


# UniMixer: Unified Patch-Wise and Global Inter-Series Dependency Modeling for Multivariate Time Series Forecasting

Jiaqi Ye<sup>1</sup>, Ciyi Liu<sup>1</sup>, Xinxing Zhou<sup>1</sup>, Rongjie Shen<sup>2</sup>, and Yanlong Wen <sup>1</sup>

<sup>1</sup> College of Computer Science, Nankai University, Tianjin, China  
yjq@mail.nankai.edu.cn  
{liuciyi, zhouxinxing}@dbis.nankai.edu.cn  
wenyl@nankai.edu.cn

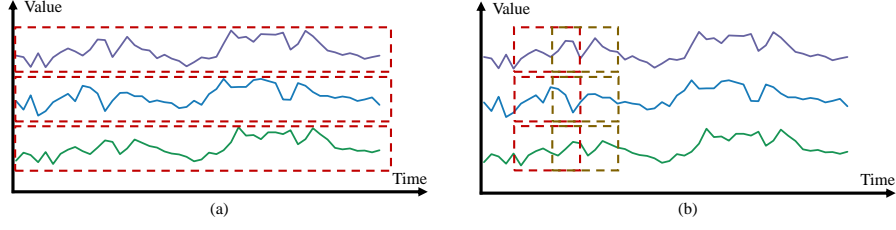
<sup>2</sup> China Railway Design Corporation, Tianjin, China  
752850985@qq.com

**Abstract.** Multivariate time series forecasting is crucial in domains such as finance, energy, and transportation, requiring effective modeling of temporal dynamics and inter-variable dependencies. Existing methods often emphasize either local or global dependency modeling but struggle to seamlessly integrate both and accurately capture complex inter-variable relationships. To address these challenges, we propose UniMixer, a unified framework designed to effectively model dependencies within multivariate time series data. UniMixer integrates three key components: the Patch-Wise Mixer, which captures local temporal patterns; the Global Context Enhancer, which models long-range inter-series relationships; and the Correlation Token Mapper, which explicitly encodes inter-variable correlations. This design achieves a balance between local detail preservation and global dependency understanding, demonstrating strong performance across various forecasting tasks. Extensive experiments conducted on eight real-world multivariate datasets demonstrate that UniMixer achieves superior performance compared to state-of-the-art (SOTA) methods on most datasets, highlighting its effectiveness and adaptability. By unifying local and global dependency modeling, UniMixer establishes a robust foundation for advancing multivariate time series forecasting, while offering insights into inter-variable relationships. Our code is available at: <https://github.com/CYD-y/UniMixer>.

**Keywords:** Multivariate Time Series · Forecasting · Neural Networks.

## 1 Introduction

Multivariate time series forecasting is a critical task with broad applications in domains such as finance [18], energy systems [12], transportation [23], and weather [1]. Accurate predictions can provide invaluable insights, enabling proactive decision-making and efficient resource allocation. Recent advances in deep



**Fig. 1.** A comparison of inverted embedding and patch embedding. (a) Inverted embedding, which maps an entire time series to a token. It makes the original data remain intact without losing information. (b) Patch embedding, which first divides a time series into patches and then maps each patch to a token.

learning have transformed the field, with Transformer-based architectures emerging as a prominent solution due to their capacity to model complex dependencies within sequential data. Despite their potential, significant challenges remain, including the computational inefficiency of traditional self-attention mechanisms [29], the difficulty in capturing local temporal dynamics [19], and the limited scalability to long forecasting horizons.

Recent studies have proposed various strategies to address these challenges. Some [19,27] use patch-based embeddings, dividing the time series into patches to capture local temporal dependencies Fig 1 (b). This improves prediction and reduces complexity but sacrifices global semantics, hindering multivariate relationship modeling. In contrast, others [15] use an inverted architecture, treating individual time series as tokens Fig 1 (a), preserving global context and leveraging Transformer mechanisms to learn inter-variable relationships. However, this excessive field expansion neglects local information, leading to poor performance on some datasets. While these approaches improve performance, they often focus on either local or global dependencies without effectively integrating both.

Furthermore, the explicit and comprehensive modeling of inter-series relationships remains underexplored. Some studies [19] have attempted to use channel-independent strategies to avoid the additional noise introduced by capturing multivariate relationships. However, these strategies fail to effectively leverage the dependencies between variables. Some studies [27] have attempted to use Router mechanisms to capture correlations between patches. While this approach simplifies computation, it also hinders the precise modeling of inter-variable correlations, leading to suboptimal results. Some studies [7] focus on modeling inter-variable correlations at a single time step. This approach can be ineffective when there are significant correlations with slight time lags between variables. Overall, these approaches are limited in scenarios requiring nuanced multivariate interactions.

In this paper, we introduce UniMixer, a novel framework that unifies patch-wise and global inter-series dependency modeling to address the limitations of existing methods. UniMixer incorporates three key components that work syner-

gistically: Patch-Wise Mixer, Global Context Enhancer, and Correlation Token Mapper. Different from methods focus on uncovering multivariable relationships at individual time steps [7], Patch-Wise Mixer models the temporal dependencies and inter-variable relationships at the patch level. This approach mitigates the risk of learning erroneous features caused by the limited scope of individual time steps, enabling more effective modeling of both temporal and inter-variable dependencies. To capture dependencies from a global perspective, the Global Context Enhancer module leverages an inverted data embedding and employs a Transformer architecture to model inter-variable relationships, incorporating them as auxiliary information to support prediction. To provide explicit guidance on the correlations between variables in multivariate time series data, we utilize instantaneous correlation to measure variable dependencies and incorporate it as auxiliary information by concatenating it with the embeddings.

Our contributions are summarized as follows:

1. We propose UniMixer, a unified architecture that integrates patch-wise and global inter-series dependency modeling, effectively addressing the challenges of local and global dependency integration as well as inter-variable relationship representation.
2. We introduce the Correlation Token Mapper, a novel mechanism that explicitly encodes inter-variable correlations, improving the model’s ability to capture meaningful relationships, particularly in datasets with many variables or intricate dependencies, thereby enhancing forecasting accuracy.
3. We conduct extensive experiments on eight public datasets, demonstrating that UniMixer significantly outperforms SOTA methods in terms of forecasting accuracy and robustness. Specifically, it attains top-1 performance in 46 out of 80 cases (57.5%) and top-2 performance in 28 out of 80 cases (35.0%), showcasing its robustness across diverse forecasting horizons and datasets.

By addressing longstanding challenges in the field, UniMixer lays the groundwork for more robust and efficient approaches to multivariate time series forecasting.

## 2 Related Works

Time series forecasting has long been investigated through traditional statistical approaches [3,22], yet these methods exhibit inherent limitations in generalizability and long-term prediction accuracy. The advent of deep learning has enabled significant advancements through recurrent [6,24,21,20,2] and convolutional architectures [6,24,21,20,2], which demonstrate enhanced modeling capacity. Nevertheless, CNN-based models face challenges in capturing long-range temporal dependencies, while RNNs are prone to error accumulation during propagation.

The recent success of Transformer architectures in natural language processing (NLP) [8] and computer vision (CV) [9] has catalyzed innovative adaptations for time series analysis. Representative works include LogTrans [11] with LogSparse self-attention for efficient local feature extraction, Informer [29] employing ProbSparse attention with knowledge distillation for long-sequence pro-

cessing, Autoformer [26] utilizing decomposition strategies for key feature representation, and FEDformer [30] integrating wavelet transforms to balance accuracy and complexity.

In order to solve the problem of excessive computational complexity of Transformer in time series forecasting task, PatchTST [19] introduced patch-based embeddings by segmenting time series into shorter subsequences, achieving state-of-the-art performance while preserving local semantics. Crossformer [27] further advanced this paradigm through unified intra/inter-variable relationship modeling via patch-based self-attention.

Despite these advances, DLinear [16] revealed a counterintuitive result: univariate linear models often outperform complex Transformer-based architectures in long-term forecasting. This finding has reignited interest in linear models, leading to innovative developments such as TiDE [5], TSMixer [4], and HDMixer [7], which achieve superior performance through specialized mixing mechanisms.

In multivariate forecasting, an important issue is whether to consider inter-variable dependencies. While many Transformer-based approaches concatenate cross-variable tokens at each time step [29,14], this practice may distort attention mechanisms and degrade performance [15]. Recent studies have explored alternative strategies: PatchTST [19] and DLinear [5] adopt channel-independent processing, while Crossformer [27] and HDMixer [7] enhance variable interactions through MLP-based mixing layers. The most promising approach emerges from iTransformer [15], which transposes multivariate inputs into unified tokens, effectively capturing cross-variable correlations while addressing receptive field limitations and computational complexity. This architecture achieves superior empirical results across diverse datasets.

### 3 Methodology

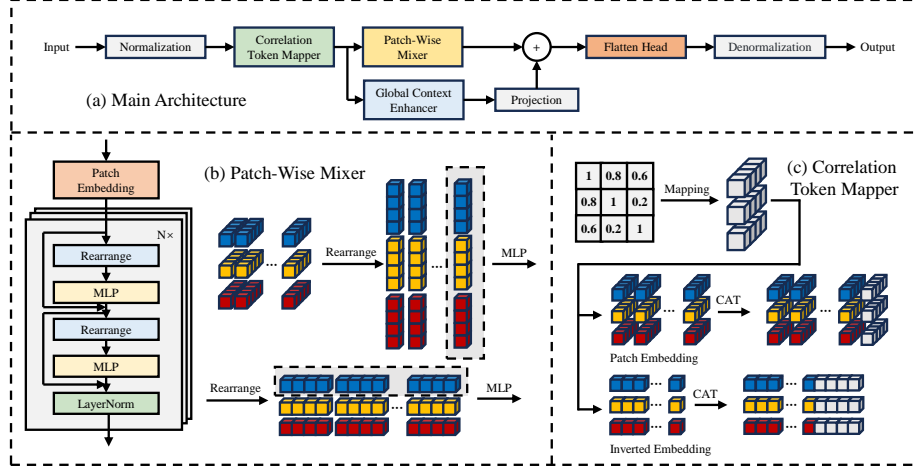
In this section, we will first briefly perform a problem definition and then detail the overall architecture of UniMixer. As shown in the Fig 2 (a), the key components of the model include Patch-Wise Mixer, Global Context Enhancer, and Correlation Token Mapper, which we will explain in detail later.

#### 3.1 Problem Definition

Given a multivariate time series  $X_L = \{x_1, x_2, \dots, x_L\}$ , where each  $x_t \in \mathbb{R}^M$  represents an  $M$ -dimensional vector at time  $t$ , and  $L$  represents the length of lookback window, the objective of multivariate time series forecasting is to predict the future values  $X_T = \{x_{L+1}, x_{L+2}, \dots, x_{L+T}\}$  for a specified horizon  $T$ .

#### 3.2 Patch-Wise Mixer

In the Patch-Wise Mixer (PWM) module, we use a patch-wise MLP-based architecture to capture temporal relationships and variable correlations between individual patches, rather than only capturing these relationships between individual time steps. The specific structure of PWM is shown in the Fig 2 (b).



**Fig. 2.** Our proposed method UniMixer. (a) Main architecture of UniMixer. UniMixer is composed of three key components: Patch-Wise Mixer, Global Context Enhancer, and Correlation Token Mapper. (b) The detailed structure of Patch Wise Mixer. The ‘Rearrange’ in the figure indicates the adjustment of the embedding dimensions, while the dashed boxes represent the dimensions processed by the MLP. (c) The detailed structure of the Correlation Token Mapper. This module first maps the computed instantaneous correlation coefficient matrix and then concatenates it to both the Patch Embedding and the Inverted Embedding.

**Patching and embedding.** Following PatchTST [19], we first divide the timing data into patches. Denote the length of a Patch as  $P$  and the overlap region between two Patches as  $S$ . Then input sequence  $X_L \in \mathbb{R}^{M \times L}$  will be mapped to  $X_P \in \mathbb{R}^{M \times N \times P}$  where  $N$  is the number of patches and  $N = \lfloor \frac{(L-P)}{S} \rfloor + 2$ . We then map it to the new vector space and add the positional encoding information. The formula is as follows:

$$X_{patch} = X_L \cdot W + P \quad (1)$$

where  $P \in \mathbb{R}^{M \times N \times D}$  is position encoding,  $W \in \mathbb{R}^{P \times D}$  denote the Linear that performs the mapping, and  $D$  is the dimension of embedding.

**Temporal and multivariate relationship capture in patches.** After patching and embedding, the input to PWM is represented as a three-dimensional vector  $X_{patch} \in \mathbb{R}^{M \times N \times D}$ . PWM is a linear-based architecture composed of multiple Patch-Wise(PW) block stacks designed to capture relationships between time and variables. We take  $X_{patch}$  as the input to the first PW block. The reason for using Linear instead of Transformer is that in some scenarios, Linear-based models can be more efficient and have better performance [5], whereas our experiments indicated that using Transformer in our architecture leads to suboptimal performance in certain datasets that are more susceptible to overfitting.

To work with patches rather than time steps, at each layer we first adjust the input dimensions to  $\mathbb{R}^{N \times (M \times D)}$  since we are a Linear-based architecture that will operate on one of the dimensions. Without the dimension adjustment, it would operate only on dimension  $M$  or dimension  $D$ , which would result in an extremely limited feeling field that would make it unable to cope with noise problems due to lagged relationships between variables or the absence of significant multivariate relationships at a single time step. Simultaneously, we convert the raw inputs of the blocks into the appropriate dimensions for residual connection. Similarly, in the modelling of the time dimension, we transform its dimensions to  $\mathbb{R}^{M \times (N \times D)}$  so that it can capture relationships in the time dimension in a patch perspective. Subsequently, there is no longer a need to perform a separate change capture for dimensions within a patch, as the initial two instances have already accounted for this. At the end of a block, we convert it back to  $\mathbb{R}^{M \times N \times D}$  dimensions and then perform a LayerNorm. The formalized representation of this block is given as follows:

$$X_{out_1} = \text{GELU}(X_{in} \cdot W_{n_1}) \cdot W_{n_2} + X_{in} \quad (2)$$

$$X_{out_2} = \text{GELU}(X_{out_1} \cdot W_{t_1}) \cdot W_{t_2} + X_{in} \quad (3)$$

$$X_{out} = \text{LayerNorm}(X_{out_2}) \quad (4)$$

where  $X_{in} \in \mathbb{R}^{N \times (M \times D)}$  is the output from the previous block,  $\text{GELU}(\cdot)$  is a nonlinear activation function,  $W_{n_1} \in \mathbb{R}^{(M \times D) \times D_{ff}}$ ,  $W_{n_2} \in \mathbb{R}^{D_{ff} \times (M \times D)}$  is the MLP for capturing relationships between variables,  $W_{t_1} \in \mathbb{R}^{(N \times D) \times D_{ff}}$ ,  $W_{t_2} \in \mathbb{R}^{D_{ff} \times (N \times D)}$  is the MLP for capturing temporal relationships, and  $D_{ff}$  is the hidden dimension of MLP. We use the output  $X_{out}$  of each block as the input  $X_{in}$  for the next block.

### 3.3 Global Context Enhancer

Although the Patch-Wise Mixer performs the capture of inter-multivariate relationships, because it performs the division of patches, it results in the loss of some global information. Therefore, we propose Global Context Enhancer (GCE) to capture the inter-multivariate relationships in the global view to provide additional information for prediction.

**Inverted Data Embedding.** In the GCE module, to better explore multi-variable relationships within a global context, we adopt the approach used by iTransformer [15], embedding the entire sequence as a single token. Such an approach can better preserve global information and avoid the loss of global information caused by dividing patches. The formula is as follows:

$$X_{emb} = X_L \cdot W \quad (5)$$

where  $W \in \mathbb{R}^{L \times (K \times D)}$  denote the Linear that performs the mapping, and  $K$  is a hyperparameter used to adjust the embedding dimension. After mapping, our embedding data becomes  $X_{emb} \in \mathbb{R}^{M \times (K \times D)}$ .

**Global variable relationship capture.** Subsequently, in order to model the relationship between the multivariables, we input embedding into a vanilla Transformer to obtain the multivariate relationship:

$$\begin{aligned} X_Q &= X_{emb} \cdot W_1, \quad X_K = X_{emb} \cdot W_2, \quad X_V = X_{emb} \cdot W_3 \\ W_1, W_2, W_3 &\in \mathbb{R}^{(K \times D) \times (K \times D)} \end{aligned} \quad (6)$$

$$AttnScore = \text{softmax}\left(\frac{X_Q X_K^T}{\sqrt{d_k}}\right) X_V \quad (7)$$

$$X_{rel} = \text{LayerNorm}(X_{emb} + AttnScore) \quad (8)$$

Like PW block, we also stack Transformer blocks, using the output of each block as input to the next. The reason for using a vanilla Transformer instead of Linear is that Transformers do have a more significant advantage in capturing complex multivariate relationships, and we tried replacing them with Linear, but experienced performance degradation in some datasets. After obtaining the multivariable relationships  $X_{rel}$  from a global perspective, we map them to the dimensions corresponding to the PWM output  $\mathbb{R}^{M \times N \times D}$  using a linear layer. This serves as auxiliary information, which is subsequently concatenated to enhance the output.

### 3.4 Correlation Token Mapper

Although the first two modules capture the relationships between variables, their inputs lack explicit guidance on variable correlations. This absence may lead to overconfidence in the relationships between weakly correlated variables, resulting in biased representations and, consequently, suboptimal performance. To explicitly guide the modeling of inter-variable correlations, we propose the Correlation Token Mapper (CTM) module, as illustrated in Fig 2 (c).

**Variable correlation calculation.** LIFT [28] proposed the use of cross-correlation coefficients to evaluate the correlation between paired variables. However, in LIFT, these coefficients are primarily utilized to compute lagged relationships between variables. Directly using cross-correlation coefficients to assess variable correlations may similarly overestimate the relationships between certain pairs of variables. To address this issue, we employ instantaneous correlation coefficients to evaluate inter-variable correlations more accurately. Following LIFT [28], we also use the Fast Fourier Transform to simplify the calculations:

$$S = \text{FFT}(X_L) \cdot \text{conj}(\text{FFT}(X_L)) \quad (9)$$

$$R_{ij} = \frac{1}{L} \cdot \text{IFFT}(S)[0] \quad (10)$$

$$R = [R_{ij}]_{M \times M} \quad (11)$$

where  $\text{FFT}(\cdot)$  is Fast Fourier Transform,  $\text{conj}$  means the conjugate of the computed Fourier Transform,  $\text{IFFT}(\cdot)$  denotes the inverse Fourier transform,  $L$  is

the length of the input sequence. We multiply by  $\frac{1}{L}$  for normalisation, take the zeroth of the result to get the instantaneous correlation coefficient, and then we concatenate all the  $R_{ij}$  to get the final relationship matrix  $R \in \mathbb{R}^{M \times M}$ . Each element  $R_{ij}$  represents the correlation between variable  $i$  and variable  $j$ .

**Mapping and Concatenation** To introduce explicit inter-variable correlation information and without affecting the original data representation, we map the relationship matrix  $R$  through a Linear and concatenate it into the embedding data:

$$Corr = R \cdot W \quad (12)$$

where  $W \in \mathbb{R}^{M \times D}$  represents a Linear for mapping, and  $Corr \in \mathbb{R}^{M \times D}$  stands for the correlation of each of the  $M$  variables with the other variables.

For PWM, we concatenate it directly as a token to the input embedding:

$$X_{patch} = \text{CAT}(X_{patch}, Corr) \quad (13)$$

where  $\text{CAT}(\cdot)$  denotes the concatenation of dimension  $N$ , making the output become  $X_{patch} \in \mathbb{R}^{M \times (N+1) \times D}$ , and the component dimensions in the PWM are changed accordingly. Before predicting, we discard this additional dimension to bring it back to  $\mathbb{R}^{M \times N \times D}$ .

For GCE, we concatenate it on dimension  $D$ , since using it as a token would result in a concatenation on dimension  $M$  that overpowers the original input data:

$$X_{emb} = \text{CAT}(X_{emb}, Corr) \quad (14)$$

The output after concatenation becomes  $X_{emb} \in \mathbb{R}^{M \times ((K+1) \times D)}$ , and the component dimensions in the GCE are changed accordingly.

### 3.5 Forecast.

After the computation of the above components, we concatenate  $X_{out}$  and  $X_{ref}$  and get the predictions through a Flatten MLP:

$$\hat{Y} = \text{GELU}(\text{FLAT}(\text{CAT}(X_{out}, X_{ref}) \cdot W_1) \cdot W_2) \cdot W_3 \quad (15)$$

where  $W_1 \in \mathbb{R}^{2D \times D}$ ,  $W_2 \in \mathbb{R}^{D \times D_{ff}}$ ,  $W_3 \in \mathbb{R}^{D_{ff} \times T}$  are the corresponding Linear, and  $\text{FLAT}(\cdot)$  represents a flattening operation that transforms the dimension from  $\mathbb{R}^{M \times N \times D}$  to  $\mathbb{R}^{M \times (N \times D)}$ .

## 4 Experiments

In this section, we first specify the various settings of the experiment and then illustrate the results by comparing our method to 11 SOTA baselines. Furthermore, we demonstrate the effectiveness of the individual components of our model through comprehensive ablation experiments.



#### 4.1 Experimental Settings

**Datasets** We conducted extensive experiments on eight widely used real-world multivariate time series forecasting datasets, including ETT (4 subsets), Weather, Electricity, Exchange used by Autoformer [26] and Solar-Energy datasets proposed in LSTNet [10]. Details of the datasets are given in Table 1. For a fair comparison, we follow the same data processing combined training-validation-testing set partitioning protocol as in TimesNet [25], with a 6:2:2 partitioning ratio on the ETT dataset and a 7:1:2 partitioning ratio on the other datasets.

**Table 1.** The statistic of the eight datasets used in our experiments.

Dataset	ETTh1&2	ETTM1&2	Weather	Electricity	Solar-Energy	Exchange
Dim	7	7	21	321	137	8
Timesteps	17420	69680	52696	26304	52560	7588
Frequency	Hourly	15min	10min	Hourly	10min	Daily

**Metrics** Following TimesNet [25], we use MSE and MAE as our evaluation metrics. For a fair comparison, we follow the same evaluation protocol, setting the look-back window for all models to  $L = 96$  and the prediction horizons to  $T \in \{96, 192, 336, 720\}$ .

**Baselines** We carefully choose 11 well-acknowledged forecasting models as our benchmark, including (1) Transformer-based methods: Autoformer [26], FEDformer [30], Stationary [17], Crossformer [27], PatchTST [19], iTransformer [15]; (2) Linear-based methods: DLinear [16], TiDE [5], HDMixer [7]; and (3) TCN-based methods: SCINet [13], TimesNet [25].

#### 4.2 Experiments Results

The comprehensive forecasts are presented in the Table 2. A lower MSE/MAE indicates a higher level of accuracy in the predictions. The experimental results demonstrate that UniMixer exhibits exceptional predictive performance across most forecasting tasks. It attains top-1 performance in 46 out of 80 cases (57.5%) and top-2 performance in 28 out of 80 cases (35.0%), showcasing its robustness across diverse forecasting horizons and datasets.

**Experiments on ETTh1&ETTM1** On ETTh1, UniMixer shows limited advantages, excelling only at specific prediction lengths. FEDformer and PatchTST dominate in MSE, while HDMixer performs better in MAE. This likely stems from ETTh1’s smaller size, where UniMixer’s cross-variable modeling may cause

**Table 2.** Multivariate long-term forecasting result comparison. We use prediction lengths  $T \in \{96, 192, 336, 720\}$ , and input length  $L = 96$ . The best results are in **bold** and the second bests are underlined.

Method		UniMixer HDMxier iTransformer PatchTST Crossformer TiDE TimesNet DLinear SCINet FEDformer Stationary Autoformer												
Dataset	PL	Metric	(Ours)	(2024)	(2024)	(2023)	(2023)	(2023)	(2023)	(2023)	(2022)	(2022)	(2022)	(2021)
ETTh1	96	MSE	0.394	0.387	0.386	0.377	0.423	0.479	0.384	0.386	0.654	0.376	0.513	0.449
		MAE	0.411	0.396	0.405	0.397	0.448	0.464	0.402	0.400	0.599	0.419	0.491	0.459
	192	MSE	0.437	0.440	0.441	0.426	0.471	0.525	0.436	0.437	0.719	0.420	0.534	0.500
		MAE	0.435	0.427	0.436	0.432	0.474	0.492	0.429	0.432	0.631	0.448	0.504	0.482
	336	MSE	0.481	0.481	0.487	0.469	0.570	0.565	0.491	0.481	0.778	0.459	0.588	0.521
		MAE	0.453	0.446	0.458	0.457	0.546	0.515	0.469	0.459	0.659	0.465	0.535	0.496
	720	MSE	0.484	0.484	0.503	0.519	0.653	0.594	0.521	0.519	0.836	0.506	0.643	0.514
		MAE	0.469	0.471	0.491	0.504	0.621	0.558	0.500	0.516	0.699	0.507	0.616	0.512
	Avg	MSE	0.449	0.448	0.454	0.448	0.529	0.541	0.458	0.456	0.747	0.440	0.570	0.496
		MAE	0.442	0.435	0.448	0.448	0.522	0.507	0.450	0.452	0.647	0.460	0.537	0.487
ETTh2	96	MSE	0.297	0.289	0.297	0.309	0.745	0.400	0.340	0.333	0.707	0.358	0.476	0.346
		MAE	0.346	0.338	0.349	0.359	0.584	0.440	0.374	0.387	0.621	0.397	0.458	0.388
	192	MSE	0.371	0.377	0.380	0.381	0.877	0.528	0.402	0.477	0.860	0.429	0.512	0.456
		MAE	0.394	0.392	0.400	0.407	0.656	0.509	0.414	0.476	0.689	0.439	0.493	0.452
	336	MSE	0.416	0.418	0.428	0.426	1.043	0.643	0.452	0.594	1.000	0.496	0.552	0.482
		MAE	0.428	0.428	0.432	0.433	0.731	0.571	0.452	0.541	0.744	0.487	0.551	0.486
	720	MSE	0.425	0.425	0.427	0.436	1.104	0.874	0.462	0.831	1.249	0.463	0.562	0.515
		MAE	0.443	0.442	0.445	0.456	0.763	0.679	0.468	0.657	0.838	0.474	0.560	0.511
	Avg	MSE	0.377	0.377	0.383	0.388	0.942	0.611	0.414	0.559	0.954	0.437	0.526	0.450
		MAE	0.403	0.400	0.407	0.414	0.684	0.550	0.427	0.515	0.723	0.449	0.516	0.459
ETTm1	96	MSE	0.320	0.334	0.334	0.324	0.404	0.364	0.338	0.345	0.418	0.379	0.386	0.505
		MAE	0.358	0.364	0.368	0.365	0.426	0.387	0.375	0.372	0.438	0.419	0.398	0.475
	192	MSE	0.366	0.374	0.377	0.372	0.450	0.398	0.374	0.380	0.439	0.426	0.459	0.553
		MAE	0.383	0.382	0.391	0.392	0.451	0.404	0.387	0.389	0.450	0.441	0.444	0.496
	336	MSE	0.398	0.405	0.426	0.398	0.532	0.428	0.410	0.413	0.490	0.445	0.495	0.621
		MAE	0.406	0.402	0.420	0.409	0.515	0.425	0.411	0.413	0.485	0.459	0.464	0.537
	720	MSE	0.467	0.468	0.491	0.457	0.666	0.487	0.478	0.474	0.595	0.543	0.585	0.671
		MAE	0.444	0.437	0.459	0.444	0.589	0.461	0.450	0.453	0.550	0.490	0.516	0.561
	Avg	MSE	0.388	0.395	0.407	0.388	0.513	0.419	0.400	0.403	0.485	0.448	0.481	0.588
		MAE	0.398	0.396	0.410	0.403	0.496	0.419	0.406	0.407	0.481	0.452	0.456	0.517
ETTm2	96	MSE	0.171	0.180	0.180	0.185	0.287	0.207	0.187	0.193	0.286	0.203	0.192	0.255
		MAE	0.255	0.262	0.264	0.268	0.366	0.305	0.267	0.292	0.377	0.287	0.274	0.339
	192	MSE	0.237	0.245	0.250	0.246	0.414	0.290	0.249	0.284	0.399	0.269	0.280	0.281
		MAE	0.299	0.304	0.309	0.305	0.492	0.364	0.309	0.362	0.445	0.328	0.339	0.340
	336	MSE	0.300	0.303	0.311	0.311	0.597	0.377	0.321	0.369	0.637	0.325	0.334	0.339
		MAE	0.338	0.342	0.348	0.348	0.542	0.422	0.351	0.427	0.591	0.366	0.361	0.372
	720	MSE	0.404	0.401	0.412	0.418	1.730	0.558	0.408	0.554	0.960	0.421	0.417	0.433
		MAE	0.397	0.398	0.407	0.414	1.042	0.524	0.403	0.522	0.735	0.415	0.413	0.432
	Avg	MSE	0.278	0.282	0.288	0.290	0.757	0.358	0.291	0.350	0.571	0.305	0.306	0.327
		MAE	0.322	0.327	0.332	0.334	0.610	0.404	0.333	0.401	0.537	0.349	0.347	0.371
Weather	96	MSE	0.153	0.171	0.174	0.176	0.158	0.202	0.172	0.196	0.221	0.217	0.173	0.266
		MAE	0.199	0.222	0.214	0.218	0.230	0.261	0.220	0.255	0.306	0.296	0.223	0.336
	192	MSE	0.204	0.214	0.221	0.221	0.206	0.242	0.219	0.237	0.261	0.276	0.245	0.307
		MAE	0.249	0.259	0.254	0.256	0.277	0.298	0.261	0.296	0.340	0.336	0.285	0.367
	336	MSE	0.262	0.276	0.278	0.280	0.272	0.287	0.280	0.283	0.309	0.339	0.321	0.359
		MAE	0.292	0.300	0.296	0.298	0.335	0.335	0.306	0.335	0.378	0.380	0.338	0.395
	720	MSE	0.344	0.352	0.358	0.356	0.398	0.351	0.365	0.345	0.377	0.403	0.414	0.419
		MAE	0.345	0.350	0.347	0.348	0.418	0.386	0.359	0.381	0.427	0.428	0.410	0.428
	Avg	MSE	0.241	0.253	0.258	0.258	0.259	0.271	0.259	0.265	0.292	0.309	0.288	0.338
		MAE	0.271	0.283	0.278	0.280	0.315	0.320	0.287	0.317	0.363	0.360	0.314	0.382
Electricity	96	MSE	0.148	0.195	0.148	0.180	0.219	0.237	0.168	0.197	0.247	0.193	0.169	0.201
		MAE	0.248	0.283	0.240	0.273	0.314	0.329	0.272	0.282	0.345	0.308	0.273	0.317
	192	MSE	0.175	0.189	0.162	0.187	0.231	0.236	0.184	0.196	0.257	0.201	0.182	0.222
		MAE	0.273	0.285	0.253	0.280	0.322	0.330	0.289	0.285	0.355	0.315	0.286	0.334
	336	MSE	0.187	0.198	0.178	0.204	0.246	0.249	0.198	0.209	0.269	0.214	0.200	0.231
		MAE	0.286	0.295	0.269	0.296	0.337	0.344	0.300	0.301	0.369	0.329	0.304	0.338
	720	MSE	0.219	0.233	0.225	0.246	0.280	0.284	0.220	0.245	0.299	0.246	0.222	0.254
		MAE	0.309	0.322	0.317	0.328	0.363	0.373	0.320	0.333	0.390	0.355	0.321	0.361
	Avg	MSE	0.182	0.204	0.178	0.204	0.244	0.251	0.192	0.212	0.268	0.214	0.193	0.227
		MAE	0.279	0.296	0.270	0.294	0.334	0.344	0.295	0.300	0.365	0.327	0.296	0.338
Solar-Energy	96	MSE	0.197	0.287	0.203	0.234	0.310	0.312	0.250	0.290	0.237	0.242	0.215	0.884
		MAE	0.230	0.312	0.237	0.286	0.331	0.399	0.292	0.378	0.344	0.342	0.249	0.711
	192	MSE	0.229	0.325	0.233	0.267	0.734	0.339	0.296	0.320	0.280	0.285	0.254	0.834
		MAE	0.272	0.332	0.261	0.310	0.725	0.416	0.318	0.398	0.380	0.380	0.272	0.692
	336	MSE	0.251	0.363	0.248	0.290	0.750	0.368	0.319	0.353	0.304	0.282	0.290	0.941
		MAE	0.276	0.378	0.273	0.315	0.735	0.430	0.330	0.415	0.389	0.376	0.296	0.723
	720	MSE	0.254	0.357	0.249	0.289	0.769	0.370	0.338	0.356	0.308	0.357	0.285	0.882
		MAE	0.281	0.380	0.275	0.317	0.765	0.425	0.337	0.413	0.388	0.427	0.295	0.717
	Avg	MSE	0.233	0.333	0.233	0.270	0.641	0.347	0.301	0.330	0.282	0.291	0.261	0.885
		MAE	0.265	0.351	0.262	0.307	0.639	0.417	0.319	0.401	0.375	0.381	0.381	0.711
Exchange	96	MSE	0.084	0.086	0.086	0.097	0.256	0.094	0.107	0.088	0.267	0.148	0.111	0.197
		MAE	0.204	0.206	0.206	0.215	0.367	0.218	0.234	0.218	0.396	0.278	0.237	0.323
	192	MSE	0.172	0.173	0.177	0.181	0.470	0.184	0.226	0.176	0.351	0.271	0.219	0.300
		MAE	0.297	0.297	0.299	0.303	0.509	0.307	0.344	0.315	0.459	0.315	0.335	0.369
	336	MSE	0.321	0.328	0.331	0.343	1.268	0.349	0.367	0.313	1.324	0.460	0.421	0.509
		MAE	0.412	0.414	0.417	0.426	0.883	0.431	0.448	0.427	0.853	0.427	0.476	0.524
	720	MSE	0.828	0.856	0.847	0.936	1.767	0.852	0.964	0.839	1.058	1.195	1.092	1.447
		MAE	0.689	0.728	0.691	0.726	1.068	0.698	0.746	0.695	0.797	0.695	0.769	0.941
	Avg	MSE	0.351	0.361	0.360	0.389	0.940	0.370	0.416	0.354	0.750	0.519	0.461	0.613
		MAE	0.401	0.411	0.403	0.418	0.707	0.413	0.443	0.414	0.626	0.429	0.454	0.539
1 <sup>st</sup>	2 <sup>nd</sup>	Count	46 / 28	19 / 22	15 / 20	3 / 8	0 / 3	0 / 0	0 / 3	1 / 3	0 / 0	4 / 0	0 / 0	0 / 0

overfitting. This hypothesis is supported by ETTm1 results: UniMixer achieves optimal/near-optimal performance across all horizons as data volume increases, while HDMixer maintains MAE advantages and PatchTST excels in MSE. The findings highlight dataset scale’s critical role in architectures modeling variable interactions, suggesting future focus on regularization for small datasets.

**Experiments on ETTh2&ETTM2** The results on ETTh2&m2 are similar to those on ETTh1&m1, but the clearer multivariate relationships in ETTh2&m2 likely lead to better overall performance. On ETTh2, despite some prediction lengths not performing as well as HDMixer, the results were still optimal or near-optimal across all lengths. On ETTm2, the advantage was more pronounced, with UniMixer achieving optimal results for nearly all prediction lengths, significantly outperforming HDMixer, as well as HDMixer and iTransformer, which do not leverage inter-variable correlations effectively. This demonstrates the effectiveness of our patch-wise multivariate relationship learning strategy combined with global multivariate correlation information.

**Experiments on Weather&Exchange** Small-scale Weather and Exchange datasets present distinct characteristics: Exchange and Weather both feature higher variable counts than ETT series, exhibiting complex interdependencies and smoother trends. On Exchange, limited data induces overfitting in most baselines, while linear models thrive on trend smoothness. UniMixer’s linear-based Patch-Wise Mixer leverages this trait, augmented by Global Context Enhancer for dependency modeling, achieving optimal performance across most horizons. iTransformer’s full-sequence tokenization captures global patterns, yielding near-optimal results. Weather’s smoother patterns, stronger multivariate relationships, and ample data enable UniMixer’s full-horizon dominance over HDMixer, iTransformer, and PatchTST. Notably, Crossformer’s Weather-specific near-optimal performance underscores effective multivariate modeling’s criticality here.

**Experiments on Electricity&Solar-Energy** High-dimensional variables in Electricity & Solar-Energy datasets complicate multivariate relationship modeling, causing traditional models like HDMixer and PatchTST to underperform. iTransformer, however, excels by tokenizing entire sequences to capture global interactions, achieving superior results. UniMixer combines local patch analysis with global dependency modeling through its Global Context Enhancer and explicit Correlation Token Mapper, enabling dual-view prediction. This hybrid approach allows UniMixer to achieve near-optimal performance across most predictions and optimal results in select cases.

### 4.3 Ablation Studies

In order to validate the effectiveness of UniMixer’s individual component designs, we conducted detailed ablation experiments, including experiments with

removing components (w/o) and replacing components (replace). The specific ablation study results are shown in the Table 3. In the table, w/o PWM means removing the Patch-Wise Mixer component and using only the prediction head and other modules for prediction; w/o GCE means removing the Global Context Enhancer component; w/o CTM means removing the Correlation Token Mapper component; Replace Patch-Wise Mixer means replacing Patch-Wise Mixer with the Hierarchical Dependency Explorer component proposed in HDMixer.

**Table 3.** Ablation of our model component across three datasets with input length  $L = 96$ . The best results are in **bold**.

Dataset		ETTM2				Weather				Solar-Energy			
Prediction Length		96	192	336	720	96	192	336	720	96	192	336	720
Full	MSE	<b>0.171</b>	<b>0.239</b>	<b>0.300</b>	<b>0.408</b>	<b>0.153</b>	<b>0.204</b>	<b>0.262</b>	<b>0.344</b>	<b>0.197</b>	<b>0.229</b>	<b>0.251</b>	<b>0.254</b>
	MAE	<b>0.255</b>	<b>0.301</b>	<b>0.338</b>	<b>0.401</b>	<b>0.199</b>	<b>0.249</b>	<b>0.292</b>	<b>0.345</b>	<b>0.230</b>	<b>0.272</b>	<b>0.276</b>	<b>0.281</b>
w/o PWM	MSE	0.180	0.246	0.302	0.413	0.173	0.221	0.276	0.352	0.208	0.238	0.278	0.272
	MAE	0.263	0.304	0.339	0.404	0.216	0.258	0.298	0.350	0.233	0.274	0.296	0.286
w/o GCE	MSE	0.174	0.241	0.303	0.417	0.160	0.208	0.269	0.353	0.210	0.243	0.267	0.270
	MAE	0.258	<b>0.301</b>	0.340	0.406	0.206	0.252	0.295	0.349	0.261	0.289	<b>0.276</b>	0.288
w/o CTM	MSE	0.176	0.244	0.304	0.408	0.157	0.205	0.265	0.353	0.199	0.232	0.255	0.260
	MAE	0.261	0.302	0.341	0.403	0.204	0.248	0.294	0.352	0.236	0.278	0.286	0.288
Replace PWM	MSE	0.180	0.247	0.314	0.412	0.180	0.225	0.285	0.363	0.221	0.251	0.256	0.255
	MAE	0.264	0.308	0.351	0.402	0.223	0.262	0.306	0.357	0.282	0.286	0.287	0.286

**Ablation study of Patch-Wise Mixer.** After removing the Patch-Wise Mixer, all evaluation metrics for prediction lengths exhibited a more pronounced decline across all three datasets, particularly in the Weather dataset, where the MSE reduction reached 13.1% at a prediction length of 96. This suggests that relying solely on the additional information and prediction headers from the other components significantly hampers prediction accuracy. The Patch-Wise Mixer, therefore, is essential for effectively capturing both multivariate relationships and temporal dynamics.

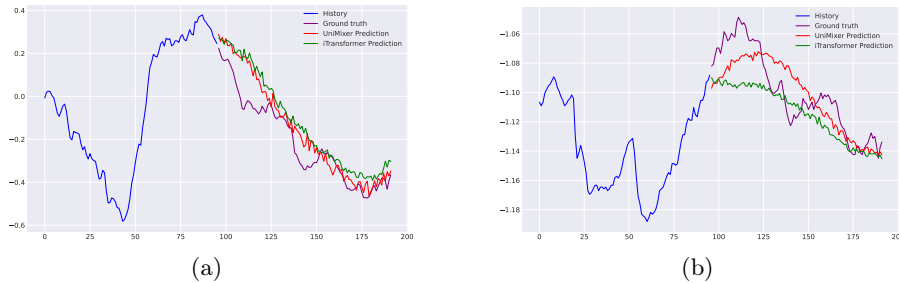
Additionally, we conducted experiments replacing the Patch-Wise Mixer with the Hierarchical Dependency Explorer module proposed in HDMixer, which focuses solely on multivariate relational changes at a single time step with a limited receptive field. The results showed a significant decline in the performance metrics across all datasets. This decrease is likely due to the small receptive field, which causes the module to learn incorrect inter-multivariate relationships in certain areas. These learned relationships are inconsistent with those captured by other modules, leading to an overall performance drop.

**Ablation study of Global Context Enhancer.** The Global Context Enhancer addresses the issue of incomplete capture of multivariate relationships due to the limited sensing field of the Patch-Wise Mixer by leveraging a global

field of view to capture these relationships as additional auxiliary information, thereby enhancing multivariate time series forecasting. Experimental results reveal a slight decrease in performance indicators for the ETTm2 and Weather datasets, which contain fewer variables, while a more substantial decline is observed in the Solar-Energy dataset, which includes a larger number of variables. These findings provide strong evidence supporting the effectiveness of this module design.

**Ablation study of Correlation Token Mapper.** Correlation Token Mapper is proposed to add additional auxiliary information to the original sequence so that the model can better learn the relationship between the variables, as can be seen from the experimental results, in the prediction of the results can play a certain degree of enhancement, proving that the design of this module is effective.

#### 4.4 Visual Experiments

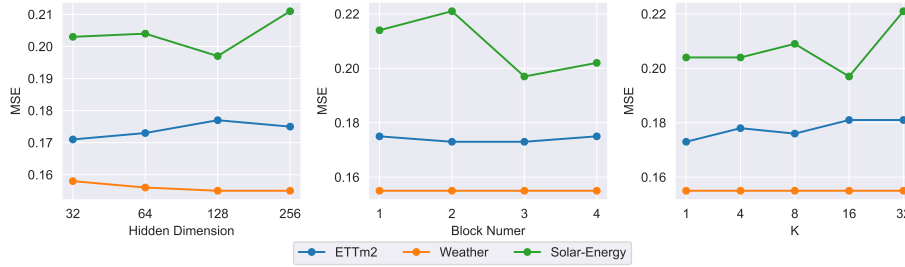


**Fig. 3.** Visualisation of the forecasting results on the Weather dataset, where the input length  $L = 96$  and the prediction length  $T = 96$ .

Figure 3 provides a visual representation of the comparison between UniMixer’s forecasting results and those of iTransformer. As can be seen from Figure 3(a), in terms of the forecasting of details, our model demonstrates better forecasting results than the iTransformer and shows a better fit to the curve of the true values. From Figure 3(b), it can be observed that in the forecasting of the overall trend, benefiting from the capture of the relationships among variables, our model also exhibits better forecasting capabilities and can more accurately predict the future trends.

#### 4.5 Hyperparameter Sensitivity

We evaluate the hyperparameter sensitivity of UniMixer from the perspective of the following factors: the hidden dimension  $D$ , the number of block layers  $L$ ,



**Fig. 4.** Hyperparameter sensitivity of UniMixer. The results are recorded with the lookback window length  $L = 96$  and the forecast window length  $T = 96$ .

and the ratio of MLP to Transformer Hidden Dimension  $K$ . As illustrated in the Fig 4, the impact of parameter selection on the results varies significantly across different datasets. When the number of variables is large (Solar-Energy), each parameter has a significant impact on the final performance. The increase in the number of dimensions and the number of blocks leads to a more accurate representation, also leads to faster overfitting, especially if the number of variables is large, and the impact of this effect of the change will be stronger. Therefore, larger dimensions and block sizes are not always advantageous; sometimes, smaller values can achieve better performance or equivalent results with reduced computational cost.

## 5 Conclusion

In this paper, we introduce UniMixer, a unified framework that effectively integrates local temporal dependency modeling with global inter-series relationship modeling for multivariate time series forecasting. By combining the Patch-Wise Mixer, Global Context Enhancer, and Correlation Token Mapper, UniMixer captures both local temporal dynamics and long-range dependencies, enhancing forecasting accuracy and robustness. Extensive experiments on several real-world datasets demonstrate its superior performance over state-of-the-art methods, particularly in handling complex, high-dimensional datasets and long-term forecasting horizons. UniMixer not only provides a novel approach to multivariate time series forecasting but also lays a solid foundation for future advancements in this area.

**Acknowledgments.** This research is supported by Nankai University-ENNEW Joint R&D Project, National Engineering Research Center for Digital Construction and Evaluation Technology of Urban Rail Transit, Development of a platform for quantity statistics and budget preparation of urban rail transit projects based on big data analysis (No. 2022A02158007). Computation is supported by the Supercomputing Center of Nankai University (NKSC).

## References

1. Angryk, R.A., Martens, P.C., Aydin, B., Kempton, D., Mahajan, S.S., Basodi, S., Ahmadzadeh, A., Cai, X., Filali Boubrahimi, S., Hamdi, S.M., et al.: Multivariate time series dataset for space weather data analytics. *Scientific data* (2020)
2. Borovykh, A., Bohte, S., Oosterlee, C.W.: Conditional time series forecasting with convolutional neural networks. *ArXiv Preprint ArXiv:1703.04691* (2017)
3. Box, G.E., Jenkins, G.M., MacGregor, J.F.: Some recent advances in forecasting and control. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **23**(2), 158–179 (1974)
4. Chen, S.A., Li, C.L., Yoder, N., Arik, S.O., Pfister, T.: Tsmixer: An all-mlp architecture for time series forecasting. *ArXiv Preprint ArXiv:2303.06053* (2023)
5. Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., Yu, R.: Long-term forecasting with tide: Time-series dense encoder. *Trans. Mach. Learn. Res.* **2023** (2023)
6. Hochreiter, S.: Long short-term memory. *Neural Computation MIT-Press* (1997)
7. Huang, Q., Shen, L., Zhang, R., Cheng, J., Ding, S., Zhou, Z., Wang, Y.: Hd-mixer: Hierarchical dependency with extendable patch for multivariate time series forecasting. In: *Thirty-Eighth AAAI Conference on Artificial Intelligence*. pp. 12608–12616 (2024)
8. Kalyan, K.S., Rajasekharan, A., Sangeetha, S.: Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542* (2021)
9. Khan, S.H., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: A survey. *ACM Comput. Surv.* **54**(10s), 200:1–200:41 (2022)
10. Lai, G., Chang, W., Yang, Y., Liu, H.: 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)
11. Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y., Yan, X.: Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: *Advances in Neural Information Processing Systems*. pp. 5244–5254 (2019)
12. Liu, C., Sun, B., Zhang, C., Li, F.: A hybrid prediction model for residential electricity consumption using holt-winters and extreme learning machine. *Applied Energy* **275**, 115383 (2020)
13. Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., Xu, Q.: Scinet: Time series modeling and forecasting with sample convolution and interaction. In: *Advances in Neural Information Processing Systems* (2022)
14. Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A.X., Dustdar, S.: Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In: *The Tenth International Conference on Learning Representations* (2022)
15. Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M.: itransformer: Inverted transformers are effective for time series. In: *The Twelfth International Conference on Learning Representations* (2024)
16. Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M.: itransformer: Inverted transformers are effective for time series forecasting. In: *The Twelfth International Conference on Learning Representations* (2024)
17. Liu, Y., Wu, H., Wang, J., Long, M.: Non-stationary transformers: Exploring the stationarity in time series forecasting. In: *Advances in Neural Information Processing Systems* (2022)

18. Nazareth, N., Reddy, Y.V.R.: Financial applications of machine learning: A literature review. *Expert Systems with Applications* **219**, 119640 (2023)
19. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. In: *The Eleventh International Conference on Learning Representations* (2023)
20. Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., Cottrell, G.W.: A dual-stage attention-based recurrent neural network for time series prediction. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. pp. 2627–2633 (2017)
21. Rangapuram, S.S., Seeger, M.W., Gasthaus, J., Stella, L., Wang, Y., Januschowski, T.: Deep state space models for time series forecasting. In: *Advances in Neural Information Processing Systems*. pp. 7796–7805 (2018)
22. Sims, C.A.: Macroeconomics and reality. *Econometrica: Journal of the Econometric Society* pp. 1–48 (1980)
23. Tedjopurnomo, D.A., Bao, Z., Zheng, B., Choudhury, F.M., Qin, A.K.: A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering* **34**(4), 1544–1561 (2020)
24. Wen, R., Torkkola, K., Narayanaswamy, B., Madeka, D.: A multi-horizon quantile recurrent forecaster. *ArXiv Preprint ArXiv:1711.11053* (2017)
25. Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: Timesnet: Temporal 2d-variation modeling for general time series analysis. In: *The Eleventh International Conference on Learning Representations* (2023)
26. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: *Advances in Neural Information Processing Systems*. pp. 22419–22430 (2021)
27. Zhang, Y., Yan, J.: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: *The Eleventh International Conference on Learning Representations* (2023)
28. Zhao, L., Shen, Y.: Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators. In: *The Twelfth International Conference on Learning Representations* (2024)
29. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: *Thirty-Fifth AAAI Conference on Artificial Intelligence*. pp. 11106–11115 (2021)
30. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: *International Conference on Machine Learning*. vol. 162, pp. 27268–27286 (2022)