PP-Mixer: A Multi-Scale Pooled Patch MLP Framework for Long-Term Time Series Forecasting

1st Shuo Zhang
School of Software Technology
Zhejiang University
Ningbo, China
shuo016719@gmail.com

2nd Feihu Yan
School of Computer Science and Technology
Zhejiang University
Hangzhou, China
yanfeihu@zju.edu.cn

3rd Wenlong Xu
School of Software Technology
Zhejiang University
Ningbo, China
xuwenlong@zju.edu.cn

4th Yanliang Tan
School of Software Technology
Zhejiang University
Ningbo, China
22451182@zju.edu.cn

5th Junlin An
School of Software Technology
Zhejiang University
Ningbo, China
22451110@zju.edu.cn

6th Xuelin Cheng*
School of Software Technology
Zhejiang University
Ningbo, China
cxlin@zju.edu.cn

***Abstract*—Long-term time series forecasting(LSTF) is crucial for applications such as energy management and traffic forecasting, yet remains challenging due to complex multi-scale temporal dependencies, including long-term trends and mixed periodic patterns. Recently patch-based methods improve local temporal modeling by partitioning sequences into subseries, but fixed-length patching struggles to adapt to diverse and varying periodic structures in real-world data. To address this issue, we propose PP-Mixer, a fully MLP-based framework for LTSF. PP-Mixer employs a Fourier-guided multi-scale pooled patch embedding to adaptively construct multi-scale representations aligned with dominant temporal periods. A multi-scale decomposition mixer module then models and integrates seasonal and trend components across scales, while a dual-dimension mixing block enables efficient temporal modeling and feature interaction within each scale. Experiments on eight real-world benchmarks show that PP-Mixer consistently outperforms Transformer-based and MLP-based baselines, with ablation and visualization analyses confirming the effectiveness of its period-aware and multi-scale design.**

***Index Terms*—Time series forecasting, Multi-Scale modeling, Period-aware patching,Deep learning**

I. Introduction

Long-term time series forecasting plays a crucial role in a wide range of real-world applications, including energy management, traffic monitoring, finance, and climate analysis [1]. Accurate long-horizon forecasting enables early identification of future trends and anomalies, which is essential for informed decision-making and effective resource allocation. However, LTSF remains challenging because real-world time series exhibit complex and intertwined temporal dynamics across multiple scales, where short-term fluctuations, seasonal patterns, and long-term trends often coexist and interact [2].

Transformer-based models have achieved strong performance by capturing long-range dependencies through self-attention, but their quadratic computational complexity limits scalability on long sequences [3]. This has motivated increasing interest in MLP-based architectures for long-term time series forecasting, which avoid attention mechanisms while remaining competitive on many benchmarks [4]. However, purely MLP-based designs often face an expressivity bottleneck: without appropriate representations and interaction mechanisms, they struggle to effectively model the heterogeneous temporal patterns present in real-world data, especially under long forecasting horizons.

A central issue lies in how time series should be tokenized and represented. A single timestamp in time series can simultaneously reflect multiple overlapping contexts (e.g., daily, weekly, and seasonal cycles) and is therefore noisy and semantically weak as an atomic modeling unit [5], [6]. Patch-based methods address this by aggregating consecutive points into segments, improving locality modeling [7]. Nevertheless, most existing approaches employ fixed patch lengths or manually specified scale sets, implicitly assuming that temporal structures are uniform across datasets and variables [4].

In practice, temporal patterns are neither uniform nor clean. Time series frequently exhibit data-dependent dominant periods that vary across domains and variables, making fixed patching misaligned with intrinsic periodic structure [8], [9]. Besides, real-world sequences are often noisy and redundant at the point level [10]; even when patch boundaries are properly aligned, directly tokenizing raw segments can result in noisy and unreliable representations, thereby impairing long-range forecasting stability.These observations suggest that an effective forecasting framework should not only align segmentation with dominant periods, but also construct compact and robust representations that suppress redundancy and noise.

Beyond representation, another key challenge lies in modeling interactions: accurate LTSF requires both cross-scale and within-scale modeling [11]. Different temporal scales capture complementary characteristics of time series, where finer scales reflect short-term fluctuations and high-frequency dy-

Corresponding author

namics, while coarser scales characterize long-term trends and low-frequency patterns [12]. A forecasting model must therefore enable coherent information integration across scales, rather than treating each scale independently.Meanwhile, within each scale, strong performance in multivariate forecasting further depends on effectively modeling two types of dependencies: temporal dependencies along the time axis and cross-variable correlations along the feature axis [13]. Although attention-based mechanisms are powerful in capturing generic pairwise interactions, their quadratic complexity and token-level coupling introduce substantial overhead when applied to long sequences and multi-scale representations. More importantly, explicit pairwise modeling may be unnecessary when the dominant dependencies are structured along orthogonal temporal and feature dimensions.This observation motivates the adoption of lightweight, dimension-wise interaction mechanisms. By decoupling temporal mixing from feature mixing, pure MLP-based designs can efficiently capture long-range temporal patterns and cross-variable relationships with linear complexity, while offering greater scalability and simplicity. Such dimension-wise mixing provides a natural and effective inductive bias for modeling intra-scale interactions in long-term time series forecasting.

In this paper, we revisit the above challenges and introduce **PP-Mixer**, a concise and fully MLP-based framework for long-term time series forecasting. PP-Mixer is designed to address the misalignment between fixed representations and the intrinsic temporal patterns of real-world time series, while retaining the efficiency advantages of MLP-based architectures. By jointly modeling multi-scale representations and dimension-wise interactions, PP-Mixer aims to capture both cross-scale dynamics and complex intra-scale dependencies in a unified and efficient manner.Specifically, our contributions can be summarized as follows:

- We identify the limitation of fixed patching in modeling real-world multi-periodic time series and propose a multi-scale pooled patch embedding module that produce compact and robust multi-scale representations.
- We design a decomposable mixing mechanism within a fully MLP-based architecture that jointly models cross-scale interactions via seasonal–trend integration and intra-scale dependencies through efficient temporal and cross-variable mixing.
- Extensive experiments on long-term forecasting benchmarks show that PP-Mixer consistently surpasses strong Transformer and MLP baselines,and further ablation and visualization studies provide detailed evidence of its effectiveness.

II. Related work

A. Time Series Forecasting Models

Time series forecasting has advanced rapidly in recent years, with models spanning Transformer-based, CNN-based, and MLP-based architectures. Transformer-based methods are widely adopted for long-range modeling: PatchTST [7] introduces patching to enhance local semantics, iTransformer [14] inverts temporal and variate dimensions to better capture multivariate correlations, and Crossformer [15] explicitly models cross-dimension dependencies via two-stage attention. CNN-based approaches leverage convolutional inductive biases for efficiency, among which MICN [16] combines multi-scale local and global convolution, while TimesNet [6] exploits multi-periodicity by transforming one-dimensional time series into two-dimensional representations and modeling intra-period and inter-period variations with two-dimensional convolutions. Recently, MLP-based models have shown strong competitiveness: TimeMixer [4] proposes a fully MLP-based multiscale mixing framework with decomposition, and PatchMLP [17] demonstrates that patching alone can endow MLPs with competitive forecasting ability. In parallel, linear models such as DLinear [18] and RLinear [19] revisit linear mapping and reveal its effectiveness in capturing long-term trends and periodic patterns. Overall, existing methods explore patching, multiscale modeling, and periodicity from different perspectives, but most rely on fixed partitions or predefined scales, limiting their adaptability to dominant periodic structures in real-world time series.

B. Patch-based Methods for Time Series

Patch-based methods provide an effective means of enhancing locality modeling and have been widely adopted across domains. Vision Transformer [20] first proposed partitioning images into patches, which are then tokenized and fed into a Transformer encoder to capture hierarchical local and global information. The same idea has been introduced into time series forecasting to improve long-sequence modeling. Early work such as PatchTST [7] divides time series into fixed-length patches, which are then input into a Transformer encoder to maintain long-range dependencies while strengthening the capture of local patterns. Subsequently, PatchMLP [17] manually selects a set of patch lengths for multi-scale modeling, extending patching ideas into MLP-based architectures. TimeMixer [4] adopts a set of power-of-two patch lengths and incorporates seasonal–trend decomposition, effectively modeling cross-scale temporal dynamics. Collectively, these methods demonstrate that patching alleviates the limitations of single-point representations in time series forecasting and significantly enhances the ability of models to capture complex temporal patterns.

C. MLP-Based Mixing Architectures for Time Series Forecasting

Recent studies have investigated fully MLP-based architectures as efficient alternatives to attention for sequence modeling, with MLP-Mixer [21] serving as a representative paradigm that performs dimension-wise mixing via stacked MLP layers. This design is particularly suitable for time series forecasting, where dependencies naturally arise along two orthogonal axes: the temporal dimension within each variable and the feature dimension across variables. Temporal mixing facilitates long-range dependency modeling and periodic pattern extraction, while feature mixing captures cross-variable correlations.

Building upon this philosophy, several works have adapted MLP-style mixing to long-term forecasting, such as U-Mixer [22] with hierarchical representations and TS-Mixer [23] with patch-wise processing, demonstrating that pure MLP-based dual-dimension mixing can provide an effective and efficient foundation for time series forecasting.

III. Methodology

The task of time series forecasting can be formally defined as follows. Given the historical observation sequence $\mathcal{X} = \{x_1, \cdots, x_L \mid x_i \in \mathbb{R}^M\}$, the objective is to predict the future sequence $\mathcal{Y} = \{x_{L+1}, \cdots, x_{L+T} \mid x_i \in \mathbb{R}^M\}$, where L and T denote the lengths of the input and output sequences, respectively, and M is the number of variables in the multivariate time series. The forecasting problem is thus to learn a mapping function $f : \mathbb{R}^{L\times M} \to \mathbb{R}^{T\times M}$ such that the predicted sequence $\hat{\mathcal{Y}} = f(\mathcal{X})$ approximates the ground truth $\mathcal{Y}$.

As discussed above, A key challenge in time series forecasting lies in modeling the complex and intertwined temporal dynamics that manifest across multiple scales in real-world sequences. To address this, we propose PP-Mixer, a novel architecture designed to improve forecasting accuracy by jointly capturing cross-scale dependencies and intra-scale temporal correlations. As illustrated in Figure 1, PP-Mixer comprises three main components: a multi-scale pooled patch embedding (MPPE) block, a decomposition mixer block, and a dual-dimension mixing (DDM) block. The details of each component are described in the following subsections.

A. Multi-Scale Pooled Patch Embedding Block

In time series forecasting, a single timestamp often provides limited semantic information, making it difficult to capture complex temporal dynamics. Patch-based representations mitigate this issue by aggregating consecutive time points into higher-level tokens. However, existing approaches typically adopt either a fixed patch length or a manually predefined set of patch sizes. Such handcrafted designs are not adaptive to the intrinsic periodic patterns of real-world time series, and often fails to adapt to the inherent multiple periodic patterns present in different datasets. Consequently,these methods have limited capacity to jointly capture short-term temporal variations and long-term periodic or trend structures.

To address this limitation, we propose a **Multi-Scale Pooled Patch Embedding Block**, which constructs adaptive multi-scale representations guided by the intrinsic periodicity of the input sequence.Specifically, we first apply the Fast Fourier Transform [24] (FFT) to identify several dominant frequencies, and derive their corresponding periods. These periods are then used to determine scale-specific patch lengths, ensuring that the partitioning process is aligned with the dominant temporal cycles of the data.Subsequently, we perform FFT-based downsampling to obtain period-aware multi-scale representations. Figure 2 depicts the architecture of this module. For each dominant period, the input sequence is divided into non-overlapping, cycle-aligned patches, within which average pooling is applied to compress redundant observations and suppress noise. Pooled patch provides a compact yet robust summary of temporal dynamics at the corresponding scale. By performing pooled patch embedding across multiple dominant periods, the model learns scale-specific representations that preserve intra-cycle temporal variations within short dominant periods, while maintaining stable and coherent representations of seasonal and trend structures across longer temporal horizons.

Formally, given an input time series $\mathbf{X} \in \mathbb{R}^{T\times C}$, we compute the amplitude spectrum via FFT and select the top-K dominant frequencies:

$$\mathbf{A} = \mathrm{Amp}(\mathrm{FFT}(\mathbf{X})), \tag{1}$$

$$f_1, \dots, f_K = \underset{f_i \in 1,\dots,\lfloor T/2 \rfloor}{\mathrm{argTopK}} (\mathbf{A}), \tag{2}$$

with the corresponding periods $p_k = T/f_k$. For each p_k, pooled patch representations $\mathbf{Z}^{(k)} \in \mathbb{R}^{N_k\times C}$ are obtained by average pooling over $N_k = \lfloor T/p_k \rfloor$ non-overlapping patches, and projected into a d-dimensional embedding space:

$$\mathbf{E}^{(k)} = \mathbf{Z}^{(k)}\mathbf{W}_e, \quad \mathbf{W}_e \in \mathbb{R}^{C\times d}. \tag{3}$$

Finally, embeddings from all scales are collected together with the raw sequence representation:

$$\mathcal{L} = \mathbf{E}^{(0)}, \mathbf{E}^{(1)}, \dots, \mathbf{E}^{(K)}, \quad \mathbf{E}^{(0)} = \mathbf{X}\mathbf{W}_e. \tag{4}$$

Including the original sequence preserves fine-grained short-term variations that may be smoothed by pooling operations.

B. Decomposition Mixer Block

Real-world time series exhibit complex temporal dynamics arising from the interaction of seasonal patterns, long-term trends, and irregular variations.Decomposition techniques are widely adopted to disentangle these components [5], [25], thereby reducing modeling complexity and enabling models to focus on substructures with distinct statistical properties, particularly those that are more predictable. However, in the presence of noise and mixed nonlinear dynamics, directly applying decomposition to raw observations can be unstable, often leading to component entanglement and degraded decomposition quality [26].To obtain more reliable and predictable decompositions, we perform decomposition on multi-scale pooled patch embeddings rather than on the original time series. These embeddings effectively smooths high-frequency noise and compresses redundant information, yielding more compact and stable representations. Decomposing such representations facilitates a clearer separation between seasonal variations and long-term trends, allowing subsequent modeling stages to focus on more predictable components at different scales.

Given a scale-specific embedding $\mathbf{E}^{(k)} \in \mathbb{R}^{N_k\times d}$, we apply a moving-average operator to decompose it into trend and seasonal components:

$$\mathbf{T}^{(k)} = \mathrm{MA}\left(\mathbf{E}^{(k)}\right), \qquad \mathbf{S}^{(k)} = \mathbf{E}^{(k)} - \mathbf{T}^{(k)}. \tag{5}$$

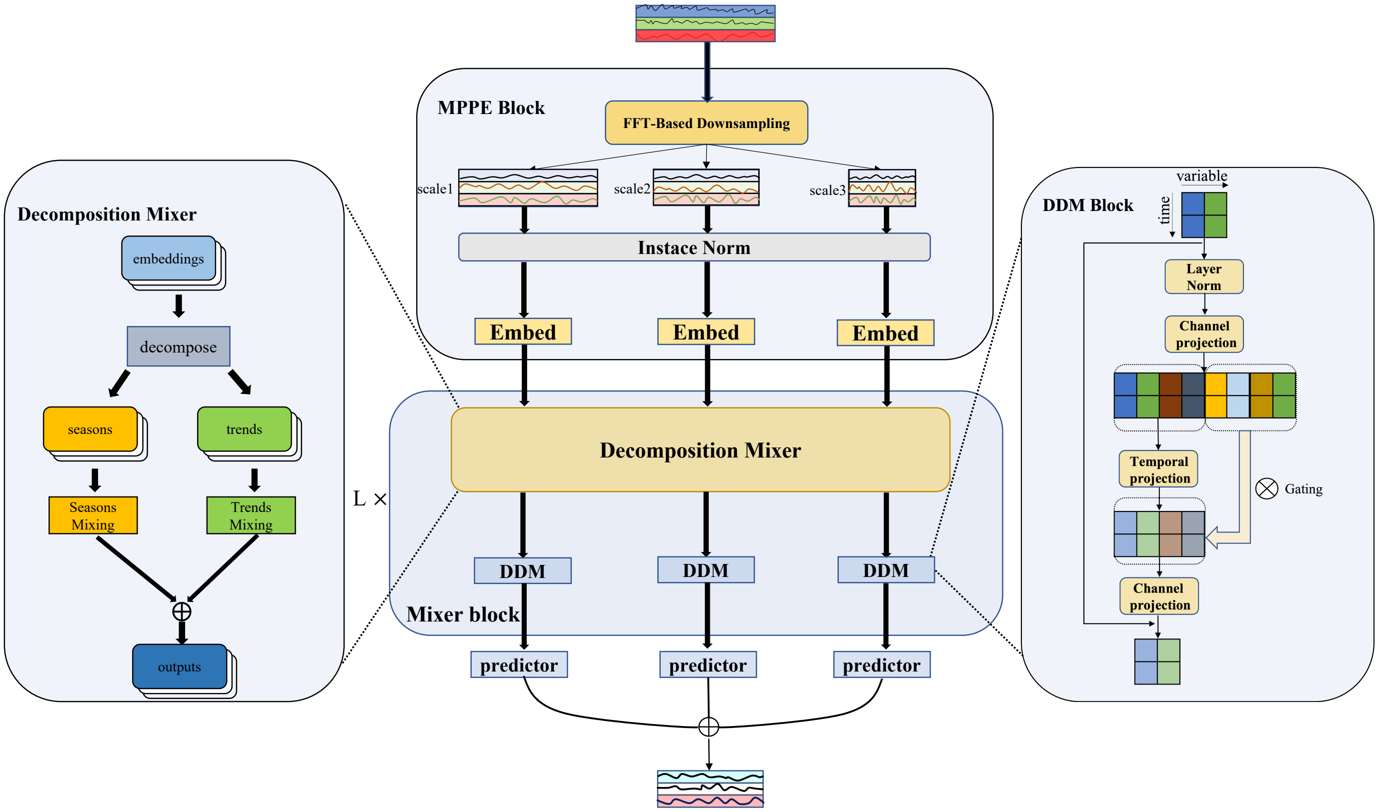

Fig. 1. Overall structure of the proposed PP-Mixer

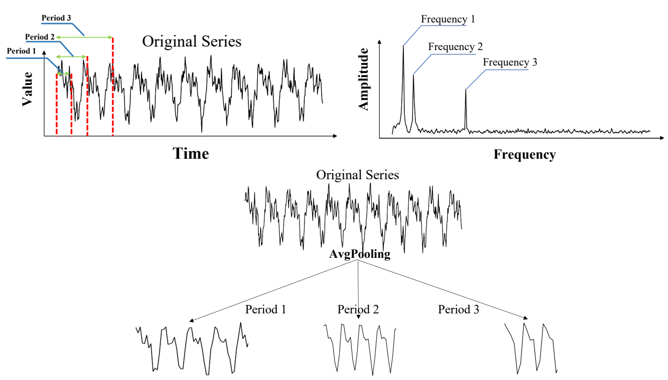

Fig. 2. Overall structure of the FFT-Based Downsampling

The resulting components are integrated across scales through a directional cross-scale mixing mechanism. Coarse-scale periodicity is inherently compositional, making it natural to aggregate information from finer temporal resolutions. We therefore adopt a bottom-up aggregation strategy, progressively propagating fine-scale periodic cues to coarser representations:

$$\tilde{\mathbf{S}}^{(k+1)} = \mathbf{S}^{(k+1)} + \phi\Big(\mathbf{S}^{(k)}\Big), \tag{6}$$

where $\phi(\cdot)$ denotes a learnable projection from finer to coarser scales. In contrast, trend components capture long-term structural dynamics that are dominated by low-frequency variations rather than local fluctuations. Such global trends are typically more reliably estimated at coarser temporal resolutions, where short-term noise is suppressed. Motivated by this property, we propagate trend information in a top-down manner, injecting coarse-scale trend context to guide and regularize finer-scale representations:

$$\tilde{\mathbf{T}}^{(k)} = \mathbf{T}^{(k)} + \psi\Big(\mathbf{T}^{(k+1)}\Big), \tag{7}$$

where $\psi(\cdot)$ transfers global trend guidance from coarser to finer scales. Finally, the enhanced seasonal and trend components are recombined at each scale to form the output representation:

$$\mathbf{H}^{(k)} = \tilde{\mathbf{S}}^{(k)} + \tilde{\mathbf{T}}^{(k)}. \tag{8}$$

By performing decomposition on multi-scale pooled embeddings rather than raw observations, the proposed module yields smoother and more interpretable seasonal and trend components at each resolution. The subsequent directional mixing further enables complementary propagation of high-frequency periodic details and long-term structural information across scales. Collectively, these components enable coherent integration of local variations and global dynamics, leading to more expressive representations and improved forecasting performance.

C. Dual-Dimension Mixing Block

Given a scale-specific representation $H^{(k)} \in \mathbb{R}^{N_k \times d}$, the proposed **Dual-Dimension Mixing (DDM) Block** is designed to fuse dependencies along both the temporal dimension (intra-variable) and the feature dimension (inter-variable). To this end, we implement DDM with a lightweight gated design that first performs channel mapping and branching, followed by temporal and feature mixing in a sequential manner.

a) Channel Mapping & Branching.: At each scale, the input $\mathbf{H}^{(k)}$ is first normalized and mapped by a pointwise channel projection that expands features to $4d$. We then evenly split the projected features into two branches along the channel dimension, $[\mathbf{U}^{(k)}, \mathbf{V}^{(k)}] = \mathbf{H}^{(k)}\mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{d\times 4d}$ and $\mathbf{U}^{(k)}, \mathbf{V}^{(k)} \in \mathbb{R}^{N_k\times 2d}$. The $\mathbf{V}^{(k)}$ branch serves as a dynamic gate via a GELU function, while $\mathbf{U}^{(k)}$ carries the features to be temporally mixed.

b) Intra-Variable (Temporal) Mixing: To model dependencies along the temporal dimension within each variable, we apply a lightweight linear projection along the time axis to the $\mathbf{U}^{(k)}$ branch, yielding $\widehat{\mathbf{U}}^{(k)}$.The gated temporal output is then formed by element-wise modulation $\mathbf{G}^{(k)} = \widehat{\mathbf{U}}^{(k)} \odot \sigma(\mathbf{V}^{(k)})$, where $\sigma(\cdot)$ denotes the GELU activation function, which adaptively controls how much temporally mixed information is passed forward.

c) Inter-Variable (Feature) Mixing: Temporal cues alone are insufficient under strong cross-variable interactions. We therefore aggregate features across variables using a channel projection $\mathcal{M}_C$ that is shared across tokens, yielding $\widetilde{\mathbf{G}}^{(k)} = \mathcal{M}_C(\mathbf{G}^{(k)})$. This step integrates co-movement patterns among variables without incurring quadratic complexity.

d) Residual Connections: The residual shortcut in DDM not only stabilizes optimization and alleviates vanishing gradients, but also complements the gating mechanism by preserving the original temporal signal, thereby ensuring information fidelity across scales.

Overall, the embeddings produced by the MPPE block are processed through a stack of L Mixer blocks, each consisting of a Decomposition Mixer and a DDM block. After these transformations, scale-specific projections are applied at each scale to generate individual forecasts, which are then aggregated by summation to produce the final prediction.

IV. Experiments

We conduct a comprehensive evaluation of the proposed PP-Mixer across multiple long-term time series forecasting tasks. This assessment not only verifies the generalization capability of our framework on diverse domains, but also provides further insights into the effectiveness of each component within PP-Mixer when applied to LSTF.

a) Datasets: We evaluate PP-Mixer on eight widely used LSTF benchmark datasets, including ETT (ETTh1, ETTh2, ETTm1, ETTm2), Electricity, Traffic, Weather, and Solar-Energy. These datasets span different domains such as energy consumption, transportation, meteorology, and renewable energy, ensuring a comprehensive evaluation of model generality.

b) Baselines: We compare PP-Mixer with nine representative baselines, including iTransformer [14], Crossformer [15], PatchMLP [17], PatchTST [7], TimeMixer [4], DLinear [18], RLinear [19], MICN [16], and TimesNet [6], covering Transformer-based, MLP-based, and CNN-based forecasting models.

c) Experimental Setup: Forecasting horizons are set to $\{96, 192, 336, 720\}$ steps. We evaluate the predictive performance using Mean Squared Error (MSE) and Mean Absolute Error (MAE), where lower values indicate better accuracy.

d) Implementation Details: All experiments are implemented in PyTorch and conducted on a single NVIDIA RTX 4090 GPU. We train the model using the mean squared error (MSE) loss throughout the optimization process.

A. Main results

To evaluate the forecasting capability of PP-Mixer in comparison with other time series forecasting models, we conduct extensive experiments in this section. Table I summarizes the long-term forecasting performance evaluated using the MSE and MAE metrics, where lower values indicate better predictive accuracy. The best results are highlighted in **bold**, while the second-best results are underlined. Compared with other forecasting models, PP-Mixer consistently achieves the best overall performance in long-term forecasting tasks. In particular, on the Electricity and Solar-Energy datasets, PP-Mixer reduces MSE by **4.5%** and **5.9%** compared with the patch-based MLP model PatchMLP, and by **12.4%** and **15.4%** compared with TimeMixer, respectively. Furthermore, relative to the best-performing Transformer-based model iTransformer, PP-Mixer achieves overall reductions of **6.0%** in MSE and **3.4%** in MAE.

B. Ablation studies

To investigate the impact of each module in PP-Mixer on model performance, we conduct a series of ablation studies. The ablation study results are summarized in Table II. In these experiments, we systematically ablate individual components. Specifically, ② removes the decomposition mixer block and directly feeds the embeddings into the DDM block for prediction. ⑥⑦⑧⑨ removes the FFT-based patch partitioning in the multi-scale pooled patch embedding module and instead adopts patch lengths that are powers of two. ③ and ④ respectively remove the variable-dimension interaction and the temporal-dimension interaction in the DDM module. ⑤ completely removes the DDM module, thereby eliminating intra-scale information interaction.

The experimental results indicate that manually defined patch lengths perform poorly. This is mainly because the the original time series exhibits complex multi-period temporal patterns, and simple patch partitioning fails to adequately align multi-period characteristics, whereas partitioning patches based on dominant periods can better address this issue. Removing the decomposition mixer module leads to a substantial performance degradation, demonstrating that modeling inter-scale information interactions is crucial for learning diverse temporal dependencies. Moreover, both intra-variable and inter-variable interactions in DDM further improve model performance by enhancing interactions along the temporal and variable dimensions within each scale; models without intra-scale interactions suffer from a pronounced performance drop.

Models		PP-Mixer		PatchMLP		TimeMixer		iTransformer		PatchTST		RLinear		TimesNet		Crossformer		MICN		DLinear	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.362**	**0.390**	0.368	0.397	0.389	0.402	0.386	0.405	0.414	0.419	0.386	0.390	0.384	0.402	0.423	0.448	0.426	0.446	0.397	0.412
	192	**0.414**	**0.421**	0.421	0.426	0.442	0.429	0.441	0.436	0.460	0.445	0.437	0.421	0.436	0.429	0.471	0.474	0.454	0.464	0.446	0.441
	336	**0.462**	0.450	0.473	0.466	0.499	0.469	0.487	0.458	0.501	0.466	0.479	**0.446**	0.491	0.469	0.570	0.546	0.493	0.487	0.489	0.467
	720	0.481	0.476	0.488	0.487	0.506	0.481	0.503	0.491	0.500	0.488	**0.481**	**0.470**	0.521	0.500	0.653	0.621	0.526	0.526	0.513	0.510
	Avg	**0.430**	0.434	0.438	0.444	0.459	0.445	0.454	0.448	0.469	0.454	0.446	**0.434**	0.458	0.450	0.529	0.522	0.475	0.480	0.461	0.457
ETTh2	96	0.280	0.336	**0.275**	**0.333**	0.290	0.342	0.297	0.349	0.302	0.348	0.288	0.338	0.340	0.374	0.745	0.584	0.372	0.424	0.340	0.394
	192	0.365	0.388	**0.360**	**0.384**	0.363	0.385	0.380	0.400	0.388	0.400	0.374	0.390	0.402	0.414	0.877	0.656	0.492	0.492	0.482	0.479
	336	**0.386**	**0.407**	0.409	0.425	0.417	0.431	0.428	0.432	0.426	0.433	0.415	0.426	0.452	0.452	1.043	0.731	0.607	0.555	0.591	0.541
	720	**0.404**	**0.421**	0.422	0.441	0.430	0.447	0.427	0.445	0.431	0.446	0.420	0.440	0.462	0.468	1.104	0.763	0.824	0.655	0.839	0.661
	Avg	**0.359**	**0.388**	0.367	0.396	0.375	0.401	0.383	0.407	0.387	0.407	0.374	0.398	0.414	0.427	0.942	0.684	0.574	0.531	0.563	0.519
ETTm1	96	**0.315**	**0.356**	0.316	0.360	0.326	0.363	0.334	0.368	0.329	0.367	0.355	0.376	0.338	0.375	0.404	0.426	0.365	0.387	0.346	0.374
	192	0.360	**0.376**	**0.358**	0.377	0.369	0.389	0.377	0.391	0.367	0.385	0.391	0.392	0.374	0.387	0.450	0.451	0.403	0.408	0.382	0.391
	336	**0.381**	**0.396**	0.388	0.402	0.396	0.407	0.426	0.420	0.399	0.410	0.424	0.415	0.410	0.411	0.532	0.515	0.436	0.431	0.415	0.415
	720	0.446	**0.438**	**0.444**	0.442	0.462	0.449	0.491	0.459	0.454	0.439	0.487	0.450	0.478	0.450	0.666	0.589	0.489	0.462	0.473	0.451
	Avg	**0.376**	**0.392**	0.377	0.395	0.388	0.402	0.407	0.410	0.387	0.400	0.414	0.407	0.400	0.406	0.513	0.495	0.423	0.422	0.404	0.408
ETTm2	96	**0.169**	0.253	0.172	**0.250**	0.175	0.258	0.180	0.264	0.175	0.259	0.182	0.265	0.187	0.267	0.287	0.366	0.197	0.296	0.193	0.293
	192	**0.234**	**0.294**	0.240	0.305	0.239	0.301	0.250	0.309	0.241	0.302	0.246	0.304	0.249	0.309	0.414	0.492	0.284	0.361	0.284	0.361
	336	0.290	**0.336**	**0.288**	0.337	0.298	0.340	0.311	0.348	0.305	0.343	0.307	0.342	0.321	0.351	0.597	0.542	0.381	0.429	0.382	0.429
	720	**0.361**	**0.383**	0.364	0.385	0.393	0.400	0.412	0.407	0.402	0.400	0.407	0.398	0.408	0.403	1.730	1.042	0.549	0.522	0.558	0.525
	Avg	**0.264**	**0.317**	0.266	0.319	0.276	0.325	0.288	0.332	0.281	0.326	0.286	0.327	0.291	0.333	0.757	0.610	0.353	0.402	0.354	0.402
Electricity	96	**0.141**	**0.238**	0.152	0.244	0.169	0.258	0.148	0.249	0.181	0.270	0.201	0.281	0.168	0.272	0.219	0.314	0.180	0.293	0.210	0.302
	192	**0.152**	**0.251**	0.160	0.258	0.172	0.275	0.169	0.264	0.188	0.274	0.201	0.283	0.184	0.289	0.231	0.322	0.189	0.302	0.210	0.305
	336	**0.173**	**0.268**	0.177	0.271	0.196	0.284	0.178	0.279	0.204	0.293	0.215	0.298	0.198	0.300	0.246	0.337	0.198	0.312	0.223	0.319
	720	**0.215**	**0.303**	0.224	0.311	0.239	0.335	0.225	0.317	0.246	0.324	0.257	0.331	0.220	0.320	0.280	0.363	0.217	0.330	0.258	0.350
	Avg	**0.170**	**0.265**	0.178	0.271	0.194	0.288	0.180	0.277	0.205	0.290	0.219	0.298	0.192	0.295	0.244	0.334	0.196	0.309	0.225	0.319
Traffic	96	0.408	0.267	0.396	**0.264**	0.480	0.288	**0.395**	0.268	0.462	0.295	0.649	0.389	0.593	0.321	0.644	0.429	0.577	0.350	0.650	0.396
	192	0.419	0.280	**0.408**	**0.273**	0.473	0.296	0.417	0.276	0.466	0.296	0.601	0.366	0.617	0.336	0.665	0.431	0.589	0.356	0.598	0.370
	336	0.436	0.281	**0.421**	**0.279**	0.500	0.303	0.433	0.283	0.482	0.304	0.609	0.369	0.629	0.336	0.674	0.420	0.594	0.358	0.605	0.373
	720	0.478	0.308	**0.443**	**0.285**	0.562	0.315	0.467	0.302	0.514	0.322	0.647	0.387	0.640	0.350	0.683	0.424	0.613	0.361	0.645	0.394
	Avg	0.435	0.284	**0.417**	**0.275**	0.504	0.301	0.428	0.282	0.481	0.304	0.626	0.378	0.620	0.336	0.667	0.426	0.593	0.356	0.625	0.383
Weather	96	**0.157**	**0.205**	0.167	0.209	0.167	0.210	0.174	0.214	0.177	0.218	0.192	0.232	0.172	0.220	0.195	0.271	0.198	0.261	0.195	0.252
	192	**0.202**	**0.247**	0.216	0.257	0.213	0.254	0.221	0.254	0.225	0.259	0.240	0.271	0.219	0.261	0.209	0.277	0.239	0.299	0.237	0.295
	336	**0.243**	**0.285**	0.248	0.294	0.271	0.299	0.278	0.296	0.278	0.297	0.292	0.307	0.280	0.306	0.273	0.332	0.285	0.336	0.282	0.331
	720	**0.337**	**0.341**	0.344	0.351	0.349	0.352	0.358	0.347	0.354	0.348	0.364	0.353	0.365	0.359	0.379	0.401	0.351	0.388	0.345	0.382
	Avg	**0.235**	**0.270**	0.244	0.278	0.250	0.279	0.258	0.278	0.259	0.281	0.272	0.291	0.259	0.287	0.264	0.320	0.268	0.321	0.265	0.315
Solar	96	**0.184**	**0.254**	0.199	0.261	0.203	0.293	0.203	0.267	0.234	0.286	0.322	0.339	0.250	0.292	0.232	0.302	0.257	0.325	0.290	0.378
	192	**0.210**	**0.265**	0.220	0.289	0.255	0.280	0.253	0.275	0.267	0.310	0.359	0.356	0.296	0.318	0.371	0.410	0.278	0.354	0.320	0.398
	336	**0.229**	**0.279**	0.240	0.291	0.275	0.288	0.248	0.293	0.290	0.315	0.397	0.369	0.319	0.330	0.495	0.515	0.298	0.375	0.353	0.415
	720	**0.209**	**0.268**	0.225	0.293	0.252	0.281	0.249	0.275	0.289	0.317	0.397	0.356	0.338	0.337	0.526	0.542	0.515	0.379	0.357	0.413
	Avg	**0.208**	**0.267**	0.221	0.284	0.246	0.286	0.238	0.278	0.270	0.307	0.369	0.356	0.301	0.319	0.406	0.442	0.283	0.358	0.330	0.401
1^{st} Count		**29**	**29**	9	8	0	0	1	0	0	0	1	3	0	0	0	0	0	0	0	0

TABLE I

OVERALL COMPARISON OF DIFFERENT BENCHMARKS MODELS. WE COMPARE PP-MIXER WITH SOTA BENCHMARKS UNDER IDENTICAL EXPERIMENTAL SETTINGS. THE INPUT SEQUENCE LENGTH IS SET TO 96 FOR ALL BASELINES WITH PREDICTION LENGTH $S \in \{96, 192, 336, 720\}$. AVG MEANS THE AVERAGE RESULTS FROM ALL FOUR PREDICTION LENGTHS.

C. Sensitivity Analysis

We analyze the sensitivity of PP-Mixer to four key hyperparameters, including the number of mixer blocks L, the dimension of embeddings D, the learning rate lr, and the number of main periods. As shown in Fig 3. (a)–(d), performance is evaluated in terms of MSE on five benchmark datasets: ETTh1, ETTh2, ETTm1, Electricity, and Weather. Overall, PP-Mixer exhibits stable performance across a wide range of hyperparameter settings, demonstrating strong robustness to hyperparameter variations.

Increasing the number of mixer blocks does not consistently improve performance and may even lead to slight degradation, indicating that simply stacking more layers is not necessarily beneficial. With respect to the embedding dimension, overly large values tend to cause overfitting, while excessively small dimensions limit representation capacity; empirically, D = 32 provides a favorable trade-off between accuracy and model complexity. PP-Mixer also maintains stable performance over a broad range of learning rates, with lr = 0.01 offering a good balance between convergence stability and training efficiency. In addition, using three dominant periods yields a reliable balance between predictive accuracy and modeling complexity, which is therefore adopted as the default configuration.

D. Visualization of predictions

Figure 4 compares PP-Mixer and TimeMixer on the Electricity (ECL) dataset in both time and frequency domains. In the time domain, PP-Mixer tracks the ground truth closely, with accurate trend, phase alignment, and peak–valley locations, whereas TimeMixer shows evident phase drift and amplitude errors. In the frequency domain, the ground truth exhibits pronounced dominant low-frequency peaks, indicating strong periodicity. PP-Mixer largely preserves these peaks with well-matched locations and energy, suggesting effective recovery of intrinsic periods. By contrast, TimeMixer yields

Case	MPPE	Decomposition Mixer	Inter Variable	Intra Variable	ETTh1		electricity		Solar-energy	
					MSE	MAE	MSE	MAE	MSE	MAE
①	✓	✓	✓	✓	**0.430**	**0.434**	**0.170**	**0.265**	**0.208**	**0.267**
②	✓	×	✓	✓	0.443	0.446	0.182	0.276	0.224	0.288
③	✓	✓	×	✓	0.435	0.439	0.172	0.269	0.213	0.276
④	✓	✓	✓	×	0.434	0.437	0.174	0.267	0.215	0.279
⑤	✓	✓	×	×	0.437	0.441	0.176	0.274	0.219	0.283
⑥	×	✓	✓	✓	0.435	0.440	0.173	0.270	0.216	0.275
⑦	×	×	✓	✓	0.445	0.448	0.184	0.283	0.231	0.291
⑧	×	✓	×	✓	0.436	0.445	0.175	0.274	0.219	0.279
⑨	×	✓	✓	×	0.438	0.441	0.176	0.271	0.222	0.278

TABLE II

ALL ABLATION STUDIES EMPLOY IDENTICAL PARAMETER SETTINGS WITH A FIXED HISTORICAL HORIZON LENGTH OF 96. THE REPORTED MSE AND MAE RESULTS ARE AVERAGED OVER FOUR PREDICTION HORIZONS 96, 192, 336, 720.

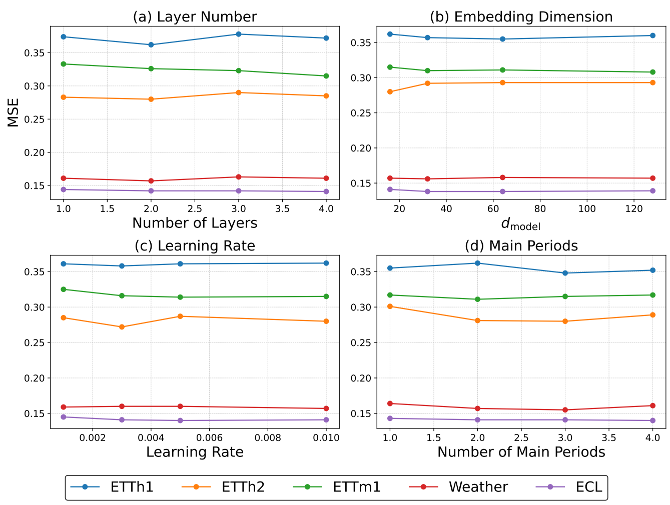

Fig. 3. Hyperparameter sensitivity analysis was carried out regarding the number of decomposition mixer blocks, the hidden dimension of embeding ,the learning rate and the main periods of FFT.The results were collected with T = 96 for the historical length and S = 96 for the perdiction length.

a more diffuse spectrum with weakened or shifted dominant components, implying inferior periodic modeling. These frequency-domain discrepancies manifest in the time domain: period misalignment in TimeMixer accumulates phase errors over long horizons, while PP-Mixer maintains stable synchronization across cycles.The results underscore the importance of dominant-period alignment for long-term forecasting and validate PP-Mixer's period-aware patching and multi-scale mixing in mitigating period mismatch.

V. Conclusion

In this work, we propose PP-Mixer, a fully MLP-based framework for long-term time series forecasting that leverages Fourier-guided multi-scale pooled patch representations and decomposable cross-scale mixing. By combining seasonal–trend integration with lightweight dual-dimension interactions, PP-Mixer captures both cross-scale dynamics and intra-scale dependencies with low computational overhead. Extensive experiments on eight benchmarks demonstrate consistent improvements over strong Transformer- and MLP-based baselines. These results suggest that period-aware multi-scale representations, together with simple dimension-wise mixing, provide an effective alternative to attention-heavy architectures for long-horizon forecasting.

Acknowledgment

This work is supported by the Joint Innovation Laboratory for Future Observation, Zhejiang University, focusing on the research of cloud-native observability technology based on GuanceCloud(H25008).

References

[1] 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," *Proc. AAAI Conf. Artificial Intelligence*, vol. 35, no. 12, pp. 11 106–11 115, 2021.

[2] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2019.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[4] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Timemixer: Decomposable multiscale mixing for time series forecasting," in *International Conference on Learning Representations*, 2023.

[5] 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*, 2021.

[6] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *International Conference on Learning Representations*, 2023.

[7] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *International Conference on Learning Representations*, 2023.

[8] Y. Hu, P. Liu, P. Zhu, D. Cheng, and T. Dai, "Adaptive multi-scale decomposition framework for time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 16, 2025, pp. 17 359–17 367.

[9] S. Lin, W. Lin, X. Hu, W. Wu, R. Mo, and H. Zhong, "Cyclenet: Enhancing time series forecasting through modeling periodic patterns," in *Advances in Neural Information Processing Systems*, 2024, pp. 106 315–106 345, neurIPS 2024 Spotlight. [Online]. Available: https://arxiv.org/abs/2409.18479

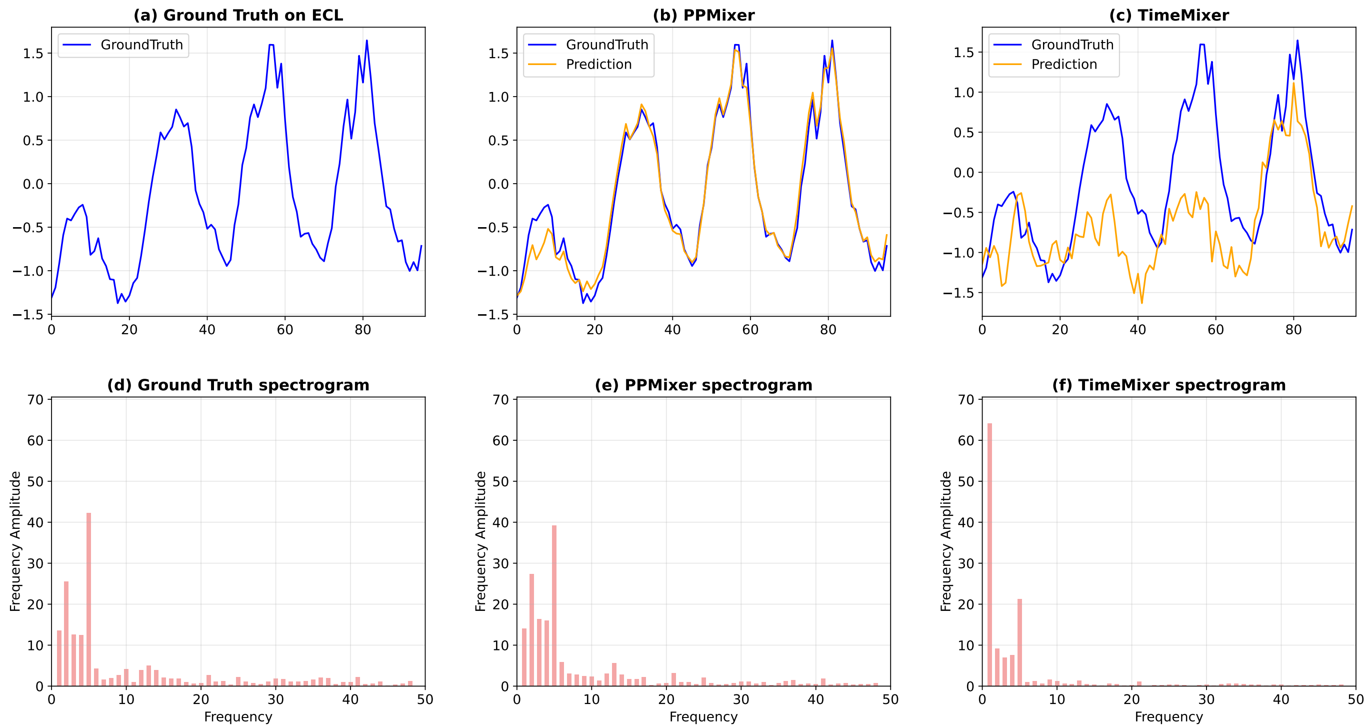

Fig. 4. Visualization of forecasting results and frequency spectra on the ECL dataset. PP-Mixer adopts period-aware adaptive patch partitioning guided by dominant periods, while TimeMixer uses fixed patch lengths that are powers of two; orange lines denote model predictions and blue lines represent the ground truth.

[10] C. Shen, Y. He, and J. Qin, "Robust temporal nonnegative matrix factorization forecasting model for multidimensional time series," *Entropy*, vol. 26, no. 1, p. 92, 2024. [Online]. Available: https://www.mdpi.com/1099-4300/26/1/92

[11] Y. Xu, W. Zhang, H. Lai, S. Zhang, and R. Xiao, "Mpmixer: Multi-scale patch mixer for long-term time series forecasting," *Neurocomputing*, vol. 646, p. 130398, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231225010707

[12] B. Gao, Q. Ni, F. Ju, Y. Chen, and Z. Zhao, "A multi-scale representation learning framework for long-term time series forecasting," *arXiv*, 2025, preprint. [Online]. Available: https://arxiv.org/abs/2505.08199

[13] M. Kraus, F. Divo, D. S. Dhami, and K. Kersting, "xlstm-mixer: Multivariate time series forecasting by mixing via scalar memories," *arXiv*, 2025, preprint. [Online]. Available: https://arxiv.org/abs/2410.16928

[14] 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*, 2023. [Online]. Available: https://arxiv.org/abs/2310.06625

[15] Y. Zhang, J. Yan, W. Wang, and J. Zhu, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *International Conference on Learning Representations*, 2023.

[16] 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.

[17] 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.

[18] A. Zeng, S. Qi, C. Liu, Y. Zhang, X. Hao, Q. Yang, M. Chen, L. Liu, J. Han, and Z. Lin, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023.

[19] W. Li, X. Meng, C. Chen, and J. Chen, "Rlinear: Rethinking the linear model for time-series forecasting," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2305.04800

[20] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[21] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, and J. Uszkoreit, "Mlp-mixer: An all-mlp architecture for vision," in *International Conference on Learning Representations*, 2021.

[22] Y. Zhang, X. Chen, S. Pan, and M. Long, "U-mixer: An all-mlp architecture for multivariate time series forecasting," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2303.16664

[23] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam, "Tsmixer: An all-mlp architecture for time series forecasting," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2306.09364

[24] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[25] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proceedings of the International Conference on Machine Learning*, 2022.

[26] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition procedure based on loess," *Journal of Official Statistics*, vol. 6, no. 1, pp. 3–73, 1990.