Fractional-Order Differential Equation-Driven Transformer for Time Series Forecasting via Gauss-Jacobi Quadrature

Shengxiang Zhu[1], Jiuhong Luan[1], Zhonglian Wei[1], Chong He[1], Zhicheng Zhang[1], Junjie Hu[1,*]
[1]College of Computer Science, Sichuan University, Chengdu, China
Email: zhushengxiang@stu.scu.edu.cn, jiuhongluan@gmail.com, wei_zhonglian@163.com, 2023223049254@stu.scu.edu.cn, 2708310559@qq.com, hujunjie@scu.edu.cn
[*] *Corresponding author*

***Abstract*—Long-term time series forecasting faces the challenge of efficiently capturing long-range dependencies and adaptively forgetting historical information in fields such as finance, meteorology, energy, and healthcare. Traditional integer-order models, which rely on exponential decay mechanisms, struggle to retain long-range correlations and exhibit poor robustness to short-term noise. To address these issues, fractional-order differential equations (FDEs) naturally model non-Markovian memory effects through power-law decay mechanisms, enabling gradual forgetting of irrelevant short-term fluctuations while enhancing the memory of key historical patterns. This significantly improves the modeling capacity for complex dynamics. In this paper, we propose FODEformer, an architecture that integrates Gauss-Jacobi quadrature accelerated fractional-order differential equations into the self-attention framework. By utilizing the Caputo fractional-order derivative in its integral form and employing a variable transformation to map historical dependencies to a reasonable interval, FODEformer ensures robustness through fixed-node spectral approximation. Experimental results on 8 datasets show that FODEformer achieves significant performance improvements on multiple benchmark datasets, especially in terms of long-range dependency capturing and noise robustness, outperforming existing models.**

***Index Terms*—Long-term Time Series Forecasting, Fractional-Order Differential Equations, Self-Attention Mechanism, Gauss-Jacobi Quadrature**

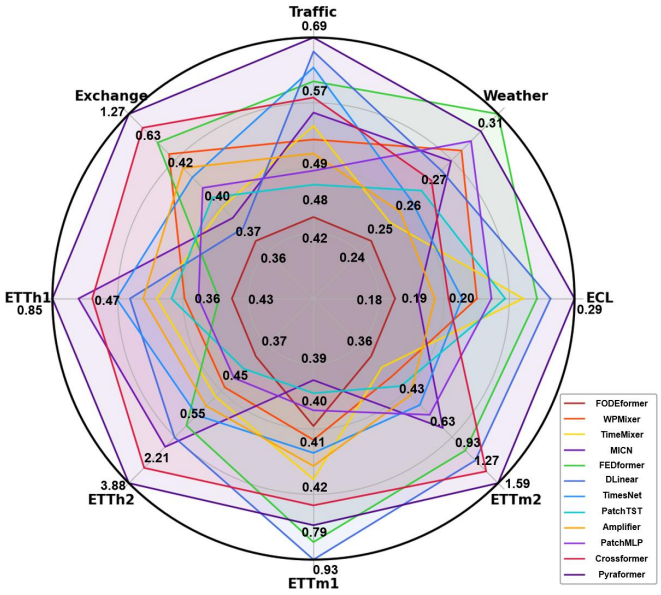

Fig. 1. The radar chart of the comparative performance of multivariate long-term forecasting models, evaluated by Mean Squared Error (MSE), across eight benchmark datasets. Each axis is independently normalized for clarity.

I. Introduction

Multivariate time series forecasting aims to accurately predict future trends by analyzing historical data, with widespread applications in financial market volatility analysis [1], weather forecasting [2], traffic flow optimization [3], and energy load scheduling [4]. As the scale and complexity of data continue to grow rapidly, traditional statistical models have gradually been replaced by modern machine learning and deep learning methods. In recent years, deep learning frameworks, particularly the Transformer [5] architecture, have successfully leveraged self-attention mechanisms to capture long-range dependencies in sequences, becoming the mainstream tool for handling long time series.

Although Transformer architectures hold great potential in capturing long-range dependencies, they still face many challenges when processing long sequences. First, the self-attention mechanism in standard Transformers tends to focus on recent historical data, limiting their ability to learn distant key patterns [6]. Second, the exponential decay characteristic of Transformer models makes it difficult to adaptively forget irrelevant past information, causing models to be susceptible to short-term noise and over-memorize irrelevant historical patterns, which leads to performance degradation in long-term forecasting tasks [7]. Thus, enhancing the long-term dependency capture ability of Transformers while reducing the interference from irrelevant patterns remains a key challenge in this field.

Traditional integer-order models perform stably in short-term forecasting and capture local patterns well, but they generally face bottlenecks when modeling long-range dependencies [8]. These models often rely on local recursion or fixed decay mechanisms, which struggle to balance long-range correlations and noise robustness across different time scales. For instance, standard autoregressive models and recurrent

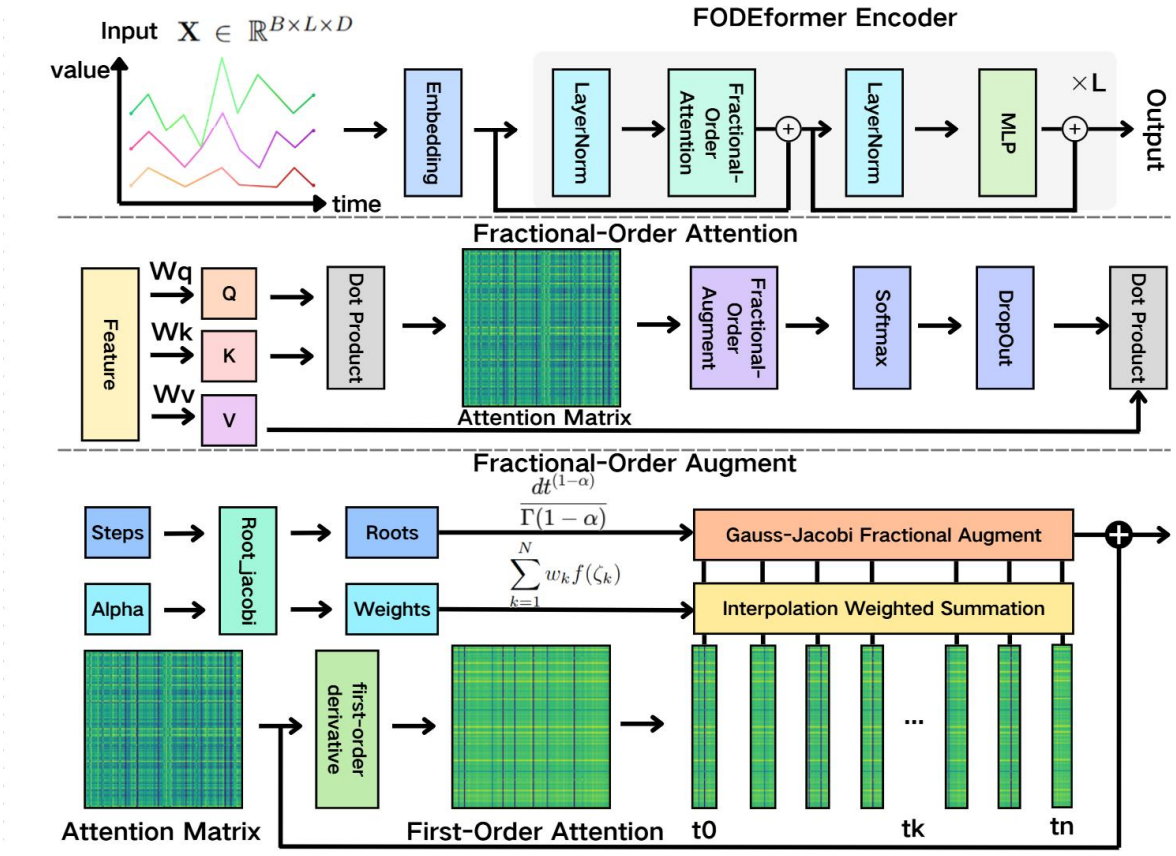

Fig. 2. The proposed FODEformer architecture.

neural networks (RNNs) [9] are limited by attention mechanisms, making them susceptible to short-term fluctuations and irrelevant patterns, resulting in performance degradation in long time series forecasting. To address this issue, fractional-order differential equations (FDEs) [10] provide a new mathematical framework for modeling long-range dependencies. Unlike integer-order models, FDEs possess a power-law decay property that effectively simulates non-Markovian memory effects, adapting to a broader range of time scale variations. FDEs achieve progressive forgetting of irrelevant short-term fluctuations while preserving memory of key long-range patterns, thereby improving robustness and accuracy in complex dynamic systems [11].

To efficiently implement the fractional-order mechanism, this paper introduces the Gauss-Jacobi quadrature [12] method as the core tool for fractional-order derivative robustness. The Gauss-Jacobi method maps non-local integrals to a finite interval through a variable transformation and employs fixed-node spectral approximation, which not only maintains high accuracy but also significantly reduces computational complexity [13]. This makes it feasible to seamlessly integrate the fractional-order mechanism into Transformer's self-attention computation. Through this innovation, the model achieves non-local, memory-dependent weighting of historical dependencies, enhancing its ability to capture long-range dependencies and mitigating the performance degradation caused by short-term noise in traditional self-attention mechanisms.

Based on this idea, we propose FODEformer, a novel fractional-order driven Transformer architecture. Our main contributions are as follows:

- By introducing fractional-order differential equations, we implement power-law memory weighting for attention scores, enhancing the Transformer's ability to model long-range dependencies in long-term time series forecasting, and improving long-term memory and robustness in dynamic systems.
- By integrating Gauss-Jacobi quadrature for spectrally accurate approximation of Caputo fractional derivatives into self-attention, FODEformer mathematically embeds rigorous power-law memory, enabling faithful modeling of non-Markovian dynamics and scale-invariant long-range correlations in real-world time series.
- Experimental results show that FODEformer outperforms traditional Transformer architectures in terms of prediction accuracy and long-range dependency modeling across multiple benchmark datasets, providing a more robust solution for long-term time series forecasting.

II. METHOD

A. Vanilla Transformer Self-Attention

The standard Transformer self-attention mechanism captures global dependencies within sequences through the query-key-value paradigm. Given an input time series $\mathbf{X} \in \mathbb{R}^{B\times L\times D}$, where B is the batch size, L is the sequence length, and D is the feature dimension, the queries, keys, and values for the h-th attention head are obtained via linear projections as follows:

$$\mathbf{Q}^{(h)} = \mathbf{X}\mathbf{W}_Q^{(h)}, \quad \mathbf{K}^{(h)} = \mathbf{X}\mathbf{W}_K^{(h)}, \quad \mathbf{V}^{(h)} = \mathbf{X}\mathbf{W}_V^{(h)}, \tag{1}$$

where $\mathbf{W}_Q^{(h)}, \mathbf{W}_K^{(h)}, \mathbf{W}_V^{(h)} \in \mathbb{R}^{D\times d}$ are learnable projection matrices, $d = D/H$ is the dimension per head, and H denotes the number of attention heads. The attention matrix for the h-th head is then computed as:

$$\mathbf{A}^{(h)} = \frac{\mathbf{Q}^{(h)}\mathbf{K}^{(h)\top}}{\sqrt{d}} + \mathbf{M}, \tag{2}$$

where $\mathbf{A}^{(h)} \in \mathbb{R}^{B\times L\times L}$ and $\mathbf{M}$ is an optional causal or padding mask used to enforce temporal causality or handle variable-length inputs. The output of the h-th head is obtained by aggregating the value vectors according to the attention weights:

$$\mathbf{O}^{(h)} = \mathrm{softmax}\left(\mathbf{A}^{(h)}\right)\mathbf{V}^{(h)}, \tag{3}$$

where $\mathbf{O}^{(h)} \in \mathbb{R}^{B\times L\times d}$. This formulation enables each time step to adaptively aggregate information from all positions in the sequence, thereby modeling long-range dependencies.

B. Gauss–Jacobi Quadrature for Fractional Derivatives

Traditional numerical methods for fractional derivatives, such as the Gürnwald-Letnikov difference [14], typically result in global uniform forgetting, which fails to effectively capture the scale-invariant memory effects commonly observed in complex systems. In contrast, power-law decay provides a more natural mechanism for describing anomalous diffusion and long-range correlations. To address this, we propose using Gauss-Jacobi quadrature to achieve spectral accuracy in approximating the Caputo fractional derivative, thus introducing a strict power-law forgetting characteristic that is highly consistent with non-Markovian dynamics. For the Caputo fractional derivative of order $\alpha \in (0, 1)$, its definition is given by:

$${}^{C}D_t^{\alpha}x(t) = \frac{1}{\Gamma(1-\alpha)}\int_0^t (t-\tau)^{-\alpha}\frac{dx(\tau)}{d\tau}\,d\tau, \tag{4}$$

where C represents the Caputo type, t is the current time, $x(t)$ is the function to be differentiated, τ is the dummy integration variable, and $\Gamma(\cdot)$ represents the Gamma function. This integral form encapsulates the cumulative memory effect from the initial time to the current time, though the kernel $(t-\tau)^{-\alpha}$ exhibits a singularity at the upper bound $\tau = t$. To facilitate high-precision numerical approximation, we introduce a linear variable transformation $\tau = t(1+\zeta)/2$, mapping the integration interval $[0, t]$ to the standard interval $[-1, 1]$, where ζ is a normalized coordinate introduced for numerical integration. After some straightforward calculations, we obtain the equivalent expression:

$${}^{C}D_t^{\alpha}x(t) = \frac{(t/2)^{1-\alpha}}{\Gamma(1-\alpha)}\int_{-1}^{1}(1+\zeta)^{-\alpha}\frac{dx(\frac{t}{2}(1+\zeta))}{d\frac{t}{2}(1+\zeta)}d\zeta. \tag{5}$$

Let $f(\zeta) = \frac{dx(\frac{t}{2}(1+\zeta))}{d\frac{t}{2}(1+\zeta)}$ represent the derivative term in the integrand. The above integral can then be rewritten as a weighted form:

$$\int_{-1}^{1} f(\zeta)\,(1+\zeta)^{-\alpha}\,d\zeta, \tag{6}$$

where the weight function is $(1+\zeta)^{-\alpha}$ and $(1-\zeta)^0$ is the constant 1. For this specific Jacobi-type weight function $(1-\zeta)^0(1+\zeta)^{-\alpha}$, we use the corresponding N-point Gauss-Jacobi quadrature rule to approximate the integral:

$$\int_{-1}^{1} f(\zeta)(1+\zeta)^{-\alpha}\,d\zeta \approx \sum_{k=1}^{N} w_k f(\zeta_k), \tag{7}$$

where $\zeta_k \in (-1, 1)$ and $w_k > 0$ are the k-th Gauss-Jacobi node and corresponding weight, both precomputed using the Golub-Welsch algorithm [15] based on the eigenvalue problem of the tridiagonal Jacobi matrix associated with the Jacobi polynomial. By substituting the above Gauss-Jacobi quadrature formula into the expression for the Caputo derivative, we obtain the final spectral-accuracy numerical approximation scheme:

$${}^{C}D_t^{\alpha}x(t) \approx \frac{(t/2)^{1-\alpha}}{\Gamma(1-\alpha)}\sum_{k=1}^{N} w_k f(\zeta_k). \tag{8}$$

The derivative term $f(\zeta_k)$ can be computed accurately using finite differences or automatic differentiation, thereby forming a computationally efficient numerical scheme that strictly adheres to the power-law memory model.

C. Fractional-Order Attention (FOA) Mechanism

The standard Transformer model's discrete softmax attention mechanism implicitly assumes exponential decay in attention scores, limiting its ability to model long-tailed dependencies and scale-invariant forgetting in time series. By integrating the Gauss-Jacobi quadrature method for fractional derivatives, the model can effectively embed power-law memory into attention scores, enhancing the capture of long-range dependencies while aligning with anomalous diffusion and non-Markovian processes. In the attention module, the original attention matrix $\mathbf{A}^{(h)} \in \mathbb{R}^{B\times L\times L}$ is enhanced by applying the Caputo fractional derivative along the query dimension, treating L as discrete time steps. The fractional-order adjustment is computed as follows, where $\Delta t = 1$ is the unit step size and the derivative term is approximated by finite differences:

$$\mathbf{A}'^{(h)}_{b,t,:} = \mathbf{A}^{(h)}_{b,t,:} + \frac{(\Delta t)^{1-\alpha}}{\Gamma(1-\alpha)}\sum_{k=1}^{N} w_k \frac{\partial \mathbf{A}^{(h)}_{b,:,:}}{\partial t_k}, \tag{9}$$

TABLE I

MULTIVARIATE LONG-TERM FORECASTING COMPARISON. ALL REPORTED RESULTS ARE AVERAGED OVER FOUR PREDICTION HORIZONS $H \in \{96, 192, 336, 720\}$ WITH A FIXED LOOKBACK LENGTH $T = 96$. THE BEST AND SECOND-BEST RESULTS ARE IN **BOLD** AND UNDERLINED, RESPECTIVELY.

Model	FODEformer (Ours)		WPMixer (2025) [16]		TimeMixer (2024) [17]		MICN (2023) [18]		FEDformer (2022) [19]		DLinear (2023) [20]		TimesNet (2022) [21]		PatchTST (2022) [22]		Amplifier (2025) [23]		PatchMLP (2025) [24]		Crossformer (2023) [25]		Pyraformer (2022) [26]	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	**0.181**	**0.279**	0.198	0.287	0.221	0.307	0.186	0.296	0.224	0.335	0.226	0.318	0.194	0.295	0.204	0.293	0.192	0.292	0.202	0.298	0.193	0.287	0.288	0.382
ETTh1	**0.439**	**0.444**	0.451	0.454	0.457	0.448	0.578	0.533	0.440	0.458	0.461	0.456	0.463	0.454	0.451	0.449	0.460	0.459	0.448	0.453	0.568	0.538	0.845	0.761
ETTh2	**0.379**	**0.405**	0.408	0.423	0.408	0.424	0.573	0.524	0.443	0.454	0.564	0.519	0.412	0.429	0.402	0.413	0.410	0.428	0.407	0.417	2.212	1.166	3.881	1.596
ETTm1	0.406	0.410	0.409	0.425	0.410	0.415	**0.396**	0.421	0.784	0.629	0.928	0.707	0.409	0.417	**0.396**	**0.408**	0.409	0.419	0.399	0.410	0.541	0.521	0.681	0.583
ETTm2	**0.287**	**0.330**	0.290	0.333	0.289	0.334	0.354	0.397	1.115	0.753	1.185	0.776	0.293	0.331	0.291	0.335	0.291	0.332	0.297	0.346	1.366	0.741	1.681	0.889
Weather	**0.247**	**0.275**	0.270	0.296	0.253	0.279	0.266	0.317	0.311	0.361	0.265	0.318	0.259	0.286	0.261	0.281	0.255	0.283	0.271	0.287	0.263	0.323	0.273	0.341
Traffic	**0.421**	**0.284**	0.537	0.355	0.544	0.359	0.546	1.048	0.613	0.379	0.672	0.418	0.625	0.332	0.480	0.310	0.527	0.329	0.489	0.319	0.570	0.307	0.688	0.393
Exchange	**0.361**	**0.401**	0.545	**0.401**	0.389	0.412	0.376	0.438	0.627	0.564	0.369	0.418	0.421	0.444	0.396	0.421	0.423	0.446	0.402	0.431	0.929	0.706	1.267	0.885

where the explanations of the equations are provided in Eq. 4 and Eq. 7, and $t_k = \frac{t}{2}(1 + \zeta_k)$ denotes the transformed evaluation point. This quadrature ensures spectral accuracy for smooth functions, with error decaying exponentially. The derivative $\frac{\partial \mathbf{A}^{(h)}_{b,:,:}}{\partial t_k}$ can be calculated via first-order forward differences:

$$\frac{\partial \mathbf{A}^{(h)}_{b,:,:}}{\partial t_k} \approx \frac{\mathbf{A}^{(h)}_{b,\lceil t_k \rceil,:} - \mathbf{A}^{(h)}_{b,\lfloor t_k \rfloor,:}}{\Delta t}, \tag{10}$$

where linear interpolation is used to handle the non-integer t_k, thus enforcing power-law forgetting: lower α values slow down decay and amplify long-range retention, while higher α values accelerate the prioritization of recent memories. Finally, the enhanced fractional attention is used in the softmax operation:

$$\mathbf{O} = \text{softmax}\left(\mathbf{A}'^{(h)}\right)\mathbf{V}^{(h)}, \tag{11}$$

integrated into the Transformer layer to enhance its ability to represent non-stationary time series.

III. EXPERIMENTS

A. Experimental settings

We conduct extensive experiments on eight widely-used time series datasets for long-term forecasting, including the Electricity [21], ETTh1, ETTh2, ETTm1, ETTm2 [27], Weather, Traffic, and Exchange [28] datasets as shown in Table II. These datasets cover a variety of domains such as electricity consumption, transformer temperature and load, weather conditions, traffic flow, and exchange rates, and are widely used in the literature for multivariate long-term forecasting tasks. All experiments are implemented in PyTorch 2.0 and Python 3.10 on Ubuntu 20.04 and conducted on two NVIDIA RTX 3090 24GB GPUs for 10 epochs with a batch size of 32. The Adam optimizer [29] is used with a learning rate of 0.001. To ensure fair comparisons, all baseline models are configured based on their official settings and evaluated using a unified evaluation framework. Each experiment is repeated three times with average results reported. Model performance is evaluated using Mean Square Error (MSE) and Mean Absolute Error (MAE).

TABLE II

STATISTICS OF THE DATASETS USED FOR LONG-TERM FORECASTING, INCLUDING THE NUMBER OF VARIABLES, PREDICTION LENGTH, AND DATASET SIZE (TRAIN/VAL/TEST).

Dataset	Dim	Frequency	Dataset Size
Electricity	321	hourly	(18317, 2633, 5261)
ETTh1	7	hourly	(8545, 2881, 2881)
ETTh2	7	hourly	(8545, 2881, 2881)
ETTm1	7	15-min	(34465, 11521, 11521)
ETTm2	7	15-min	(34465, 11521, 11521)
Weather	21	10-min	(36792, 5271, 10540)
Traffic	862	hourly	(12185, 1757, 3509)
Exchange	8	daily	(6334, 1888, 3780)

B. Experimental Results

We evaluate the proposed model, all results are averaged over four prediction horizons, $H \in \{96, 192, 336, 720\}$, with a fixed lookback length of $T = 96$. As shown in Table I, FODEformer achieves the best or second-best performance across the majority of datasets, demonstrating excellent results in both MSE and MAE, particularly in scenarios with significant long-range dependencies or complex cross-variable interactions. Compared to TimesNet [21], FODEformer reduces MSE and MAE by approximately 4.6% and 3.8% on the Weather dataset, respectively. On the Traffic dataset, FODEformer outperforms Amplifier [23], achieving a reduction of about 20.9% in MSE and 13.7% in MAE. In contrast, channel-independent methods such as DLinear [20] and PatchTST [22] show significantly poorer performance on datasets like Traffic and Exchange, where cross-variable coupling is strong, failing to capture the dynamic interdependencies among multiple variables effectively. These results demonstrate that by embedding Caputo fractional derivatives with precise Gauss-Jacobi quadrature, FODEformer naturally implements a power-law forgetting mechanism, significantly enhancing its ability to

TABLE III

PERFORMANCE COMPARISON OF VARIOUS MODELS WITH AND WITHOUT FOA ON MULTIPLE BENCHMARK DATASETS. RESULTS ARE AVERAGED OVER FOUR PREDICTION HORIZONS $H \in \{96, 192, 336, 720\}$ WITH LOOKBACK LENGTH $T = 96$. BEST RESULTS ARE IN **BOLD**. DETAILED PER-HORIZON RESULTS ARE PROVIDED IN THE SUPPLEMENTARY MATERIAL.

Dataset	Case	Transformer		iTransformer		FEDformer		Informer		Autoformer		Reformer		Pyraformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	Original	0.417	0.478	0.257	0.318	0.225	0.335	0.351	0.431	0.238	0.346	0.331	0.409	0.288	0.383
	w/FOA	**0.181**	**0.279**	**0.182**	**0.280**	**0.220**	**0.329**	**0.335**	**0.421**	**0.228**	**0.338**	**0.318**	**0.401**	**0.287**	**0.380**
ETTh1	Original	1.184	0.878	0.478	0.467	0.439	0.458	1.078	0.812	0.511	0.495	1.020	0.746	0.845	0.761
	w/FOA	**0.439**	**0.444**	**0.441**	**0.442**	**0.418**	**0.439**	**0.972**	**0.718**	**0.497**	**0.475**	**0.973**	**0.717**	**0.813**	**0.731**
ETTh2	Original	3.181	1.372	0.441	0.450	0.443	0.454	4.674	1.776	0.454	0.462	2.578	1.249	3.881	1.596
	w/FOA	**0.379**	**0.405**	**0.389**	**0.412**	**0.426**	**0.428**	**4.423**	**1.681**	**0.441**	**0.454**	**2.474**	**1.201**	**3.694**	**1.528**
ETTm1	Original	0.982	0.819	0.442	0.421	0.784	0.629	0.885	0.701	0.562	0.505	1.014	0.729	0.681	0.583
	w/FOA	**0.406**	**0.410**	**0.412**	**0.417**	**0.746**	**0.603**	**0.839**	**0.664**	**0.534**	**0.488**	**0.951**	**0.693**	**0.645**	**0.565**
ETTm2	Original	1.232	0.786	0.316	0.344	1.115	0.753	1.387	0.847	0.322	0.364	1.992	1.031	1.681	0.889
	w/FOA	**0.287**	**0.330**	**0.289**	**0.331**	**1.062**	**0.715**	**1.324**	**0.809**	**0.306**	**0.343**	**1.894**	**0.999**	**1.594**	**0.836**
Weather	Original	0.624	0.558	0.272	0.309	0.312	0.361	0.612	0.545	0.344	0.387	0.546	0.523	0.274	0.341
	w/FOA	**0.247**	**0.275**	**0.248**	**0.279**	**0.311**	**0.363**	**0.521**	**0.508**	**0.339**	**0.382**	**0.520**	**0.501**	**0.237**	**0.291**
Traffic	Original	0.664	0.364	0.555	0.377	0.614	0.379	0.949	0.534	0.660	0.410	0.713	0.396	0.688	0.393
	w/FOA	**0.421**	**0.284**	**0.423**	**0.286**	**0.602**	**0.371**	**0.871**	**0.491**	**0.592**	**0.364**	**0.701**	**0.387**	**0.684**	**0.381**
Exchange	Original	1.381	0.897	0.425	0.444	0.628	0.564	1.566	1.029	0.510	0.501	1.587	1.031	1.267	0.885
	w/FOA	**0.361**	**0.401**	**0.361**	**0.402**	**0.592**	**0.551**	**1.417**	**0.972**	**0.371**	**0.435**	**1.223**	**0.939**	**0.779**	**0.665**

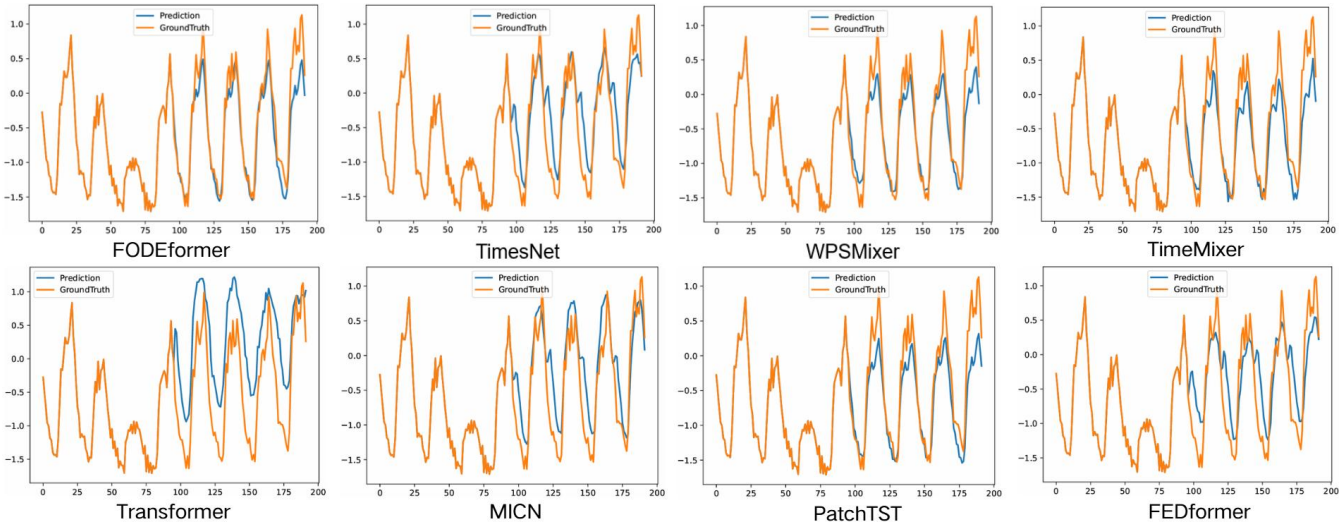

Fig. 3. Visualization of multivariate long-term forecasting results on the Electricity dataset with a prediction horizon $H = 96$ and a fixed lookback length $T = 96$. The orange line represents the ground truth, the blue line represents the prediction of the models.

model complex non-stationary temporal patterns. It outperforms existing state-of-the-art methods relying on local attention or linear projections, showing stronger robustness and overall predictive accuracy in multivariate long-term forecasting tasks.

C. Ablation Study

To evaluate the effectiveness of the FOA module, we conducted an ablation study comparing models with and without FOA on several benchmark datasets as shown in Table III. Results show that FOA, as an effective enhancement, consistently improves performance in time series forecasting. On stable datasets like Electricity and Weather, FOA reduces MSE by over 50% and improves MAE by 42%. iTransformer [30] also saw nearly a 30% reduction in error, demonstrating FOA's broad applicability. The benefits of FOA are even more pronounced in challenging, long-sequence or volatile scenarios. For example, on the ETTh2 dataset, FOA reduces the MSE from 3.181 to 0.379, an 88% drop. Similar improvements were observed on Exchange, where MSE decreased by nearly 74%. Even in models with strong self-correlation mechanisms, such as FEDformer [19] and Autoformer [31], FOA further reduces error by 1%-5%, especially in longer forecasting horizons. These results confirm FOA's ability to model complex multivariable dynamics, making it a valuable enhancement for time series forecasting models.

D. Visualization Analysis

Visualization as shown in Figure 3 reveals that FODEformer's predictions closely match the ground truth, maintain-

ing precise phase and amplitude alignment in high-frequency oscillations at later sequence segments. This stems from the fractional-order attention's power-law forgetting, which adaptively preserves critical historical information and mitigates exponential decay in standard Transformers. In contrast, TimeNet [21], WPSMixer [16], and TimeMixer [17] exhibit phase shifts, amplitude underestimation, and accumulating errors over longer horizons, while Transformer and MICN [18] show pronounced smoothing and sluggish responses to extrema. PatchTST [20] and FEDFormer [19] partially alleviate long-range issues but fail to fully capture power-law persistent correlations, resulting in deviations in complex oscillations. These comparisons confirm FODEformer's clear superiority in long-term prediction fidelity.

IV. CONCLUSION

In this paper, we introduce FODEformer, a novel time series forecasting model that incorporates fractional-order derivatives into the attention mechanism. By embedding Caputo fractional derivatives with precise Gauss-Jacobi quadrature, FODEformer naturally implements a power-law forgetting mechanism, significantly enhancing its ability to model complex non-stationary temporal patterns. Experimental results on multiple benchmark datasets demonstrate that FODEformer outperforms existing state-of-the-art methods in terms of prediction accuracy and long-range dependency modeling, providing a more effective solution for long-term time series forecasting tasks.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 62476184.

REFERENCES

[1] W. Li and K. E. Law, "Deep learning models for time series forecasting: A review," *IEEE Access*, vol. 12, pp. 92 306–92 327, 2024.

[2] Z. Karevan and J. A. Suykens, "Transductive lstm for time-series prediction: An application to weather forecasting," *Neural Networks*, vol. 125, pp. 1–9, 2020.

[3] İ. Arslan, C. H. Dağıdır, and Ü. Işlak, "An overview of time series point and interval forecasting based on similarity of trajectories, with an experimental study on traffic flow forecasting," *arXiv preprint arXiv:2309.10613*, 2023.

[4] L. Sun, Y. Lin, N. Pan, Q. Fu, L. Chen, and J. Yang, "Demand-side electricity load forecasting based on time-series decomposition combined with kernel extreme learning machine improved by sparrow algorithm," *Energies*, vol. 16, no. 23, p. 7714, 2023.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[6] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," *arXiv preprint arXiv:2202.07125*, 2022.

[7] A. Szałata, K. Hrovatin, S. Becker, A. Tejada-Lapuerta, H. Cui, B. Wang, and F. J. Theis, "Transformers in single-cell omics: a review and new perspectives," *Nature methods*, vol. 21, no. 8, pp. 1430–1443, 2024.

[8] L. Su, X. Zuo, R. Li, X. Wang, H. Zhao, and B. Huang, "A systematic review for transformer-based long-term series forecasting," *Artificial Intelligence Review*, vol. 58, no. 3, p. 80, 2025.

[9] L. R. Medsker, L. Jain *et al.*, "Recurrent neural networks," *Design and applications*, vol. 5, no. 64-67, p. 2, 2001.

[10] O. Brandibur, R. Garrappa, and E. Kaslik, "Stability of systems of fractional-order differential equations with caputo derivatives," *Mathematics*, vol. 9, no. 8, p. 914, 2021.

[11] M. Ayaz, T. Ali, M. Hijji, I. Baig, T. Alhmiedat, and E.-H. M. Aggoune, "Predictive modeling of complex networks using deep learning and fractional dynamics," *AIMS MATHEMATICS*, vol. 10, no. 11, pp. 26 717–26 743, 2025.

[12] A. Gil, J. Segura, and N. M. Temme, "Noniterative computation of gauss–jacobi quadrature," *SIAM Journal on Scientific Computing*, vol. 41, no. 1, pp. A668–A693, 2019.

[13] S. Jahanshahi, E. Babolian, D. F. Torres, and A. Vahidi, "A fractional gauss–jacobi quadrature rule for approximating fractional integrals and derivatives," *Chaos, Solitons & Fractals*, vol. 102, pp. 295–304, 2017.

[14] F. M. Atici, S. Chang, and J. Jonnalagadda, "Grünwald-letnikov fractional operators: From past to present," *Fract. Differ. Calc*, vol. 11, no. 1, pp. 147–159, 2021.

[15] G. Meurant and A. Sommariva, "Fast variants of the golub and welsch algorithm for symmetric weight functions in matlab," *Numerical Algorithms*, vol. 67, no. 3, pp. 491–506, 2014.

[16] M. M. N. Murad, M. Aktukmak, and Y. Yilmaz, "Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 18, 2025, pp. 19 581–19 588.

[17] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. Zhou, "Timemixer: Decomposable multiscale mixing for time series forecasting," *arXiv preprint arXiv:2405.14616*, 2024.

[18] H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, and Y. Xiao, "Micn: Multi-scale local and global context modeling for long-term series forecasting," in *The eleventh international conference on learning representations*, 2023.

[19] 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*. PMLR, 2022, pp. 27 268–27 286.

[20] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.

[21] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[22] Y. Nie, "A time series is worth 64words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[23] J. Fei, K. Yi, W. Fan, Q. Zhang, and Z. Niu, "Amplifier: Bringing attention to neglected low-energy components in time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 11, 2025, pp. 11 645–11 653.

[24] P. Tang and W. Zhang, "Unlocking the power of patch: Patch-based mlp for long-term time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 12, 2025, pp. 12 640–12 648.

[25] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The eleventh international conference on learning representations*, 2023.

[26] 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 *# PLACEHOLDER_PARENT_METADATA_VALUE#*, 2022.

[27] 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 *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[28] G. Lai, W.-C. 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*, 2018, pp. 95–104.

[29] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[31] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.