor $a_{c,h}$, let the two highlighted observations occur at $t_- = a_{c,h} - \Delta$ and $t_+ = a_{c,h} + \Delta$, with $\Delta > 0$ and $x_- = x_+ = v$. Their Gaussian kernel values are both $\exp(-\gamma\Delta^2)$, so they receive the same normalized weight κ in the weighted sum. The RTF function satisfies

$$f(-\Delta) = -f(\Delta). \tag{16}$$

Writing $\eta = f(\Delta)$, the enriched values are

$$x'_{c,h,-} = v - \eta, \qquad x'_{c,h,+} = v + \eta. \tag{17}$$

Thus, observations with equal values and equal distances receive opposite signed temporal adjustments. Panel (c) depicts the case $\eta > 0$.

For the full observation set, the temporal component of the token is

$$r_{c,h} = \sum_i k_{c,h,i}\, f(t_{c,i} - a_{c,h}), \tag{18}$$

$$z_{c,h} = \sum_i k_{c,h,i} x_{c,i} + r_{c,h}. \tag{19}$$

At the illustrated anchor, the highlighted pair contributes $\kappa(-\eta) + \kappa\eta = 0$ to $r_{c,h}$, reflecting its temporal symmetry. Its total contribution to $z_{c,h}$ is $2\kappa v$. More generally, if the pair has values u and v, its contribution is $\kappa(u + v)$. Exchanging u and v therefore preserves the token at this anchor. Other observations contribute according to their own offsets, and other anchors can assign the same pair different weights.

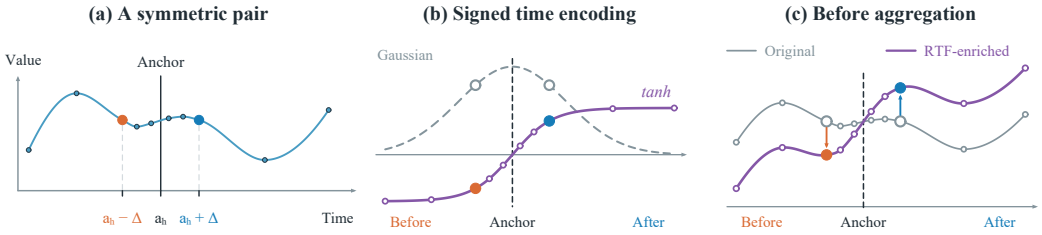

Figure 6: RTF encoding around an anchor. (a) The highlighted observations have equal values and lie at equal distances on opposite sides of the anchor. (b) The Gaussian kernel is symmetric, whereas the tanh encoding gives opposite-sign values at opposite offsets. (c) RTF shifts the highlighted values in opposite directions before aggregation. The anchor a_h in the figure denotes $a_{c,h}$ for the channel shown.

The token-level effect is especially direct when two anchors attend to the same observation at opposite offsets. Consider a single valid observation of value v at time t_0, with anchors $a_{c,h_1} = t_0 - \Delta$ and $a_{c,h_2} = t_0 + \Delta$. The normalized weight is one at each anchor. Value-only aggregation gives v at both anchors, while RTF yields

$$z_{c,h_1} = v + f(\Delta), \qquad z_{c,h_2} = v - f(\Delta). \tag{20}$$

For $f(\Delta) \neq 0$, the two token values differ by $2f(\Delta)$, encoding their opposite temporal orientations. When $|\theta_2(t_{c,i} - a_{c,h})| \ll 1$ for the contributing observations, $f(d) \approx \theta_1\theta_2 d$, so the temporal component locally approximates

$$r_{c,h} \approx \theta_1\theta_2 \left(\sum_i k_{c,h,i} t_{c,i} - a_{c,h} \right). \tag{21}$$

This term measures the displacement of the weighted observation times from the anchor. The tanh saturation bounds the adjustment at larger offsets, while the learned coefficients control its magnitude and sensitivity.

C Dataset details

C.1 Datasets for forecasting

This appendix supplements Section 5.1 and restores the dataset-specific descriptions and preprocessing details compressed from the main text.

PhysioNet [41] comprises 12,000 irregular multivariate time series, each representing a different ICU patient and containing 41 clinical variables recorded over the first 48 hours of admission. The maximum length of the time series is 216, the average value is 75.15, and the sparsity is 85.74%. We designate the initial 24 hours of each series as observed inputs and forecast the remaining 24 hours.

MIMIC [42], a publicly available critical-care database, yields 23,457 patient records following the preprocessing pipeline of [14]. Each IMTS spans the first 48 hours post-admission and includes 96 variables. The maximum length of the time series is 643, the average value is 55.96, and the sparsity is 96.72%. We use the first half of the series for observation and predict the latter half.

Human Activity consists of irregularly sampled three-dimensional positional data from four body-worn sensors (left ankle, right ankle, belt, chest) across five subjects performing various activities (e.g., walking, sitting, lying, standing). To simulate realistic forecasting, we segment the raw recordings into 5,400 IMTS of 4,000 ms duration, using the first 3,000 ms to predict the final 1,000 ms. The maximum length of the time series is 130, the average value is 1.64, and the sparsity is 75.00%.

USHCN contains daily observations of five climatic variables from over 1,100 U.S. weather stations between 1996 and 2000, which comprises 1,114 samples, has an average observed value of 0.018 and a sparsity of 77.98%. Following the preprocessing of [43], we extract 26,736 series, whose maximum length is 214, each covering a 24-month history to forecast the subsequent month's measurements.

C.2 Datasets for classification and interpolation

For classification, we use the five released 80/10/10 splits of P19 and PAM. P19 is the sepsis classification benchmark derived from the PhysioNet 2019 challenge [44, 20]. PAM is the eight-class activity classification benchmark derived from PAMAP2 [45, 20]. For interpolation, we use the PhysioNet and Human Activity benchmarks with the 60/20/20 splits of ISTS-PLM [20].

D Baseline details

This appendix supplements Section 5.2 and preserves the per-model descriptions omitted from the main text.

To ensure comprehensive evaluation, we include twenty baselines spanning multivariate forecasting, irregular classification/forecasting, covering state-of-the-art architectures: GNNs and Transformers for spatio-temporal dependencies, set-based attention for asynchronous observations, neural differential equations for continuous-time modeling, and pretrained language-model adaptations. These encompass graph learning, time-frequency analysis, interpolation strategies, and stochastic processes addressing regular/irregular sampling, alignment issues, and missing data patterns in complex temporal scenarios.

All baseline results in the main benchmark table are obtained under the same hardware environment and the same evaluation pipeline. For each method, we follow the officially reported best hyperparameter choices and task-objective settings whenever they are explicitly specified in the original paper or released implementation; under this principle, each baseline is run with the standard five random seeds used throughout this work. Minor numerical deviations from the values reported in the corresponding baseline papers can still occur in practice due to differences in hardware platforms, CUDA and library versions, and accumulated floating-point effects. These discrepancies are especially noticeable on extremely sparse datasets, where repeated normalization, rescaling, masking, and near-zero denominators can amplify small numerical perturbations. In our experiments, however, such differences remain minor and do not affect the overall empirical conclusions or the comparative trends discussed in the main text.

All baseline results in the main benchmark table are obtained under the same hardware environment and the same evaluation pipeline. For each method, we follow the officially reported best hyperparameter choices and task-objective settings whenever they are explicitly specified in the original paper or released implementation; under this principle, each baseline is run with the standard five random seeds used throughout this work. Minor numerical deviations from the values reported in the corresponding baseline papers can still occur in practice due to differences in hardware platforms, CUDA and library versions, and accumulated floating-point effects. These discrepancies are especially noticeable on extremely sparse datasets, where repeated normalization, rescaling, masking, and near-zero denominators can amplify small numerical perturbations. In our experiments, however, such differences remain minor and do not affect the overall empirical conclusions or the comparative trends discussed in the main text.

In addition, we do not include IMTS-Mixer in the main benchmark baseline set. A brief discussion of this exclusion is provided in Appendix A.

D.1 Models for multivariate time series forecasting

DLinear [31]: DLinear proposes a decomposition-reconstruction framework that disentangles time series into trend and residual components, employing dual linear networks to separately model low-frequency trends and high-frequency fluctuations for decoupled temporal dynamics learning.

TimesNet [32]: TimesNet introduces a modular period-aware architecture that maps 1D time series to 2D temporal-frequency planes via time-frequency transforms, jointly modeling intra-period patterns and inter-period evolution to capture multi-scale spatio-temporal dependencies.

PatchTST [11]: A Transformer variant based on channel independence, utilizing local sequence patching to enhance local pattern extraction while maintaining global attention mechanisms for cross-timestep dependency modeling.

Crossformer [33]: Crossformer proposes a Transformer-based architecture for multivariate time series forecasting that explicitly decouples cross-temporal and cross-dimension dependency modeling through two dedicated modules.

GraphWavenet [34]: GraphWavenet integrates adaptive graph convolution with diffusion convolution, dynamically learning feature-wise topological relationships through self-adaptive adjacency matrices while capturing long-range temporal dependencies via gated temporal convolutions.

MTGNN [35]: MTGNN approaches multivariate time series forecasting from a graph neural network perspective. It uses a graph learning module to extract one-way variable relationships and form a graph structure. Then, it employs graph and time convolution modules to capture spatial and temporal dependencies. The model jointly learns graph structures and time series modeling in an end-to-end training manner.

StemGNN [36]: StemGNN maps spatio-temporal signals to spectral domains using graph Fourier transforms and discrete Fourier transforms, achieving unified representation learning of feature correlations and temporal periodicity in frequency space.

CrossGNN [37]: CrossGNN constructs multi-scale series via an Adaptive Multi-Scale Identifier to reduce temporal noise, applies a Cross-Scale GNN to model clearer trends across scales, and employs a Cross-Variable GNN to capture homogeneous and heterogeneous inter-variable relations. By focusing on high-saliency edges while constraining low-saliency ones, it achieves linear time and space complexity and robust forecasting performance.

FourierGNN [38]: FourierGNN builds a hypervariate graph treating each series value as a node in a fully connected space-time graph and replaces graph convolutions with a Fourier Graph Operator that performs frequency-domain matrix multiplications. By stacking FGO layers, it captures unified spatiotemporal dependencies with log-linear complexity and efficient high-order diffusion learning for forecasting.

D.2 Models for irregular time series classification

GRU-D [9]: GRU-D enhances GRU architecture with time decay mechanisms and missing value imputation strategies, jointly modeling observations, missingness patterns, and temporal intervals for dynamic representation of irregularly sampled sequences.

SEFT [17]: SEFT transforms irregular time series into set encodings, leveraging set functions and set attention mechanisms to extract robust features from asynchronous observations.

RainDrop [18]: RainDrop implements neural message passing with temporal self-attention to model inter-sensor dependencies, adaptively estimating missing measurements through neighboring observations for collaborative reasoning.

Warpformer [8]: Warpformer is a transformer-based model which employs a specific input representation that encodes both intra-series irregularity and inter-series discrepancy via adaptive warping modules to align series at each scale. It then uses customized attention modules and stacks multiple warping-attention blocks to produce multi-scale embeddings balancing coarse- and fine-grained temporal patterns for forecasting.

D.3 Models for irregular time series forecasting

mTAND [10]: mTAND embeds each irregular observation into a continuous-time latent space and applies a time-attention mechanism to compute weighted interpolations at reference timestamps. The resulting fixed-length representations enable efficient interpolation and classification of multivariate sparse time series.

Latent-ODE [6]: Latent ODE models irregularly sampled multivariate time series by first encoding observations into a latent initial state using an ODE-RNN encoder; it then generates a continuous-time latent trajectory by numerically solving a neural ODE, and employs a decoder to reconstruct observations at arbitrary timestamps. This framework inherently accommodates variable sampling intervals.

CRU [7]: CRUs handle irregularly-sampled multivariate time series by modeling hidden-state evolution via linear stochastic differential equations and closed-form Kalman-filter–derived update steps.

Within an encoder–decoder framework, CRUs propagate continuous latent states between observations and apply gating at observation times, enabling efficient integration of noisy, asynchronous data.

Neural Flow [14]: Neural Flows replace costly numerical solvers by directly parameterizing ODE solution curves with neural networks under precise flow-validity conditions, using architectures like residual, coupling, or autoregressive flows to ensure invertibility and expressiveness. This approach yields significant computational speedups while preserving generalization for tasks such as time-series modeling and density estimation.

T-PatchGNN [30]: t-PatchGNN transforms each irregular univariate series into uniformly-resolved patch sequences, enabling local semantics extraction and inter-series correlation modeling without sequence explosion. A time-adaptive GNN then learns dynamic inter-series dependencies over learned, time-varying graphs, yielding robust forecasts across many scientific domains.

ISTS-PLM [20]: ISTS-PLM adapts pretrained language models to irregular time series by replacing fixed positional embeddings with continuous-time embeddings, allowing timestamp information to be injected without assuming regular sampling. Its strength lies in transferring the representational capacity of large sequence backbones to irregular observations through continuous-time embedding design. Relative to IMTS-Tokenizer, ISTS-PLM primarily advances the downstream backbone, whereas our method focuses on the tokenizer itself and constructs irregularity-aware tokens before backbone modeling. In our benchmark protocol, ISTS-PLM is treated as a strong PLM-based irregular-forecasting baseline and is evaluated following its reported optimization target and best available parameter settings under the common five-seed protocol described above.

Hi-Patch [21]: Hi-Patch introduces hierarchical intra- and inter-patch graph attention layers for multi-scale irregular time-series modeling, capturing inter-channel dependencies through graph-based message passing across patch representations. Its main emphasis is hierarchical graph reasoning over patch-level representations, which makes it a strong comparison point for multi-scale irregular forecasting rather than a tokenizer-focused method. Relative to IMTS-Tokenizer, Hi-Patch relies more heavily on explicit graph-style interaction after patch construction, whereas our framework resolves irregularity earlier through anchor-based token formation and then applies standard downstream modeling. In our benchmark protocol, Hi-Patch is evaluated under the same hardware and five-seed setting, using the best task-oriented hyperparameter configuration reported by the original work or its official implementation.

E Parameter settings

E.1 Final forecasting configurations

This section supplements Section 5.7. We use Adam together with learning-rate decay and validation-based early stopping; the exact dataset-specific learning rates, milestones, batch sizes, patience values, and related settings listed in Table 6 should be regarded as the authoritative training configuration.

E.2 Parameter grids for the sensitivity analysis

Table 7 lists the values of the base anchor count H and minimum kernel precision $\gamma_{\min}$ examined in the sensitivity analyses in Appendix F. Table 6 reports the configurations used for the main forecasting experiments.

Table 6: Model Hyperparameters by Dataset

Hyperparameter	Activity	MIMIC	PhysioNet	USHCN
hid_dim	12	16	24	2
Number of scales (m)	2	2	2	4
num_layers	4	2	2	5
Learning-rate schedule	plateau	plateau	plateau	milestone [8]
patch_size	1500	12	12	3
history	3000	24	24	24
num_var	12	96	41	5
dropout	0.1	0.2	0.0	0.0
batch_size	32	32	32	192
lr	1e-3	1e-3	1e-3	1e-3
patience	12	12	12	12

Table 7: Parameter values examined in the sensitivity analyses.

Dataset	Base anchor count H	Minimum precision $\gamma_{\min}$
Activity	$24, 26, \ldots, 40$	$0.8, 0.9, \ldots, 1.2$
MIMIC-III	$6, 8, 9, 10, 12$	$0.05, 0.10, \ldots, 0.30$
PhysioNet	$6, 8, 9, 10$	$0.05, 0.10, 0.15, 0.20$
USHCN	$6, 7, \ldots, 12$	$0.8, 0.9, \ldots, 1.2$

F Hyperparameter sensitivity analysis

This appendix supplements Section 5.5 and expands the hyperparameter analyses summarized in the main text.

This appendix presents comprehensive hyperparameter sensitivity analyses conducted across four benchmark datasets. For each dataset, we systematically evaluate the impact of four key hyperparameters: base tokenizer width H (also the number of anchors per channel at the first scale), network depth L, the minimum gamma constraint $\gamma_{\min}$, and the number of scales m. All experiments are repeated with 5 random seeds to ensure statistical robustness, and we report both the mean performance and standard deviation across configurations.

F.1 USHCN dataset

The USHCN (U.S. Historical Climatology Network) dataset contains climate observations with 5 variables. We conducted 875 experiments spanning 175 configurations, achieving a 100% success rate. The hyperparameter search space is defined as: $H \in \{6, 7, 8, 9, 10, 11, 12\}$, $L \in \{1, 2, 3, 4, 5\}$, $\gamma_{\min} \in \{0.8, 0.9, 1.0, 1.1, 1.2\}$, with $\gamma_{\max} = 2.0$ fixed throughout.

Table 8 presents the effect of hidden dimension H on model performance. The results indicate that smaller hidden dimensions ($H = 6$–8) consistently yield lower MSE values, with $H = 6$ achieving the best average performance (MSE = 0.5032 ± 0.0025). Performance gradually degrades as H increases, suggesting that the relatively low-dimensional nature of USHCN (5 variables) does not require large hidden representations.

Table 9 examines the influence of network depth L. The model exhibits remarkable stability across different depth configurations, with MSE values ranging narrowly from 0.5059 to 0.5080. This robustness suggests that USHCN's relatively simple temporal patterns can be effectively captured by shallow architectures, while deeper networks provide no significant advantage.

Table 10 reports the sensitivity to the minimum gamma constraint $\gamma_{\min}$. Performance remains relatively stable across the tested range (0.8–1.2), with $\gamma_{\min} = 1.0$ achieving marginally better results. This indicates that USHCN benefits from moderate temporal anchoring, consistent with the regular sampling patterns characteristic of climate observation networks.

Table 8: Effect of hidden dimension H on USHCN dataset. Each row aggregates results from 25 configurations sharing the same H value. MSE and MAE statistics are computed across all configurations within each group.

H	#Configs	MSE (mean±std)	MAE (mean±std)
6	25	0.5032±0.0025	0.3142±0.0041
7	25	0.5037±0.0022	0.3138±0.0036
8	25	0.5049±0.0023	0.3153±0.0049
9	25	0.5079±0.0039	0.3185±0.0052
10	25	0.5089±0.0047	0.3197±0.0054
11	25	0.5101±0.0033	0.3223±0.0051
12	25	0.5116±0.0041	0.3214±0.0045

Table 9: Effect of network depth L on USHCN dataset. Each row aggregates results from 35 configurations sharing the same L value.

L	#Configs	MSE (mean±std)	MAE (mean±std)
1	35	0.5059±0.0039	0.3165±0.0049
2	35	0.5064±0.0047	0.3172±0.0063
3	35	0.5080±0.0050	0.3196±0.0056
4	35	0.5079±0.0038	0.3179±0.0054
5	35	0.5078±0.0049	0.3184±0.0057

Table 10: Effect of minimum gamma constraint $\gamma_{\min}$ on USHCN dataset. Each row aggregates results from 35 configurations sharing the same $\gamma_{\min}$ value.

$\gamma_{\min}$	#Configs	MSE (mean±std)	MAE (mean±std)
0.8	35	0.5077±0.0037	0.3178±0.0046
0.9	35	0.5069±0.0038	0.3178±0.0051
1.0	35	0.5065±0.0046	0.3168±0.0047
1.1	35	0.5077±0.0048	0.3185±0.0069
1.2	35	0.5071±0.0055	0.3185±0.0066

F.2 Activity dataset

The Activity dataset comprises human activity recognition data with 12 sensor variables. We conducted 675 experiments spanning 135 configurations, achieving a 100% success rate. The hyperparameter search space is defined as: $H \in \{24, 26, 28, 30, 32, 34, 36, 38, 40\}$, $L \in \{4, 5, 6\}$, $\gamma_{\min} \in \{0.8, 0.9, 1.0, 1.1, 1.2\}$, with $\gamma_{\max} = 2.0$ fixed throughout.

Table 11 illustrates the effect of hidden dimension on model performance. Smaller values of H (24–28) consistently outperform larger ones, with $H = 24$ achieving the lowest average MSE (2.787×10^{-3}). The performance degradation with increasing H suggests potential overfitting when the model capacity exceeds the intrinsic complexity of the activity recognition task.

Table 12 presents the network depth analysis. The model demonstrates stable performance across $L \in \{4, 5, 6\}$, with all configurations achieving comparable MSE values within a narrow range. This stability indicates that moderate network depths are sufficient for capturing the temporal dynamics inherent in activity recognition data.

Table 13 examines the sensitivity to $\gamma_{\min}$. A clear monotonic trend emerges: larger $\gamma_{\min}$ values consistently yield better performance, with $\gamma_{\min} = 1.2$ achieving the lowest average MSE (2.793×10^{-3}).

Table 11: Effect of hidden dimension H on Activity dataset. Each row aggregates results from 15 configurations sharing the same H value.

H	#Configs	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)
24	15	2.787±0.006	3.360±0.006
26	15	2.790±0.007	3.362±0.006
28	15	2.792±0.008	3.365±0.012
30	15	2.794±0.008	3.371±0.007
32	15	2.797±0.006	3.375±0.008
34	15	2.805±0.008	3.379±0.009
36	15	2.805±0.007	3.383±0.011
38	15	2.819±0.010	3.392±0.012
40	15	2.816±0.011	3.391±0.010

Table 12: Effect of network depth L on Activity dataset. Each row aggregates results from 45 configurations sharing the same L value.

L	#Configs	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)
4	45	2.800±0.012	3.374±0.013
5	45	2.799±0.014	3.373±0.014
6	45	2.802±0.015	3.379±0.016

Table 13: Effect of minimum gamma constraint $\gamma_{\min}$ on Activity dataset. Each row aggregates results from 27 configurations sharing the same $\gamma_{\min}$ value.

$\gamma_{\min}$	#Configs	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)
0.8	27	2.806±0.012	3.381±0.014
0.9	27	2.804±0.013	3.378±0.014
1.0	27	2.802±0.013	3.376±0.014
1.1	27	2.798±0.012	3.374±0.014
1.2	27	2.793±0.012	3.369±0.014

Table 14: Effect of hidden dimension H on PhysioNet dataset. Each row aggregates results from 12 valid configurations sharing the same H value.

H	#Configs	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)
6	12	5.095±0.063	3.761±0.040
8	12	5.008±0.061	3.716±0.032
9	12	4.947±0.061	3.690±0.023
10	12	5.012±0.063	3.713±0.032

F.3 PhysioNet dataset

The PhysioNet dataset contains 41 physiological variables recorded from ICU patients. We attempted five training runs for each of 60 configurations, of which 240 of the 300 runs completed successfully (80%). The sensitivity analysis covers 48 configurations with $H \in \{6, 8, 9, 10\}$, $L \in \{3, 4, 5\}$, and $\gamma_{\min} \in \{0.05, 0.10, 0.15, 0.20\}$. We fixed $\gamma_{\max} = 2.0$ throughout this analysis.

Table 14 reports the effect of hidden dimension on PhysioNet. The intermediate value $H = 9$ achieves the best average performance (MSE = 4.947×10^{-3}), outperforming both smaller ($H = 6$) and larger ($H = 10$) alternatives. This suggests an optimal trade-off between model capacity and generalization for the moderate dimensionality of PhysioNet data.

Table 15 analyzes the network depth sensitivity. Shallower networks ($L = 3$) slightly outperform deeper configurations, achieving the lowest average MSE (4.994×10^{-3}). The marginal differences

Table 15: Effect of network depth L on PhysioNet dataset. Each row aggregates results from 16 valid configurations sharing the same L value.

L	#Configs	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)
3	16	4.994±0.076	3.718±0.038
4	16	5.038±0.082	3.721±0.053
5	16	5.014±0.081	3.722±0.030

Table 16: Effect of minimum gamma constraint $\gamma_{\min}$ on PhysioNet dataset. Each row aggregates results from 12 valid configurations sharing the same $\gamma_{\min}$ value.

$\gamma_{\min}$	#Configs	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)
0.05	12	5.008±0.094	3.712±0.032
0.10	12	5.023±0.085	3.718±0.043
0.15	12	5.005±0.070	3.715±0.048
0.20	12	5.025±0.078	3.736±0.037

Table 17: Effect of hidden dimension H on MIMIC-III dataset. Each row aggregates results from valid configurations sharing the same H value. The reduced count for $H = 12$ reflects partial experimental success.

H	#Configs	MSE ($\times 10^{-2}$)	MAE ($\times 10^{-2}$)
6	18	1.604±0.006	6.936±0.033
8	18	1.609±0.008	6.930±0.027
9	18	1.612±0.009	6.948±0.031
10	18	1.618±0.010	6.949±0.027
12	12	1.630±0.010	6.990±0.043

across depth values indicate that PhysioNet's temporal patterns can be effectively captured without deep architectures.

Table 16 presents the $\gamma_{\min}$ sensitivity analysis. Unlike USHCN and Activity, PhysioNet exhibits a preference for smaller $\gamma_{\min}$ values, with $\gamma_{\min} = 0.05$ and 0.15 yielding the lowest average MSE values among the tested settings. This pattern reflects the highly irregular sampling characteristic of clinical time series, where flexible temporal anchoring is advantageous.

F.4 MIMIC-III dataset

The MIMIC-III dataset contains 96 clinical variables recorded from ICU patients. We attempted five training runs for each of 126 configurations. Of the 630 runs, 416 completed successfully (66.0%). We retained the 84 configurations with at least three successful runs for the sensitivity analysis. These configurations used $H \in \{6, 8, 9, 10, 12\}$, $L \in \{4, 5, 6\}$, and $\gamma_{\min} \in \{0.05, 0.10, 0.15, 0.20, 0.25, 0.30\}$, with $\gamma_{\max} = 3.0$ throughout. Among the 18 configurations with $H = 12$, 12 met the inclusion criterion.

Table 17 presents the hidden dimension analysis. Smaller dimensions ($H = 6$–8) consistently outperform larger ones, with $H = 6$ achieving the best average MSE (1.604×10^{-2}). The performance degradation at $H = 12$ may be attributed to increased model complexity relative to the available training data.

Table 18 examines network depth effects. In contrast to other datasets, MIMIC-III benefits from deeper architectures, with $L = 6$ achieving the lowest average MSE (1.609×10^{-2}). This finding aligns with the high dimensionality (96 variables) and complex inter-variable dependencies characteristic of ICU monitoring data, which require deeper networks to model effectively.

Table 19 reports the $\gamma_{\min}$ sensitivity. Similar to PhysioNet, MIMIC-III performs optimally with smaller $\gamma_{\min}$ values, achieving the best results at $\gamma_{\min} = 0.25$ and $\gamma_{\min} = 0.05$. The preference for

Table 18: Effect of network depth L on MIMIC-III dataset. Each row aggregates results from valid configurations sharing the same L value.

L	#Configs	MSE ($\times 10^{-2}$)	MAE ($\times 10^{-2}$)
4	29	1.616±0.010	6.953±0.035
5	29	1.615±0.011	6.960±0.033
6	26	1.609±0.013	6.928±0.035

Table 19: Effect of minimum gamma constraint $\gamma_{\min}$ on MIMIC-III dataset. Each row aggregates results from valid configurations sharing the same $\gamma_{\min}$ value. Configuration counts vary due to differential experimental success rates.

$\gamma_{\min}$	#Configs	MSE ($\times 10^{-2}$)	MAE ($\times 10^{-2}$)
0.05	13	1.612±0.010	6.938±0.041
0.10	15	1.615±0.015	6.957±0.037
0.15	14	1.613±0.012	6.947±0.027
0.20	15	1.616±0.010	6.938±0.035
0.25	13	1.610±0.009	6.942±0.037
0.30	14	1.614±0.013	6.963±0.039

Table 20: Effect of the number of scales m on model performance. Results are reported as mean ± std over 5 random seeds. **Bold**: best performance per dataset.

Dataset	Metric	$m = 1$	$m = 2$	$m = 3$	$m = 4$
PhysioNet	MSE ($\times 10^{-3}$)	5.39±0.04	**4.82±0.05**	4.90±0.16	5.08±0.11
	MAE ($\times 10^{-2}$)	3.79±0.12	**3.65±0.05**	3.78±0.10	3.85±0.08
MIMIC	MSE ($\times 10^{-2}$)	1.68±0.03	**1.59±0.02**	1.64±0.02	1.94±0.06
	MAE ($\times 10^{-2}$)	7.04±0.12	**6.93±0.07**	7.18±0.09	8.23±0.25
Activity	MSE ($\times 10^{-3}$)	2.54±0.02	**2.52±0.00**	2.69±0.02	14.76±0.24
	MAE ($\times 10^{-2}$)	3.09±0.27	**3.08±0.00**	3.18±0.03	7.42±0.12
USHCN	MSE ($\times 10^{-1}$)	5.26±0.04	5.35±0.04	5.13±0.04	**4.98±0.04**
	MAE ($\times 10^{-1}$)	3.23±0.04	3.30±0.05	3.18±0.04	**3.07±0.06**

flexible temporal anchoring reflects the highly irregular sampling patterns inherent in clinical data collection.

F.5 Multi-scale sensitivity

Multi-scale modeling captures temporal patterns at different granularities, from fine-grained local fluctuations to coarse-grained global trends. We investigate the impact of the number of scales $m \in \{1, 2, 3, 4\}$ on model performance across all four datasets. Figure 5 visualizes the comparative performance across different scale configurations.

Table 20 shows that $m = 2$ yields the lowest mean MSE on PhysioNet, MIMIC-III, and Activity among the four scale counts tested. Compared with $m = 1$, using two scales reduces mean MSE by 10.6% on PhysioNet, 5.4% on MIMIC-III, and 0.8% on Activity. On Activity, mean MSE rises from 2.52 at $m = 2$ to 14.76 at $m = 4$ (in units of 10^{-3}).

On USHCN, mean MSE increases from 5.26 at $m = 1$ to 5.35 at $m = 2$, then decreases to 5.13 at $m = 3$ and 4.98 at $m = 4$ (in units of 10^{-1}). Four scales also yield the lowest mean MAE on this dataset.

These findings suggest practical guidelines for setting m: (1) $m = 2$ serves as a robust default for most time series forecasting tasks; (2) datasets with long-range temporal dependencies and clear multi-scale patterns (e.g., climate data) may benefit from larger m values; (3) high-frequency sensor data with localized patterns should avoid excessive scales to prevent overfitting to noise.

Table 21: Relative range of marginal mean MSE across the sensitivity grids, in percent.

Dataset	H	$\gamma_{\min}$
Activity	1.15	0.47
MIMIC-III	1.62	0.37
PhysioNet	2.99	0.40
USHCN	1.67	0.24

Table 22: Aggregate significance analysis across the eight dataset-metric tasks. Relative improvement is computed in terms of MSE reduction.

Comparison	Rel. Impr. (%)	95% Bootstrap CI	Wilcoxon p	Paired t-test p	Effect size d_z
vs ISTS-PLM	2.28	[1.90, 2.82]	0.0313	5.37×10^{-4}	3.78
vs Hi-Patch	3.40	[2.33, 4.47]	0.0313	2.39×10^{-3}	2.54

Table 23: Average rank across the eight dataset-metric tasks. Lower is better.

Method	Average Rank
IMTS-Tokenizer	**1.19**
ISTS-PLM	2.38
Hi-Patch	2.44

F.6 Variation in marginal mean MSE across the sensitivity grids

For each candidate value of H, we average MSE over the valid configurations with that value. We apply the same procedure to $\gamma_{\min}$. Table 21 reports the relative range $100(\max \bar{e} - \min \bar{e})/\min \bar{e}$ for each parameter, where $\bar{e}$ denotes a group mean. The largest range across the four datasets is 2.99% for H and 0.47% for $\gamma_{\min}$.

G Additional result analyses

This appendix collects auxiliary analyses that complement the main benchmark results. The following material is not required to follow the primary empirical conclusions, but it sharpens the interpretation of statistical reliability and behavior outside the standard irregular benchmarks.

G.1 Aggregate statistical significance

This subsection supplements Section 5.3. To assess whether the observed gains over the strongest recent baselines remain reliable beyond overlapping mean$\pm$std intervals on individual tasks, we aggregate the eight dataset-metric comparisons against ISTS-PLM and Hi-Patch and perform paired significance tests over the five-seed outcomes. Table 22 reports the resulting aggregate test statistics, and Table 23 provides the corresponding average-rank summary across the same tasks.

As further summarized in Table 23, the proposed tokenizer also attains the best average rank across the same eight dataset-metric tasks. Together, the aggregate evidence indicates that, although some individual mean$\pm$std intervals overlap, the overall gains against the strongest recent baselines remain statistically reliable and consistently favor the proposed tokenizer.

G.2 Regular-sampling and controlled missingness studies

This subsection complements the discussion in Section J by broadening the supplementary evidence beyond the four standard irregular benchmarks. We examine three controlled settings: random masking on five regularly sampled fMRI datasets, a synthetic benchmark generated to preserve realistic trend and periodic structure, and a structured block-missing benchmark designed to mimic bursty sensor failures. Tables 24, 25, and 26 report the corresponding results. These studies are supplementary because they do not alter the main benchmark protocol, but they help clarify the range of regimes in which anchor-based tokenization remains effective.

Table 24: Random masking on five regularly sampled fMRI datasets. Lower MSE is better. "OOM" indicates out-of-memory.

Sparsity	Dataset	Ours	Hi-Patch	ISTS-PLM	CRU	Latent-ODE	Neural-flow	iTransformer
0%	fMRI1	3.71±0.03	3.79±0.00	3.73±0.02	3.93±0.17	9.13±4.14	15.49±10.14	4.17±0.14
	fMRI2	3.52±0.21	OOM	3.66±0.01	3.85±0.10	7.10±4.01	17.19±13.78	3.98±0.14
	fMRI3	2.69±0.02	2.75±0.00	2.76±0.01	2.82±0.08	2.91±0.12	2.84±0.09	3.26±0.05
	fMRI4	3.12±0.06	3.41±0.04	3.40±0.02	3.16±0.75	3.50±0.37	4.14±0.98	4.10±0.09
	fMRI5	3.26±0.04	3.35±0.04	3.36±0.09	3.57±0.76	3.63±0.57	3.55±0.30	3.71±0.12
50%	fMRI1	4.36±0.10	4.63±0.11	4.67±0.18	4.39±0.53	7.56±2.51	20.50±11.55	5.20±0.28
	fMRI2	4.12±0.21	4.47±0.05	4.46±0.10	4.70±0.58	6.87±1.82	32.63±14.87	5.05±0.32
	fMRI3	2.54±0.01	2.76±0.00	2.77±0.01	3.01±0.09	3.11±0.17	3.32±0.74	3.08±0.05
	fMRI4	3.41±0.03	3.55±0.04	3.55±0.02	3.35±0.64	3.79±0.70	5.25±2.07	4.37±0.12
	fMRI5	3.13±0.05	3.32±0.06	3.33±0.06	3.46±0.52	3.90±0.68	5.16±2.01	3.99±0.20
90%	fMRI1	5.45±0.54	5.48±0.13	5.49±0.25	7.33±1.82	12.34±4.55	29.45±14.52	10.27±0.76
	fMRI2	6.55±0.19	6.60±0.23	6.64±0.19	8.60±3.05	13.06±4.48	34.62±13.73	8.05±0.09
	fMRI3	3.86±0.05	3.94±0.02	3.92±0.02	4.02±0.48	4.49±0.70	5.44±2.33	6.16±0.22
	fMRI4	3.53±0.08	3.60±0.05	3.56±0.03	3.55±0.67	4.06±0.60	5.57±2.56	6.43±0.24
	fMRI5	3.25±0.03	3.35±0.04	3.32±0.04	3.62±0.67	4.07±0.59	5.73±2.18	6.65±0.13

Table 25: Random masking on the synthetic benchmark generated from real-series statistics. Lower MSE is better.

Sparsity	Ours	Hi-Patch	ISTS-PLM	CRU	Latent-ODE	Neural-flow	iTransformer
0%	1.68±0.04	2.06±0.00	2.06±0.02	2.32±0.55	2.78±0.55	7.92±8.12	2.06±0.02
50%	1.85±0.12	2.53±0.14	2.50±0.13	2.64±0.75	2.69±0.85	9.92±8.07	2.47±0.14
90%	2.66±0.29	3.75±1.10	3.69±1.06	5.44±2.09	6.30±2.21	21.83±15.72	4.43±1.09

For the real-data fMRI study, results are averaged over five regularly sampled datasets rather than a single masked realization. We evaluate sparsity levels of 0%, 50%, and 90%. For the masked settings, each configuration is repeated over three independent masking realizations and five training seeds so that the reported mean±std reflects both masking and optimization variability. All baselines use either their default settings or grid-searched hyperparameters under the same protocol. Table 24 summarizes the resulting random-masking performance on the five fMRI datasets.

As shown in Table 24, IMTS-Tokenizer remains competitive across these real-data settings even without irregular sampling, and its advantage remains visible under increased masking. Importantly, the advantage is not restricted to comparison against a single regular-series baseline: the same protocol also includes strong irregular-series baselines such as Hi-Patch, ISTS-PLM, CRU, Latent-ODE, and Neural-flow.

To test whether the trend persists under controlled dynamics, we further construct a synthetic benchmark with 10 channels and length 1,000 using a WaveNet/TCN-style generator. The generator preserves trend and periodic structure extracted from real data while causally modeling the residual dynamics, allowing the same masking protocol to be evaluated in a more controlled environment. Table 25 reports the corresponding synthetic-benchmark results. We use five training seeds per configuration. At 50% and 90% random missingness, we also average over three independent mask realizations.

As further summarized in Table 25, the same trend persists on the controlled synthetic benchmark, indicating that the advantage is not tied only to the specific idiosyncrasies of the real fMRI datasets.

Random masking is still optimistic because it removes points independently. We therefore add a structured block-missing benchmark with synchronized multi-channel gaps and independent heavy-tailed single-channel block failures under the same overall sparsity level. This setting is more demanding because contiguous temporal evidence and cross-channel context may disappear simultaneously, which better reflects bursty sensor failures than i.i.d. masking. Table 26 presents the results under this structured block-missing protocol. The structured masks draw on prior work on block missingness in multivariate time series [46].

As shown in Table 26, these supplementary studies sharpen the regime analysis in two ways. First, on the real-data fMRI benchmark, the median relative MSE reduction over iTransformer reaches 18.63%, showing that the tokenizer remains effective even before irregularity is introduced. Second, under

Table 26: Structured block-missing benchmark on five fMRI datasets and the synthetic WaveNet benchmark. Lower MSE is better.

Sparsity	Dataset	Ours	Hi-Patch	ISTS-PLM	CRU	Latent-ODE	Neural-flow	iTransformer
50%	fMRI1	4.74±0.08	4.89±0.08	4.96±0.07	6.05±0.93	9.85±2.92	26.33±10.81	7.29±0.28
	fMRI2	4.18±0.19	4.33±0.05	4.34±0.03	4.54±0.60	7.76±5.35	26.83±11.54	6.64±0.28
	fMRI3	2.95±0.02	2.91±0.00	2.94±0.01	3.02±0.32	3.22±0.35	3.24±0.54	3.51±0.07
	fMRI4	3.47±0.06	3.58±0.03	3.67±0.09	3.64±0.16	3.60±0.57	5.56±1.46	6.70±0.14
	fMRI5	3.06±0.08	3.11±0.05	3.24±0.02	3.66±0.30	3.68±0.80	5.86±3.05	7.76±0.11
90%	fMRI1	11.80±2.07	14.99±5.52	11.91±3.80	68.28±93.63	36.84±27.12	38.12±21.47	31.30±0.34
	fMRI2	5.14±0.25	5.42±0.17	5.35±0.09	16.69±11.30	13.28±3.77	34.12±19.54	8.75±0.28
	fMRI3	2.81±0.02	2.91±0.03	2.89±0.03	4.33±1.83	4.02±0.95	3.56±0.65	4.25±0.16
	fMRI4	4.62±0.20	4.81±0.11	4.71±0.11	5.82±2.21	4.80±0.47	9.49±3.59	14.00±0.61
	fMRI5	4.33±0.21	4.53±0.14	4.38±0.14	4.89±1.18	4.71±0.95	8.07±1.56	12.68±0.18
50%	WaveNet	1.95±0.17	2.84±0.01	2.95±0.10	4.03±1.19	5.93±0.91	26.18±8.60	6.75±0.19
90%	WaveNet	4.53±0.29	10.67±1.79	10.68±1.36	11.26±7.95	13.76±14.12	22.42±16.63	16.45±0.07

Table 27: Exact one-sided sign tests over benchmark settings for the supplementary regular-sampling and missingness studies. Ties and OOM cases are excluded from the tested count.

Setting group	Comparison	Wins / tested settings	Exact sign-test p
Real-data random masking	vs iTransformer	15 / 15	3.05×10^{-5}
Real-data random masking	vs Hi-Patch	14 / 14	6.10×10^{-5}
Structured block missingness	vs iTransformer	12 / 12	2.44×10^{-4}
Structured block missingness	vs ISTS-PLM	11 / 12	3.17×10^{-3}

structured block missingness, the corresponding median reduction rises to 54.39%, indicating that the advantage grows when missingness becomes more realistic and temporally correlated. The per-setting win counts are also consistent with this trend: IMTS-Tokenizer records 13 of 14 non-OOM wins over Hi-Patch on the real-data masking studies and 11 of 12 wins over ISTS-PLM on the block-missing benchmark. These results therefore strengthen the interpretation that the proposed tokenizer is especially beneficial once temporal evidence becomes sparse, fragmented, or unevenly distributed.

To complement these descriptive summaries with a formal statistical check, we also report exact one-sided sign tests over benchmark settings. This analysis is appropriate here because the supplementary studies combine heterogeneous masking protocols and dataset families, so the key question is whether the direction of improvement remains consistently favorable across settings rather than whether a single pooled effect size dominates all regimes. As shown in Table 27, IMTS-Tokenizer achieves uniformly favorable or near-uniformly favorable outcomes against both regular-series and strong irregular-series baselines, providing additional statistical support for the broader-baseline comparisons in this appendix.

Under this setting-level test, the superiority over iTransformer remains statistically reliable in both the real-data random-masking and structured block-missing regimes, while the comparisons against Hi-Patch and ISTS-PLM also retain strong directional support. Together with the median relative improvements and full result tables above, these statistics make the broader-baseline conclusion explicit rather than merely descriptive. The synthetic trajectory provides a separate comparison in which the missingness pattern can be varied while keeping the underlying dynamics fixed. Relative to the better of Hi-Patch and ISTS-PLM in each setting, the MSE reductions are 18.4% without masking, 26.0% and 27.9% with random missingness at 50% and 90%, and 31.3% and 57.5% with block missingness at the same two levels (Tables 25 and 26). At 50% random missingness, iTransformer has a lower mean MSE than either of those two references (2.47), and the reduction relative to iTransformer is 25.1%.

G.3 Test-time shifts in sampling density and missingness pattern

We use the synthetic trajectory from Appendix G.2 to test how forecasting changes under a shift in the observation pattern. Let R_{50} and R_{90} denote 50% and 90% random point missingness, and let B_{50} denote 50% structured block missingness. For each source condition, training and validation use only that condition. We evaluate the checkpoint selected by source validation MSE on both the matched and shifted test masks. The underlying trajectory, targets, partition, hyperparameters, and

Table 28: Test-time mask shifts on the synthetic trajectory. We evaluate each shift using three paired test masks. MSE columns report $100 \times \mathrm{MSE}$. Relative changes use the unrounded matched error as the denominator. Coverage is the percentage of anchor instances with an observation inside the Gaussian half-maximum radius.

	Forecasting error			Anchor coverage (%)	
Source $\rightarrow$ test	Matched	Shifted	Change (%)	Matched	Shifted
$\mathrm{R}_{50} \rightarrow \mathrm{R}_{90}$	2.9756	3.0607	$+2.86$	100.00	98.06
$\mathrm{R}_{90} \rightarrow \mathrm{R}_{50}$	3.0119	2.9777	-1.13	98.06	100.00
$\mathrm{R}_{50} \rightarrow \mathrm{B}_{50}$	2.9756	3.2685	$+9.84$	100.00	88.61
$\mathrm{B}_{50} \rightarrow \mathrm{R}_{50}$	2.9966	3.0003	$+0.12$	88.61	100.00

training budget are held fixed. The observations retained under R_{90} form a subset of those retained under R_{50}, and the R_{50} and B_{50} masks retain the same number of observations.

For random point masks, increasing test missingness from 50% to 90% raises MSE by 2.86%, while decreasing it from 90% to 50% lowers MSE by 1.13% (Table 28). Changing the pattern at fixed missingness has a larger effect in one direction. Moving from random to block masks raises MSE by 9.84% and reduces anchor coverage by 11.39 percentage points. The reverse shift increases MSE by only 0.12%. The loss of local observational support therefore coincides with the largest increase in error on this trajectory.

G.4 Classification and interpolation

We trained a separate model for each task while retaining the Gaussian IRAM tokenizer, cross-channel attention, and the corresponding forecasting tokenizer settings. For classification, we replaced the forecasting head with masked mean pooling over observed variables and a classifier. The P19 classifier also used the static covariates. For interpolation, we withheld 30% of the observed timestamp rows and used the forecasting head to predict at those times. We computed MSE on the originally observed targets and averaged it across variables [20]. Table 29 reports our results on the four benchmarks described in Appendix C.2.

Table 29: IMTS-Tokenizer results with the forecasting tokenizer settings. Entries give the mean and standard deviation over five released splits for classification and five training seeds for interpolation.

Task	Dataset	Metric	IMTS-Tokenizer
Classification	P19	AUROC (%)	89.4 ± 1.7
Classification	PAM	Accuracy (%)	95.8 ± 0.9
Interpolation	PhysioNet	MSE (10^{-3})	4.58 ± 0.07
Interpolation	Human Activity	MSE (10^{-3})	1.89 ± 0.01

Table 30: Ablation study results. MSE/MAE reported as mean±std over 5 seeds. **Bold**: best per dataset. Underline: second-best per dataset.

Method	PhysioNet		MIMIC-III		Human Activity		USHCN	
	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-2}$	MAE$\times 10^{-2}$	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-1}$	MAE$\times 10^{-1}$
Full Model	**4.82 ± 0.05**	**3.65 ± 0.05**	**1.59 ± 0.02**	**6.93 ± 0.07**	**2.52 ± 0.00**	**3.08 ± 0.00**	**4.98 ± 0.04**	**3.07 ± 0.06**
w/o VSA	4.99 ± 0.11	3.71 ± 0.09	1.61 ± 0.04	7.05 ± 0.08	2.87 ± 0.02	3.44 ± 0.03	5.02 ± 0.04	3.19 ± 0.06
w/o RTF	5.08 ± 0.19	3.75 ± 0.10	1.66 ± 0.01	7.07 ± 0.07	2.73 ± 0.01	3.25 ± 0.01	5.12 ± 0.09	3.21 ± 0.14
w/o AC	5.00 ± 0.15	3.67 ± 0.08	1.62 ± 0.03	6.95 ± 0.10	2.87 ± 0.03	3.45 ± 0.01	5.01 ± 0.05	3.17 ± 0.16
w/o CA	5.57 ± 0.03	3.93 ± 0.02	1.72 ± 0.01	7.43 ± 0.03	3.20 ± 0.01	3.51 ± 0.01	5.15 ± 0.02	3.12 ± 0.09
w/o multi-scale	5.39 ± 0.04	3.79 ± 0.12	1.68 ± 0.03	7.04 ± 0.12	2.54 ± 0.02	3.09 ± 0.27	5.26 ± 0.04	3.23 ± 0.04
w/o channel share	5.40 ± 0.10	3.86 ± 0.09	1.68 ± 0.09	7.16 ± 0.10	2.65 ± 0.01	3.16 ± 0.01	5.11 ± 0.07	3.20 ± 0.06
rp-MLP	5.40 ± 0.08	3.71 ± 0.05	1.69 ± 0.01	7.17 ± 0.08	2.73 ± 0.02	3.17 ± 0.05	5.13 ± 0.06	3.11 ± 0.04

H Ablation experiment details

This appendix supplements Section 5.4 and restores the extended ablation evidence omitted from the main text.

This appendix provides comprehensive ablation study results that complement the main text analysis. We systematically evaluate the contribution of each proposed component and examine alternative design choices for key modules.

H.1 Experimental protocol

All ablation experiments follow a consistent protocol to ensure fair comparison. We employ five random seeds $\{42, 43, 44, 45, 46\}$ for each configuration, reporting mean and standard deviation across runs. The single-module and progressive ablation campaign encompasses 300 individual runs spanning 15 configurations across 4 benchmark datasets. We achieved an overall success rate of 98.0% (294/300), with the 6 failures occurring on MIMIC-III due to GPU memory constraints under specific seed and batch size combinations.

H.2 Complete ablation results

Table 30 presents the complete ablation results with both MSE and MAE metrics. This table extends the main text Table 2 by including MAE values, thereby providing a more comprehensive view of model behavior under different ablation conditions. Removing cross-channel attention (CA) increases mean MSE on all four datasets, and removing RTF increases mean MSE on both clinical benchmarks. Three additional observations are important for interpreting the compressed main-text discussion. First, removing multi-scale processing leads to clear degradation, most notably on PhysioNet, indicating that hierarchical temporal resolution is beneficial rather than incidental. Second, sharing one tokenizer across channels also hurts performance, especially on Activity, which supports the need for channel-specific tokenization under asynchronous observations. Third, the rp-MLP variant consistently underperforms the full model, showing that replacing the attention interaction module with an MLP weakens the representation even when the tokenizer itself is preserved.

H.3 Tokenizer replacement and progressive module removal

This subsection supplements Section 5.4. To isolate the contribution of the tokenizer backbone from auxiliary modules, we start from the full model and remove VSA, boundary/diversity regularization

Table 31: Progressive module-removal and tokenizer-replacement ablations. MSE reported as mean±std over 5 seeds.

Method	USHCN ($\times 10^{-1}$)	Activity ($\times 10^{-3}$)	PhysioNet ($\times 10^{-3}$)	MIMIC ($\times 10^{-2}$)
Full Model	**4.98 ± 0.04**	**2.52 ± 0.00**	**4.82 ± 0.05**	**1.59 ± 0.02**
w/o VSA	5.02 ± 0.04	2.87 ± 0.02	4.99 ± 0.11	1.61 ± 0.04
w/o VSA + AC	5.03 ± 0.06	2.87 ± 0.01	5.02 ± 0.16	1.61 ± 0.02
w/o VSA + AC + RTF	5.05 ± 0.08	2.90 ± 0.12	5.95 ± 0.12	1.64 ± 0.01
w/o VSA + AC + RTF + CA	5.19 ± 0.04	2.91 ± 0.00	14.16 ± 0.02	4.47 ± 0.01
w/o VSA + AC + RTF + CA + multi-scale (Tokenizer Only)	5.74 ± 0.01	2.90 ± 0.01	14.43 ± 0.05	4.48 ± 0.02
Linear interpolation + Attention	8.00 ± 0.03	4.53 ± 0.01	18.93 ± 0.02	7.09 ± 0.00
IRAM + MLP (rp-MLP)	5.13 ± 0.06	2.73 ± 0.02	5.40 ± 0.08	1.69 ± 0.01
Linear interpolation + standalone MLP	6.05 ± 0.04	4.06 ± 0.01	7.23 ± 0.19	5.84 ± 0.02

Table 32: Relative MSE reductions (%) from the reported configuration means. Each row uses the named reference model as the denominator. The first row retains the attention predictor. The other rows compare complete configurations.

Model	Reference	USHCN	Activity	PhysioNet	MIMIC-III
Full	Interp. + Attn.	37.8	44.4	74.5	77.6
rp-MLP	Interp. + MLP	15.2	32.8	25.3	71.1
Anchor core	Interp. + Attn.	28.3	36.0	23.8	36.8

(AC), RTF, cross-channel attention (CA), and multi-scale modeling step by step while keeping the same anchor-based tokenizer backbone. We also replace IRAM with linear interpolation while retaining the attention predictor. Table 31 reports the progressive ablations alongside this comparison, the rp-MLP variant with IRAM, and a standalone MLP trained on interpolated inputs. This analysis is complementary to the rp-MLP ablation in Table 30: the rp-MLP variant tests replacing the attention interaction module, whereas the present subsection tests removing or replacing the tokenizer front end.

With the attention predictor retained, the full IRAM front end reduces MSE by 37.8–77.6% relative to linear interpolation (Table 32). With IRAM retained, replacing attention with an MLP increases MSE by 3.0–12.0% relative to the full model. Equivalently, restoring attention reduces MSE by 2.9–10.7% when the MLP variant is the reference. For example, the change from 4.82 to 5.40 on PhysioNet is a 12.0% increase, whereas the reverse change is a 10.7% reduction.

The rp-MLP variant also has lower mean MSE than the standalone interpolation–MLP reference on all four datasets. The anchor-only configuration in Table 31, which retains Gaussian aggregation and omits cross-channel attention, has 23.8–36.8% lower mean MSE than interpolation with attention. Restoring the full architecture reduces MSE relative to the core by 66.6% on PhysioNet and 64.5% on MIMIC-III, compared with 13.1% on Activity and 13.2% on USHCN. The largest step on the two clinical benchmarks occurs when the model also loses cross-channel attention after VSA, AC, and RTF have been removed. MSE rises from 5.95 to 14.16 on PhysioNet and from 1.64 to 4.47 on MIMIC-III. These changes describe the specified removal order.

H.4 RTF encoding variants

We evaluate five encoding functions for the Relative Time Feature (RTF) module to determine the optimal temporal representation. Table 33 compares Tanh, Linear, Gaussian, Sinusoidal, and MLP encodings across all datasets. The superscript notation indicates cases where fewer than 5 seeds completed successfully.

Table 33 shows that Tanh yields the lowest mean MSE on PhysioNet, MIMIC-III, and Activity. On Activity, Tanh achieves a mean MSE of 2.52, compared with 2.73 for Gaussian encoding (in units of 10^{-3}). On USHCN, Tanh, linear, and sinusoidal encodings share the lowest mean MSE at the reported precision (4.98×10^{-1}).

Table 33: RTF encoding comparison. MSE reported as mean ± std. Superscripts indicate number of successful seeds when fewer than 5.

	PhysioNet ($\times 10^{-3}$)	**MIMIC-III** ($\times 10^{-2}$)	**Activity** ($\times 10^{-3}$)	**USHCN** ($\times 10^{-1}$)
Tanh	$\mathbf{4.82 \pm 0.05}$	$\mathbf{1.59 \pm 0.02}$	$\mathbf{2.52 \pm 0.00}$	$\mathbf{4.98 \pm 0.04}$
Linear	5.00 ± 0.13	1.63 ± 0.02	2.85 ± 0.02	$\mathbf{4.98 \pm 0.07}$
Gaussian	4.99 ± 0.17	1.63 ± 0.01	2.73 ± 0.01	5.08 ± 0.04
Sinusoidal	5.10 ± 0.13	1.63 ± 0.02	2.86 ± 0.01	$\mathbf{4.98 \pm 0.03}$
MLP	5.13 ± 0.10	1.63 ± 0.01	2.79 ± 0.05	5.00 ± 0.05

Table 34: Relative MSE increase of `w/o AC+CA` under matched anchor-number and initialization settings. Lower is better.

Configuration	Setting	**Activity**	**PhysioNet**	**USHCN**	**MIMIC**
Anchor number	$H = 2$	+13.72%	+6.06%	+4.78%	+2.13%
	$H = 4$	+14.95%	+7.21%	+2.31%	+7.21%
	$H = 6$	+13.16%	+7.19%	+3.23%	+7.16%
	$H = 8$	+12.31%	+5.50%	+1.65%	+6.55%
	$H = 12$	+12.15%	+9.61%	+0.28%	+6.27%
Initialization	VSA	+13.34%	+10.36%	+3.23%	+6.48%
	Boundary-Aware	+10.53%	+7.43%	+1.86%	+5.79%
	Uniform	+12.05%	+6.09%	+1.24%	+6.34%
	Random	+14.68%	+10.23%	+1.30%	+6.83%

H.5 Matched AC+CA removal under fixed configurations

To verify that the observed robustness is not merely a consequence of a favorable anchor configuration, we additionally evaluate the relative MSE increase caused by removing AC and CA under matched anchor-number and initialization settings. Table 34 shows that the degradation remains consistent across all tested configurations, confirming that the stability of the full model arises from the complete anchor design rather than from anchor selection alone.

Across matched settings, removing AC and CA increases MSE by 10.53–14.95% on Activity, 5.50–10.36% on PhysioNet, 0.28–4.78% on USHCN, and 2.13–7.21% on MIMIC. These results show that the robustness of the full model cannot be attributed to a single favorable anchor configuration, but instead depends on the joint contribution of data-aligned initialization, regularization, and cross-channel interaction.

H.6 Analysis of RTF contribution

The contribution of RTF varies across datasets with different temporal characteristics. On clinical datasets (PhysioNet and MIMIC-III), removing RTF results in 4–6% MSE degradation, validating its role in preserving fine-grained temporal structure where precise timing carries predictive significance. For the Human Activity dataset, RTF similarly contributes to improved performance, as the full model (MSE = 2.52) outperforms the ablated variant (MSE = 2.73), representing an 8% improvement. This confirms that even for periodic sensor data, the relative temporal encoding provides valuable positional information that enhances prediction accuracy. The USHCN climate dataset shows comparable RTF benefits, with removal causing approximately 3% degradation. These results collectively demonstrate that RTF encoding serves as a consistent contributor across diverse temporal domains.

H.7 Sensitivity to the temporal kernel

We replace the Gaussian kernel in the full tokenizer with the Laplacian or Cauchy profile defined in Appendix B.1. All three variants use the same attention predictor and the same masked normalization over valid observations. For a given precision γ, they have the same half-maximum radius.

All three kernel variants yield lower mean MSE than interpolation on all four datasets (Table 35). The Gaussian has the lowest mean on USHCN, Activity, and PhysioNet. On MIMIC-III, the Laplacian

Table 35: Temporal kernel comparison with the attention predictor retained. Entries are mean test MSE over five training seeds, in the units indicated. Bold marks the lowest mean in each column.

Front end	USHCN (10^{-1})	Activity (10^{-3})	PhysioNet (10^{-3})	MIMIC-III (10^{-2})
Linear interpolation	8.00	4.53	18.93	7.09
IRAM, Gaussian	**4.98**	**2.52**	**4.82**	1.59
IRAM, Laplacian	6.08	2.78	5.37	1.48
IRAM, Cauchy	5.96	2.77	5.36	**1.47**

Table 36: Complexity of the dominant temporal interaction step.

Method	Core Temporal Operation	Complexity
IRAM (ours)	Gaussian aggregation: L obs $\rightarrow$ H anchor tokens	$O(C \cdot L \cdot H)$
Latent-ODE	ODE integration over irregular timestamps	$O(C \cdot L \cdot d^2 \cdot K)$
mTAND	Time-attention over (L, R) pairs	$O(L \cdot R \cdot d)$
ISTS-PLM	Full self-attention over tokenized sequence	$O(L^2 \cdot d_{\mathrm{LLM}})$

and Cauchy variants have lower means than the Gaussian by 6.9% and 7.5%, respectively. This exception is consistent with a benefit from assigning more weight to distant observations on the sparsest benchmark.

I Computational efficiency details

This appendix supplements Section 5.6 and records the detailed efficiency comparison summarized in the main text.

I.1 Core operation complexity

This subsection supplements Section 5.6. Besides runtime profiling, it is useful to compare the dominant temporal interaction step of each method. Table 36 summarizes the corresponding complexity terms. IRAM aggregates L irregular observations into H anchor tokens with complexity linear in both L and C when H is fixed and small, which helps explain the empirical efficiency advantage of the proposed tokenizer.

I.2 Expanded baseline efficiency comparison

Table 37 expands the efficiency comparison to a broader set of competitive baselines, including graph-based and continuous-time models, across all four datasets.

Across all four datasets, IMTS-Tokenizer consistently achieves the lowest training time. Its memory footprint and parameter count are also substantially lower than those of large PLM-style and graph-based baselines, although some lightweight continuous-time models remain more compact on selected datasets.

Table 37: Expanded computational efficiency comparison. OOM: out of memory on 24GB GPU.

Dataset	Method	Memory (MB)	Time (s)	Params (K)
PhysioNet	MTGNN	2038	1090.39	2940
	StemGNN	OOM	OOM	—
	CRU	2468	931.44	86.5
	Latent-ODE	666	3313.17	127.2
PhysioNet	tPatchGNN	3482	35.1	362
	ISTS-PLM	16088	442.0	1241
	Hi-Patch	23516	32.5	140
	Ours	1706	14.6	32
MIMIC	MTGNN	9697	1548.07	11658.2
	StemGNN	OOM	OOM	—
	CRU	6799	2495.81	97.1
	Latent-ODE	1311	5623.8	135.3
MIMIC	tPatchGNN	14643	57.4	363
	ISTS-PLM	OOM	OOM	1283
	Hi-Patch	21650	387.2	409
	Ours	5576	52.7	58
Activity	MTGNN	645	450.11	1813.8
	StemGNN	OOM	OOM	—
	CRU	440	439.3	16.1
	Latent-ODE	307	1604.61	32.5
	tPatchGNN	399	14.5	390
	ISTS-PLM	3986	63.0	1237
	Hi-Patch	12004	35.4	363
	Ours	694	6.4	320
USHCN	MTGNN	584	1752.81	1592.4
	StemGNN	OOM	OOM	—
	CRU	3635	853.74	15.4
	Latent-ODE	1690	128.63	35.1
	tPatchGNN	2263	77.7	398
	ISTS-PLM	2024	175.4	1201
	Hi-Patch	7376	84.6	400
	Ours	900	29.9	595

J More discussion

J.1 Limitation

The current framework primarily addresses forecasting. We also evaluate its adaptation to classification and interpolation through separate training runs (Appendix G.4). Additionally, while experiments demonstrate efficiency advantages on moderate-scale benchmarks, scalability to datasets with significantly higher channel dimensions or longer observation histories remains to be validated.

Appendix G.3 evaluates one such shift by varying the density and pattern of missing observations on a synthetic trajectory.

J.2 Attention collapse analysis

This subsection provides an auxiliary technical analysis that is not central to the main-text narrative. Under extreme sparsity, we observe attention collapse, namely that attention distributions across anchors converge to highly overlapping observation subsets. We trace this phenomenon to three factors: anchor drift, temporal information loss after weighted aggregation, and softmax amplification that further sharpens already similar attention patterns.

These factors motivate the three stabilizing design choices retained in the main text. VSA reduces anchor drift by initializing anchors near channel-specific observation regions, RTF preserves offset information that would otherwise be washed out during aggregation, and the boundary/diversity regularizers prevent degenerate anchor configurations during training.

J.3 Future work

For future work, we plan to generalize the IMTS-Tokenizer into a truly multi-objective framework by introducing shared and task-specific embedding streams that can be dynamically reweighted according to each temporal reasoning task. We will investigate joint optimization protocols that merge forecasting, interpolation, and classification losses, thereby enabling unified representation learning across diverse irregular-sampling challenges. Furthermore, we aim to integrate cross-modal attention mechanisms to incorporate spatial context and categorical metadata alongside temporal embeddings, enhancing the model's capacity to capture complex dependencies in real-world multimodal sequences.

We also aim to develop a formal theoretical analysis characterizing the conditions under which attention collapse occurs and establishing convergence guarantees under varying sparsity regimes. Moreover, we will explore hierarchical compression techniques to reduce the embedding size for ultra-high-dimensional series, ensuring that IMTS-Tokenizer remains both scalable and efficient when applied to large-scale irregular time series applications.

Addressing Distribution Shifts. Real-world time series often exhibit non-stationary behavior due to evolving data distributions over time. Recent advances in spatio-temporal learning have demonstrated that adaptive normalization techniques can effectively mitigate such distribution shifts [47, 48]. Integrating similar mechanisms into our anchor-based tokenization framework represents a promising direction, where the normalization statistics could be adaptively adjusted based on the temporal context of each observation window.

Model Calibration. An orthogonal direction involves post-hoc calibration of forecasting outputs. Recent work has shown that universal calibrating strategies can substantially improve prediction quality without modifying the underlying model architecture [49]. Exploring such calibration techniques as a complementary enhancement to IMTS-Tokenizer may yield further performance gains with minimal computational overhead.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state that the paper proposes IMTS-Tokenizer for irregular multivariate time-series forecasting, introduces IRAM/RTF/VSA/regularization as the main components, and supports these claims with benchmark, ablation, significance, and efficiency results. The scope is forecasting, and the paper explicitly notes current limitations and future extensions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in the main conclusion and in the supplemental limitation subsection, including the current focus on forecasting, the need to validate scalability on higher-dimensional or longer-history datasets, and possible sensitivity to distribution shift. They are covered in Appendix J.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results such as theorems, lemmas, or proofs; it is primarily a methodological and empirical study.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper discloses the main ingredients needed to reproduce the central claims: the model design and objectives in Section 4, dataset setup in Section 5.1, baseline protocol in Section 5.2, and the implementation summary in Section 5.7. Detailed preprocessing, baseline descriptions, authoritative hyperparameter settings, ablation details, and efficiency statistics are further provided in Appendix C, Appendix D, Appendix E, Appendix H, and Appendix I. In addition, the code package as *Supplementary Material* provides the practical reproduction materials for running the reported experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code for IMTS-Tokenizer is publicly available at `https://github.com/baimao00001/IMTS-Tokenizer`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the benchmark datasets and chronological train/validation/test splits in Section 5.1, the baseline families and rerun protocol in Section 5.2, and the optimizer, early stopping, GPU type, and seed count in Section 5.7. The authoritative dataset-specific hyperparameters are listed in Appendix E, while Appendix C and Appendix D provide the detailed preprocessing and baseline descriptions needed to understand the reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The main result and ablation tables report mean $\pm$ std over five random seeds (Table 3 and Table 2), and Appendix G.1 further provides paired Wilcoxon tests, paired t-tests, bootstrap confidence intervals, and average-rank comparisons against the strongest baselines.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Section 5.7 states that all experiments were conducted on NVIDIA RTX 4090 GPUs, and Appendix I reports per-dataset training memory and training time together with OOM cases on a 24GB GPU. Combined with the authoritative dataset-specific settings in Appendix E, this provides the hardware and runtime information needed to reproduce the reported experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The work is an empirical study on irregular time-series forecasting using existing benchmark datasets and standard evaluation protocols. It does not involve new data collection from participants, intervention on human subjects, or deployment-oriented claims that would conflict with the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The work may have positive societal value by improving forecasting under irregular sampling in domains such as healthcare, human activity analysis, and climate monitoring, where reliable prediction from sparse asynchronous observations is important. At the same time, the paper recognizes that deployment in high-stakes settings requires caution, since erroneous predictions, distribution shifts, or misuse of forecasts could lead to misleading decisions if the model is applied without appropriate validation and oversight.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a high-risk generative model, a scraped dataset, or another asset that would require special safeguards for responsible release.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original datasets and baseline methods, and the supplementary code package includes a README that lists the relevant dataset access links, license requirements, attribution information, and usage notes for the existing assets used in this work. All datasets are used and documented in accordance with their respective licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The submission includes a new code asset for the proposed method as supplementary material, together with accompanying documentation describing reproduction, usage, and relevant implementation details.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The work does not involve crowdsourcing experiments or research with human subjects conducted by the authors.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not report crowdsourcing or human-subject research conducted by the authors, so IRB approval is not applicable in this setting.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of the proposed method or the core experimental methodology in this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.