TiVaT: Joint-Axis Attention for Time Series Forecasting with Lead-Lag Dynamics

Junwoo Ha[1]*, **Hyukjae Kwon**[2]*, **Sungsoo Kim**[2]*, **Kisu Lee**[2]*& **Ha Young Kim**[2]†
[1]Department of AI, [2]Graduate School of Information
Yonsei University
Seoul, Yonsei-ro 50, South Korea
{gkwnsdn0402,kwonhj1015,kss8421,kisu0928,hayoung.kim}@yonsei.ac.kr

Abstract

Multivariate time series (MTS) forecasting plays a crucial role in various real-world applications, yet simultaneously capturing both temporal and inter-variable dependencies remains a challenge. Conventional Channel-Dependent (CD) models handle these dependencies separately, limiting their ability to model complex interactions such as lead-lag dynamics. To address these limitations, we propose TiVaT (Time-Variable Transformer), a novel architecture that integrates temporal and variate dependencies through its Joint-Axis (JA) attention mechanism. TiVaT's ability to capture intricate variate-temporal dependencies, including asynchronous interactions, is further enhanced by the incorporation of Distance-aware Time-Variable (DTV) Sampling, which reduces noise and improves accuracy through a learned 2D map that focuses on key interactions. TiVaT effectively models both temporal and variate dependencies, consistently delivering strong performance across diverse datasets. Notably, it excels in capturing complex patterns within multivariate time series, enabling it to surpass or remain competitive with state-of-the-art methods. This positions TiVaT as a new benchmark in MTS forecasting, particularly in handling datasets characterized by intricate and challenging dependencies.

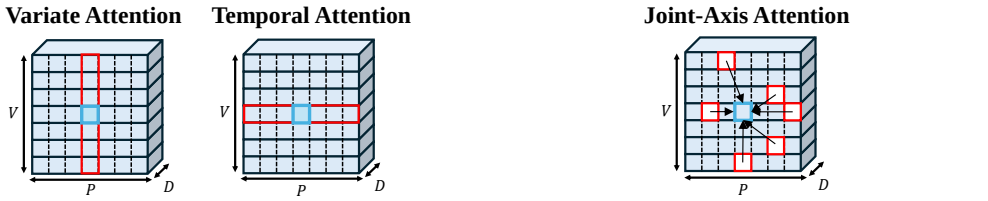
1 Introduction

Multivariate time series (MTS) forecasting a pivotal role in numerous real-world applications, including finance, climate modeling, traffic management, and energy demand forecasting(Lu & Xu, 2024; Nguyen et al., 2023; Jin et al., 2023; Yuan et al., 2023). With advancements of deep learning, models developed based on MLP(Oreshkin et al., 2019; Zeng et al., 2023; Challu et al., 2023; Li et al., 2023), RNN (Lai et al., 2018; Qin et al., 2017; Salinas et al., 2020), CNN (Wu et al., 2022; Luo & Wang, 2024; Wang et al., 2023), and Transformers (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022; Nie et al., 2022) have significantly improved long-term temporal dependency modeling. However, handling both temporal and inter-variable dependencies in MTS remains a challenge.

MTS models are typically classified as either Channel-Independent (CI) or Channel-Dependent (CD) based on how they handle inter-variable relationships. CI models process variables independently, which makes them resilient to noise and overfitting but neglects crucial inter-variable dependencies required for complex datasets. Recent CD models, such as iTransformer (Liu et al., 2023) and CARD (Wang et al., 2024b), use Transformer architectures to model these dependencies, improving predictive accuracy. However, these CD models handle temporal and inter-variable dependencies separately, limiting their ability to capture more complex interactions between variables and temporal dynamics, such as lead-lag relationships.

The main reason CD models struggle to address both temporal and inter-variable dependencies simultaneously is the significant increase in computational cost and model complexity. Joint modeling of all variate-temporal dependencies in high-dimensional datasets becomes computationally

*Equal contibution
†Corresponding author



(a) Conventional CD models (b) TiVaT

Figure 1: In figures, P represents the patched time axis, V refers to the variate axis, and D denotes the dimensional space. (a) illustrates conventional CD models that handle temporal and variate dependencies separately. This separate treatment of the two axes is implicit and may struggle to capture complex lead-lag relationships between variables. In contrast, (b) presents the TiVaT model, which simultaneously considers both temporal and variate dependencies through its JA Attention mechanism. This approach enables TiVaT to better capture complex interactions, including lead-lag relationships between variables.

prohibitive and risks overfitting by capturing irrelevant interactions. The limitations in computational efficiency prevent CD models from fully capturing the complex interdependencies present in real-world scenarios, such as financial markets and climate systems.

To address the limitations of conventional CD models, we introduce TiVaT (Time-Variable Transformer), a novel architecture designed to explicitly capture both temporal and variate dependencies in MTS forecasting. At the core of TiVaT is the Joint-Axis (JA) attention mechanism, which integrates temporal and variate interactions, enabling the model to simultaneously capture complex variate-temporal dependencies, including critical lead-lag dynamics that conventional approaches often miss. As shown in Fig.1a, conventional CD models process temporal and variate dependencies independently, limiting their ability to fully capture intricate interactions across both axes. In contrast, Fig.1b highlights how TiVaT simultaneously considers both temporal and variate dependencies, allowing it to better capture complex, asynchronous interactions across time steps and variables, such as lead-lag relationships.

A key feature of TiVaT is the integration of offsets inspired by deformable attention (Zhu et al., 2020) and Distance-aware Time-Variable (DTV) Sampling. The DTV Sampling technique constructs a learned 2D map to capture both spatial and temporal distances between variables and time steps, dynamically selecting critical variate-temporal points. This not only reduces computational overhead but also mitigates noise, contributing to improved accuracy, allowing TiVaT to scale efficiently to high-dimensional datasets without sacrificing performance. Additionally, TiVaT incorporates time series decomposition to capture long-term trends and cyclical patterns, further enhancing forecasting accuracy and interpretability. The combination of JA attention and DTV Sampling provides a more scalable and robust solution for complex MTS forecasting.

In summary, while TiVaT still faces computational cost challenges, it represents a significant advancement in MTS forecasting as the first model to simultaneously handle both temporal and variate dependencies, setting a new standard in multivariate time series forecasting. The main contributions of this study are as follows:

- TiVaT introduces a JA attention mechanism that addresses temporal and variate dependencies simultaneously, effectively modeling variate-temporal interactions like lead-lag dynamics.
- TiVaT leverages DTV Sampling to reduce noise and enhance accuracy by using a learned 2D map that focuses on key interactions across the temporal and variate axes.
- TiVaT consistently outperforms or remains competitive with state-of-the-art models across a wide range of datasets, particularly demonstrating strong capability in capturing complex patterns. Its robustness and versatility make it highly suited for managing intricate interactions in MTS forecasting.

2 RELATED WORKS

MTS forecasting requires models to capture both temporal dependencies and inter-variable relationships. CI models focus on processing each variable independently, primarily addressing temporal dependencies while ignoring interactions between variables. This independence reduces the risk of overfitting and makes these models more resilient to noise, but it also limits their ability to capture complex cross-variable dependencies, which are critical in multivariate datasets. or example, N-BEATS (Oreshkin et al., 2019) models each variable separately by decomposing time series into trend and seasonal components, while DLinear (Zeng et al., 2023) applies linear transformations independently across variables, optimizing for computational efficiency. Similarly, TimeMixer (Wang et al., 2024a), an MLP-based model, processes temporal and feature dimensions sequentially through independent MLP layers. While TimeMixer enhances scalability by simplifying model complexity, its lack of explicit attention to inter-variable dependencies limits its effectiveness in scenarios where such relationships significantly impact forecasting performance. Although these CI models are computationally efficient, their failure to consider inter-variable dependencies can lead to suboptimal results in datasets where variable interactions play a crucial role.

To address the limitations of CI models, CD models capture inter-variable dependencies, which improves forecasting performance when variable interactions play a critical role. CD models can be broadly divided into two types based on the architectures they employ: non-Transformer-based models and Transformer-based models.

Non-Transformer-based models include those that utilize GNN or CNN to capture inter-variable dependencies. Likewise, MTGNN (Wu et al., 2020) combines GNNs and convolutions to capture variable correlations but faces similar limitations in its ability to integrate temporal dependencies with inter-variable interactions. Additionally, TimesNet (Wu et al., 2022) offers an alternative approach by using a 2D tensor transformation to model both temporal and variate dependencies simultaneously. However, TimesNet relies on tensor operations rather than attention mechanisms, limiting its ability to efficiently handle long-range dependencies across high-dimensional datasets. ModernTCN (Luo & Wang, 2024) leverages a temporal convolutional network (TCN) architecture, using dilated convolutions to capture variable interactions across multiple time steps. While this structure is effective for modeling short- to medium-range dependencies, it treats temporal and inter-variable dependencies separately. This separation limits its ability to model dynamic inter-variable interactions over time, particularly in datasets where long-range or asynchronous interactions are critical.

On the other hand, Transformer-based models leverage the self-attention mechanism, which allows them to capture long-range dependencies across both time steps and variables. For instance, Informer (Zhou et al., 2021) improves the efficiency of self-attention through sparse attention, allowing it to handle large-scale time series data. Fedformer (Zhou et al., 2022) and Autoformer (Wu et al., 2021) further enhance the model's ability to capture long-term temporal patterns by decomposing the time series into trend and seasonal components. However, despite these advancements, most Transformer-based models, including CrossFormer (Zhang & Yan, 2023), iTransformer (Liu et al., 2023), and CARD (Zhao & Shen, 2024), still treat temporal and variate dependencies separately. While these models are effective at modeling inter-variable relationships, their separate treatment of temporal dependencies limits their capacity to fully capture complex interactions like lead-lag dynamics, where variables influence each other asynchronously across time steps.

TiVaT addresses the limitations of both non-Transformer-based and Transformer-based CD models by introducing a JA attention mechanism that simultaneously captures both temporal and variate dependencies. Unlike previous models that treat these dependencies independently, TiVaT's unified attention framework allows it to model intricate variate-temporal interactions, including lead-lag dynamics. Through this integrated approach, TiVaT offers a comprehensive and efficient solution for MTS forecasting, addressing the limitations of existing CI and CD models and significantly improving forecasting performance in complex multivariate datasets.

3 METHODOLOGY

MTS forecasting is a task that predicts future values for each variate leveraging historical data. Formally, given past data $\mathbf{X} = \{\mathbf{x}_{T-L_H+1}, ..., \mathbf{x}_T\} \in \mathbb{R}^{L_H \times V}$ with V variates and L_H time steps for a time point T, the objective is to predict future data $\mathbf{Y} = \{\mathbf{x}_{T+1}, ..., \mathbf{x}_{T+L_F}\} \in \mathbb{R}^{L_F \times V}$

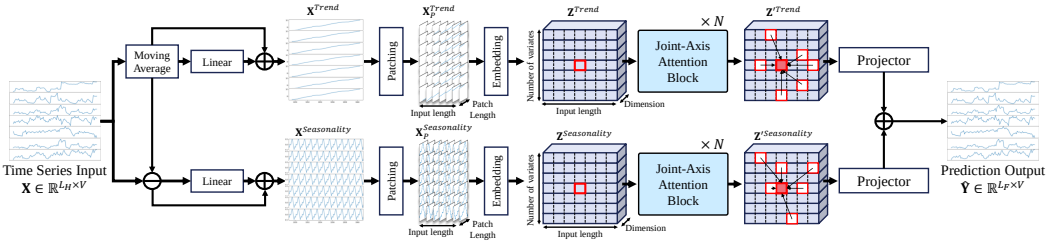


Figure 2: Overview of TiVaT.

for L_F time steps for each variate. MTS forecasting is challenging because it requires capturing complex relationships along both the variate and temporal axes. Especially, previous transformer-based studies (Zhang & Yan, 2023; Wang et al., 2024b) based on the self-attention module focuses either on the relationships between data points at a specific time step t, denoted as $\mathbf{X}_{(t,:)} \in \mathbb{R}^V$, or across time steps for a specific variate v, denoted as $\mathbf{X}_{(:,v)} \in \mathbb{R}^{L_H}$, but not on both simultaneously. The approaches based on self-attention, while effective for certain tasks, fail to address the fact that complex interactions across both different time points and variates often affect each other. For instance, the relationship between a data point $\mathbf{X}_{(t,v)}$ and other points $\mathbf{X}_{(t',v')}$, where $t' \neq t$ and $v' \neq v$, is overlooked. However, in MTS, phenomena like systemic time lags and cross-axis among variates are common, making it essential to consider the effects of other variates at different times. Thus, it becomes crucial to develop a model capable of handling both axes concurrently.

3.1 Structure Overview

Fig. 2 describes overview of our proposed TiVaT. The model is designed to effectively capture complex cross-axiss across both variate and temporal axes simultaneously through joint-axis attention. Time series data can be decomposed into trend and seasonality components, each of which has distinct characteristics that represent long- and short-term patterns, respectively (Cleveland et al., 1990; Wang et al., 2024a). Thus, we first apply the seasonality-trend (ST) decomposition method to the normalized time series input data. This simplifies complex cross-axiss in the raw time series to those in long- and short-term patterns, enabling more effective identification of the relationships.

We decompose the given input sequence for each variate into two components: the moving average, which represents the trend $\mathbf{X}^{Trend} \in \mathbb{R}^{L_H \times V}$, and the remainder, which is treated as seasonality $\mathbf{X}^{Seasonality} \in \mathbb{R}^{L_H \times V}$. To preserve the temporal information and enrich its pattern information, we then pass each component through a separate linear layer $Linear(\cdot)$ with the residual connections, as following Eq:

$$\begin{aligned} \mathbf{X}^{Trend} &= MA(\mathbf{X}) \\ \mathbf{X}^{Seasonality} &= \mathbf{X} - \mathbf{X}^{Trend} \\ \mathbf{X}^{Trend} &= \mathbf{X}^{Trend} + Linear(\mathbf{X}^{Trend}) \\ \mathbf{X}^{Seasonality} &= \mathbf{X}^{Seasonality} + Linear(\mathbf{X}^{Seasonality}), \end{aligned} \tag{1}$$

where MA represents the moving average for the temporal axis for each variate.

The decomposed components $\mathbf{X}^{Trend}$ and $\mathbf{X}^{Seasonality}$ are processed through separate sibling architectures to reduce confusion arising from the difference of long-term and short-term properties. The architectures consist of patching, embedding, JA attention blocks, and projector. We adopt patching and embedding (Nie et al., 2022; Zhang & Yan, 2023; Wang et al., 2024b; Cao et al., 2023) to reduce the computational burden caused by long temporal sequences and to extract local semantic information. When each component is patched with a patch length of L_P and a stride of S along the temporal axis, the input length L_H is reduced to $L_N = \lfloor \frac{L_H - L_P}{S} + 1 \rfloor$ and a new dimension corresponding to the patch length is introduced, resulting in the patched input $X_P \in \mathbb{R}^{L_N \times V \times L_P}$. Subsequently, we feed the patched input to an embedding layer $E : \mathbb{R}^{L_N \times V \times L_P} \rightarrow \mathbb{R}^{L_N \times V \times D}$ to make input tokens $Z \in \mathbb{R}^{L_N \times V \times D}$ for Transformer-based architecture. TiVaT learns the relationships among the input tokens with Transformer encoder architecture, and it outputs the predictions

for trend and seasonality components through a linear layer-based projector $Proj : \mathbb{R}^{L_N \times V \times D} \rightarrow \mathbb{R}^{L_F \times V}$. The final prediction $\hat{\mathbf{Y}} \in \mathbb{R}^{L_F \times V}$ is obtained by aggregating each prediction with an element-wise sum, as follows:

$$\hat{\mathbf{Y}} = Proj(Enc(E(X_P^{Trend}) + PE)) + Proj(Enc(E(X_P^{Seasonality}) + PE)), \tag{2}$$

where Enc and PE present the Transformer encoder blocks and positional encoding, respectively. The projectors for each component include a flattening to extract predictions per variate.

A core component of TiVaT is the proposed JA attention module. The JA attention block is a Transformer encoder block structure, replacing self-attention with the JA attention module. Inspired by the deformable attention module (Zhu et al., 2020), the JA attention module is designed to capture relationships between a feature vector $Z_{(t,v)}$ and other feature vectors $Z_{(t',v')}$, where $t \neq t'$, $v \neq v'$, or both. Unlike the deformable attention, the JA attention module uses offset points as guidelines to minimize information loss and uses the DTV sampling to capture relationships with other points that are highly relevant. This results in overcoming the limitation that existing self-attention-based models can only grasp the relationship along a single axis (temporal $Z_{(:,v)}$ or variate $Z_{(t,:)}$). Furthermore, it is efficient compared to full attention of a data point to entire other points, as it avoids processing less relevant points. The following subsection provides a detailed explanation of the technical implementation of the JA attention module, including how offset points are calculated and how DTV sampling is performed.

3.2 Joint-Axis Attention Module

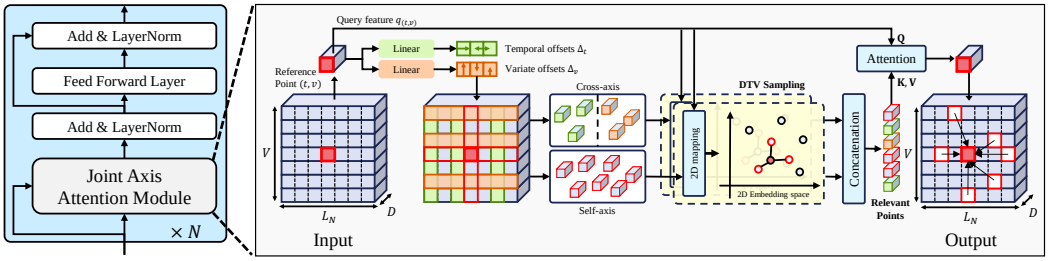


Figure 3: Joint Axis Attention Block.

The deformable attention module was introduced to tackle the inefficiencies of self-attention operations in the computer vision domain. This module extracts offset points based on the query feature $q_{(t,v)}$ at the reference point (t, v) on a feature map, and the attention operation is performed accordingly. Since the offsets are extracted simultaneously along both the width and height axes of the three-dimensional feature map, the module can efficiently consider all axes while also offering computational efficiency compared to self-attention at every location.

However, in time series data, each data point can influence predictive performance due to temporal dependencies. As a result, relying solely on the extracted offset points to reflect the information related to each reference point may lead to information loss. Thus, our JA attention module is designed to prevent information loss in time series data while preserving the deformable attention's property, enabling simultaneous consideration of both axes.

As shown in Fig. 3, an input tensor for JA attention is the patched and embedded time series $Z \in \mathbb{R}^{L_N \times V \times D}$. Here, when we map each dimension of the input to the computer vision domain, the temporal and variable axes may serve as width and height axes in terms of the feature map, respectively. Furthermore, the offset points, which are extracted considering both axes in the deformable attention, can indentify relationships between the reference point and the other data points across both axes. Thus, the JA attention leverages the offset point-based attention to efficiently capture information across temporal and variate axes simultaneously.

Unlike traditional deformable attention, we use the offset points solely as guidelines for selecting candidates for attention operations with each reference point. Relying exclusively on these offset points may not fully capture the complex relationships and interactions with their reference point, as

only a limited number of feature vectors are included in an attention operation without any auxiliary constraints. To mitigate this risk, the JA attention module incorporates feature vectors from the guidelines, ensuring that the attention mechanism considers a broader range of information beyond just the offset points. Furthermore, the DTV sampling in the JA attention module improves efficiency in the attention by selecting more relevant feature vectors in the guideline based on their semantic significance.

In the JA attention module, for each reference point (t, v), the query feature $q_{(t,v)}$ is a D-dimensional feature vector at the corresponding point in the input feature map Z. The temporal and variate offset points, Δ_t and Δ_v, are extracted by passing the query feature through their respective linear layers. Δ_t and Δ_v are initially set to unconstrained real numbers and then normalized into their respective temporal and variate ranges. This process ensures that the offset points cover the entire area of the feature map. After the normalization, we use nearest interpolation to convert the continuous offset points to their discrete counterparts. These offset points serve as guidelines to construct the cross-axis pool, which consist of candidate features for attention operation with the query feature. In other words, if a temporal offset Δ_t is determined for the query feature $q_{(t,v)}$, all variate feature vectors $Z_{(t+\Delta_t,:)}$ at the time of $t + \Delta_t$ are included in the cross-axis pool. Similarly, if a variate offset Δ_v is determined, all temporal feature vectors $Z_{(:,v+\Delta_v)}$ for the variate $v + \Delta_v$ are added to the cross-axis pool. As relevant information differs depending on the number of variates or patch length, we determine the number of Δ_t and Δ_v as hyperparameters, defined as a percentage of the number of elements on each axis.

In addition, using the entire set of feature vectors as key and value features in the attention mechanism can significantly increase computational complexity compared to traditional attention modules, as the attention operation scales quadratically with the number of input elements. To address this, we apply our DTV sampling based on a top K sampling method to the feature vectors. This DTV sampling uses Euclidean distance in the two-dimensional (2D) embedding space to ensure interpretability, selecting features that are closely related to the query. By measuring proximity, it identifies the most relevant points, providing a clear rationale for the selection of meaningful features in the attention mechanism. Moreover, this sampling technique helps reduce noise by excluding less relevant or unrelated feature vectors, which not only improves computational efficiency but also enhances the quality of the attention mechanism by focusing on the most significant features.

Formally, our sampling method first projects the query feature $q_{(t,v)}$ and the other features $Z_{(t',v')}$ from the cross-axis pool associated with the query into a 2D embedding space, resulting in $q^2_{(t,v)}$ and $Z^2_{(t',v')}$, respectively. Subsequently, the indices I_q of the relevant points in the cross-axis pool are determined based on the Euclidean distance, denoted as $Dist$, as in Eq. 3. The relevant feature vectors $R_q \in \mathbb{R}^{K\times D}$, corresponding to the selected indices I_q in the cross-axis pool for their query feature $q_{(t,v)}$, are concatenated and used as key and value features in the attention operation.

$$I_q = \underset{(t',v')}{\arg \operatorname{TopK}}(Dist(q^2_{(t,v)}, Z^2_{(t',v')})). \tag{3}$$

Meanwhile, in MTS analysis, the value of a reference point (t, v) is often considered to be most related to the values of all time steps of the same variate ($Z_{(:,v)}$) and the values of all variates at the same time step ($Z_{(t,:)}$). Thus, we construct an additional pool, we call it the self-axis pool, to incorporate this inductive bias during training our model. The self-axis pool consists of $Z_{(:,v)}$ and $Z_{(t,:)}$, and this pool is used in the same way with the offset points-based pools. As in the offset points-based pools, the DTV sampling is also leveraged for the self-axis pool, and we denote the relevant feature vectors in the self-axis pool as $R^{self}_q \in \mathbb{R}^{K\times D}$.

Here, we use the same K value for both the cross-axis pool and the self-axis pool in the sampling process. This setup introduces an inductive bias that emphasizes the importance of values in the self-axis (temporal and variate relationships within the same reference point) in the MTS domain. By maintaining a consistent proportion of relevant points in the self-axis, we improve the model's ability to effectively incorporate these values into subsequent attention operations. This inductive bias is essential, as it allows the model to better capture inherent temporal dependencies and inter-variate correlations, which are critical for each reference point.

Finally, the relevant feature vectors R_q and R^{self}_q are concatenated and integrated into their query feature $q_{(t,v)}$ using a cross-attention layer. First, the linear projections generate query $\mathbf{Q}$, key $\mathbf{K}$, and

value $\mathbf{V}$, according to the following Equation:

$$\begin{aligned}\mathbf{Q} &= Proj^{q}(q_{(t,v)}) \in \mathbb{R}^{1\times D} \\ \mathbf{K} &= Proj^{k}(Concat(R_q, R_q^{self})) \in \mathbb{R}^{2K\times D} \\ \mathbf{V} &= Proj^{v}(Concat(R_q, R_q^{self})) \in \mathbb{R}^{2K\times D},\end{aligned} \tag{4}$$

where $Concat$ represents a concatentation layer and $Proj^i$ for $i \in \{q, k, v\}$ are the linear projections for query, key, and value, respectively.

Subsequently, the query feature $q_{(t,v)}$ incorporates relevant information from various points using the scaled dot product attention mechanism. In this process, attention weights are computed between the query and the concatenated key-value features. The updated query feature $q'_{(t,v)}$ is then calculated as shown in Eq. 5. Specifically, the attention scores are computed by taking the scaled dot product of the query matrix ($\mathbf{Q}$) and the transpose of the key matrix ($\mathbf{K}^T$), followed by applying these attention weights to the value matrix ($\mathbf{V}$) to update the query feature.

$$q'_{(t,v)} = \text{Softmax}\left(\frac{1}{\sqrt{D}}\mathbf{Q}\cdot\mathbf{K}^T\right)\cdot\mathbf{V} \tag{5}$$

This updated query feature $q'_{(t,v)}$ incorporates information from lagged points $Z_{(t',v')}$, which exist at different time steps ($t' \neq t$) and across different variates ($v' \neq v$). As a result, each query feature is enriched with information from both the temporal and variate axes simultaneously, allowing TiVaT to overcome the limitations of traditional attention mechanisms that can only capture information along one axis at a time.

4 EXPERIMENTS

We conduct a comprehensive set of experiments to assess the performance of TiVaT across a wide range of time series forecasting tasks, illustrating the model's broad applicability. Furthermore, we provide empirical evidence that highlights the efficacy of the proposed JA Attention module, which enhances the model's ability to capture intricate dependencies in multivariate time series data. Additionally, we demonstrate the effectiveness of the DTV Sampling, which improves the model's focus on key variate-temporal interactions.

Dataset Our experimental evaluation leverages six real-world datasets that are widely used in time-series forecasting research, ensuring a rigorous and thorough comparison with state-of-the-art models. These datasets include ECL, ETT (with four subsets), Exchange, Traffic, and Weather, which are used by Autoformer (Wu et al., 2021) for long-term forecasting. These datasets have been extensively benchmarked in prior works, including iTransformer (Liu et al., 2023). Detailed descriptions of each dataset and their characteristics are provided in Appendix A.1.

Baselines We carefully select eight previously successful forecasting models as our benchmarks, categorized into two groups: (1) CD models: iTransformer (Liu et al., 2023), CARD (Wang et al., 2024b), CrossFormer (Zhang & Yan, 2023), ModernTCN (Luo & Wang, 2024), TimesNet (Wu et al., 2022). (2) CI models: PatchTST (Nie et al., 2022), TimeMixer (Wang et al., 2024a), Dlinear (Zeng et al., 2023) .

Unified experiment settings For a fair comparison, we compare our model's performance with the results reported in previous studies. In cases where the input sequence length differs, we conduct experiments using the parameters provided by the respective papers to ensure consistency in performance comparison.

Implementation Details All experiments are conducted using PyTorch (Paszke et al., 2019) on multiple NVIDIA A100 GPUs with 80GB of memory. The model training is performed using L2 loss. Additionally, the number of reference points is determined by Per_{Δ_t}, which selects a percentage of the time axis, and Per_{Δ_v}, which selects a percentage of the variable axis, ensuring both efficiency and optimal performance depending on the dataset.

4.1 MAIN RESULTS

4.1.1 MULTIVARIATE TIME SERIES FORECASTING

Based on the results shown in Table 1, with the best performance highlighted in **red** and the second-best in **undelined**. TiVaT demonstrates its unique strength in simultaneously capturing temporal and variate dependencies, while effectively accounting for lead-lag relationships, making it particularly well-suited for datasets with complex multivariate interactions. On ETTh1, TiVaT achieves an MSE of 0.434, outperforming iTransformer (0.454) and CARD (0.442), underscoring its ability to handle complex seasonal and trend patterns in energy-related datasets. Across the other ETT datasets, TiVaT remains highly competitive, securing a second-best MSE of 0.382 on ETTm1, still surpassing CARD (0.383), further showcasing its robustness in structured time series forecasting.

Beyond the ETT datasets, TiVaT excels on more volatile and dynamic datasets. On Exchange, it achieves the best result with an MSE of 0.349, while on ECL, TiVaT leads with an MSE of 0.166, effectively capturing intricate lead-lag and inter-variable dynamics. Even on the more complex Traffic and Weather datasets, TiVaT remains highly competitive, securing second-best results with MSEs of 0.430 and 0.240, respectively. These results collectively highlight TiVaT's versatility and adaptability across diverse multivariate time series forecasting tasks, making it a highly competitive model for both structured and complex datasets.

Table 1: Results for long-term forecasting. The reported values are averaged over four different prediction horizons: {96, 192, 336, 720}. Lower MSE or MAE values indicate better predictive performance. The input sequence length is fixed at 96 across all experiments. Full detailed results can be found in Appendix A.4.1

Models	**TiVaT (Ours)**		iTransformer (2024)		CARD (2024)		TimeMixer (2024)		ModernTCN (2024)		Crossformer (2023)		PatchTST (2023)		TimesNet (2023)		DLinear (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.434**	0.435	0.454	0.447	0.442	**0.429**	0.447	0.440	**0.435**	**0.431**	0.529	0.522	0.469	0.454	0.458	0.450	0.456	0.452
ETTh2	0.374	0.412	0.383	0.407	0.368	0.390	**0.364**	0.395	**0.354**	**0.388**	0.942	0.684	0.387	0.407	0.414	0.427	0.559	0.515
ETTm1	**0.382**	0.397	0.407	0.410	0.383	0.383	**0.381**	**0.395**	0.386	0.400	0.513	0.496	0.387	0.400	0.400	0.406	0.403	0.407
ETTm2	0.278	0.325	0.288	0.332	**0.271**	**0.316**	**0.275**	**0.323**	0.279	**0.323**	0.757	0.610	0.281	0.326	0.291	0.333	0.350	0.401
Exchange	**0.349**	**0.398**	0.360	**0.403**	0.369	0.409	0.397	0.414	**0.354**	0.419	0.940	0.707	0.367	0.404	0.416	0.443	0.353	0.414
ECL	**0.166**	**0.262**	0.178	0.270	**0.169**	**0.258**	0.182	0.272	0.205	0.287	0.244	0.334	0.205	0.290	0.168	0.272	0.197	0.282
Traffic	**0.430**	0.299	**0.428**	**0.282**	0.453	**0.282**	0.484	**0.297**	0.626	0.378	0.550	0.304	0.481	0.304	0.620	0.336	0.625	0.383
Weather	**0.240**	**0.270**	0.258	0.278	**0.239**	**0.261**	0.240	0.271	**0.240**	0.273	0.259	0.315	0.259	0.281	0.172	0.220	0.196	0.255

4.2 ANALYSIS

4.2.1 ABLATION STUDY ON JA ATTENTION MECHANISM

Ablation on JA Attention. The JA Attention mechanism is designed to effectively capture the unique characteristics of multivariate time series data, where information at specific time points and variables often holds more importance compared to others. By simultaneously addressing temporal and variate dependencies, JA Attention efficiently captures complex variate-temporal interactions that traditional models may miss. As shown in Table 2, we conducted an ablation study comparing JA Attention with two alternative methods: "Full Attention" (using the vanilla transformer model (Vaswani et al., 2017)) and "2-Stage Attention" (using the Crossformer encoder(Zhang & Yan, 2023)). The results clearly demonstrate that JA Attention consistently outperforms both methods across all datasets (ETTh1, ETTm1, ETTm2, Exchange, Weather) in terms of MSE and MAE. By accurately capturing variate-temporal dependencies, JA Attention improves predictive performance and enables more efficient attention allocation, making it a superior solution for time series forecasting.

Ablation on Offset-Points Based Guidelines The results in Table 3 confirm the effectiveness of using offset points as guidelines rather than as direct bases, as in deformable attention mechanisms (Zhu et al., 2020). This approach is tailored for time series data to capture both temporal and variate dependencies while avoiding information loss. The table compares using points only, guidelines without sampling, and guidelines with sampling. Across datasets (ETTh1, ETTh2, ETTm1, ETTm2, Exchange, Weather), using points alone results in suboptimal performance due to information loss. For instance, on ETTh1, the MSE remains at 0.383 with points and guidelines without sampling, but improves to 0.380 with sampling. Similar trends are seen in the Exchange and Weather datasets, where sampling yields the best MSE results (0.083 and 0.156, respectively). These results

Table 2: Ablation Study on JA Attention. The results are presented with a prediction length of $L_F = 96$ and a lookback length of $L_H = 96$. "2-Stage Attention" refers to the method using the Crossformer encoder, "Full Attention" refers to the method using the Transformer encoder, and "JA Attention" refers to our proposed method.

Attention Method	ETTh1		ETTm1		ETTm2		Exchange		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
2-Stage Attention	0.390	0.405	0.325	0.363	0.177	0.259	0.092	0.212	**0.156**	0.202
Full Attention	0.392	0.404	0.326	0.363	0.183	0.265	0.084	0.202	0.161	0.204
JA Attention	**0.380**	**0.399**	**0.314**	**0.354**	**0.173**	**0.259**	**0.083**	**0.201**	**0.156**	**0.201**

Table 3: Ablation Study on Offset. The results are presented with a prediction length of $L_F = 96$ and a lookback length of $L_H = 96$. "Offset as points" refers to using only the relevant points. "Guidelines without sampling" refers to using all relevant points from both the self-axis pool and the cross-axis pool. "Guidelines with sampling" refers to using DTV Sampling points from both the self-axis pool and the cross-axis pool.

offset as	ETTh1		ETTh2		ETTm1		ETTm2		Exchange		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
points	0.383	**0.399**	0.300	0.350	0.322	0.359	0.180	0.265	0.086	0.205	0.159	0.205
guidelines w/o sampling	0.383	0.403	0.297	0.348	0.322	0.361	0.179	0.261	0.085	0.203	0.159	0.205
guidelines w/ sampling	**0.380**	**0.399**	**0.290**	**0.340**	**0.314**	**0.354**	**0.173**	**0.259**	**0.083**	**0.201**	**0.156**	**0.201**

demonstrate that incorporating guidelines with sampling significantly improves the model's ability to capture meaningful relationships, enhancing overall predictive performance.

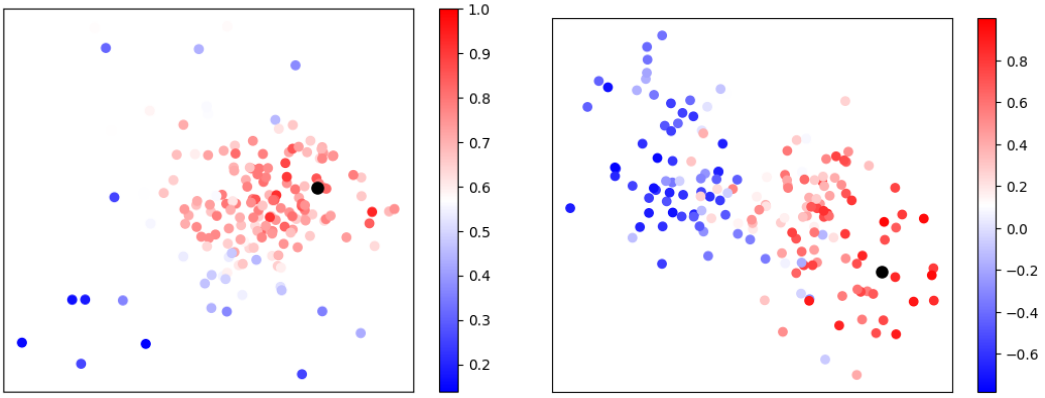
4.2.2 EFFECT OF DISTANCE-AWARE TIME-VARIATE SAMPLING

Effect of Distance-aware Time-Variate Sampling We propose DTV Sampling as a method to reduce noise by selecting features that are semantically related to the reference point. To validate its effectiveness, we compare it with a Random Sampling method, which selects features randomly without considering semantic relevance. Additionally, we perform experiments applying different sampling methods to each pool to further assess the effectiveness of DTV Sampling. The experimental results, as shown in Table 4, indicate that when random sampling is applied to both axes, the overall performance is suboptimal due to the failure to extract features that are semantically related to the reference points. However, when DTV Sampling is applied to a single axis, the performance improves over random sampling, and when applied to both axes, it achieves the best performance. This confirms that DTV Sampling effectively captures features that are semantically aligned with the reference points.

Fig. 4 illustrates a qualitative analysis of the 2D embedding space used for DTV Sampling on the ETTh1 dataset. All features are projected into this 2D space, and cosine similarity is employed to measure the semantic relevance between the reference point and other features. The reference point is represented in black, with points increasingly similar (higher cosine similarity) displayed in red, and less similar points in blue. DTV Sampling selects the top K closest points based on the distance between the reference point and other features in the 2D embedding space. Points with higher cosine similarity cluster near the reference point, while those with lower similarity are farther away. These results further demonstrate that DTV Sampling effectively captures semantically relevant features, leading to improved performance.

Table 4: Ablation Study on DTV Sampling in the Predict-96 Task (ETT, Weather). A check mark ✓indicates that a particular sampling method was used. The last row in the table corresponds to our proposed method.

Self-Axis pool		Cross-Axis pool		ETTh1		ETTh2		ETTm1		ETTm2		Weather	
Random Sampling	DTV Sampling	Random Sampling	DTV Sampling	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
✓		✓		0.388	0.402	0.292	0.342	0.322	0.361	0.177	0.261	**0.156**	**0.201**
✓			✓	0.383	0.400	0.291	0.342	0.325	0.363	**0.173**	**0.258**	**0.156**	**0.201**
	✓	✓		0.385	0.401	0.292	0.342	0.317	0.357	0.178	0.261	0.160	0.205
	✓		✓	**0.380**	**0.399**	**0.290**	**0.340**	**0.314**	**0.354**	**0.173**	0.259	**0.156**	**0.201**



(a) (b)

Figure 4: Qualitative results of DTV Sampling. This is the result of projecting all data points into a 2D embedding space. The color of the points is expressed by calculating cosine similarity. The closer it is to red, the higher the similarity, and the closer it is to blue, the lower the similarity. The black dot is the projections of the reference point into the 2D embedding space. (a) 2D embedding space in Trend. (b) 2D embedding space in Seasonality.

Effect of Separate Sampling We propose a Separate Sampling technique to better capture important information from the self-axis pool, and we validate its effectiveness by comparing three methods: (1) using all features without sampling (w/o sampling), (2) sampling the top 2K features by combining the self-axis and cross-axis pools (w/ common sampling), and (3) our proposed Separate Sampling method (w/ separate sampling). In the w/o sampling method, all features from both pools are used without filtering, while w/ common sampling combines features from both pools and samples the top 2K. The w/ separate sampling approach samples each pool independently. Table 5 shows that w/o sampling performs worse than both sampling methods, confirming the effectiveness of DTV Sampling in extracting relevant information. Additionally, the superior performance of the proposed w/ separate sampling method compared to w/ common sampling highlights the importance of self-axis information. This confirms that separate sampling is better suited to capturing key information from the self-axis, leading to improved model performance.

Table 5: Considering the inherent characteristics of time series data, an experiment was conducted to evaluate the effectiveness of the proposed separate sampling method. This is used for the purpose of further reflecting self-axis information based on TMS's analysis that information on the same time and same variable is more important. The results with prediction length $L_F = 96$ and lookback length $L_H = 96$.

Sampling Method	ETTm1		ETTm2		ETTh1		ETTh2		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o sampling	0.322	0.361	0.179	0.262	0.382	**0.399**	0.292	0.343	0.159	0.205
w/ common sampling	0.315	**0.353**	0.175	**0.259**	0.390	0.402	0.292	**0.342**	0.160	0.205
w/ separate sampling	**0.314**	0.354	**0.174**	**0.259**	**0.380**	**0.399**	**0.290**	**0.340**	**0.156**	**0.201**

5 Conclusion

In this work, we introduced TiVaT, a novel model that addresses the limitations of conventional CD models by simultaneously capturing temporal and variate dependencies in multivariate time series forecasting. Through its JA attention mechanism, TiVaT effectively models complex variate-temporal interactions, including lead-lag dynamics, which are often missed by conventional CD models. The integration of DTV Sampling further enhances the model's performance by reducing noise and improving accuracy, enabling it to focus on key interactions across the temporal and variate axes. Extensive experiments demonstrate that TiVaT consistently outperforms or remains highly competitive with state-of-the-art models across diverse datasets, showcasing its robustness

and versatility. TiVaT sets a new standard in MTS forecasting, making it a promising solution for handling complex, real-world datasets with intricate dependencies.

REFERENCES

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*, 2023.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 6989–6997, 2023.

Robert B Cleveland, William S Cleveland, Jean E McRae, Irma Terpenning, et al. Stl: A seasonal-trend decomposition. *J. off. Stat*, 6(1):3–73, 1990.

Di Jin, Jiayi Shi, Rui Wang, Yawen Li, Yuxiao Huang, and Yu-Bin Yang. Trafformer: unify time and space in traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8114–8122, 2023.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Zhe Li, Zhongwen Rao, Lujia Pan, and Zenglin Xu. Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing. *arXiv preprint arXiv:2302.04501*, 2023.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Minrong Lu and Xuerong Xu. Trnn: An efficient time-series recurrent neural network for stock price prediction. *Information Sciences*, 657:119951, 2024.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *The Twelfth International Conference on Learning Representations*, 2024.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*, 2023.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*, 2017.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

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 *Advances in Neural Information Processing Systems*, volume 30, 2017.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*, 2023.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024a.

Xue Wang, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. Card: Channel aligned robust blend transformer for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024b.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

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 international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Yue Yuan, Zhihua Chen, Zhe Wang, Yifu Sun, and Yixing Chen. Attention mechanism-based transfer learning model for day-ahead energy demand forecasting of shopping mall buildings. *Energy*, 270:126878, 2023.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

Lifan Zhao and Yanyan Shen. Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators. *arXiv preprint arXiv:2401.17548*, 2024.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A APPENDIX

A.1 DATASET

We conduct experiments on five real-world datasets to evaluate the performance of the proposed TiVaT, including:

ETT (Li et al., 2021): This dataset contains seven factors of electricity transformer data from July 2016 to July 2018, with four subsets. ETTh1 and ETTh2 are recorded hourly, while ETTm1 and ETTm2 are recorded every 15 minutes. Exchange (Wu et al., 2021): This dataset collects panel data of daily exchange rates from eight countries, covering the period from 1990 to 2016. Weather (Wu et al., 2021): It includes 21 meteorological factors collected every 10 minutes from the Weather Station of the Max Planck Biogeochemistry Institute in 2020. ECL (Wu et al., 2021): This dataset records hourly electricity consumption data from 321 clients. Traffic (Wu et al., 2021): It collects hourly road occupancy rates measured by 862 sensors on San Francisco Bay area freeways from January 2015 to December 2016. We follow the data processing steps and train-validation-test splitting method outlined in TimesNet (Wu et al., 2023). The datasets for training, validation, and testing are strictly separated in chronological order to prevent any risk of data leakage. For the forecasting configurations, we use a fixed lookback window of 96 time steps across the ETT, Weather, ECL, and Traffic datasets, with prediction horizons set to $\{96, 192, 336, 720\}$.The details of datasets are provided in Table 6.

Table 6: Detailed descriptions of each dataset. 'Dim' represents the number of variables in each dataset. 'Dataset Size' indicates the total number of time points divided into (Train, Validation, Test) splits, respectively. 'Prediction Length' refers to the number of future time steps to forecast, with four prediction scenarios provided for each dataset. 'Frequency' specifies the time interval at which the data points are sampled.

Dataset	**Dim**	**Prediction Length**	**Dataset Size**	**Frequency**	**Information**
ETTh1, ETTh2	7	{96, 192, 336, 720}	(8545, 2881, 2881)	Hourly	Electricity
ETTm1, ETTm2	7	{96, 192, 336, 720}	(34465, 11521, 11521)	15min	Electricity
Exchange	8	{96, 192, 336, 720}	(5120, 665, 1422)	Daily	Economy
Weather	21	{96, 192, 336, 720}	(36792, 5271, 10540)	10min	Weather
ECL	321	{96, 192, 336, 720}	(18317, 2633, 5261)	Hourly	Electricity
Traffic	862	{96, 192, 336, 720}	(12185, 1757, 3509)	Hourly	Transportation
PEMS04	307	{12, 24, 48, 96}	(10172, 3375, 3375)	5min	Transportation
PEMS08	170	{12, 24, 48, 96}	(10690, 3548, 3548)	5min	Transportation

Table 7: Model configurations of TiVaT

	Num blocks	Patch	Stride	Model dim	FFN dim	Learning rate	Per_{Δ_t}	Per_{Δ_v}	Num R_q^{self}	Num R_q
ECL	3	16	8	512	512	0.0005	0.2	0.2	40	40
ETTh1	2	8	4	128	1024	0.0001	0.2	0.2	40	20
ETTh2	2	8	4	128	256	0.0001	0.6	0.4	20	20
ETTm1	2	8	4	128	1024	0.0001	0.6	0.2	40	40
ETTm2	2	8	4	128	256	0.0001	0.2	0.2	20	40
Exchange	2	8	4	128	256	0.0001	0.2	0.2	40	40
Traffic	4	16	8	128	512	0.001	0.1	0.2	40	40
Weather	3	16	8	512	1024	0.0001	0.4	0.8	40	40

A.2 CONFIGURATION

All experiments were implemented using PyTorch (Paszke et al., 2019) and conducted on a multi-GPU setup consisting of NVIDIA A100 GPUs, each with 80GB of memory. For optimization, we employed the ADAM optimizer (Kingma & Ba, 2015) with an initial learning rate selected from the range $\{10^{-3}, 5 \times 10^{-4}, 10^{-4}\}$, and utilized L2 loss as the loss function. The batch size was chosen from $\{4, 16, 32\}$ across all experimental configurations. The number of Joint Axis Attention Blocks in our model was varied, with values selected from $\{2, 3, 4\}$, while the dimension of the series representation was chosen from 128, 256, 512, 1024. Within each Joint Axis Attention Block, the parameter representing the percentage of the temporal axis (Per_{Δ_t}) and variable (Per_{Δ_v}) axis was set between 0.1 and 0.8. Additionally, the number of parameters for both the self-axis(Num R_q^{self})

and the temporal-variable axis(Num R_q) was selected from the set $\{20, 40, 60, 80\}$. Detailed model configuration information is presented in Table 7.

A.3 Time Series Decomposition with Residual Connections

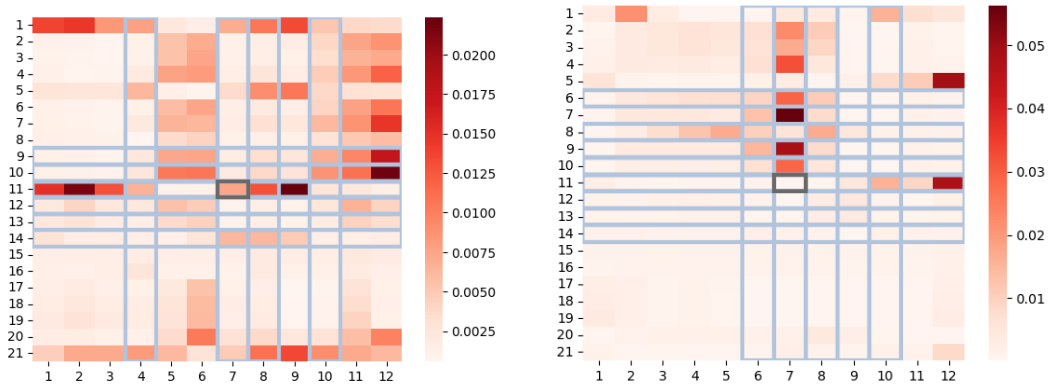
In this part, we highlight the effectiveness of residual connections in time series decomposition. As shown in Table 8, incorporating residual connections after the decomposed components (trend and season) are passed through a linear layer leads to improved performance across all datasets. Specifically, applying residual connections to both trend and seasonal components yields the best results, as indicated by the lowest MSE and MAE values. These results suggest that residual connections enhance the model's ability to capture more intricate temporal patterns by improving the representational capacity of the decomposed components, allowing for more accurate forecasting.

Table 8: Effect of residual connections in timeseries decomposition, The results with prediction length $L_F = 96$ and lookback length $L_H = 96$.

Decomposition Method	Exchange		ETTh1		ETTh2		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o reisual connections	0.088	0.206	0.392	0.406	0.300	0.350	**0.156**	0.203
Residual connections with trend	**0.083**	**0.201**	0.384	0.401	0.291	0.341	0.159	0.204
Residual connections with season	0.084	0.203	0.383	0.400	0.301	0.351	0.160	0.206
Residual connections with trend and season	**0.083**	0.202	**0.380**	**0.399**	**0.290**	**0.340**	**0.156**	**0.201**

A.4 Qualitative results of JA attention's Guidelines.

When performing an attention operation with a specific time, a specific variable, all other times, and all variables, irrelevant information can act as noise. Fig. 5 is a heatmap obtained by performing Attention operations with a specific reference point(dark gray box) and all other times and variables. Since there has been no previous study that applied vanilla self-attention, we experimented with using vanilla self-attention to extract heatmap. If there is a strong correlation with a specific reference point, it is expressed in red, and if there is no correlation, it is expressed in white. As can be seen in Fig. 5, it can be seen that for a specific reference point, there are parts where other variables at different times are not important and can act as noise. Additionally, the Guidelines (light gray box) were extracted through TiVaT's JA attention, and it can be seen that the extracted Guidelines are formed to reflect as much important information as possible.



(a) (b)

Figure 5: The dark gray box represents the corresponding query, the heatmap between this query and the entire data, and the light gray box represents the guideline obtained through TiVaT's JA attention. (a) Heatmap and guideline in Trend. (b) Heatmap and Guidelines in Seasonality.

A.4.1 FULL RESULT

Table 9: Multivariate forecasting results with prediction $L_F \in \{96, 192, 336, 720\}$and fixed look-back length $L_H = 96$ for all baselines. AVG represents the average value of results across the four prediction lengths.

Models		Ours		iTransformer		CARD		TimeMixer		ModernTCN		Crossformer		PatchTST		TimesNet		DLinear	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	**0.136**	**0.232**	0.148	0.240	**0.141**	**0.233**	0.153	0.247	0.183	0.267	0.219	0.314	0.181	0.270	0.168	0.272	0.197	0.282
	192	**0.157**	**0.253**	0.162	0.253	**0.160**	**0.250**	0.166	0.256	0.188	0.272	0.231	0.322	0.188	0.274	0.184	0.289	0.196	0.285
	336	**0.174**	0.271	0.178	**0.269**	**0.173**	**0.263**	0.185	0.277	0.202	0.287	0.246	0.337	0.204	0.293	0.198	0.300	0.209	0.301
	720	**0.197**	**0.292**	0.225	0.317	**0.197**	**0.284**	0.225	0.310	0.245	0.320	0.280	0.363	0.246	0.324	**0.220**	0.320	0.245	0.333
	Avg	**0.166**	**0.262**	0.178	0.270	**0.169**	**0.258**	0.182	0.272	0.205	0.287	0.244	0.334	0.205	0.290	0.192	0.295	0.212	0.300
ETTh1	96	**0.380**	**0.399**	0.386	0.405	0.383	**0.391**	**0.375**	0.400	0.387	0.399	0.423	0.448	0.414	0.419	0.384	0.402	0.386	0.400
	192	**0.425**	0.431	0.441	0.436	0.435	**0.420**	**0.429**	**0.421**	0.442	0.428	0.471	0.474	0.460	0.445	0.436	0.429	0.437	0.432
	336	**0.469**	0.450	0.487	0.458	0.479	**0.442**	0.484	0.458	**0.445**	**0.429**	0.570	0.546	0.501	0.466	0.491	0.469	0.481	0.459
	720	**0.463**	**0.460**	0.503	0.491	**0.471**	**0.461**	0.498	0.482	0.477	0.468	0.653	0.621	0.500	0.488	0.521	0.500	0.519	0.516
	Avg	**0.434**	0.435	0.454	0.447	0.442	**0.429**	0.447	0.440	**0.438**	**0.431**	0.529	0.522	0.469	0.454	0.458	0.450	0.456	0.452
ETTh2	96	0.290	0.340	0.297	0.349	**0.281**	**0.330**	**0.289**	0.341	**0.281**	**0.335**	0.745	0.584	0.302	0.348	0.340	0.374	0.333	0.387
	192	0.373	0.391	0.380	0.400	**0.363**	**0.381**	0.372	0.392	**0.361**	**0.387**	0.877	0.656	0.388	0.400	0.402	0.414	0.477	0.476
	336	0.411	0.426	0.428	0.432	0.411	0.418	**0.386**	**0.414**	**0.350**	**0.388**	1.043	0.731	0.426	0.433	0.452	0.452	0.594	0.541
	720	**0.416**	0.440	0.427	0.445	**0.416**	**0.431**	**0.412**	**0.434**	0.422	0.440	1.104	0.763	0.431	0.446	0.462	0.468	0.831	0.657
	Avg	0.374	0.412	0.383	0.407	0.368	**0.390**	**0.364**	0.395	**0.354**	**0.388**	0.942	0.684	0.387	0.407	0.414	0.427	0.559	0.515
ETTm1	96	**0.314**	**0.354**	0.334	0.368	**0.316**	**0.347**	0.320	0.357	0.322	0.362	0.404	0.426	0.329	0.367	0.338	0.375	0.345	0.372
	192	**0.362**	0.384	0.377	0.391	0.363	**0.370**	**0.361**	**0.381**	0.365	0.387	0.450	0.451	0.367	0.385	0.374	0.387	0.380	0.389
	336	**0.392**	**0.403**	0.426	0.420	**0.392**	**0.390**	**0.390**	0.404	0.398	0.411	0.532	0.515	0.399	0.410	0.410	0.411	0.413	0.413
	720	0.460	0.444	0.491	0.459	0.458	**0.425**	**0.454**	0.441	**0.457**	0.441	0.666	0.589	**0.454**	**0.439**	0.478	0.450	0.474	0.453
	Avg	**0.382**	0.397	0.407	0.410	0.383	**0.383**	**0.381**	**0.395**	0.386	0.400	0.513	0.496	0.387	0.400	0.400	0.406	0.403	0.407
ETTm2	96	**0.173**	0.259	0.180	0.264	**0.169**	**0.248**	0.175	0.258	**0.173**	**0.255**	0.287	0.366	0.175	0.259	0.187	0.267	0.193	0.292
	192	0.241	0.303	0.250	0.309	**0.234**	**0.292**	**0.237**	0.299	0.238	**0.298**	0.414	0.492	0.241	0.302	0.249	0.309	0.284	0.362
	336	0.299	0.341	0.311	0.348	**0.294**	**0.339**	**0.298**	**0.340**	0.312	0.346	0.597	0.542	0.305	0.343	0.321	0.351	0.369	0.427
	720	0.397	0.398	0.412	0.407	**0.390**	**0.388**	**0.391**	0.396	0.394	**0.392**	1.730	1.042	0.402	0.400	0.408	0.403	0.554	0.522
	Avg	0.278	0.325	0.288	0.332	**0.271**	**0.316**	**0.275**	**0.323**	0.279	0.323	0.757	0.610	0.281	0.326	0.291	0.333	0.350	0.401
Exchange	96	**0.083**	**0.201**	0.086	0.206	**0.085**	**0.203**	**0.083**	0.204	0.102	0.227	0.256	0.367	0.088	0.205	0.107	0.234	0.088	0.218
	192	**0.174**	**0.297**	0.177	**0.299**	0.182	0.304	0.182	0.304	0.202	0.321	0.470	0.509	**0.176**	0.299	0.226	0.344	0.176	0.315
	336	0.336	**0.417**	0.331	**0.417**	0.348	0.426	0.361	0.437	0.370	0.450	1.268	0.883	**0.301**	**0.397**	0.367	0.448	**0.313**	0.427
	720	**0.800**	**0.674**	0.847	0.697	0.859	0.701	0.963	0.710	**0.740**	**0.677**	1.767	1.068	0.901	0.714	0.964	0.746	0.839	0.695
	Avg	**0.349**	**0.398**	0.360	**0.403**	0.369	0.409	0.397	0.414	0.354	0.419	0.940	0.707	0.367	0.404	0.416	0.443	**0.353**	0.414
Traffic	96	**0.411**	0.289	**0.395**	**0.268**	0.419	**0.269**	0.462	0.285	0.653	0.392	0.522	0.290	0.462	0.295	0.593	0.321	0.650	0.396
	192	**0.430**	**0.293**	**0.417**	**0.276**	0.443	**0.276**	0.473	0.296	0.599	0.365	0.530	**0.293**	0.466	0.296	0.617	0.336	0.598	0.370
	336	**0.443**	0.298	**0.433**	**0.283**	0.460	**0.283**	0.498	**0.296**	0.608	0.368	0.558	0.305	0.482	0.304	0.629	0.336	0.605	0.373
	720	**0.473**	0.315	**0.467**	**0.302**	0.490	**0.299**	0.506	0.313	0.646	0.387	0.589	0.328	0.514	0.322	0.640	0.350	0.645	0.394
	Avg	**0.439**	0.299	**0.428**	**0.282**	0.453	**0.282**	0.484	**0.297**	0.626	0.378	0.550	0.304	0.481	0.304	0.620	0.336	0.625	0.383
Weather	96	**0.156**	**0.201**	0.174	0.214	**0.150**	**0.188**	0.163	0.209	0.160	0.207	0.158	0.230	0.177	0.218	0.172	0.220	0.196	0.255
	192	**0.203**	**0.247**	0.221	0.254	**0.202**	0.238	0.208	**0.250**	0.206	0.251	0.206	0.277	0.219	0.261	0.240	0.271	0.237	0.296
	336	0.260	0.291	0.278	0.296	0.260	**0.282**	**0.251**	**0.287**	**0.255**	0.292	0.272	0.335	0.278	0.297	0.280	0.306	0.283	0.335
	720	0.341	0.343	0.358	0.347	0.343	0.353	**0.339**	**0.341**	**0.340**	**0.342**	0.398	0.418	0.354	0.348	0.365	0.359	0.345	0.381
	Avg	**0.240**	**0.270**	0.258	0.278	**0.239**	**0.261**	**0.240**	0.271	0.240	0.273	0.259	0.315	0.259	0.281	0.259	0.287	0.265	0.317