Revitalizing Canonical Pre-Alignment for Irregular Multivariate Time Series Forecasting

Ziyu Zhou[1], Yiming Huang[1], Yanyun Wang[1], Yuankai Wu[2], James Kwok[3]*, Yuxuan Liang[1]*
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]Sichuan University
[3]The Hong Kong University of Science and Technology
{zzhou651, yhuang033, ywang856}@connect.hkust-gz.edu.cn,
wuyk0@scu.edu.cn, jamesk@cse.ust.hk, yuxliang@outlook.com

Abstract

Irregular multivariate time series (IMTS), characterized by uneven sampling and inter-variate asynchrony, fuel many forecasting applications yet remain challenging to model efficiently. Canonical Pre-Alignment (CPA) has been widely adopted in IMTS modeling by padding zeros at every global timestamp, thereby alleviating inter-variate asynchrony and unifying the series length, but its dense zero-padding inflates the pre-aligned series length, especially when numerous variates are present, causing prohibitive compute overhead. Recent graph-based models with patching strategies sidestep CPA, but their local message passing struggles to capture global inter-variate correlations. Therefore, we posit that CPA should be retained, with the pre-aligned series properly handled by the model, enabling it to outperform state-of-the-art graph-based baselines that sidestep CPA. Technically, we propose **KAFNet**, a compact architecture grounded in CPA for IMTS forecasting that couples (1) a Pre-Convolution module for sequence smoothing and sparsity mitigation, (2) a Temporal **K**ernel **A**ggregation module for learnable compression and modeling of intra-series irregularity, and (3) **F**requency Linear Attention blocks for low-cost inter-series correlation modeling in the frequency domain. Experiments on multiple IMTS datasets show that KAFNet achieves state-of-the-art forecasting performance, with a $7.2\times$ parameter reduction and an $8.4\times$ training–inference acceleration.

Code — https://github.com/zhouziyu02/KAFNet

Introduction

Irregular time series are prevalent in various real-world scenarios, ranging from transportation (Zhang et al. 2024c) to meteorology (Schulz and Stattegger 1997). In modern sensing systems, sensor malfunctions, transmission errors, and cost-driven sampling strategies commonly give rise to irregular multivariate time series (IMTS) (Liu, Cao, and Chen 2024, 2025; Su et al. 2025b), in which observations are (i) unevenly spaced within each variate (intra-series irregularity) and (ii) mutually asynchronous across variates (inter-series asynchrony) (Zhang et al. 2024a,b; Yalavarthi et al. 2024). These twin forms of irregularity greatly complicate the modeling of long-term temporal dependency and inter-variate correlations, as most classical approaches (e.g., RNN) implicitly assume regularly spaced and synchronously aligned series (a.k.a., regular multivariate time series, MTS) (Wang et al. 2024; Shen et al. 2021; Shen and Kwok 2023; Ruan et al. 2025; Liu and Chen 2025; Su et al. 2025a; Fang et al. 2025; Zhou et al. 2024, 2025).

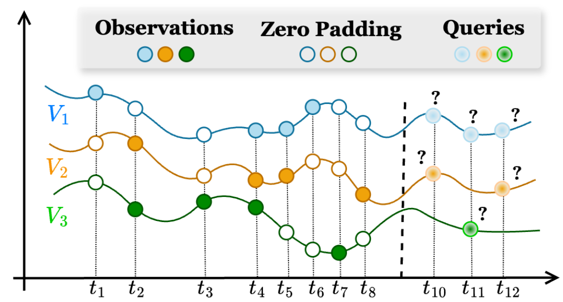


Figure 1: Illustration of Canonical Pre-Alignment (CPA).

In order to alleviate inter-series asynchrony, Canonical Pre-Alignment (CPA) has become a widely adopted preprocessing procedure in IMTS modeling (Che et al. 2018; Rubanova, Chen, and Duvenaud 2019; Shukla and Marlin 2021; Bilos et al. 2021; Tashiro et al. 2021; Zhang et al. 2022, 2023). As illustrated in Fig. 1, CPA aligns all variates onto a shared temporal grid by inserting zero-valued placeholders at every global timestamp. This simple operation provides two crucial advantages: (i) CPA effectively mitigates inter-variate asynchrony by forcing all variates to share a common timeline, thereby enabling the extraction of inter-series correlations under a unified temporal resolution, and (ii) CPA transforms the variate-length univariate series within an IMTS into fixed-size representations, preserving observing sparsity while facilitating batch training and efficient processing by sequence modeling architectures, e.g., RNN (Che et al. 2018) and Transformer (Zhang et al. 2023).

Unfortunately, despite the aforementioned advantages, CPA however suffers from poor efficiency. This inefficiency arises because CPA must identify all global timestamps in

*Corresponding authors.

an IMTS and indiscriminately pad every variate, which inevitably inflates the average sequence length. As shown in Fig. 1, each variate doubles from four to eight observations after alignment, leading to prohibitive computational overhead and memory bottlenecks, particularly when the number of variates is large (Zhang et al. 2024b). Recently, several graph-based approaches have been proposed to bypass the length-explosion issue introduced by CPA. For example, tPatchGNN (Zhang et al. 2024b) chunks an IMTS into fixed-length patches and pads each patch individually, but the rigid patch size distorts local temporal patterns. GraFITi (Yalavarthi et al. 2024) and TimeCHEAT (Liu, Cao, and Chen 2025) treat the series as a bipartite graph; yet their message-passing schemes cannot exchange information between variates that never co-occur in time (Li et al. 2025).

Considering CPA's unparalleled capacity in mitigating inter-variate asynchrony and standardizing sequence length, **we are the first to argue that CPA should not be discarded as recent graph-based detours have done, but should instead be fully exploited, provided that its efficiency issue can be properly resolved.** Therefore, we propose a new model named **KAFNet** for IMTS forecasting, proactively embracing CPA but mitigating its inefficiency.

In KAFNet, a Pre-Convolution layer first processes the pre-aligned series for primary feature extraction by smoothing the length-inflated series and enhancing local temporal patterns. Because the pre-aligned series remains long, we then route it to the Temporal **K**ernel **A**ggregation (TKA) module, where a learnable bank of Gaussian kernels softly partitions the normalized timestamps and pools observations inside each partition, compressing the pre-aligned series while modeling the intra-series irregularity in a channel-independent way. The collection of per-variate representations is fed to stacked **F**requency Linear Attention (FLA) blocks. Each FLA block integrates a frequency-enhanced linear attention mechanism with random fourier feature projection, effectively capturing inter-variate correlations with minimal computational overhead. Finally, the encoded representation is passed through a lightweight MLP-based output layer that enables query-specific forecasts, after which the entire model is trained end-to-end with a standard mean-squared-error objective. To the best of our knowledge, we provide the first study directly addressing the inefficient series-length explosion issue in CPA for IMTS forecasting. Our contributions can be articulated as follows:

- We revitalize Canonical Pre-Alignment for irregular multivariate time series forecasting by showing that, once the efficiency issue is addressed, model leverages CPA can outperform recent dominating graph-based baselines.
- We introduce KAFNet for IMTS forecasting that seamlessly integrates a Pre-Convolution module for sequence smoothing, a Temporal Kernel Aggregation module for temporal compression, and Frequency Linear Attention blocks for inter-variate correlations modeling, yielding an approach that is both compact and expressive.
- Extensive experiments on four public IMTS benchmark datasets show that KAFNet achieves superior predictive accuracy, realizing on average a **7.2**× parameter reduction and a **8.4**× training-inference acceleration compared with SOTA graph-based models (as shown in Fig. 2).

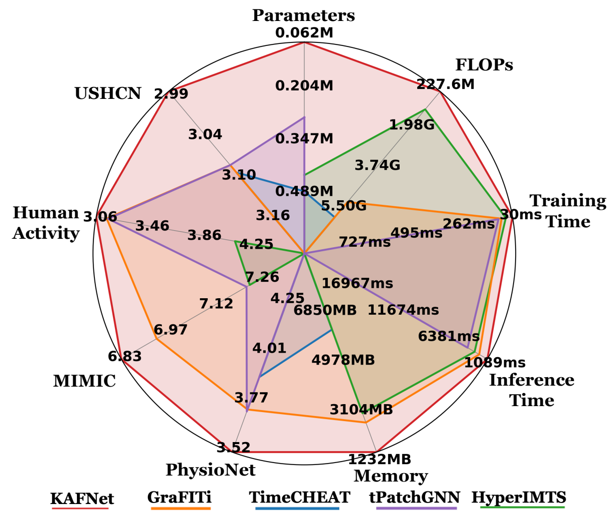


Figure 2: KAFNet delivers superior predictive accuracy (MAE) and efficiency (average) on four IMTS datasets.

Related Works

IMTS Forecasting

Deep learning for IMTS broadly follows two lines. (i) Continuous-time models parameterize dynamics with ODE/SDEs, including Neural ODE/Latent ODE (Chen et al. 2018; Rubanova, Chen, and Duvenaud 2019), CRU (Schirmer et al. 2022), and GRU-ODE (De Brouwer et al. 2019), and variants that reduce solver cost such as Neural Flows (Bilos et al. 2021); ContiFormer (Chen et al. 2023) further couples Neural ODE dynamics with attention. These methods capture irregular trajectories but can be computationally heavy due to numerical solvers. (ii) Relational/patch-based methods avoid dense alignment by operating on patches or graphs, e.g., GraFITi (Yalavarthi et al. 2024), tPatchGNN (Zhang et al. 2024b), Hi-Patch (Luo et al. 2025), and HyperIMTS (Li et al. 2025). While they alleviate length growth, their locality and message-passing often limit global inter-variate correlation modeling under a shared temporal grid. Our approach differs: we retain CPA for a unified timeline, but amortize its cost by learnable temporal compression and linear-time inter-variate modeling in the frequency domain.

CPA for IMTS Analysis

CPA aligns all variates onto a shared grid, filling missing values and recording availability with a mask; it is widely adopted across IMTS tasks, including GRU-D (Che et al. 2018), Latent ODE (Rubanova, Chen, and Duvenaud 2019), mTAND (Shukla and Marlin 2021), Neural Flows (Bilos et al. 2021), CSDI (Tashiro et al. 2021), RainDrop (Zhang et al. 2022), and Warpformer (Zhang et al. 2023). The benefit is fixed-shape inputs and mitigated inter-variate asynchrony;

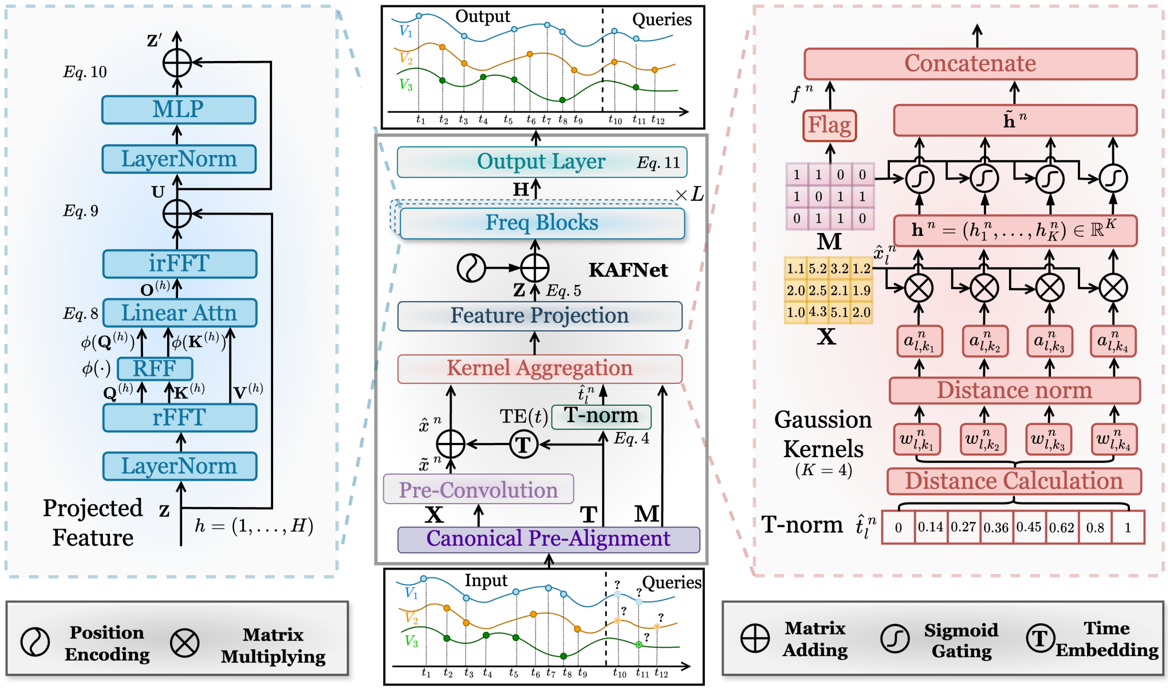


Figure 3: The main architecture of KAFNet. The input IMTS is initially processed by CPA and fed into the Pre-Convolution module ($n \in [1, N]$) for sequence smoothing, then passed through the Temporal Kernel Aggregation module for intra-series irregularity modeling and through the Frequency Linear Attention blocks for the inter-series correlations modeling. Finally, the Output Layer generates the query-specific forecasts. Linear Attn: linear attention mechanism, MLP: multi-layer perceptron.

the drawback is inflated sequence length L, which raises memory and compute, especially with many variates (Zhang et al. 2024b). Recent work mitigates this via patch-based alignment (e.g., tPatchGNN (Zhang et al. 2024b), APN (Liu et al. 2025a)) built upon PatchTST-style slicing (Nie et al. 2023), but per-variate patching can distort local patterns and overlook cross-variate correlations. In contrast, we keep full CPA and address its inefficiency directly through Temporal Kernel Aggregation (for compression) and Frequency Linear Attention (for global inter-variate modeling).

Preliminary

Problem Formulation

Definition (Irregular Multivariate Time Series). An IMTS with N variates is commonly expressed as $\mathcal{O} = \{[(t_i^n, x_i^n)]_{i=1}^{L_n}\}_{n=1}^{N}$. For the n-th variate, $(t_i^n, x_i^n) \in \mathbb{R}^2$ represents the value x_i^n recorded at time t_i^n. Observation counts L_n can vary between variates. The intervals are typically non-uniform, indicating intra-variate irregularity. Additionally, timestamps across different variates do not align, creating inter-variate asynchrony.

Problem (Irregular Multivariate Time Series Forecasting). Let $\mathcal{Q} = \{[q_j^n]_{j=1}^{Q_n}\}_{n=1}^{N}$ denote the collection of future query timestamps, where $q_j^n > \max_i t_i^n$ and Q_n is the number of queries on variate n. The forecasting task targets on learning a model $\mathcal{F}_\theta(\cdot, \cdot)$, parameterized by θ, that maps historical observations $\mathcal{O}$ and queries Q to predictions, $\mathcal{F}_\theta(\mathcal{O}, \mathcal{Q}) \to \hat{\mathcal{X}} = \{[\hat{x}_j^n]_{j=1}^{Q_n}\}_{n=1}^{N}$, where $\hat{x}_j^n$ approximates the predicted future value at time q_j^n.

Canonical Pre-Alignment

Canonical Pre-Alignment is a standard data preprocessing technique to alleviate inter-variate asynchrony (Che et al. 2018; Rubanova, Chen, and Duvenaud 2019; Shukla and Marlin 2021; Bilos et al. 2021; Tashiro et al. 2021; Zhang et al. 2022, 2023). Specifically, IMTS data $\mathcal{O}$ is transformed into a triplet $(\mathcal{T}, \mathcal{X}, \mathcal{M})$. First, $\mathcal{T} = [t_l]_{l=1}^{L} \in \mathbb{R}^L$ is obtained by merging all timestamps and sorting them, i.e., $\mathcal{T} = \bigcup_{n=1}^{N} \{t_i^n\}_{i=1}^{L_n}$. Second, the value matrix $\mathcal{X} = [[x_l^n]_{n=1}^{N}]_{l=1}^{L} \in \mathbb{R}^{L \times N}$ assigns $x_l^n = x_i^n$ if a matching observation exists, otherwise a placeholder such as zero is stored. Finally, the binary mask $\mathcal{M} = [[m_l^n]_{n=1}^{N}]_{l=1}^{L} \in \{0, 1\}^{L \times N}$ with $m_l^n = 1$ when x_l^n is observed and 0 otherwise. This aligned grid preserves the irregular sampling pattern through $\mathcal{M}$ while enabling subsequent models to process fixed-shape inputs. However, this technique may significantly increase the average sequence length, leading to scalability concerns particularly when modeling IMTS with a large number of variates (Zhang et al. 2024b).

Methodology

Our model is composed of four modular components (Fig. 3). (i) **Pre-Convolution** module, which rectifies the uneven information distribution introduced by CPA's zero-padding. (ii) **Temporal Kernel Aggregation** module, which captures temporal irregularities and compresses each lengthy, pre-aligned sequence into a concise representation. (iii) a series of **Frequency Linear Attention** blocks (Freq Blocks), which efficiently models inter-variate dependencies in the frequency domain. (iv) MLP-based **Output Layer**, which addresses arbitrary forecasting queries. The entire architecture is trained end-to-end by minimizing the mean squared error. In the following, we use $\mathbf{T}, \mathbf{X}, \mathbf{M} \in \mathbb{R}^{L \times N}$ to represent the triplet $(\mathcal{T}, \mathcal{X}, \mathcal{M})$. From $(\mathbf{T}, \mathbf{X}, \mathbf{M})$, we have a length-$L$ time series $x^n = (x_1^n, \ldots, x_L^n) \in \mathbb{R}^L$ for each variate $n \in [1, N]$. In the sequel, we will use $l \in [1, L]$ to index the time grid (observation).

Pre-Convolution for Sequence Smoothing

In CPA, zeros are introduced for padding to unify all time steps to a common time axis. However, zero-padding triggers a severe imbalance in information distribution within the pre-aligned series. Therefore, before feeding the input pre-aligned series x^n into complex architecture for deep representation learning, we propose preprocessing the sequence to reduce information sparsity while enhancing its smoothness. Specifically, we pass x^n through two lightweight convolutions acting along the time dimension:

$$\tilde{x}^n = \mathrm{Conv}_{1\times 1}\big(\sigma\big(\mathrm{Conv}_{1\times 3}(x^n)\big)\big) \in \mathbb{R}^L, \tag{1}$$

where σ is `ReLU`. For the same variate n, let $\mathbf{t}^n = (t_1^n, \ldots, t_L^n)$ denote the CPA-generated timestamps in $\mathbf{T}$. Given that the time grid indices generated by CPA no longer reflect true elapsed time, we encode continuous temporal information with a time embedding function defined for a scalar timestamp t:

$$\mathrm{TE}(t) = \big[w_s t + b_s \oplus \sin(\mathbf{w}_p t + \mathbf{b}_p) \oplus \cos(\mathbf{w}_c t + \mathbf{b}_c)\big], \tag{2}$$

which is applied element-wise to $\mathbf{t}^n$, i.e., to each t_l^n. Finally, after a linear projection with a learnable vector $\mathbf{w}_t \in \mathbb{R}^{d_{\mathrm{te}}}$, we obtain the time-aware encoded representation:

$$\hat{x}^n = \tilde{x}^n + \mathbf{w}_t^\top \mathrm{TE}(\mathbf{t}^n) \ \in \mathbb{R}^L, \tag{3}$$

where $\mathbf{w}_t^\top \mathrm{TE}(\mathbf{t}^n)$ denotes the length-L vector whose l-th element is $\mathbf{w}_t^\top \mathrm{TE}(t_l^n)$. This representation is then sent to the subsequent TKA module for explicit temporal-irregularity modeling and sequence compression.

Temporal Kernel Aggregation (TKA)

As analyzed in (Zhang et al. 2024b), CPA inevitably increases the average length of an IMTS, thereby making the modeling of the extended series less efficient. Therefore, we propose a new Temporal Kernel Aggregation (TKA) module to significantly reduce the series length while explicitly encoding the intra-series irregularity.

Specifically, after Pre-Convolution, each variate is represented by $\hat{x}_l^n \in \mathbb{R}^L$. Here, the subscript l denotes the time-grid index (observation). To obtain a fixed-length embedding of the input pre-aligned series while modeling the temporal irregularity, we first map every time index in $\hat{x}_l^n$ to the unit interval by min-max normalization:

$$\hat{t}_l^n = \frac{t_l^n - t_{\min}^n}{t_{\max}^n - t_{\min}^n} \in [0, 1], \tag{4}$$

where $t_{\min}^n$ and $t_{\max}^n$ denote the first and last canonical timestamps for variate n, respectively. Subsequently, on this normalized axis, we place K Gaussian kernels whose centers c_k's are evenly spaced and whose bandwidths $\{\sigma_k = e^{\log \alpha_k}\}$ are learnable, so that together they define a smooth partition of the timeline. Intuitively, these K Gaussian kernels form a soft temporal codebook on the normalized time axis, with each timestamp softly assigned to nearby codewords according to its Gaussian affinity. The closeness of the l-th timestamp $\hat{t}_l^n$ to kernel k is measured by $w_{l,k}^n = \exp\big[-\frac{1}{2}(\hat{t}_l^n - c_k)^2/\sigma_k^2\big]\, m_l^n$, where m_l^n is the binary mask indicating whether x_l^n is observed. Normalization then produces the coefficient: $a_{l,k}^n = \frac{w_{l,k}^n}{\sum_{j=1}^L w_{j,k}^n}$, which quantifies how strongly the l-th observation contributes to the k-th temporal region. Then, we stack $a_{l,k}^n$ over l and k, pooling the time series values $\{\hat{x}_l^n\}$ according to these coefficients through $h_k^n = \sum_{l=1}^L a_{l,k}^n\, \hat{x}_l^n$, forming $\mathbf{h}^n = (h_1^n, \ldots, h_K^n) \in \mathbb{R}^K$. This operation gives one summary per kernel, so $\mathbf{h}^n$ captures the signal present in K smooth temporal windows. A learnable gate vector $\mathbf{g} \in \mathbb{R}^K$ modulates their importance by the element-wise product $\tilde{\mathbf{h}}^n = Sigmoid(\mathbf{g}) \odot \mathbf{h}^n$. Then, the binary flag $f^n = \mathbb{I}(\sum_l m_l^n > 0)$ is concatenated as $\tilde{\mathbf{h}}^n \oplus f^n \in \mathbb{R}^{K+1}$. Finally, the Feature Projection module in Fig. 3 using a linear layer weighted $\mathbf{W}_{\mathrm{proj}} \in \mathbb{R}^{(K+1)\times d}$ with the hidden state dimension of d projects the vector through:

$$\mathbf{z}^n = (\tilde{\mathbf{h}}^n \oplus f^n)\mathbf{W}_{\mathrm{proj}} \in \mathbb{R}^d, \tag{5}$$

yielding a compact embedding whose size is independent of L yet still encodes the original irregular timing through the Gaussian weighting mechanism. Stacking all $\mathbf{z}^n$'s forms $\mathbf{Z} = [\mathbf{z}^1, \ldots, \mathbf{z}^N] \in \mathbb{R}^{N\times d}$, which serves as the input to the subsequent frequency-domain blocks.

Frequency Linear Attention (FLA) Blocks

Transforming the representation into the frequency domain has been proved effective in capturing inter-variate dependencies and global information (Wang et al. 2025; Liu et al. 2025b). Therefore, we propose multi-layer Frequency Linear Attention (FLA) blocks that capture inter-variate correlations by employing a linearized attention mechanism in the frequency domain, maintaining low computational complexity. Let $\mathbf{Z} \in \mathbb{R}^{N\times d}$ denote the input representation where each row corresponds to one variate obtained from the TKA. In each block layer, a LayerNorm first normalizes the input, and a real-valued FFT (rFFT) is applied along the hidden dimension of $\mathbf{Z}$ to convert $\mathbf{Z}$ into frequency coefficients:

$$\mathbf{C} = \mathbf{rFFT}\big(\mathrm{LayerNorm}(\mathbf{Z})\big) \in \mathbb{R}^{N\times 2d_f}, d_f = d/2, \tag{6}$$

where d is the hidden state dimension. Subsequently, FLA leverages multi-head self-attention mechanism to capture inter-variate correlation. Each head $h = (1, \ldots, H)$ computes its query, key, and value matrices by applying learned projections to the shared frequency representation through:

$$\mathbf{Q}^{(h)} = \mathbf{C}\mathbf{W}_h^Q, \mathbf{K}^{(h)} = \mathbf{C}\mathbf{W}_h^K, \mathbf{V}^{(h)} = \mathbf{C}\mathbf{W}_h^V, \tag{7}$$

where $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{2d_f\times d_h}$, and $d_h = 2d_f/H$.

To avoid the quadratic cost of computing $\exp(\mathbf{Q}^\top\mathbf{K})$ in Softmax attention (Hu et al. 2025), we approximate the Softmax kernel using a Random Fourier Feature (RFF) (Rahimi and Recht 2007) map $\phi(\cdot)$, defined as: $\phi(\mathbf{x}) = \frac{1}{\sqrt{R}}\big[\cos(\mathbf{\Omega}^\top\mathbf{x} + \mathbf{b}), \sin(\mathbf{\Omega}^\top\mathbf{x} + \mathbf{b})\big] \in \mathbb{R}^R$, where $\mathbf{\Omega} \in \mathbb{R}^{d_h \times R/2}$, $\mathbf{b} \in \mathbb{R}^{R/2}$ are randomly initialized, yielding a closed-form attention weight computation with linear complexity:

$$\mathbf{O}^{(h)} = \frac{\phi(\mathbf{Q}^{(h)})\left(\phi(\mathbf{K}^{(h)})^\top\mathbf{V}^{(h)}\right)}{\phi(\mathbf{Q}^{(h)})\left(\phi(\mathbf{K}^{(h)})^\top\right)}. \tag{8}$$

Subsequently, all multi-head outputs $\mathbf{O}^{(1)}, \ldots, \mathbf{O}^{(H)}$ are concatenated and transformed back to its original domain via inverse real FFT (irFFT). A residual connection with the input is then applied:

$$\mathbf{U} = \mathbf{Z} + \mathbf{irFFT}\big([\mathbf{O}^{(1)} \ldots \mathbf{O}^{(H)}]\big). \tag{9}$$

To further enhance representational capacity, $\mathbf{U}$ is normalized again and passed through a feed-forward network (two-layer MLP), followed by a second residual connection:

$$\mathbf{Z}' = \mathbf{U} + \mathrm{MLP}(\mathrm{LayerNorm}(\mathbf{U})). \tag{10}$$

The above operations define one Frequency Linear Attention block. By stacking L such blocks, we obtain the final inter-variate representation $\mathbf{Z}^{(L)}$, which is further projected by a learnable matrix $\mathbf{W}_a \in \mathbb{R}^{d\times d}$ to produce the output of the entire frequency attention module: $\mathbf{H} = \mathbf{W}_a\mathbf{Z}^{(L)}$. The proposed FLA blocks enable efficient and expressive modeling of dependencies among variates.

Output Layer and Training Objective

After temporal irregularity has been compressed by TKA and inter-variate correlations have been refined by the L stacked FLA blocks, KAFNet outputs the encoded representation $\mathbf{H} \in \mathbb{R}^{N\times d}$; its n-th row $\mathbf{H}^n$ is a compact summary of variate n. For each query time q_j^n, we concatenate this summary with its time embedding and directly map the result to the scalar prediction via a three-layer MLP:

$$\hat{x}_j^n = \mathrm{MLP}\Big(\mathbf{H}^n \oplus \mathrm{TE}(q_j^n)\Big). \tag{11}$$

On training, we minimize the mean-squared error (MSE) over all variates and their query points:

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{Q_n}\sum_{j=1}^{Q_n}\big(\hat{x}_j^n - x_j^n\big)^2. \tag{12}$$

Optimizing the parameters in Eq. (1) to Eq. (11) thus jointly learns temporal abstraction, inter-variate correlation, and query-aware prediction in an end-to-end manner.

Computational Complexity

Per forward pass, computation proceeds through four stages on a pre-aligned length-L series with N variates: (i) Pre-Conv, (ii) TKA that compresses each variate to d features, (iii) a length-invariant FLA block, and (iv) the output head. The total cost is

$$\begin{aligned}\Omega &= \underbrace{4NLd}_{\text{Pre-Conv}} + \underbrace{N(3LK+Kd)}_{\text{TKA}} \\ &\quad + \underbrace{2Nd\log d + 3Nd^2 + 2NdR}_{\text{FLA}} + \underbrace{QNd^2}_{\text{Output}} \\ &= N\big[(4d+3K)L + Kd + (Q+3)d^2 + 2d(\log d + R)\big].\end{aligned}$$

Only $(4d+3K)L$ depends on L. Thus, Ω scales linearly in both L and N, while post-TKA computation (FLA and output head) is length-invariant due to TKA's compression.

Experiments

Experimental Setup

Datasets. To empirical evaluate the model performance on IMTS forecasting, we employ four datasets from three distinct domains. These include two from healthcare, **PhysioNet** (Silva et al. 2012) with 41 variates and **MIMIC** (Johnson et al. 2016) with 96 variates; one from biomechanics, **Human Activity** with 12 variates; and one from climate science, **USHCN** (Menne, Williams, and Vose 2016) with 5 variates. Adhering to the protocol established in tPatchGNN (Zhang et al. 2024b), we split each dataset into training, validation, and test subsets with a 60%:20%:20% distribution.

Baselines. To facilitate a comprehensive and rigorous comparison, we select a diverse suite of baseline models, which are categorized into four distinct methodological groups. The first group, **MTS Forecasting** [1], comprises regular multivariate time series forecasting models: DLinear (Zeng et al. 2023), TimesNet (Wu et al. 2023), PatchTST (Nie et al. 2023), Crossformer (Wang et al. 2022), GraphWaveNet (Wu et al. 2019), MTGNN (Wu et al. 2020), StemGNN (Cao et al. 2020), CrossGNN (Huang et al. 2023), and FourierGNN (Yi et al. 2023). The second group consists of models designed for **IMTS Classification** [2], including GRU-D (Che et al. 2018), SeFT (Horn et al. 2020), RainDrop (Zhang et al. 2022), and Warpformer (Zhang et al. 2023). For the **IMTS Interpolation** [3] category, we select mTAND (Shukla and Marlin 2021). The final group focuses on **IMTS Forecasting** methods: Latent ODEs (Rubanova, Chen, and Duvenaud 2019), CRU (Schirmer et al. 2022), Neural Flows (Bilos et al. 2021), tPatchGNN (Zhang et al. 2024b), GraFITi (Yalavarthi et al. 2024), TimeCHEAT (Liu, Cao, and Chen 2025), and HyperIMTS (Li et al. 2025).

Implementation Details. All experiments are conducted on a single NVIDIA RTX A6000 GPU. During the training phase, models are optimized using the Adam (Kingma and Ba 2014) optimizer to minimize the loss function as introduced in Eq. (12). The predictive performance is evaluated using two standard metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE).

[1] Concat future-time queries with the encoder output and feed it into an MLP forecasting head.

[2] Replace the classification head with an MLP forecasting head.

[3] Swap interpolation targets for queries to enable extrapolation.

Dataset	PhysioNet		MIMIC		Human Activity		USHCN		Average
Metric	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$\times 10^{-1}$	**Rank**
DLinear (2023)	$41.86 \pm 0.05_{(23)}$	$15.52 \pm 0.03_{(23)}$	$4.90 \pm 0.00_{(23)}$	$16.29 \pm 0.05_{(23)}$	$4.03 \pm 0.01_{(14)}$	$4.21 \pm 0.01_{(15)}$	$6.21 \pm 0.00_{(23)}$	$3.88 \pm 0.02_{(23)}$	20.9
TimesNet (2023)	$16.48 \pm 0.11_{(22)}$	$6.14 \pm 0.03_{(22)}$	$5.88 \pm 0.08_{(22)}$	$13.62 \pm 0.07_{(22)}$	$3.12 \pm 0.01_{(11)}$	$3.56 \pm 0.02_{(11)}$	$5.58 \pm 0.05_{(14)}$	$3.60 \pm 0.04_{(18)}$	17.8
PatchTST (2023)	$12.00 \pm 0.23_{(21)}$	$6.02 \pm 0.14_{(21)}$	$3.78 \pm 0.03_{(21)}$	$12.43 \pm 0.10_{(21)}$	$4.29 \pm 0.14_{(17)}$	$4.80 \pm 0.09_{(18)}$	$5.75 \pm 0.01_{(17)}$	$3.57 \pm 0.02_{(17)}$	19.1
Crossformer (2023)	$6.66 \pm 0.11_{(13)}$	$4.81 \pm 0.11_{(16)}$	$2.65 \pm 0.10_{(17)}$	$9.56 \pm 0.29_{(17)}$	$4.29 \pm 0.20_{(18)}$	$4.89 \pm 0.17_{(19)}$	$5.25 \pm 0.04_{(7)}$	$3.27 \pm 0.09_{(11)}$	14.8
Graph Wavenet (2019)	$6.04 \pm 0.28_{(8)}$	$4.41 \pm 0.11_{(9)}$	$2.93 \pm 0.09_{(19)}$	$10.50 \pm 0.15_{(19)}$	$2.89 \pm 0.03_{(6)}$	$3.40 \pm 0.05_{(6)}$	$5.29 \pm 0.04_{(9)}$	$3.16 \pm 0.09_{(6)}$	10.3
MTGNN (2020)	$6.26 \pm 0.18_{(11)}$	$4.46 \pm 0.07_{(10)}$	$2.71 \pm 0.23_{(18)}$	$9.55 \pm 0.65_{(16)}$	$3.03 \pm 0.03_{(9)}$	$3.53 \pm 0.03_{(9)}$	$5.39 \pm 0.05_{(12)}$	$3.34 \pm 0.02_{(12)}$	12.1
StemGNN (2020)	$6.86 \pm 0.28_{(15)}$	$4.76 \pm 0.19_{(15)}$	$1.73 \pm 0.02_{(7)}$	$7.71 \pm 0.11_{(9)}$	$8.81 \pm 0.37_{(21)}$	$6.90 \pm 0.02_{(21)}$	$5.75 \pm 0.09_{(18)}$	$3.40 \pm 0.09_{(13)}$	14.9
CrossGNN (2023)	$7.22 \pm 0.36_{(17)}$	$4.96 \pm 0.12_{(17)}$	$2.95 \pm 0.16_{(20)}$	$10.82 \pm 0.21_{(20)}$	$3.03 \pm 0.10_{(10)}$	$3.48 \pm 0.08_{(8)}$	$5.66 \pm 0.04_{(16)}$	$3.53 \pm 0.05_{(15)}$	15.4
FourierGNN (2023)	$6.84 \pm 0.35_{(14)}$	$4.65 \pm 0.12_{(13)}$	$2.55 \pm 0.03_{(16)}$	$10.22 \pm 0.08_{(18)}$	$2.99 \pm 0.02_{(8)}$	$3.42 \pm 0.02_{(7)}$	$5.82 \pm 0.06_{(21)}$	$3.62 \pm 0.07_{(20)}$	14.6
GRU-D (2018)	$5.59 \pm 0.09_{(4)}$	$4.08 \pm 0.05_{(6)}$	$1.76 \pm 0.03_{(9)}$	$7.53 \pm 0.09_{(7)}$	$2.94 \pm 0.05_{(7)}$	$3.53 \pm 0.06_{(10)}$	$5.54 \pm 0.38_{(13)}$	$3.40 \pm 0.28_{(14)}$	8.8
SeFT (2020)	$9.22 \pm 0.18_{(19)}$	$5.40 \pm 0.08_{(19)}$	$1.87 \pm 0.01_{(11)}$	$7.84 \pm 0.08_{(11)}$	$12.20 \pm 0.17_{(22)}$	$8.43 \pm 0.07_{(22)}$	$5.80 \pm 0.19_{(20)}$	$3.70 \pm 0.11_{(22)}$	18.3
RainDrop (2021)	$9.82 \pm 0.08_{(20)}$	$5.57 \pm 0.06_{(20)}$	$1.99 \pm 0.03_{(15)}$	$8.27 \pm 0.07_{(15)}$	$14.92 \pm 0.14_{(23)}$	$9.45 \pm 0.05_{(23)}$	$5.78 \pm 0.22_{(19)}$	$3.67 \pm 0.17_{(21)}$	19.5
Warpformer (2023)	$5.94 \pm 0.35_{(6)}$	$4.21 \pm 0.12_{(7)}$	$1.73 \pm 0.04_{(8)}$	$7.58 \pm 0.13_{(8)}$	$2.79 \pm 0.04_{(5)}$	$3.39 \pm 0.03_{(5)}$	$5.25 \pm 0.05_{(8)}$	$3.23 \pm 0.05_{(8)}$	6.9
mTAND (2021)	$6.23 \pm 0.24_{(10)}$	$4.51 \pm 0.17_{(12)}$	$1.85 \pm 0.06_{(10)}$	$7.73 \pm 0.13_{(10)}$	$3.22 \pm 0.07_{(12)}$	$3.81 \pm 0.07_{(12)}$	$5.33 \pm 0.05_{(10)}$	$3.26 \pm 0.10_{(10)}$	10.8
Latent-ODE (2019)	$6.05 \pm 0.57_{(9)}$	$4.23 \pm 0.26_{(8)}$	$1.89 \pm 0.19_{(13)}$	$8.11 \pm 0.52_{(14)}$	$3.34 \pm 0.11_{(13)}$	$3.94 \pm 0.12_{(13)}$	$5.62 \pm 0.03_{(15)}$	$3.60 \pm 0.12_{(19)}$	13.0
CRU (2022)	$8.56 \pm 0.26_{(18)}$	$5.16 \pm 0.09_{(18)}$	$1.97 \pm 0.02_{(14)}$	$7.93 \pm 0.19_{(12)}$	$6.97 \pm 0.78_{(20)}$	$6.30 \pm 0.47_{(20)}$	$6.09 \pm 0.17_{(22)}$	$3.54 \pm 0.18_{(16)}$	17.5
Neural Flow (2021)	$7.20 \pm 0.07_{(16)}$	$4.67 \pm 0.04_{(14)}$	$1.87 \pm 0.05_{(12)}$	$8.03 \pm 0.19_{(13)}$	$4.05 \pm 0.13_{(15)}$	$4.46 \pm 0.09_{(16)}$	$5.35 \pm 0.05_{(11)}$	$3.25 \pm 0.05_{(9)}$	13.3
tPatchGNN (2024)	$\underline{4.98 \pm 0.08}_{(2)}$	$\underline{3.72 \pm 0.03}_{(2)}$	$\underline{1.69 \pm 0.03}_{(2)}$	$7.22 \pm 0.09_{(3)}$	$\underline{2.66 \pm 0.03}_{(2)}$	$3.15 \pm 0.02_{(3)}$	$5.00 \pm 0.04_{(3)}$	$\underline{3.08 \pm 0.04}_{(2)}$	$\underline{2.4}$
tPatchGNN* (2024)	$6.41 \pm 0.07_{(12)}$	$3.89 \pm 0.07_{(5)}$	$1.71 \pm 0.03_{(5)}$	$7.43 \pm 0.11_{(6)}$	$2.76 \pm 0.03_{(4)}$	$3.23 \pm 0.04_{(4)}$	$5.00 \pm 0.05_{(4)}$	$3.09 \pm 0.03_{(3)}$	5.4
GraFITi* (2024)	$6.02 \pm 0.06_{(7)}$	$3.73 \pm 0.03_{(3)}$	$1.71 \pm 0.02_{(4)}$	$\underline{6.94 \pm 0.03}_{(2)}$	$2.73 \pm 0.03_{(3)}$	$\underline{3.14 \pm 0.02}_{(2)}$	5.0S9 $\pm 0.03_{(5)}$	$3.09 \pm 0.04_{(4)}$	3.8
TimeCHEAT* (2025)	$5.05 \pm 0.08_{(3)}$	$3.89 \pm 0.04_{(4)}$	$1.70 \pm 0.01_{(3)}$	$7.40 \pm 0.09_{(5)}$	$4.06 \pm 0.08_{(16)}$	$4.65 \pm 0.04_{(17)}$	$\mathbf{4.42} \pm \mathbf{0.04}_{(1)}$	$3.10 \pm 0.04_{(5)}$	6.8
HyperIMTS* (2025)	$\mathbf{4.59} \pm \mathbf{0.03}_{(1)}$	$4.50 \pm 0.04_{(11)}$	$1.72 \pm 0.03_{(6)}$	$7.23 \pm 0.04_{(4)}$	$4.36 \pm 0.06_{(19)}$	$4.12 \pm 0.08_{(14)}$	$5.21 \pm 0.03_{(6)}$	$3.21 \pm 0.03_{(7)}$	8.5
KAFNet (Ours)	$5.88 \pm 0.01_{(5)}$	$\mathbf{3.52} \pm \mathbf{0.01}_{(1)}$	$\mathbf{1.59} \pm \mathbf{0.02}_{(1)}$	$\mathbf{6.83} \pm \mathbf{0.08}_{(1)}$	$\mathbf{2.54} \pm \mathbf{0.08}_{(1)}$	$\mathbf{3.06} \pm \mathbf{0.07}_{(1)}$	$\underline{4.98 \pm 0.02}_{(2)}$	$\mathbf{2.99} \pm \mathbf{0.01}_{(1)}$	**1.6**

Table 1: We report mean ± standard deviation over five random seeds for MSE and MAE. The best result is in **bold**, and the second-best is underlined. Results from models marked with * are obtained from our own re-implementation under a unified setting for fair comparison, others are collected from (Zhang et al. 2024b). The subscript denotes the rank; when two models achieve the same MSE or MAE, we rank them according to their standard deviations.

Main Results

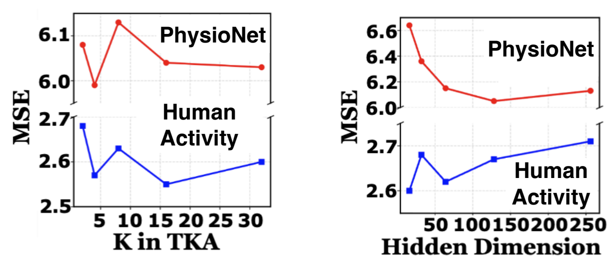
Table 1 summarizes the results. We observe that KAFNet consistently outperforms the state-of-the-art methods. Our improvements pass the Friedman test on dataset-level ranks $\alpha = 0.05$, with Nemenyi post-hoc confirming KAFNet's pairwise superiority over the compared baselines. This superior performance stems from its bold adoption of CPA, which mitigates inter-series asynchrony and unifies temporal resolution across variates. Moreover, KAFNet excels on high-dimensional IMTS, e.g., the MIMIC dataset with 96 variates where pre-aligned sequences can become prohibitively long.

Hyper-parameter Analysis

We focus on two key hyper-parameters in KAFNet: (1) the number of Gaussian kernels in the TKA module; and (2) the hidden state dimension used in Eq. (5), FLA blocks and the final Output Layer. Fig. 4 illustrates how the MSE varies as each of these parameters is changed. Notably, adding more kernels to TKA does not always improve accuracy because too many kernels can overlap excessively along the normalized timeline. Similarly, increasing the hidden state dimension can boost capacity but may incur heavier computation and even degrade performance if over-parameterized.

Ablation Study

Given that KAFNet comprises several modular components for IMTS representation learning including CPA, the Pre-Convolution module, TKA, and FLA blocks, we conduct an ablation study on four public IMTS datasets to assess each component's contribution. As shown in Table 2, removing any single module consistently degrades performance compared to the full KAFNet. Specifically, the removal of CPA leads to a significant decline in forecasting performance, thereby underscoring its essential role in IMTS modeling by alleviating inter-variate asynchrony. Moreover, two architecture-agnostic designs: Pre-Convolution and T-Norm both yield notable improvements and could be adopted in other IMTS models. Furthermore, substituting our FLA with conventional softmax attention (SA) also results in inferior accuracy, underscoring the effectiveness of the proposed FLA in fostering the modeling of inter-variate correlations.



(a) Number of Kernels in TKA. (b) Hidden State Dimension.

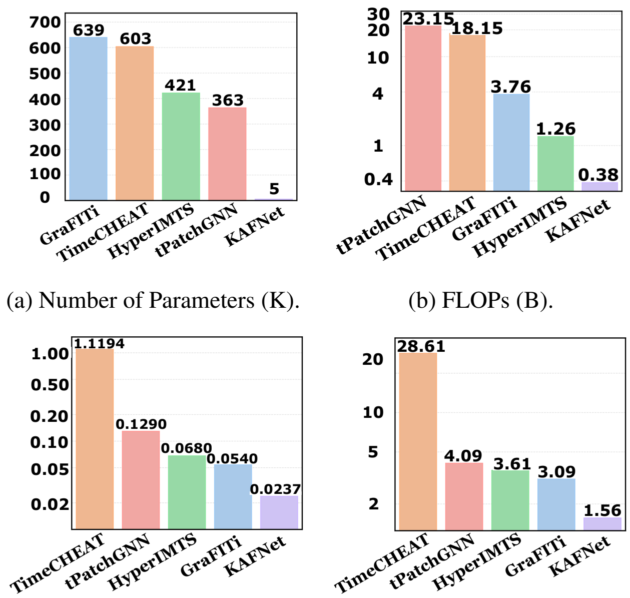
Figure 4: Sensitivity of MSE to (a) the number of Gaussian kernels in TKA and (b) the hidden state dimension in Eq. (5), FLA and the Output Layer, on two IMTS datasets.

Efficiency Analysis

As shown in Fig. 2, KAFNet consistently outperforms strong lightweight baselines in efficiency. We therefore present a detailed analysis on the MIMIC dataset, evaluating efficiency across four dimensions: (i) number of trainable pa-

Dataset	PhysioNet		MIMIC		Human Activity		USHCN	
Metric	MSE$\times10^{-3}$	MAE$\times10^{-2}$	MSE$\times10^{-2}$	MAE$\times10^{-2}$	MSE$\times10^{-3}$	MAE$\times10^{-2}$	MSE$\times10^{-1}$	MAE$\times10^{-1}$
KAFNet	$\mathbf{5.88 \pm 0.01}$	$\mathbf{3.52 \pm 0.01}$	$\mathbf{1.59 \pm 0.02}$	$\mathbf{6.83 \pm 0.08}$	$\mathbf{2.54 \pm 0.08}$	$\mathbf{3.06 \pm 0.07}$	$\mathbf{4.98 \pm 0.02}$	$\mathbf{2.99 \pm 0.01}$
w/o CPA	6.21 ± 0.05	3.88 ± 0.04	1.69 ± 0.02	6.98 ± 0.05	2.70 ± 0.05	3.20 ± 0.03	5.04 ± 0.02	3.10 ± 0.02
w/o Pre-Conv	6.42 ± 0.02	3.70 ± 0.04	1.62 ± 0.02	6.91 ± 0.03	2.66 ± 0.03	3.17 ± 0.04	5.06 ± 0.02	3.05 ± 0.03
w/o T-Norm	6.37 ± 0.02	3.83 ± 0.08	1.73 ± 0.03	7.50 ± 0.10	2.66 ± 0.04	3.18 ± 0.04	5.14 ± 0.05	3.08 ± 0.05
w/o TKA	6.95 ± 0.04	3.98 ± 0.08	1.74 ± 0.03	7.29 ± 0.10	4.21 ± 0.03	4.15 ± 0.04	5.07 ± 0.03	3.14 ± 0.05
w/o FLA	6.26 ± 0.06	3.78 ± 0.08	1.79 ± 0.04	7.75 ± 0.09	2.71 ± 0.08	3.12 ± 0.12	5.23 ± 0.04	3.15 ± 0.06
w/o FLA & w/ SA	6.08 ± 0.02	3.64 ± 0.03	1.67 ± 0.05	7.11 ± 0.02	2.57 ± 0.06	3.09 ± 0.07	5.20 ± 0.06	3.02 ± 0.04

Table 2: Ablation results on four datasets evaluated by MSE and MAE (mean $\pm$ standard). The best results are in **bold**.



(a) Number of Parameters (K).

(b) FLOPs (B).

(c) Training Time (s).

(d) Inference Time (s).

Figure 5: Comparison of the number of parameters (K), FLOPs (B), average training time per batch per epoch (s), and total inference time (s) of KAFNet and four strong baselines for IMTS forecasting. All statistics are collected on the MIMIC dataset with a batch size of 32 to ensure a fair comparison. Lower values indicate higher efficiency.

rameters, (ii) FLOPs, (iii) training time per batch per epoch, and (iv) inference time per batch. As shown in Fig. 5, KAFNet delivers a clear efficiency advantage over graph-based baselines. Panels (a) and (b) reveal that KAFNet attains the best predictive performance with only 5K parameters and 0.38B FLOPs (several orders of magnitude fewer than its counterparts), while panels (c) and (d) confirm that it also achieves the fastest training and inference.

Comparison of FLA and SA

To illustrate FLA's superiority over the vanilla Softmax Attention (SA), Fig. 6 compares their attention maps. The distributions of attention scores produced by FLA and by conventional SA diverge markedly. The FLA map spans almost the entire color scale, indicating that each query variate assigns sharply different weights to different keys. In contrast, the SA map is dominated by values confined to a narrow, low-magnitude band, with only a few isolated pixels exhibiting slightly higher scores. This broader dynamic range allows FLA to selectively amplify or suppress inter-variate dependencies. Moreover, as Table 3 shows, FLA also achieves explicit computational savings compared to SA.

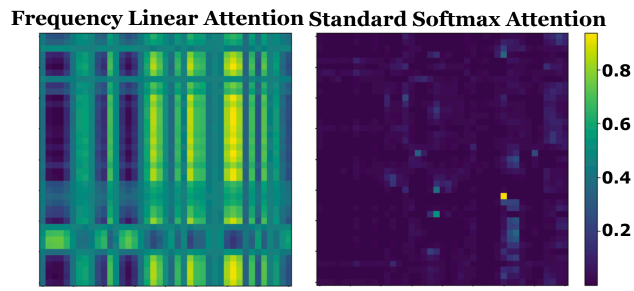


Figure 6: Visualization of attention maps collected from PhysioNet: the left panel shows maps from KAFNet with FLA, while the right panel shows maps from KAFNet in which FLA is replaced by standard Softmax Attention (SA).

	Memory ↓	Parameters ↓	FLOPs ↓	Train. Speed ↓	Infer. Speed ↓
FLA	**890MB**	**118.4K**	**360.5M**	**0.023s**	**1.23s**
SA	1022MB	125.4K	378.2M	0.039s	1.30s

Table 3: Efficiency Analysis on the PhysioNet dataset.

Conclusion and Future Work

This paper revisits Canonical Pre-Alignment (CPA) for irregular multivariate time series (IMTS) forecasting and resolves the long-standing tension between CPA's alignment benefits and its computational overhead. We present **KAFNet**, a compact CPA-based forecasting model that restores the advantages of pre-aligned modeling while remaining highly efficient. Across representative IMTS forecasting benchmarks, KAFNet achieves strong predictive accuracy with fewer parameters and lower training/inference cost than leading graph-based alternatives. However, our study is restricted to forecasting, and the evaluated datasets cover a limited set of domains. Future work will focus on extending the approach to other IMTS-related downstream tasks.

Acknowledgments

This work is mainly supported by the National Natural Science Foundation of China (No. 62402414). This work is also supported by the Guangdong Basic and Applied Basic Research Foundation (No. 2025A1515011994), the National Natural Science Foundation of China (No. 62406206), Guangzhou Municipal Science and Technology Project (No. 2023A03J0011), the Guangzhou Industrial Information and Intelligent Key Laboratory Project (No. 2024A03J0628), a grant from State Key Laboratory of Resources and Environmental Information System, and Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things (No. 2023B1212010007), and in part by the Research Grants Council of the Hong Kong Special Administrative Region (Grant 16200021).

References

Bilos, M.; Sommer, J.; Rangapuram, S. S.; Januschowski, T.; and Günnemann, S. 2021. Neural Flows: Efficient Alternative to Neural ODEs. In *Advances in Neural Information Processing Systems*.

Cao, D.; Wang, Y.; Duan, J.; Zhang, C.; Zhu, X.; Huang, C.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; and Zhang, Q. 2020. Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting. In *Advances in Neural Information Processing Systems*.

Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; and Liu, Y. 2018. Recurrent Neural Networks for Multivariate Time Series with Missing Values. In *Scientific Reports*.

Chen, T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. 2018. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems*.

Chen, Y.; Ren, K.; Wang, Y.; Fang, Y.; Sun, W.; and Li, D. 2023. ContiFormer: Continuous-Time Transformer for Irregular Time Series Modeling. In *Advances in Neural Information Processing Systems*.

De Brouwer, E.; Simm, J.; Arany, A.; and Moreau, Y. 2019. GRU-ODE-Bayes: Continuous Modeling of Sporadically-Observed Time Series. In *Advances in Neural Information Processing Systems*.

Fang, Y.; Miao, H.; Liang, Y.; Deng, L.; Cui, Y.; Zeng, X.; Xia, Y.; Zhao, Y.; Pedersen, T. B.; Jensen, C. S.; Zhou, X.; and Zheng, K. 2025. Unraveling Spatio-Temporal Foundation Models via the Pipeline Lens: A Comprehensive Review. arXiv preprint arXiv:2506.01364.

Horn, M.; Moor, M.; Bock, C.; Rieck, B.; and Borgwardt, K. M. 2020. Set Functions for Time Series. In *International Conference on Machine Learning*.

Hu, J.; Pan, Y.; Du, J.; Lan, D.; Tang, X.; Wen, Q.; Liang, Y.; and Sun, W. 2025. Comba: Improving Bilinear RNNs with Closed-loop Control. In *Advances in Neural Information Processing Systems*.

Huang, Q.; Shen, L.; Zhang, R.; Ding, S.; Wang, B.; Zhou, Z.; and Wang, Y. 2023. CrossGNN: Confronting Noisy Multivariate Time Series via Cross Interaction Refinement. In *Advances in Neural Information Processing Systems*.

Johnson, A. E. W.; Pollard, T. J.; Shen, L.; Lehman, L. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. MIMIC-III: A Freely Accessible Critical Care Database. *Scientific Data*.

Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.

Li, B.; Luo, Y.; Liu, Z.; Zheng, J.; Lv, J.; and Ma, Q. 2025. HyperIMTS: Hypergraph Neural Network for Irregular Multivariate Time Series Forecasting. In *International Conference on Machine Learning*.

Liu, J.; Cao, M.; and Chen, S. 2024. MuSiCNet: A Gradual Coarse-to-Fine Framework for Irregularly Sampled Multivariate Time Series Analysis. arXiv preprint arXiv:2412.01063.

Liu, J.; Cao, M.; and Chen, S. 2025. TimeCHEAT: A Channel Harmony Strategy for Irregularly Sampled Multivariate Time Series Analysis. In *AAAI Conference on Artificial Intelligence*.

Liu, J.; and Chen, S. 2025. TimesURL: Self-Supervised Contrastive Learning for Universal Time Series Representation Learning. In *AAAI Conference on Artificial Intelligence*.

Liu, X.; Qiu, X.; Wu, X.; Li, Z.; Guo, C.; Hu, J.; and Yang, B. 2025a. Rethinking Irregular Time Series Forecasting: A Simple yet Effective Baseline. arXiv preprint arXiv:2505.11250.

Liu, Y.; Zheng, Z.; Cheng, J.; Tsung, F.; Zhao, D.; Rong, Y.; and Li, J. 2025b. CirT: Global Subseasonal-to-Seasonal Forecasting with Geometry-inspired Transformer. In *International Conference on Learning Representations*.

Luo, Y.; Zhang, B.; Liu, Z.; and Ma, Q. 2025. Hi-Patch: Hierarchical Patch GNN for Irregular Multivariate Time Series. In *International Conference on Machine Learning*.

Menne, M. J.; Williams, C. N. J.; and Vose, R. S. 2016. Long-Term Daily and Monthly Climate Records from Stations Across the Contiguous United States (U.S. Historical Climatology Network). *United States Historical Climatology Network Dataset*.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations*.

Rahimi, A.; and Recht, B. 2007. Random features for large-scale kernel machines. In *Advances in neural information processing systems*.

Ruan, W.; Zhong, S.; Wen, H.; and Liang, Y. 2025. Vision-Enhanced Time Series Forecasting via Latent Diffusion Models. arXiv preprint arXiv:2502.14887.

Rubanova, Y.; Chen, T. Q.; and Duvenaud, D. 2019. Latent Ordinary Differential Equations for Irregularly-Sampled Time Series. In *Advances in Neural Information Processing Systems*.

Schirmer, M.; Eltayeb, M.; Lessmann, S.; and Rudolph, M. 2022. Modeling Irregular Time Series with Continuous Recurrent Units. In *International Conference on Machine Learning*.

Schulz, M.; and Stattegger, K. 1997. SPECTRUM: Spectral analysis of unevenly spaced paleoclimatic time series. *Computers & Geosciences*.

Shen, L.; and Kwok, J. 2023. Non-autoregressive conditional diffusion models for time series prediction. In *International Conference on Machine Learning*.

Shen, L.; Yu, Z.; Ma, Q.; and Kwok, J. T. 2021. Time series anomaly detection with multiresolution ensemble decoding. In *AAAI Conference on Artificial Intelligence*.

Shukla, S. N.; and Marlin, B. M. 2021. Multi-Time Attention Networks for Irregularly Sampled Time Series. In *International Conference on Learning Representations*.

Silva, I.; Moody, G.; Scott, D. J.; Celi, L. A.; and Mark, R. G. 2012. Predicting In-Hospital Mortality of ICU Patients: The PhysioNet/Computing in Cardiology Challenge 2012. In *Computing in Cardiology*.

Su, X.; Liu, F.; Chang, Y.; Tanin, E.; Sarvi, M.; and Qi, J. 2025a. Dualcast: A model to disentangle aperiodic events from traffic series. In *International Joint Conference on Artificial Intelligence*.

Su, X.; Sarvi, M.; Liu, F.; Tanin, E.; and Qi, J. 2025b. Generalising Traffic Forecasting to Regions without Traffic Observations. *arXiv preprint arXiv:2508.08947*.

Tashiro, Y.; Song, J.; Song, Y.; and Ermon, S. 2021. CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation. In *Advances in Neural Information Processing Systems*.

Wang, Q.; Xia, Y.; Zhong, S.; Li, W.; Wu, Y.; Cheng, S.; Zhang, J.; Zheng, Y.; and Liang, Y. 2025. AirRadar: Inferring Nationwide Air Quality in China with Deep Neural Networks. In *AAAI Conference on Artificial Intelligence*.

Wang, W.; Yao, L.; Chen, L.; Lin, B.; Cai, D.; He, X.; and Liu, W. 2022. CrossFormer: A Versatile Vision Transformer Hinging on Cross-scale Attention. In *International Conference on Learning Representations*.

Wang, Y.; Du, D.; Hu, H.; Liang, Z.; and Liu, Y. 2024. TSFool: Crafting Highly-Imperceptible Adversarial Time Series through Multi-Objective Attack. arXiv preprint arXiv:2209.06388.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *International Joint Conference on Artificial Intelligence*.

Yalavarthi, V. K.; Madhusudhanan, K.; Scholz, R.; Ahmed, N.; Burchert, J.; Jawed, S.; Born, S.; and Schmidt-Thieme, L. 2024. GraFITi: Graphs for Forecasting Irregularly Sampled Time Series. In *AAAI Conference on Artificial Intelligence*.

Yi, K.; Zhang, Q.; Fan, W.; He, H.; Hu, L.; Wang, P.; An, N.; Cao, L.; and Niu, Z. 2023. FourierGNN: Rethinking Multivariate Time Series Forecasting from a Pure Graph Perspective. In *Advances in Neural Information Processing Systems*.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are Transformers Effective for Time Series Forecasting? In *AAAI Conference on Artificial Intelligence*.

Zhang, J.; Zheng, S.; Cao, W.; Bian, J.; and Li, J. 2023. Warpformer: A Multi-scale Modeling Approach for Irregular Clinical Time Series. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Zhang, W.; Yin, C.; Liu, H.; and Xiong, H. 2024a. Unleash the power of pre-trained language models for irregularly sampled time series. *arXiv preprint arXiv:2408.08328*.

Zhang, W.; Yin, C.; Liu, H.; Zhou, X.; and Xiong, H. 2024b. Irregular Multivariate Time Series Forecasting: A Transformable Patching Graph Neural Networks Approach. In *International Conference on Machine Learning*.

Zhang, W.; Zhang, L.; Han, J.; Liu, H.; Fu, Y.; Zhou, J.; Mei, Y.; and Xiong, H. 2024c. Irregular traffic time series forecasting based on asynchronous spatio-temporal graph convolutional networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Zhang, X.; Zeman, M.; Tsiligkaridis, T.; and Zitnik, M. 2022. Graph-Guided Network for Irregularly Sampled Multivariate Time Series. In *International Conference on Learning Representations*.

Zhou, Z.; Hu, J.; Wen, Q.; Kwok, J. T.; and Liang, Y. 2025. Multi-Order Wavelet Derivative Transform for Deep Time Series Forecasting. arXiv preprint arXiv:2505.11781.

Zhou, Z.; Lyu, G.; Huang, Y.; Wang, Z.; Jia, Z.; and Yang, Z. 2024. SDformer: Transformer with Spectral Filter and Dynamic Attention for Multivariate Time Series Long-term Forecasting. In *International Joint Conference on Artificial Intelligence*.