2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Fourier U-shaped Network for Multi-variate Time Series Forecasting

Baowen Xu[1,2], Xuelei Wang[1*], Chengbao Liu[1], and Shuo Li[1,2]

[1]Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China

***Abstract*—Multi-variate time series forecasting plays a crucial role in addressing key tasks across various domains, such as early warning, pre-planning, resource scheduling, and other critical tasks. Thus, accurate multi-variate time series forecasting is of significant importance in guiding practical applications and facilitating these essential tasks. Recently, Transformer-based multi-variate time series forecasting models have demonstrated tremendous potential due to their outstanding performance in long-term time predictions. However, Transformer-based models for multi-variate time series forecasting often come with high time complexity and computational costs. Therefore, we propose a low time complexity model called Fourier U-shaped Network (F-UNet) for multi-variate time series forecasting, which is non-Transformer based. Specifically, F-UNet is composed of low time complexity neural network components, such as Fourier neural operator and feed-forward neural network, arranged in a U-shaped architecture. F-UNet conducts channel and temporal modeling separately for the multi-variate time series. The U-Net constructed based on Fourier neural operators is employed to achieve channel interactions, while linear layers are used to realize temporal interactions. Experimental results on several real-world datasets demonstrate that F-UNet outperforms existing Transformer-based models with higher efficiency in multi-variate time series forecasting.**

***Index Terms*—Multivariate time series forecasting, U-Net, fourier transform**

I. Introduction

With the continuous advancement of information and intelligence, various technologies such as sensors and monitoring devices have been widely applied, resulting in a large amount of time series data. These abundant data resources provide a solid foundation for conducting time series forecasting. The widespread use of time series data has promoted research and applications in time series forecasting. Time series forecasting is a method that utilizes historical data models to predict future trends. By analyzing time series data, we can identify patterns and trends, and based on these patterns, establish predictive models to anticipate future occurrences. This holds significant importance in decision-making, resource planning, risk management, and other aspects. For instance, based on weather forecasts, people can make disaster preparations in advance; using traffic flow predictions, individuals can choose the optimal travel routes; and through forecasting future weather conditions based on historical data, precision agriculture can enhance crop yields and reduce costs by adjusting planting times and methods.

Compared with traditional statistical methods and machine learning methods, deep learning models have better adaptability in time series prediction, can handle complex non-linear relationships, handle large amounts of data, and adapt to different data types, showing better prediction results. Transformer-based models have emerged in many excellent works on multi-variate time series prediction tasks because they can better solve long-term prediction problems. However, the quadratic complexity of the sequence length L in memory and time limits the performance improvement of the Transformer-based models. Therefore, current Transformer-based methods focus on reducing the complexity of attention computing. LogTrans [1] reduces complexity to $\mathcal{O}(L(\log(L))^2)$ by introducing local convolution in Transformer. Informer [2] reduces complexity to $\mathcal{O}(L\log(L))$ by utilizing ProbeSpare attention based on KL divergence. Autoformer [3] reduces complexity to $\mathcal{O}(L\log(L))$ by proposing an Auto-correlation mechanism. FEDformer [4] reduces complexity to $\mathcal{O}(L)$ by replacing the self-attention block with a frequency enhancement block.

Although Transformer-based models have made significant progress in reducing complexity, recent studies have shown that non-Transformer models, including SCINet [5], DLinear [6], and MTS-Mixers [7], outperform Transformer-based models in terms of predictive performance. The multi-variate time series forecasting model based on non-Transformer exhibits advantages such as low time complexity and high prediction accuracy. Encouraged by the outstanding performance and promising prospects of excellent non-Transformer multi-variate time series forecasting models, we propose a novel non-Transformer Fourier U-shaped Network (F-UNet) for multi-variable time series prediction. Specifically, the multi-variate time series data consists of both channel and temporal dimensions, and we separately model the channel interaction and temporal interaction. The Fourier U-Net dynamically captures local and global spatial correlations for channel interactions, while linear layer is utilized to model the temporal correlations for temporal interactions. The main contributions of this paper are as follows.

1) In multi-variate time series data, there are two dimensions: channel and temporal. We perform separate operations for channel interaction and temporal interaction. Specifically, for channel interaction, we use the Fourier

This work is supported in part by the National Key Research and Development Program of China under Grant 2022YFB3305401
*Corresponding author

U-Net as the framework to capture local and global spatial correlations in the channel dimension. For temporal interaction, we use linear layer as the framework to capture temporal correlations in the time dimension.

2) We propose a Fourier U-shaped Network to finely process local spatial information and global spatial information to better capture channel dependencies.
3) We propose a non-Transformer model with low time complexity, which, compared to the state-of-the-art Transformer-based multi-variate time series prediction models, achieves state-of-the-art predictive performance on several common real-world multi-variate time series datasets with very few parameters and fast inference speed.

The rest of the paper is organized as follows. Related Work is discussed in Section II. Fourier U-shaped network architecture is described in Section III. Experimental Results and Analysis are provided in Section IV. Finally, the paper is concluded in Section V.

II. Related Work

A. Multivariate Time Series Forecasting

Multivariate time series prediction refers to the prediction of data at a certain time or period of time in the future based on historical data. This technology is widely used in economic, financial, meteorological, traffic, and other fields. Due to the important application significance of multivariate time series, many excellent models have been generated. ARIMA [8] uses a differential method to convert non-stationary processes into stationary processes for prediction. DeepAR [9] combines RNN and autoregressive methods to model the probability distribution of future sequences. The multivariate time series prediction models based on Transformer can effectively conduct long-term modeling, demonstrating strong potential. Recently, in terms of time complexity and prediction accuracy, non-Transformer-based models [5]–[7] have surpassed Transformer-based models [5]–[7], and non-Transformer-based models may once again be great.

B. U-Net

U-Net [10] is a deep learning architecture developed for image modeling, segmentation, and generation in computer vision. The U-Net architecture consists of two main components: an encoder network and a decoder network. The encoder network is designed to extract features and downsample them to lower resolutions. The decoder network is designed to obtain a feature map from the encoder network and sample it to a higher resolution. Local and global information is processed in a more distributed manner. Downsampling corresponds to sequential processing of information more globally, while upsampling corresponds to fine-grained global information and adding local information by skipping connections. A typical U-Net architecture is shown in Fig.1.

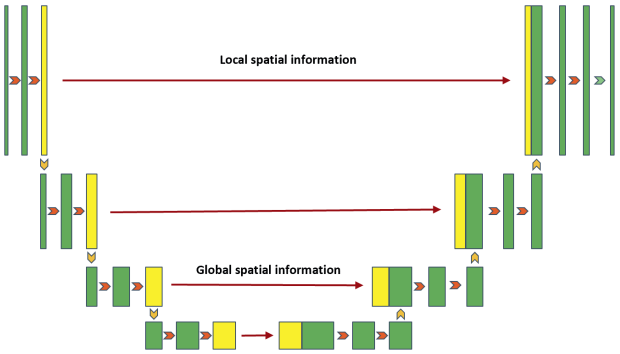


Fig. 1. U-Net based architectures.

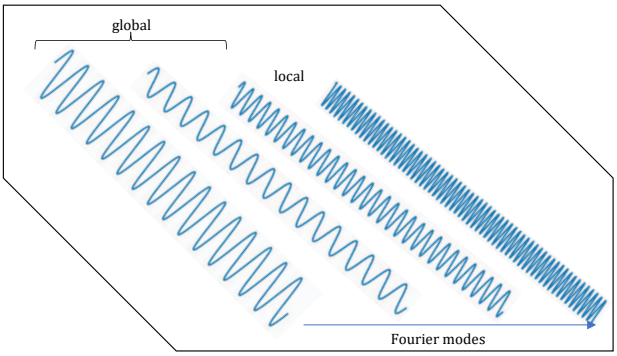


Fig. 2. Fourier spatial interaction.

C. Fourier Neural Operator Layer

The Fourier Neural Operator (FNO) Layer [11] consists of fast Fourier transform and weight multiplication. As shown in Fig.2, the low Fourier modes provide global information, while the high Fourier modes provide local information. The Fourier neural operator layer simultaneously processes global and local information by multiplying the weights of different modes. Discrete Fourier Transform (DFT) in Fourier space is briefly described below.

Discrete Fourier Transform (DFT) converts n-dimensional complex signal $f(x) = f(x_1, ..., x_n)$ at $M_1 \times ... \times M_n$ grid points into its complex Fourier modes $\hat{f}(\epsilon_1, ...\epsilon_n)$, Specifically, as follows:

$$\hat{f}(\epsilon_1, ...\epsilon_n) = \mathcal{F}_f(\epsilon_1, ...\epsilon_n) = \Sigma_{m_1=0}^{M_1} ... \Sigma_{m_n=0}^{M_n} f(x) e^{-2\pi i \cdot (\frac{m_1 \epsilon_1}{M_1} + ... + \frac{m_n \epsilon_n}{M_n})}. \quad (1)$$

where $(\epsilon_1, ...\epsilon_n) \in \mathbb{Z}_{M_1} \times ... \mathbb{Z}_{M_n}$.

III. Fourier U-shaped Network Architecture

In this section, first briefly state the problem definition of multivariate time series prediction; Secondly, briefly state the series stationarization; Finally, a detailed introduction is given to the overall architecture of F-UNet.

A. Problem definition

Given a historical multivariate time series $\mathcal{X}_h = [x_1, ..., x_T] \in \mathbb{R}^{T \times C}$, where T represents the length of the

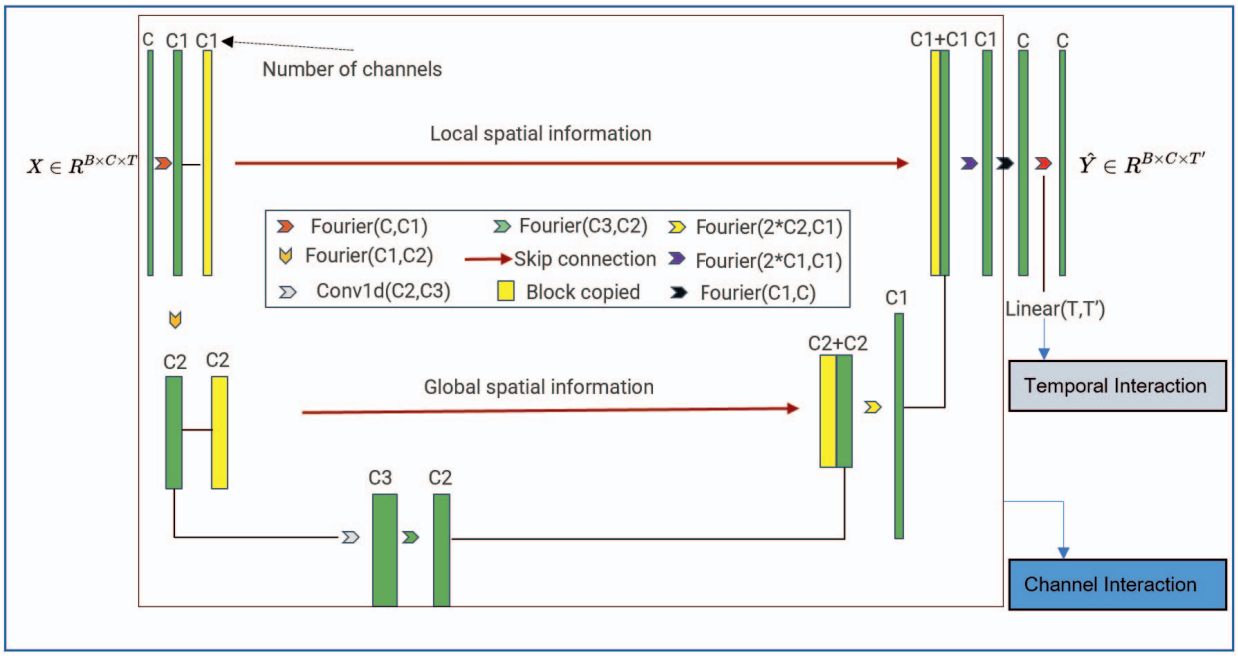


Fig. 3. An illustration of the proposed F-UNet architecture.

historical time series and C represents the number of features, the multivariate time series prediction task aims to predict the value of the next T' time steps, *i.e.*, $\hat{\mathcal{X}}_f = [\hat{x}_{T+1}, ..., \hat{x}_{T+T'}] \in \mathbb{R}^{T'\times C}$. The task of multivariate time series prediction is to learn a mapping function f, *i.e.*, $\mathcal{X}_h \xrightarrow{f} \hat{\mathcal{X}}_f$.

B. Series Stationarization

Although stationarity is crucial for the predictability of time series, the majority of real-world time series are non-stationary, possibly exhibiting distributional shifts, where statistical characteristics such as mean and variance often vary over time. Most existing works remove non-stationary information (such as mean and variance) from the input sequences during data pre-processing to reduce data distribution differences and improve predictive model performance. However, applying the removal of non-stationary information through normalization to the model inputs can introduce another issue, preventing the model from capturing the original data distribution, *i.e.*, removing non-stationary information that may be highly valuable for predictions.

We adopt the Reversible Instance Normalization (*RevIN*) [12], which explicitly reintroduces the removed non-stationary information back into the model. This effectively addresses the problem of distributional shifts in time series forecasting. *RevIN* is a general normalization and de-normalization method with a symmetric structure of learnable affine transformations. It consists of two corresponding operations [13]: the normalization module handles non-stationary sequences caused by varying means and standard deviations, making the sequences more stable for predictability; the de-normalization module transforms the model output back to the original statistical data, *i.e.*, the model output retains the non-stationary information of the original sequences, preventing excessive over-stabilization.

1) Normalization Module: This module is to convert the original data into stationarized input, which follows a relatively stable distribution. The Normalization Module can be formulated as follows:

$$\begin{cases} \mu_t = E_t[x_h^t] = \frac{1}{T}\Sigma_{j=1}^{T} x_{t+j} \\ \sigma_t^2 = Var[x_h^t] = \frac{1}{T}\Sigma_{j=1}^{T}(x_{t+j} - \mu_t)^2 \\ x_h^{norm,t} = \frac{x_h^t - \mu_t}{\sqrt{\sigma_t^2 + \varepsilon}} \cdot \alpha + \beta \end{cases} \tag{2}$$

where $u_t, \sigma_t^2 \in \mathbb{R}^{1\times C}$, $\alpha, \beta \in \mathbb{R}^C$ are learnable affine parameter vectors. ε is a positive scalar that can be ignored.

2) De-normalization Module: Let $NN_p(\cdot)$ represents the neural network, where p is all the parameters of the neural network. The stationary input sequence $x_h^{norm,t}$ is input into the neural network model $NN_p(\cdot)$ to obtain the stationary output $\hat{x}_f^{norm,t}$ of the model, *i.e.*, $\hat{x}_f^{norm,t} = NN_p(x_h^{norm,t})$. The De-normalization Module can be formulated as follows:

$$\hat{x}_f^t = \sqrt{\sigma_t^2 + \varepsilon} \cdot (\frac{\hat{x}_f^{norm,t} - \beta}{\alpha}) + \mu_t \tag{3}$$

De-normalization Module returns the original data distribution information to the model output by scaling and shifting by an amount equal to the shift and scaling of the input data in the Normalization Module.

In summary, the stationary input sequence $\mathcal{X}_h^{norm}$ obtained through Eq.(2) is fed into the neural network model $NN_p(\cdot)$ to obtain a stationary output $\hat{\mathcal{X}}_f^{norm}$ of the model. Finally, the stationary output is processed through Eq.(3) to obtain the final

output $\hat{\mathcal{X}}_f$ of the model. The above process can be formulated as:

$$\begin{cases} \mathcal{X}_h^{norm} = \Theta_{Norm}(\mathcal{X}_h) \\ \hat{\mathcal{X}}_f^{norm} = NN_p(\mathcal{X}_h^{norm}) \\ \hat{\mathcal{X}}_f = \Theta_{deNorm}(\hat{\mathcal{X}}_f^{norm}) \end{cases} \tag{4}$$

Where Θ_{Norm} represents Normalization operation shown in Eq.(2), Θ_{deNorm} represents De-normalization operation shown in Eq.(3), and NN_p represents the neural network model.

C. Overall architecture of F-UNet

F-UNet models multivariate time series channel and temporal separately. In order to capture channel correlation, we fully consider the interaction of local and global spatial information, and design a Fourier U-Net. To capture temporal correlation, a simple linear layer is used to map historical time steps to predicted time steps. The proposed F-UNet architecture is shown in Fig.3.

The F-UNet consists of two core components: the Fourier U-Net for channel interaction and linear layers for time interaction. The overall model can be formulated as follows:

$$\mathcal{X}^{\mathcal{C}} = f_{channel}(\mathcal{X}_h). \tag{5}$$

$$\mathcal{X}_f = f_{temporal}(\mathcal{X}^{\mathcal{C}}). \tag{6}$$

Where $f_{channel}$ and $f_{temporal}$ respectively represent the parameters for channel interaction and time interaction. Detailed descriptions of channel interaction and temporal interaction are provided in III-D and III-E, respectively.

D. Channel Interaction

We perform channel interaction using the proposed Fourier U-Net. The Fourier U-Net consists of Fourier neural operators and channel-wise attention. The low Fourier modes of Fourier neural operators provide global information, while the high Fourier modes offer local information. The downsampling in the U-Net corresponds to a more global sequential processing of information, while the upsampling corresponds to fine-grained global information. As a result, the Fourier U-Net can effectively capture local and global spatial correlations. Additionally, the introduction of channel attention adjusts the importance of feature map channels, enhancing the model's focus and response to crucial features while reducing reliance on irrelevant features. This improves the expressive capacity and performance of the model in feature processing.

1) Fourier Neural Operator: The Fourier neural operator includes the fast Fourier transform and the weight multiplication in Fourier space. The low Fourier modes provide global information, while the high Fourier modes provide local information. The Fourier neural operator combines global and local information by weighting different modes simultaneously. Given the input $\mathcal{X} \in \mathbb{R}^{C_{in} \times T}$, the Fourier neural operator can be formulated as:

$$\begin{aligned} \mathcal{X}_{fourier} &= g_{fft}(\mathcal{X}) \in \mathbb{C}^{C_{in} \times d}, \\ \mathcal{X}'_{fourier}[:, :modes] &= \mathcal{W}\mathcal{X}_{fourier} \in \mathbb{C}^{C_{out} \times modes}, \\ \mathcal{X}'_{fourier}[:, modes:] &= \mathcal{Z}, \\ \mathcal{X}_{ifourier} &= g_{ifft}(\mathcal{X}'_{fourier}) \in \mathbb{R}^{C_{out} \times T}. \end{aligned} \tag{7}$$

Where g_{fft} and g_{ifft} represent the fast Fourier transform and inverse Fourier transform, respectively. $modes$ is the number of modes, $\mathcal{W} \in \mathbb{C}^{C_{in} \times C_{out} \times modes}$ denotes a complex parameter matrix, $\mathcal{Z} \in \mathbb{C}^{C_{out} \times (d - modes)}$ represents the zero matrix, that is, to filter out the invalid high-frequency part, and $d = T//2 + 1$.

2) Channel-wise Attention: Inspired by the lightweight convolutional attention module proposed by CBAM [14], we incorporate channel-wise attention as shown in Fig.4 to enhance the model's representational capacity. Channel-wise attention mainly focuses on relationships between different channels within the feature map. In CNNs, each channel represents different feature information. Some channels may be more critical for the successful execution of the task, while others may contribute less. The channel attention mechanism calculates importance weights for each channel, enabling the network to automatically learn and emphasize the most relevant feature channels for the current task. Therefore, the introduction of channel-wise attention allows for better selection of important features and suppression of irrelevant ones, thus improving the efficiency and representational power of feature representation. Specifically, given the input/output $\mathcal{X}'_{ca}/\mathcal{X}_{ca} \in \mathbb{R}^{C_n \times T}$ of channel-wise attention, it can be formulated as follows:

$$\begin{aligned} \psi_{ca}(\mathcal{X}'_{ca}) = \sigma(&W_1^n(ReLU(W_0^n(AvgPool(\mathcal{X}'_{ca})))) \\ &+ W_1^n(ReLU(W_0^n(MaxPool(\mathcal{X}'_{ca}))))), \\ \mathcal{X}_{ca} &= \psi_{ca}(\mathcal{X}'_{ca}) \cdot \mathcal{X}'_{ca}. \end{aligned} \tag{8}$$

Where $\psi_{ca}(\mathcal{X}'_{ca}) \in \mathbb{R}^{C_n \times 1}$ represents the 1-D channel attention weights, $AvgPool(\mathcal{X}'_{ca})$ and $MaxPool(\mathcal{X}'_{ca}) \in \mathbb{R}^{C_n \times 1}$

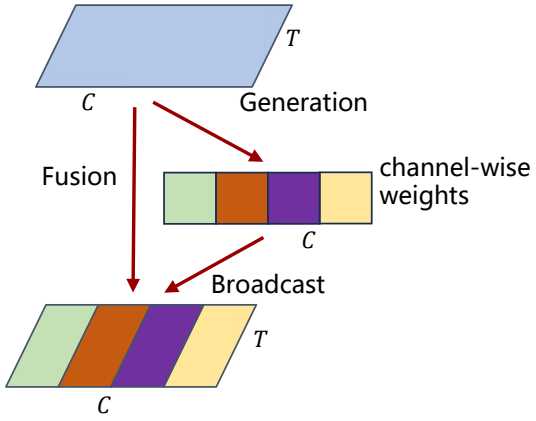


Fig. 4. Channel-wise Attention.

denote the average pooling and max pooling operations, respectively. $W_1^n \in \mathbb{R}^{C_n \times \frac{C_n}{r_n}}$ and $W_0^n \in \mathbb{R}^{\frac{C_n}{r_n} \times C_n}$ are parameter matrices, and r_n indicates the channel reduction factor. To reduce the number of parameters, the same weights W_0^n and W_1^n are shared.

In summary, the i^{th} layer upsampling or downsampling in the Fourier U-Net can be formulated as follows:

$$\begin{aligned} \hat{\mathcal{X}}_i^{\mathcal{C}} &= \pi_{fourier}(\mathcal{X}) \\ \mathcal{X}_i^{\mathcal{C}} &= \pi_{ca}(\hat{\mathcal{X}}_i^{\mathcal{C}}). \end{aligned} \tag{9}$$

Where $\pi_{fourier}$ represents all the parameters of the Fourier neural operator in Eq.(7), and π_{ca} represents all the parameters of the channel-wise attention in Eq.(8).

E. Temporal Interaction

For temporal interactions, we choose to use feed-forward neural networks to model the time dependencies. Specifically, given the output $\mathcal{X}^{\mathcal{C}} \in \mathbb{R}^{T \times C}$ for channel interactions, the output $\mathcal{X}^f \in \mathbb{R}^{T' \times C}$ of temporal interactions can be formulated as:

$$\mathcal{X}_f = W_{temp}\mathcal{X}^{\mathcal{C}} + b_{temp}. \tag{10}$$

Where $W_{temp} \in \mathbb{R}^{T' \times T}$ and $b_{temp} \in \mathbb{R}^{T'}$ are the parameters of the feed-forward neural network.

IV. Experimental Results and Analysis

A. Experimental Setup

1) Datasets: We conduct extensive experiments on several widely used real-world multivariate time prediction datasets, including *ETT* (transformer temperature) [2] (ETTh2, ETTm1, ETTm2), *Traffic*, *Electricity*, *Weather*, *ILI*, and *Exchange-Rate* [15]. Table I provides a brief description of these datasets. All experiments on these datasets in this article are conducted under the setting of multivariate time series prediction.

2) Compared methods: The baseline models selected in the experiment include five state-of-the-art Transformer-based models: FEDformer [4], Autoformer [3], Informer [2], Pyraformer [16], LogTrans [1], and a state-of-the-art non-Transformer-based model: SCINet [5].

3) Implementation details: The training of F-UNet is based on L_2 loss (Eq.(12)) and Adam optimizer. The training process uses early stop over 10 epochs, with all models implemented in PyTorch and trained and tested on a single NVIDIA GeForce 3090 32 GB GPU.

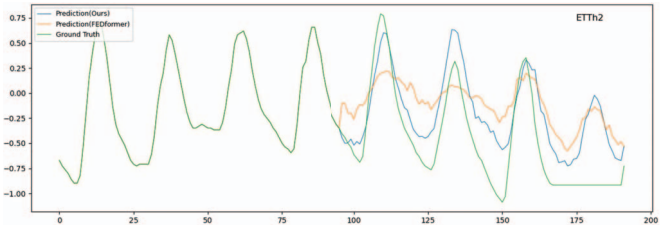
The hyper-parameter of the baseline methods are the same as those in the original paper. Table II shows the settings of hyper-parameter for different datasets.

B. Evaluation Indicator

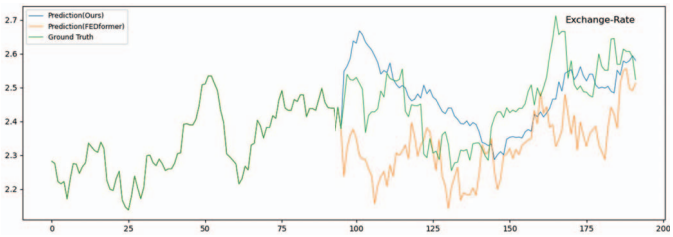
As in previous work, we use two metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE).

$$MAE(\mathcal{X}_f, \hat{\mathcal{X}}_f) = \frac{\sum_{i=1}^{T'}\sum_{j=1}^{C}|\hat{x}_{T+i}^j - x_{T+i}^j|}{T' \cdot C} \tag{11}$$

$$MSE(\mathcal{X}_f, \hat{\mathcal{X}}_f) = \frac{\sum_{i=1}^{T'}\sum_{j=1}^{C}(\hat{x}_{T+i}^j - x_{T+i}^j)^2}{T' \cdot C} \tag{12}$$



(a)



(b)

Fig. 5. Given 96 to predict 96 steps on the (a) **ETTh2**, (b) **Exchange-Rate**.

C. Results and Analyses

In the experiment, the input time steps for the *ILI* dataset are set to 36, while for other datasets, they are set to 96. The prediction time steps for the *ILI* dataset are set to 24, 36, 48, 60, whereas for other datasets, they are set to 96, 192, 336, 720. Table III summarizes the prediction results of different models on eight multivariate time series datasets. The best results are highlighted in bold, while the second best results are highlighted with an underline. Figure 5 provides a qualitative analysis comparing the prediction results of F-UNet and the state-of-the-art Transformer-based model, FEDformer.

Table III demonstrates that the prediction results of F-UNet are close to, and in some cases even superior to, the state-of-the-art baseline models. On the *Exchange*, *Weather*, and *ILI* datasets, the proposed F-UNet outperforms the state-of-the-art comparative models for all prediction time steps. On the *ETT* dataset, F-UNet outperforms the state-of-the-art Transformer-based model. On the *ECL* and *Traffic* datasets, the prediction results of F-UNet are comparable to those of the state-of-the-art Transformer-based model. Overall, the proposed F-UNet shows competitive advantages compared to the state-of-the-art Transformer-based models.

Table IV presents the parameter counts of different models with an input time step of 96 and an output time step of 96. Table V shows the inference time and average training time for different models. From Table IV, it can be observed that F-UNet has significantly fewer parameters than other models. Taking into account the results from both Table III and Table IV, we can deduce that even with a significantly lower number of parameters compared to other baseline models, F-UNet's prediction results can be close to or even superior to those of the baseline models, demonstrating its effectiveness in multivariate time series prediction tasks.

Table III indicates that F-UNet's predictive performance

TABLE I
Summary of different datasets.

Datasets	ETTh	ETTm	Traffic	Weather	Electricity	ILI	Exchange-Rate
Variants	7	7	862	21	321	7	8
Timesteps	17,420	69,680	17,544	52,696	26,304	966	7,588
Granularity	1hour	15min	1hour	10min	1hour	1 week	1day
Start time	7/1/2016	7/1/2016	1/1/2015	1/1/2020	1/1/2012	1/1/2002	1/1/1990
Task type	Multi-step	Multi-step	Multi-step	Multi-step	Multi-step	Multi-step	Multi-step
Data partition	6:2:2	6:2:2	7:1:2	7:1:2	7:1:2	7:1:2	7:1:2

TABLE II
The hyper-parameters used in different datasets.

Hyper-parameters	ETTh2	ETTm1	ETTm2	Traffic	Weather	Electricity	Exchange-Rate	ILI
Batch size	16	32	32	32	32	32	32	32
Number of Fourier modes	10	8	10	8	10	10	10	8
Number of U-Net hidden layer channels	[8,16]	[8,16]	[8,16]	[16,32]	[16,32]	[8,16]	[8,16]	[8,16]
Learning rate	5e-3	5e-3	5e-3	5e-3	5e-3	5e-3	5e-3	5e-3

TABLE III
Multivariate time series forecasting results. The length of the historical horizon is set as 36 for ILI and 96 for the others. The prediction lengths are {24,36,48,60} for ILI and {96, 192, 336, 720} for others. Black bold is state-of-the-art and underline is the second-best result.

Method		F-UNet		FEDformer		SCINet		Pyraformer		Autoformer		Informer		LogTrans	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ECL	96	0.203	**0.307**	**0.193**	0.308	0.205	0.312	0.386	0.449	0.201	0.317	0.274	0.368	0.258	0.357
	192	0.216	0.320	0.201	0.315	**0.197**	**0.308**	0.378	0.443	0.222	0.334	0.296	0.386	0.266	0.368
	336	0.228	0.329	0.214	0.329	**0.202**	**0.312**	0.376	0.443	0.231	0.338	0.300	0.394	0.280	0.380
	720	0.264	0.355	0.246	0.355	**0.234**	**0.338**	0.376	0.445	0.254	0.361	0.373	0.439	0.283	0.376
Traffic	96	0.612	**0.339**	**0.587**	0.366	0.651	0.393	0.867	0.468	0.613	0.388	0.719	0.391	0.684	0.384
	192	0.640	**0.355**	**0.604**	0.373	0.604	0.372	0.869	0.467	0.616	0.382	0.696	0.379	0.685	0.390
	336	0.663	**0.360**	0.621	0.383	**0.611**	0.375	0.881	0.469	0.622	0.387	0.777	0.420	0.734	0.408
	720	0.682	**0.367**	**0.626**	0.382	0.649	0.393	0.896	0.473	0.660	0.408	0.864	0.472	0.717	0.396
Exchange	96	**0.112**	**0.242**	0.148	0.278	0.142	0.249	1.748	1.105	0.197	0.323	0.847	0.752	0.968	0.812
	192	**0.212**	**0.329**	0.271	0.380	0.261	0.364	1.874	1.151	0.300	0.369	1.204	0.895	1.040	0.851
	336	**0.385**	**0.454**	0.460	0.500	0.457	0.490	1.943	1.172	0.509	0.524	1.672	1.036	1.659	1.081
	720	**1.040**	**0.771**	1.195	0.841	1.364	0.859	2.085	1.206	1.447	0.941	2.478	1.310	1.941	1.127
Weather	96	**0.181**	**0.232**	0.217	0.296	0.239	0.271	0.622	0.556	0.266	0.336	0.300	0.384	0.458	0.490
	192	**0.236**	**0.276**	0.276	0.336	0.283	0.303	0.739	0.624	0.307	0.367	0.598	0.544	0.658	0.589
	336	**0.289**	**0.312**	0.339	0.380	0.330	0.335	1.004	0.753	0.359	0.395	0.578	0.523	0.797	0.652
	720	**0.362**	**0.359**	0.403	0.428	0.400	0.379	1.420	0.934	0.419	0.428	1.059	0.741	0.869	0.675
ILI	24	**2.185**	**0.922**	3.228	1.260	2.782	1.106	7.394	2.012	3.483	1.287	5.764	1.677	4.480	1.444
	36	**1.766**	**0.852**	2.679	1.080	2.689	1.064	7.551	2.031	3.103	1.148	4.755	1.467	4.799	1.467
	48	**1.824**	**0.865**	2.622	1.078	2.324	0.999	7.662	2.057	2.669	1.085	4.763	1.469	4.800	1.468
	60	**2.345**	**0.955**	2.857	1.157	2.802	1.112	7.931	2.100	2.770	1.125	5.264	1.564	5.278	1.560
ETTh2	96	0.340	0.376	0.358	0.397	**0.312**	**0.355**	0.645	0.597	0.346	0.388	3.755	1.525	2.116	1.197
	192	0.431	0.425	0.429	0.439	**0.401**	**0.412**	0.788	0.683	0.456	0.452	5.602	1.931	4.315	1.635
	336	0.498	0.463	0.496	0.487	**0.413**	**0.432**	0.907	0.747	0.482	0.486	4.721	1.835	1.124	1.604
	720	0.498	0.482	**0.463**	**0.474**	0.490	0.483	0.963	0.783	0.515	0.511	3.647	1.625	3.188	1.540
ETTm1	96	0.378	0.402	0.379	0.419	**0.350**	**0.385**	0.543	0.510	0.505	0.475	0.672	0.571	0.600	0.546
	192	0.424	0.426	0.426	0.441	**0.382**	**0.400**	0.557	0.537	0.553	0.496	0.795	0.669	0.837	0.700
	336	0.455	0.443	0.445	0.459	**0.419**	**0.425**	0.754	0.655	0.621	0.537	1.212	0.871	1.124	0.832
	720	0.534	0.489	0.543	0.490	**0.494**	**0.463**	0.908	0.724	0.671	0.561	1.166	0.823	1.153	0.820
ETTm2	96	**0.199**	**0.279**	0.203	0.287	0.201	0.280	0.435	0.507	0.255	0.339	0.365	0.453	0.768	0.642
	192	**0.268**	**0.324**	0.269	0.328	0.283	0.331	0.730	0.673	0.281	0.340	0.533	0.563	0.989	0.757
	336	0.345	0.373	0.325	0.366	**0.318**	**0.352**	1.201	0.845	0.339	0.372	1.363	0.887	1.334	0.872
	720	0.464	0.439	**0.421**	**0.415**	0.439	0.423	3.625	1.451	0.422	0.419	3.379	1.388	3.048	1.328

is comparable to, and even superior to, the recent state-of-the-art Transformer-based FEDformer model, while Table V shows that F-UNet's inference time and average training time are significantly lower than those of FEDformer. Based on the results from Table III and Table V, it can be concluded that compared to the state-of-the-art Transformer-based multivariate time series prediction models, F-UNet achieves higher accuracy and efficiency with fewer parameters.

TABLE IV
Number of parameters for different models with input/output–96/96 settings.

	F-UNet	SCINet	Autoformer	Informer	FEDformer-f
Number of parameters	0.08M	0.27M	10.54M	11.33M	16.30M

TABLE V
Different models Inference Time and Average Training Time at ETTh2/Exchange-Rate with a given input of 96 steps and predict 96 steps.

Models	Average Training Time (s)	Inference Time (s)
F-UNet	5.32/2.42	2.328/1.199
FEDformer-f	76.75/49.11	8.934/4.503

V. Conclusion

This article proposes a multivariate time series prediction method based on non-Transformer models for the problem of multivariate time series prediction. Compared to Transformer-based models, this method can achieve more accurate prediction of future multivariate time series data with fewer parameter quantities through a U-shaped Fourier neural operator layer and a simple linear layer. The experimental results show that this method has high prediction accuracy and low time complexity, and can provide an effective solution for multivariate time series prediction. In the future, we will further develop multivariate time series prediction models based on non-Transformer methods, and gain a deeper understanding of the reasons why simple non-Transformer-based models outperform state-of-the-art Transformer-based models.

References

[1] R. Chen, S. Zhang, D. Li, Y. Zhang, F. Guo, W. Meng, D. Pei, Y. Zhang, X. Chen, and Y. Liu, "Logtransfer: Cross-system log anomaly detection for software systems with transfer learning," in *31st IEEE International Symposium on Software Reliability Engineering, ISSRE 2020, Coimbra, Portugal, October 12-15, 2020*, M. Vieira, H. Madeira, N. Antunes, and Z. Zheng, Eds. IEEE, 2020, pp. 37–47.

[2] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 11 106–11 115.

[3] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 22 419–22 430.

[4] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 2022, pp. 27 268–27 286.

[5] M. Liu, A. Zeng, M. Chen, Z. Xu, Q. Lai, L. Ma, and Q. Xu, "Scinet: time series modeling and forecasting with sample convolution and interaction," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5816–5828, 2022.

[6] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" *CoRR*, vol. abs/2205.13504, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2205.13504

[7] Z. Li, Z. Rao, L. Pan, and Z. Xu, "Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing," *CoRR*, vol. abs/2302.04501, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2302.04501

[8] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[9] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, ser. Lecture Notes in Computer Science, N. Navab, J. Hornegger, W. M. W. III, and A. F. Frangi, Eds., vol. 9351. Springer, 2015, pp. 234–241.

[11] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier neural operator for parametric partial differential equations," *arXiv preprint arXiv:2010.08895*, 2020.

[12] T. Kim, J. Kim, Y. Tae, C. Park, J. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[13] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Rethinking the stationarity in time series forecasting," *CoRR*, vol. abs/2205.14415, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2205.14415

[14] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[15] G. Lai, W. Chang, Y. Yang, and H. 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, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, K. Collins-Thompson, Q. Mei, B. D. Davison, Y. Liu, and E. Yilmaz, Eds. ACM, 2018, pp. 95–104.

[16] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.