

# DPWMixer: Dual-Path Wavelet Mixer for Long-Term Time Series Forecasting

Qianyang Li<sup>1</sup>, Xingjun Zhang<sup>1\*</sup>, Shaoxun Wang<sup>1</sup>, Jia Wei<sup>2</sup>

<sup>1</sup>School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China.

<sup>2</sup>Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China.

\*Corresponding author(s). E-mail(s): [xjzhang@xjtu.edu.cn](mailto:xjzhang@xjtu.edu.cn);

Contributing authors: [liqianyang@stu.xjtu.edu.cn](mailto:liqianyang@stu.xjtu.edu.cn);  
[shaoxunwang@stu.xjtu.edu.cn](mailto:shaoxunwang@stu.xjtu.edu.cn); [weijia4473@mail.tsinghua.edu.cn](mailto:weijia4473@mail.tsinghua.edu.cn);

## Abstract

Long-term time series forecasting (LTSF) is a critical task in computational intelligence. While Transformer-based models effectively capture long-range dependencies, they often suffer from quadratic complexity and overfitting due to data sparsity. Conversely, efficient linear models struggle to depict complex non-linear local dynamics. Furthermore, existing multi-scale frameworks typically rely on average pooling, which acts as a non-ideal low-pass filter, leading to spectral aliasing and the irreversible loss of high-frequency transients. In response, this paper proposes DPWMixer, a computationally efficient Dual-Path architecture. The framework is built upon a Lossless Haar Wavelet Pyramid that replaces traditional pooling, utilizing orthogonal decomposition to explicitly disentangle trends and local fluctuations without information loss. To process these components, we design a Dual-Path Trend Mixer that integrates a global linear mapping for macro-trend anchoring and a flexible patch-based MLP-Mixer for micro-dynamic evolution. Finally, An adaptive multi-scale fusion module then integrates predictions from diverse scales, weighted by channel stationarity to optimize synthesis. Extensive experiments on eight public benchmarks demonstrate that our method achieves a consistent improvement over state-of-the-art baselines. The code is available at <https://github.com/hit636/DPWMixer>.

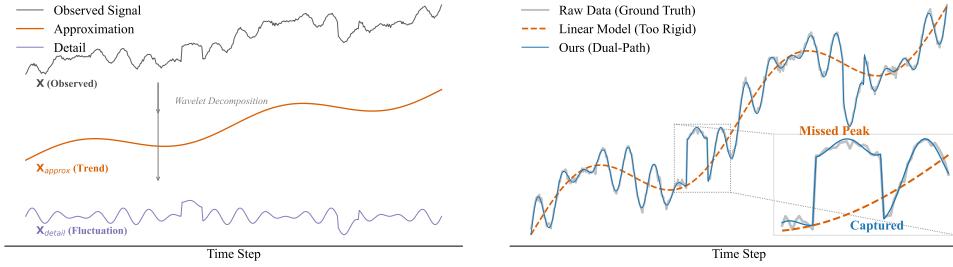
**Keywords:** Time series forecasting, Wavelet Pyramid, Dual-Path Mixer, Adaptive fusion

# 1 Introduction

Time series forecasting (TSF) is fundamental for data-driven decision-making systems, including energy grid management [1, 2], intelligent transportation systems [3, 4], and climate modeling [5]. Recently, researchers have started to investigate Long-Term Time Series Forecasting (LTSF) [6, 7], where models are asked to forecast substantial future horizons given historical context. Given the computationally demanding nature of the task, models are challenged by the involved temporal dynamics, which are characterized by non-stationarity, multi-scale periodicity, and the superposition of deterministic trends with stochastic noise.

Approaches have evolved from recurrent structures to Transformer-based models [8, 9]. To address self-attention issues for long sequences, methods such as Informer [10], Autoformer [11], and PatchTST [12] have been developed to alleviate quadratic computational complexity and to improve local semantic extraction. These methods are able to model long-range dependencies, they tend to be memory-intensive and are prone to overfitting. This is an issue due to the sparse semantic density of time series data, compared to natural language. On the other hand, Multi-Layer Perceptron (MLP) and linear-based models, such as DLinear [13] and TiDE [14], have shown that simple models can obtain competitive results with linear complexity  $\mathcal{O}(L)$ . However, the use of static weights in these models hinders their ability to model the non-linear evolutionary dynamics in rapidly changing distributions.

Despite these advancements, modeling real-world LTSF data presents two fundamental challenges derived from the intrinsic characteristics of time series signals, as illustrated in Figure 1:



(a) Spectral Aliasing (Decomposition) (b) Trend-Detail Incompatibility (Modeling)

**Fig. 1** Illustration of the intrinsic challenges in LTSF and our design intuition. **(a) Spectral Aliasing (Decomposition):** Standard pooling leads to aliasing and information loss, whereas our wavelet-based approach achieves lossless, orthogonal disentanglement. **(b) Trend-Detail Incompatibility (Modeling):** Pure linear models (Orange dashed) capture the macro-trend but fail to model local transients. Our Dual-Path architecture (Blue solid) resolves this incompatibility by harmonizing global anchoring with local refinement.

- **Spectral Aliasing in Down-sampling (Figure 1a):** To capture hierarchical dependencies, existing frameworks typically employ average pooling. However, from a signal processing perspective, pooling acts as a non-ideal low-pass filter, causing spectral aliasing, which irreversibly mixes high-frequency fluctuations into the

trend, leading to information loss. To preserve spectral integrity, a mathematically orthogonal decomposition is required.

- **Trend-Detail Incompatibility (Figure 1b):** Real-world time series emerge from a combination of deterministic global trends and stochastic local fluctuations. This inherent heterogeneity presents a fundamental challenge to single-stream architectures. As shown in (b), linear model successfully anchors the trend but is rigid to capture rapid transients (Missed Peak). A reasonable forecasting framework should explicitly decouple above two components.

Therefore, we bridge signal processing theory and deep representation learning by proposing DPWMixer (Dual-Path Wavelet Mixer), a novel architecture for modeling multi-scale signals with linear computational complexity. Specifically, we replace the lossy pooling with Lossless Haar Wavelet Pyramid [15]. Different from pooling, DWT uses orthogonal basis functions to decompose signals into approximation coefficients and detail coefficients, and DWT satisfies Parseval’s theorem (energy conservation). This prevents information loss by separating high-frequency components from trend signals. Further, to resolve conflict between trend and detail modeling, we design a Dual-Path Trend Mixer. Specifically, the module processes decomposed components into two parallel paths: a Global Linear Path anchors the prediction to macro-trend while a Local Evolution Path (implemented by a lightweight Patch-Mixer) models micro-dynamic contexts. Finally, an Adaptive Multi-Scale Fusion mechanism is implemented to weigh predictions from different scales given inputs’ characteristics. The main contributions of this paper can be summarized as follows.

- We formulate the Haar Wavelet Pyramid as a rigorous solution to aliasing problem in multi-scale forecasting. Both theoretical analysis and algorithm design demonstrate that, compared with pooling, Haar Wavelet Pyramid provides a better inductive bias to disentangle trends and details without information loss.
- We design Dual-Path Trend Mixer as a hybrid layer to explicitly resolve the conflict between modeling stable global trend and flexible local variation. The design borrows the robustness of linear model and the expressiveness of MLPs.
- We implement Adaptive Multi-Scale Fusion mechanism to weigh predictions from different resolutions adaptively, which allows the model to specialize for multivariate heterogeneity.
- Extensive evaluations on eight benchmark datasets demonstrate that DPWMixer consistently outperforms state-of-the-art baselines, including iTransformer and TimeMixer. Furthermore, our model maintains  $\mathcal{O}(L)$  time complexity, ensuring scalability for high-throughput forecasting tasks.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 details the DPWMixer methodology. Section 4 presents experimental results, and Section 5 concludes the paper.

## 2 Related Work

### 2.1 Transformer-based Models for LTSF

Transformers[9] model the global dependence with self-attention mechanism. Its canonical quadratic complexity  $\mathcal{O}(L^2)$  brings great challenges to LTSF. Many efficient variants have been proposed to alleviate this issue. Informer[10] proposes a ProbSparse attention mechanism based on KL-divergence criterion to select dominant queries, and the complexity is lowered to  $\mathcal{O}(L \log L)$ . Autoformer[11] abandons the point-wise attention and proposes an Auto-Correlation mechanism, which discovers the similarities between sub-series based on periodicity of series. FEDformer[16] explores in the Frequency Domain, and uses Fourier and Wavelet transform to select a subset of frequency components for linear complexity interaction. Pyraformer[17] constructs a pyramidal attention graph to capture multi-resolution temporal dependencies.

Recently, researchers focus on input representation instead of model complexity. PatchTST[12] divides time series into sub-series patches and uses them as input tokens. This design not only keeps the advantage of preserving local semantic information, but also greatly reduces the number of tokens, which in turn extends the look-back window. Besides, PatchTST discusses the advantages of Channel Independence, where each variate is forecasted independently. While Crossformer[18] and iTransformer[19] argue that cross-variate dependency should be considered. Particularly, iTransformer inverts the dimensions, where the whole time series of each variate is embedded into one token and applies attention over variates to capture the multivariate correlations. Transformer based models are prone to overfitting because of the sparse semantic density of time series compared with NLP. Besides, they may also be weak in learning trend information without any decomposition modules.

### 2.2 MLP and Linear-based Models

Challenging the dominance of Transformers, DLinear[13] decomposes time series into trend and seasonal components and models them with only one linear layer. The simple design shows that what matters for LTSF is that the temporal order of inputs should be preserved and more complex attention might be useless or even harmful due to the permutation invariance. RLinear[20] further studies the influence of input Reversibility and Normalization.

Among MLPs, TiDE[21] proposes a dense encoder-decoder model that integrates exogenous variables and covariates through simple MLP layers and obtains competitive results with high throughput. TSMixer[22] extends MLP-Mixer in computer vision to time series and mixes along both time and feature dimensions to model intra- and inter-variate dependencies. These models are more efficient and stable, their receptive fields are rigid. Linear models posit a relatively fixed relationship between history and future, which is unable to model the highly complex nonlinear evolutionary dynamics, especially for time series with rapidly changing distributions (distribution shift). Moreover, simple linear mappings would be a kind of low-pass filter that may smear out important high-frequency anomalies.

### 2.3 Multi-Scale and Frequency Domain Modeling

Real-world time series exhibit intricate patterns at various granularities. Multi-scale modeling aims to capture these hierarchical dependencies. SCINet[23] proposes a recursive down-sampling-convolve-interact architecture to learn multi-resolution representations, enhancing predictability. TimesNet[24] transforms 1D series into 2D tensors to utilize convolutions. MSTF [25] proposed a multi-scale temporal fusion model using time reverse blocks and dynamic combination reconstruction. However, MSTF relies on average pooling for down-sampling, which inherently leads to information loss and aliasing of high-frequency signals according to the Nyquist-Shannon sampling theorem. In the frequency domain, ScaleMixNet [26] proposes an adaptive multi-scale time-frequency fusion network using FFT and hybrid loss functions. While effective, Fourier-based methods provide a global view of frequencies and struggle to capture local transient changes typical in non-stationary time series.

A recent representative method TimeMixer[27] uses past-decomposable-mixing architecture, and average pooling to construct a multi-scale input pyramid. Although the receptive field can be enlarged by this design, since the average pooling is in fact a kind of non-invertible low-pass filter according to the Nyquist-Shannon sampling theorem, inevitable aliasing would be induced and high-frequency information (transients) necessary for forecasting volatile series will be randomly lost.

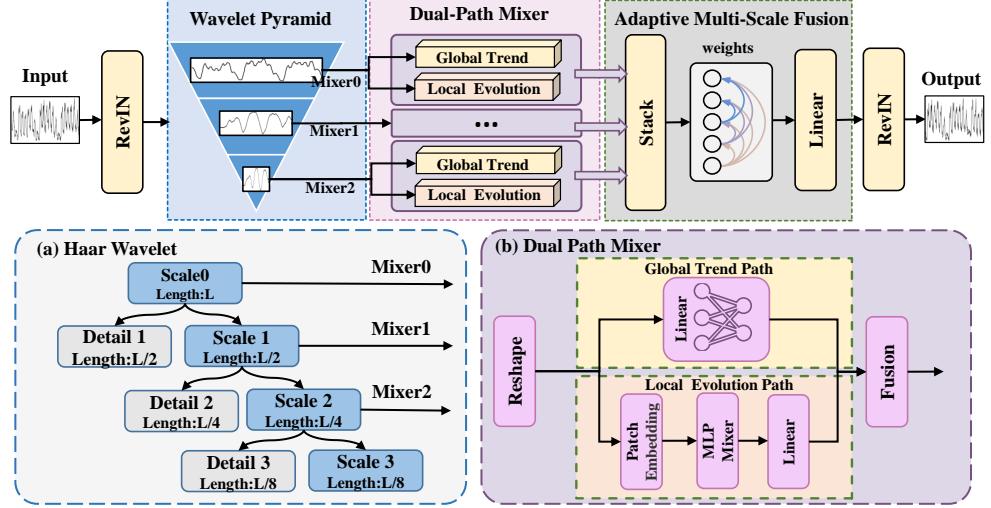
In frequency domain, FiLM[28] and ETSformer[29] use Legendre projections and exponential smoothing respectively to remember historical information. However, Fourier based methods usually offer global view on frequencies, while historical information of non-stationary data should be time-frequency localized. In contrast, our method employs Orthogonal Haar Wavelet Transforms in dual-path. Unlike pooling, wavelets can provide an orthogonal decomposition into approximation and detail coefficients. Unlike Fourier, wavelets could provide localization in both time and frequency simultaneously, which helps our model to capture trend in evolution and transient anomalies at the same time.

## 3 Methodology

### 3.1 Problem Formulation and Framework Overview

Let  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_L\} \in \mathbb{R}^{L \times C}$  denote the historical multivariate time series, where  $L$  represents the look-back window size and  $C$  denotes the number of variates. The fundamental objective of Long-Term Time Series Forecasting (LTSF) is to estimate the joint probability distribution  $P(\mathcal{Y}|\mathcal{X})$  and predict the future sequence  $\mathcal{Y} = \{\mathbf{x}_{L+1}, \dots, \mathbf{x}_{L+T}\} \in \mathbb{R}^{T \times C}$  over a forecast horizon  $T$ . We aim to approximate the optimal mapping function  $\mathcal{F}_\theta : \mathbb{R}^{L \times C} \mapsto \mathbb{R}^{T \times C}$ , parameterized by  $\theta$ , that minimizes the  $L_2$  norm distance (Mean Squared Error) between the ground truth  $\mathcal{Y}$  and the prediction  $\hat{\mathcal{Y}}$ .

To bridge these gaps, our proposed DPWMixer incorporates the following two innovations: (1) Orthogonal Multi-Resolution Decomposition that recursively splits normalized series by Lossless Haar Wavelet Pyramid to preserve spectral information resulted in down-sampling aliasing; (2) Dual-Path Representation Learning that



**Fig. 2** The overall architecture of DPWMixer. Top: The global pipeline showing orthogonal multi-scale decomposition and adaptive fusion. (a) Haar Wavelet Pyramid: The input is orthogonally decomposed into Approximation ( $\mathbf{X}$ ) and Detail ( $\mathbf{H}$ ) coefficients, preventing aliasing compared to average pooling. (b) Dual-Path Trend Mixer: A hybrid block unifying a Global Trend Path for rigid trends and a Local Evolution Path for flexible dynamics. Outputs are fused via learnable gates.

models deterministic global trend and stochastic local semantics via linear path and MLP-based path respectively at each extracted multi-scale representation. As illustrated in Figure 2, our method consists of three stages: orthogonal multi-resolution decomposition, which recursively splits normalized series into multi-scale representations via a Lossless Haar Wavelet Pyramid to preserve spectral information affected by down-sampling aliasing; dual-path representation learning, where at each resolution scale, a Dual-Path Trend Mixer models deterministic global trends and stochastic local semantics through a linear path and an MLP-based path, respectively; and adaptive spectral aggregation, which integrates multi-scale forecasts into the final prediction via an Adaptive Multi-Resolution Fusion module that adaptively aggregates different frequency bands weighted by channel-wise characteristics.

To synthesize the proposed framework, Algorithm 1 outlines the holistic forward propagation pipeline. This procedure systematically orchestrates the lossless wavelet decomposition and dual-path mixing, ensuring that multi-scale features are extracted from physically disentangled frequency bands before adaptive integration.

### 3.2 Haar Wavelet Pyramid

A limitation of existing multi-scale architectures is the adoption of average pooling for down-sampling. From the signal processing point of view, by convolution with a rectangular window function, average pooling leads to the multiplication of a Sinc function in the frequency domain. Since the Sinc function decays slowly in  $\mathcal{O}(1/f)$ , most of its energy is at low frequencies, i.e., spectral leakage. When sampling rate

---

**Algorithm 1** DPWMixer Training Procedure

---

**Require:** Historical time series  $\mathbf{X} \in \mathbb{R}^{B \times L \times C}$ , Prediction horizon  $T$ , Number of scales  $N$ .

**Ensure:** Predicted future series  $\hat{\mathbf{Y}} \in \mathbb{R}^{B \times T \times C}$ .

- 1: **1. Distribution Alignment (RevIN)**
- 2:  $\mu, \sigma \leftarrow \text{Statistics}(\mathbf{X})$
- 3:  $\mathbf{X}' \leftarrow \text{Normalize}(\mathbf{X}, \mu, \sigma)$  ▷ Instance normalization
- 4: **2. Lossless Wavelet Pyramid Construction**
- 5: Initialize scale list  $\mathcal{S} \leftarrow [\mathbf{X}']$
- 6: **for**  $j = 0$  to  $N - 1$  **do**
- 7:    $\mathbf{X}^{(j)} \leftarrow \mathcal{S}[j]$
- 8:    $\mathbf{X}_{low}, \mathbf{X}_{high} \leftarrow \text{HaarDWT}(\mathbf{X}^{(j)})$  ▷ Orthogonal decomposition
- 9:    $\mathcal{S}.\text{append}(\mathbf{X}_{low})$  ▷ Down-sampled approximation
- 10: **end for**
- 11: **3. Multi-Scale Dual-Path Mixing**
- 12: Initialize prediction list  $\mathcal{P} \leftarrow []$
- 13: **for**  $j = 0$  to  $N$  **do**
- 14:    $\mathbf{X}^{(j)} \leftarrow \mathcal{S}[j]$  ▷ Input at scale  $j$ , Length  $L/2^j$
- 15:   // Path A: Global Trend Patch
- 16:    $\mathbf{H}_{global} \leftarrow \mathbf{W}_{lin}^{(j)} \mathbf{X}^{(j)}$
- 17:   // Path B: Local Evolution Patch
- 18:    $\mathbf{Z}_{patch} \leftarrow \text{PatchEmbedding}(\mathbf{X}^{(j)})$
- 19:    $\mathbf{Z}_{mix} \leftarrow \text{MLP-Mixer}(\mathbf{Z}_{patch})$  ▷ Token & Channel mixing
- 20:    $\mathbf{H}_{local} \leftarrow \text{Projection}(\mathbf{Z}_{mix})$
- 21:   // Gated Fusion for Scale  $j$
- 22:    $\hat{\mathbf{Y}}^{(j)} \leftarrow w_g^{(j)} \mathbf{H}_{global} + w_l^{(j)} \mathbf{H}_{local}$  ▷ Combine rigidity & flexibility
- 23:    $\mathcal{P}.\text{append}(\hat{\mathbf{Y}}^{(j)})$
- 24: **end for**
- 25: **4. Adaptive Multi-Resolution Fusion**
- 26: Compute weights  $\mathbf{W} \leftarrow \text{Softmax}(\mathcal{A}_{learnable})$
- 27:  $\hat{\mathbf{Y}}' \leftarrow \sum_{j=0}^N \mathbf{W}[j] \odot \mathcal{P}[j]$  ▷ Weighted ensemble
- 28: **5. Inverse Normalization**
- 29:  $\hat{\mathbf{Y}} \leftarrow \text{DeNormalize}(\hat{\mathbf{Y}}', \mu, \sigma)$
- 30: **return**  $\hat{\mathbf{Y}}$

---

is reduced, high-frequency part cannot be suppressed enough fold back into lower frequency bands, aliasing. This distortion makes down-sampled representation not an accurate approximation of the original trend, breaking the effectiveness of learning long-term dependencies.

### 3.2.1 Orthogonal Decomposition Mechanism

To overcome the spectral aliasing issue, we employ Discrete wavelet transform(DWT) [30] with Haar basis. Different from pooling which is also an approximation process similar to convolving with a window function, Haar transform uses a pair of

Quadrature Mirror Filters (QMF) to decompose the signal space  $V_j$  into two mutually orthogonal subspaces: the approximation subspace  $V_{j+1}$  and the detail subspace  $W_{j+1}$ . The orthogonality condition guarantees that the energy of the signal is conserved (Parseval’s Theorem) and the information is losslessly split.

We employ Haar wavelet over continuous bases (e.g., Daubechies or Symlets) for two reasons. First, from the signal processing point of view, the step-function nature of Haar basis renders it better at capturing sudden changes and transients (e.g., traffic peaks or sensor faults) without introducing the ringing artifacts produced by higher order wavelets. Second, in HPC, Haar transform provides the lowest possible computational overhead among all wavelets. Given its filter length is only 2, it incurs the least floating point operations and memory access cost. Therefore, it strictly preserves the linear complexity  $\mathcal{O}(L)$  of our whole framework.

We construct the pyramid recursively. Let  $\mathbf{X}^{(0)} = \mathbf{X}'$  be the normalized input. For each decomposition level  $j$ , the approximation coefficients  $\mathbf{X}^{(j+1)}$  and detail coefficients  $\mathbf{D}^{(j+1)}$  are computed via 1D convolution with a specific stride:

$$\mathbf{X}^{(j+1)} = (\mathbf{X}^{(j)} * \mathbf{k}_{low}) \downarrow 2 \quad (1)$$

$$\mathbf{D}^{(j+1)} = (\mathbf{X}^{(j)} * \mathbf{k}_{high}) \downarrow 2 \quad (2)$$

where  $\mathbf{k}_{low} = [1/\sqrt{2}, 1/\sqrt{2}]$  and  $\mathbf{k}_{high} = [1/\sqrt{2}, -1/\sqrt{2}]$  are fixed Haar filters.

The notation  $\downarrow 2$  represents the down-sampling layer which is implemented as a strided convolution (stride=2, padding=0). Since  $L^{(j)}$  may not be divisible by 2 (i.e., non-power-of-2 length), there might be some troubles in handling arbitrary sequence lengths. To avoid wasting information due to boundary effects, we explicitly apply symmetric padding on the tail of sequence before convolution. The down-sampled outputs are always accurately calibrated without cutting information at boundaries. Finally, we transmit the approximation coefficients  $\mathbf{X}^{(j+1)}$  to the upper layers while suppressing the high-frequency noises without introducing aliasing artifacts.

### 3.3 Dual-Path Trend Mixer

Time series data inherently exhibits a kinematic duality: a deterministic macroscopic trend (e.g., seasonality, monotonic growth) superimposed with stochastic microscopic fluctuations. Single-stream models are challenged to model these two components in a disentangled manner. We design the Dual-Path Trend Mixer at each scale  $j$ , including Global Trend Modeling Path and Local Evolution Path.

#### 3.3.1 Global Trend Path

Such path is intended to model low-frequency, global trajectory of time series. Linear models have shown to be more robust when extrapolating monotonic trends compared to deep neural networks which are prone to fit local noises. Thus, we apply channel-independent linear projection to directly map entire historical sequence at scale  $j$  onto prediction horizon:

$$\mathbf{H}_{global}^{(j)} = \mathbf{W}_{lin}^{(j)} \cdot \text{Flatten}(\mathbf{X}^{(j)}) + \mathbf{b}_{lin}^{(j)} \quad (3)$$

where  $\mathbf{W}_{lin}^{(j)} \in \mathbb{R}^{T \times (L^{(j)} \cdot C)}$  is the weight matrix sharing parameters across channels. This component serves as a global constraint, ensuring that the forecasted trajectory adheres to the global inertia of the historical data.

### 3.3.2 Local Evolution Path

Although the linear path provides the global direction, it is still unable to capture the second-order non-linear dynamics and local semantic variation. This path designs a Patching + MLP-Mixer module to capture local evolution path.

- Patching Operation: The input sequence  $\mathbf{X}^{(j)} \in \mathbb{R}^{L^{(j)} \times C}$  is segmented into  $N_p$  non-overlapping patches of length  $P$ . This transforms the 2D temporal variates into a 3D tensor  $\mathcal{Z}_{patch} \in \mathbb{R}^{N_p \times P \times C}$ , preserving the local temporal context within each patch.
- MLP-Mixer Backbone: We project the patches into a hidden dimension  $D$  and process them through a stack of Mixer layers. Each layer consists of two distinct MLP blocks:

$$\mathbf{U} = \mathbf{Z} + \text{MLP}_{token}(\text{LayerNorm}(\mathbf{Z})) \quad (4)$$

$$\mathbf{Z}_{out} = \mathbf{U} + \text{MLP}_{channel}(\text{LayerNorm}(\mathbf{U})) \quad (5)$$

The Token-Mixing MLP captures temporal dependencies across different time segments, while the Channel-Mixing MLP extracts feature correlations within each patch.

- Projection: The output features are flattened and projected to the target horizon  $T$  via a linear head, yielding the local component  $\mathbf{H}_{local}^{(j)}$ .

The prediction at scale  $j$  is synthesized by fusing the deterministic and stochastic components via learnable scalar gates  $w_g$  and  $w_l$ :

$$\hat{\mathbf{Y}}^{(j)} = w_g \cdot \mathbf{H}_{global}^{(j)} + w_l \cdot \mathbf{H}_{local}^{(j)} \quad (6)$$

This formulation can be interpreted as a residual learning framework, where the MLP path learns the non-linear residuals that the linear path cannot approximate.

## 3.4 Adaptive Multi-Scale Fusion

A critical observation often overlooked in prior work is that the optimal scales are heterogeneous across different variables: e.g., for a voltage sensor signal, the signal may be mostly contaminated by high-frequency noises and hence needs to be predicted based on relatively coarse scales ( $j = 2, 3$ ); while for a traffic flow signal containing sharp and informative peaks, the model should focus on finer scales ( $j = 0, 1$ ). In this work, we explore the channel-aware adaptive fusion mechanism. Suppose  $\mathcal{A} \in \mathbb{R}^{(N+1) \times C}$  is a learnable attention parameter matrix, then the fusion weights  $\mathbf{W}_{fusion}$  will be rescaled by Softmax over scale dimension:

$$\mathbf{W}_{fusion}[k, c] = \frac{\exp(\mathcal{A}_{k,c})}{\sum_{n=0}^N \exp(\mathcal{A}_{n,c})} \quad (7)$$

where  $k$  denotes the scale index and  $c$  denotes the channel index. The final prediction  $\hat{\mathbf{Y}}$  is computed as the weighted sum of forecasts from all scales:

$$\hat{\mathbf{Y}}_{norm} = \sum_{j=0}^N \mathbf{W}_{fusion}[j] \odot \hat{\mathbf{Y}}^{(j)} \quad (8)$$

where  $\odot$  denotes element-wise multiplication with broadcasting. With such design, our model is able to automatically focus on the best frequency band for each variable and hence acts as a learnable spectral filter.

### 3.5 Complexity Analysis

We present a formal complexity analysis to show that DPWMixer remains linear despite its multi-scale design. Suppose  $L$  is the input length and  $T$  is the prediction horizon.

- Scale 0: The complexity is dominated by Mixer ( $\mathcal{O}(L \cdot D^2)$  given fixed patch size) and Linear path ( $\mathcal{O}(L \cdot T)$ ).
- Scale  $j$ : The sequence length becomes  $L/2^j$ . Therefore, the complexity of  $j$ -th mixer is  $\mathcal{O}(\frac{L}{2^j} \cdot D^2)$ .

The total complexity is the summation of geometric series:

$$\mathcal{C}_{total} = \sum_{j=0}^N \mathcal{C}_{total}(\frac{L}{2^j}) \approx \mathcal{C}_{total}(L) \cdot (1 + \frac{1}{2} + \frac{1}{4} + \dots) \leq 2 \cdot \mathcal{C}_{total}(L) \quad (9)$$

Therefore, the derivation shows that when we add the multi-scale pyramid, the computational cost only increases by a constant (at most 2 times) compared with a single-scale model. Thus, DPWMixer is proved to have  $\mathcal{O}(L)$  complexity. While the complexity of Transformer models is  $\mathcal{O}(L^2)$  or  $\mathcal{O}(L \log L)$ , DPWMixer is more efficient, especially when the look-back window is longer.

## 4 Experiments

To comprehensively evaluate the efficacy of DPWMixer, we perform extensive experiments on eight real-world benchmarks. Our evaluation is not limited to simple comparisons. We empirically validate our theoretical claims on orthogonal decomposition, dual-path modeling, and adaptive fusion, respectively.

### 4.1 Experimental Setup

#### 4.1.1 Datasets

We employ eight multivariate time series benchmarks from different domains. These datasets are challenging in different aspects in terms of periodicity, trend, and noise. We summarize statistical information in Table 1.

- *ETT* (*Electricity Transformer Temperature*) Comprising four subsets: ETTh1, ETTh2 (hourly) and ETTm1, ETTm2 (15-minutely). This dataset records the load and oil temperature of electricity transformer. This dataset has strong long-term seasonal periodicity but has high local volatility due to the change of load.
- *Electricity* This dataset records the hourly electricity consumption of 321 clients. Compared with ETT, the consumption behavior of each client has shifted to a different style over time. This set tests the robustness of models against distribution shifts.
- *Weather* This dataset records 21 different meteorological factors (e.g., air temperature, humidity) at every 10 minutes. Weather is known to be a very chaotic factor. It follows thermodynamic rules, local changes are stochastic and less trended compared with energy.
- *Exchange* This dataset records the daily exchange rate of 8 countries. This dataset is dominated by aleatoric uncertainty and has no obvious seasonal periodicity. This dataset is a stress test for financial trend.
- *Traffic* This dataset records the road occupancy rate from 862 sensors over San Francisco Bay Area freeways. This is a typical high-frequency dataset. Occupancy has sharp and non-linear peaks (rush hours) and is obviously periodic every day and every week.

**Table 1** STATISTICS OF DATASETS IN THE EXPERIMENT

Dataset	#Variates	Timesteps	Frequency	Domain
ETTh1	7	17,420	1-hour	Energy
ETTh2	7	17,420	1-hour	Energy
ETTm1	7	69,680	15-min	Energy
ETTm2	7	69,680	15-min	Energy
Electricity	321	26,304	1-hour	Energy
Weather	21	52,696	10-min	Weather
Exchange	8	7,588	Daily	Economy
Traffic	862	17,544	1-hour	Transportation

#### 4.1.2 Baselines

We benchmark DPWMixer against 9 SOTA models, categorizing baselines by their core architectures to ensure a fair evaluation. **Transformer-based Models:**

- *iTransformer* [19] Reverse the mainstream Transformer by embedding the whole time series of each variate into one token. Then apply attentions over tokens among different variates to explicitly capture the multivariate correlation. Finally, it achieves the state-of-the-art performance.
- *PatchTST* [12] Divide the time series into sub-series patches to keep the local semantic information and reduce the computational cost. Then, it adopts a channel-independent strategy to allow a shared Transformer backbone to learn the generalizable temporal pattern from different variables.

- *Crossformer* [18] Employ a Two-Stage Attention (TSA) mechanism to capture the dependency of target series from other patches spanned over both time and dimension. It effectively routes the information among patches from different variates. Different from channel-independent methods which model the interaction between channels as modeling the interaction between dimensions, patches from different variates are highly correlated.
- *FEDformer* [16] Adopt seasonal-trend decomposition and frequency domain analysis. It uses Fourier/Wavelet transform to extract k significant frequency components and then adopts the learned components as inputs of linearformer. Therefore, the seasonal-trend decomposition makes FEDformer linear complexity attention while capturing the global periodic pattern effectively.
- *TimeXer* [31] Adopts the patch-wise inverted architecture to leverage exogenous variables effectively. It utilizes an Endogenous-Exogenous Attention mechanism that adaptively aligns and aggregates information from external series to the target series, ensuring that only beneficial correlations are captured while filtering out irrelevant noise.

#### MLP/Linear-based Models:

- *TimeMixer* [27] Design a Past-Decomposable-Mixing (PDM) architecture. It uses average pooling to down-sample the series into multi-scale pyramid and then apply MLP-based mixing operations on the series at different temporal resolutions. It may lead to aliasing since it mixes information from different temporal resolutions.
- *DLinear* [13] Decompose time series into trend and seasonal components using moving average kernels. It models each component with a simple single-layer linear mapping. It shows that preserving the temporal order is more important than a complex network design on each time series.
- *FITS* [32] FITS uses frequency domain interpolation to achieve state-of-the-art performance in time series forecasting and anomaly detection tasks. This lightweight model is parameter-efficient compared with complex Transformers and more efficient, universal, and adaptive to different variable input lengths and analytical tasks.

#### CNN-based Models:

- *TimesNet* [24]: Rearranges 1D time series into 2D tensors according to multi-periodicity analysis. A parameter-efficient Inception block is used to model intra-period and inter-period variations in a complementary way, which efficiently extends 2D visual backbones to TS forecasting.

##### 4.1.3 Implementation Details

All experiments are conducted based on PyTorch in a cluster with NVIDIA RTX 4090 GPUs. Following the standard protocol of Long-Term Time Series Forecasting LTSF, we also test the forecasting performance on another three long-term prediction horizons  $T \in \{96, 192, 336, 720\}$  to cover the forecasting range from short to long.

For reproducibility and fairness, we follow the following strict protocols. Data Splitting. Following the standard benchmarks in Informer [10], we follow the 7:1:2

ratio to split all datasets into training set, validation set and testing set. For Traffic dataset, we follow 6:2:2 ratio since it could reflect the spatial-temporal dynamics more complicatedly.  $L = 96$ .

We adopt ADAM optimizer with cosine annealing learning rate scheduler. The learning rate will be selected from  $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$ , through grid search, and the batch size will be chosen from 16, 32, 64 according to the size of the corresponding dataset. To avoid over-fitting, we adopt an Early Stopping mechanism on top of the validation loss, which is implemented with patience of 5 epochs. The maximum epoch is limited in 10. The number of scales  $N$  in wavelet decomposition is set to 3, which is a rational design to balance the expansion of the receptive field and the preservation of resolution. We unify the patch length  $P$  in 16 and set the latent hidden dimension  $D$  in 128. All results reported in the paper are the mean of 3 runs with different random seeds.

**Evaluation Metrics.** We adopt two standard quantitative metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE). MSE is the mean of the squared error between the predicted value and real value. MSE will be strictly punished on the large error to identify large deviation in volatile series. While MAE is a quite robust way to evaluate the general performance of the forecast on the basis of absolute error. For both of MSE and MAE, the smaller value means the better performance.

## 4.2 Main Results

As shown in Table 2, DPWMixer demonstrates state-of-the-art performance in a variety of multivariate long-term time series forecasting tasks. Compared with other strong baselines, i.e., current SOTA Transformer-based method iTransformer and multi-scale MLP method TimeMixer, our method obtains the lowest MSE and MAE in the majority of cases.

Specifically, DPWMixer ranks first in 44 out of 64 experimental settings (aggregating across 8 datasets and 4 prediction horizons), showcasing its robust generalization capability. On datasets characterized by complex periodic patterns and rapid fluctuations, such as Weather and Electricity, our model achieves remarkable improvements. For instance, on the ETTm2 dataset (horizon 96), DPWMixer reduces MSE by 10.7% compared to iTransformer (0.169 vs. 0.180) and by 6.50 % compared to TimeMixer (0.169 vs. 0.175). These results validate our hypothesis that: the lossless Haar wavelet decomposition effectively preserves high-frequency details (e.g., traffic peaks) that are often smoothed out by the average pooling used in TimeMixer, while the dual-path mixer captures local non-linear dynamics better than the rigid attention mechanisms in Transformers.

Furthermore, on datasets with strong global trends but significant noise, such as Exchange and Weather, DPWMixer continues to lead. In the Exchange dataset, our model achieves an average MSE of 0.171 (horizon 192), significantly outperforming iTransformer (0.177) and TimeMier (0.187). These gains stem from the Global Linear Path in our architecture, which acts as a stable anchor for macroscopic trends, preventing the model from overfitting to aleatoric uncertainty—a common pitfall for deep models like iTransformer in financial data.

**Table 2** Long-term forecasting performance comparison with an input length of  $L = 96$  for prediction horizons  $T \in \{96, 192, 336, 720\}$ . Best and second-best scores are marked in **bold** and with an underline, respectively. ‘Avg’ denotes the average performance.

Models	DPWMixer			iTTransformer			TimeXer			FITS			PatchTST			Crossformer			TimesNet			DLinear			FEDformer		
	(Ours)			2024			2024			2024			2023			2023			2023			2023			2022		
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	MSE	MAE	
96	<b>0.373</b>	<b>0.391</b>	0.386	0.405	0.375	0.4	0.382	0.403	0.450	0.442	0.46	0.447	0.423	0.448	0.384	0.402	0.407	0.412	0.395	0.424							
192	<b>0.426</b>	<b>0.419</b>	0.441	0.512	<u>0.429</u>	<u>0.421</u>	0.429	0.453	0.522	0.481	0.477	0.429	0.471	0.474	0.436	0.446	0.441	0.411	0.469	0.470							
ETTh1	<b>0.462</b>	<b>0.441</b>	0.487	0.458	0.484	0.458	<u>0.468</u>	<u>0.448</u>	0.553	0.501	0.546	0.496	0.470	0.491	0.491	0.469	0.489	0.547	0.495								
720	<u>0.491</u>	<u>0.474</u>	0.503	0.491	0.498	0.482	<b>0.469</b>	<b>0.461</b>	0.545	0.517	0.544	0.517	0.653	0.621	0.521	0.500	0.513	0.510	0.598	0.544							
Avg	<b>0.438</b>	<b>0.431</b>	0.454	0.447	0.440	0.438	<u>0.437</u>	<u>0.437</u>	0.517	0.485	0.516	0.484	0.529	0.522	0.458	0.450	0.461	0.457	0.498	0.484							
96	<b>0.282</b>	<b>0.326</b>	0.286	<u>0.338</u>	0.289	0.342	0.308	0.355	0.314	0.359	0.745	0.584	0.400	0.440	0.340	0.374	0.340	0.394	0.358	0.397							
192	<b>0.361</b>	<b>0.387</b>	0.380	0.400	0.378	0.397	0.363	0.389	0.406	0.414	0.793	0.585	0.877	0.656	0.402	0.452	0.419	0.479	0.414	0.439							
ETTh2	<u>0.409</u>	<u>0.421</u>	0.428	0.432	<b>0.386</b>	<b>0.414</b>	0.414	0.423	0.446	0.447	0.927	0.643	1.043	0.731	0.452	0.482	0.591	0.541	0.496	0.487							
720	<u>0.423</u>	<b>0.426</b>	0.427	0.445	<b>0.412</b>	0.434	<u>0.414</u>	<u>0.432</u>	0.475	0.471	1.043	0.636	1.104	0.763	0.462	0.468	0.661	0.661	0.463	0.474							
Avg	0.368	<b>0.39</b>	0.383	0.497	<b>0.364</b>	0.398	0.383	0.396	0.410	0.422	0.878	0.612	0.841	0.642	0.414	0.427	0.563	0.519	0.437	0.449							
96	<b>0.316</b>	<b>0.346</b>	0.334	0.368	0.320	0.357	0.318	0.356	0.350	0.370	0.352	0.374	0.404	0.426	0.338	0.375	0.346	0.374	0.379	0.419							
192	<b>0.356</b>	<b>0.373</b>	0.404	0.393	<u>0.361</u>	0.393	0.362	0.383	0.392	0.393	0.387	0.404	0.450	0.451	0.374	0.387	0.381	0.391	0.389	0.387							
ETTh1	<b>0.378</b>	<b>0.401</b>	0.426	0.420	<u>0.39</u>	<u>0.404</u>	0.395	0.407	0.424	0.414	0.421	0.414	0.532	0.515	0.410	0.411	0.415	0.415	0.445	0.459							
720	0.452	<b>0.437</b>	0.491	0.459	0.454	<u>0.441</u>	<u>0.441</u>	0.441	0.484	<u>0.447</u>	0.462	0.449	0.666	0.589	0.478	0.450	0.473	0.451	0.543	0.490							
Avg	<b>0.375</b>	<b>0.389</b>	0.407	0.410	<u>0.381</u>	0.395	0.382	<u>0.391</u>	0.412	0.406	0.406	0.407	0.513	0.495	0.400	0.406	0.404	0.408	0.448	0.452							
96	0.169	<b>0.255</b>	0.180	0.264	0.175	0.258	0.171	<u>0.256</u>	0.184	0.268	0.183	0.270	0.287	0.366	0.187	0.267	0.193	0.286	0.203	0.287							
192	<b>0.232</b>	<b>0.297</b>	0.250	0.309	<u>0.237</u>	<u>0.299</u>	0.237	0.299	0.249	0.307	0.255	0.314	0.414	0.492	0.249	0.309	0.284	0.361	0.269	0.328							
ETTh2	<b>0.293</b>	<b>0.335</b>	0.311	0.348	0.298	0.340	<u>0.296</u>	<u>0.338</u>	0.309	0.343	0.309	0.347	0.597	0.542	0.321	0.331	0.382	0.429	0.325	0.366							
720	<b>0.389</b>	<b>0.391</b>	0.407	0.407	<u>0.391</u>	<u>0.392</u>	0.392	0.394	0.409	0.398	0.412	0.404	1.730	1.042	0.408	0.403	0.558	0.525	0.421	0.415							
Avg	<b>0.271</b>	<b>0.319</b>	0.288	0.332	0.275	0.323	<u>0.274</u>	<u>0.322</u>	0.287	0.329	0.290	0.334	0.757	0.610	0.291	0.333	0.354	0.402	0.305	0.349							

**Table 2** (continued)

Models	DPWMixer			iTrausformer			TimeMixer			TimeXer			FITS			PatchTST			Crossformer			TimesNet			DLinear			FEDformer		
	(Ours)			2024			2024			2024			2024			2023			2023			2023			2023			2022		
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	MSE	MAE	MSE	MAE		
Electricity	96	<u>0.152</u>	<u>0.245</u>	<b>0.148</b>	<b>0.24</b>	0.153	0.247	0.182	0.278	0.203	0.282	0.190	0.296	0.219	0.314	0.168	0.272	0.210	0.305	0.169	0.273									
	192	<b>0.160</b>	<b>0.251</b>	<u>0.162</u>	<u>0.253</u>	0.166	0.256	0.199	0.263	0.201	0.283	0.196	0.304	0.231	0.322	0.184	0.298	0.210	0.305	0.201	0.315									
	336	<b>0.173</b>	<b>0.266</b>	<u>0.178</u>	<u>0.269</u>	0.185	0.277	0.193	0.312	0.215	0.297	0.217	0.319	0.246	0.337	0.198	0.300	0.223	0.319	0.200	0.304									
	720	<b>0.223</b>	<b>0.306</b>	<u>0.225</u>	<u>0.310</u>	0.225	0.317	0.233	0.312	0.257	0.330	0.258	0.352	0.280	0.363	0.220	0.320	0.258	0.350	0.246	0.355									
Avg		<b>0.177</b>	<b>0.267</b>	<u>0.178</u>	<u>0.271</u>	0.182	0.274	0.202	0.290	0.219	0.298	0.216	0.318	0.244	0.334	0.192	0.304	0.225	0.319	0.214	0.327									
96		<u>0.166</u>	<u>0.209</u>	0.174	0.212	0.163	0.209	<b>0.157</b>	<b>0.205</b>	0.174	0.224	0.186	0.227	0.195	0.271	<u>0.172</u>	0.220	0.195	0.252	0.217	0.296									
192		<b>0.202</b>	<b>0.238</b>	<u>0.221</u>	<u>0.254</u>	0.208	<u>0.250</u>	<u>0.204</u>	<u>0.247</u>	0.223	0.264	0.234	0.265	0.209	0.277	0.219	0.261	0.237	0.282	0.276	0.336									
Weather		<u>0.258</u>	<b>0.275</b>	0.278	0.296	<b>0.251</b>	<u>0.278</u>	0.261	0.290	0.28	0.302	0.284	0.301	0.273	0.332	0.280	0.306	0.282	0.331	0.339	0.380									
720		<b>0.336</b>	<b>0.338</b>	0.358	0.347	<u>0.339</u>	<u>0.341</u>	0.340	0.341	0.357	0.351	0.356	0.349	0.379	0.401	0.365	0.359	0.345	0.403	0.428										
Avg		<b>0.240</b>	<b>0.265</b>	0.258	0.278	0.250	0.283	0.240	<u>0.271</u>	0.258	0.285	0.265	0.285	0.264	0.32	0.259	0.287	0.265	0.315	0.309	0.360									
Exchange	96	<b>0.084</b>	<b>0.201</b>	0.086	0.206	0.090	0.235	<u>0.085</u>	<u>0.204</u>	0.085	0.236	0.088	0.205	0.256	0.367	0.107	0.234	0.088	0.218	0.148	0.278									
	192	<b>0.171</b>	<b>0.297</b>	<u>0.177</u>	<u>0.299</u>	0.187	0.343	0.181	0.302	0.193	0.392	0.176	0.299	0.470	0.509	0.226	0.344	0.176	0.315	0.271	0.315									
	336	<u>0.341</u>	<b>0.413</b>	<b>0.331</b>	<u>0.417</u>	0.353	0.473	0.363	0.435	0.386	0.503	0.301	0.397	1.268	0.883	0.367	0.448	0.313	0.427	0.460	0.427									
	720	<b>0.842</b>	<b>0.688</b>	<u>0.847</u>	<u>0.691</u>	0.934	0.761	0.930	0.727	1.023	0.862	0.901	0.714	1.767	1.068	0.964	0.746	0.839	0.695	1.195	0.695									
Avg		<b>0.359</b>	<b>0.399</b>	<u>0.36</u>	<u>0.403</u>	0.391	0.453	0.389	0.417	0.421	0.498	0.367	0.404	0.940	0.707	0.416	0.443	0.354	0.414	0.519	0.429									
Traffic	96	0.452	<u>0.277</u>	<b>0.395</b>	<b>0.271</b>	0.462	0.285	<u>0.428</u>	<b>0.271</b>	0.725	0.484	0.526	0.347	0.644	0.429	0.593	0.321	0.650	0.396	0.587	0.366									
	192	0.469	0.291	<b>0.417</b>	<b>0.276</b>	0.473	0.296	<u>0.448</u>	<u>0.282</u>	0.737	0.512	0.522	0.332	0.665	0.431	0.617	0.336	0.598	0.370	0.604	0.373									
	336	0.486	<u>0.295</u>	<b>0.433</b>	0.298	0.498	0.296	<u>0.473</u>	<b>0.289</b>	0.730	0.495	0.517	0.334	0.674	0.420	0.629	0.336	0.605	0.373	0.621	0.383									
	720	<u>0.503</u>	0.311	<b>0.467</b>	<b>0.302</b>	0.506	0.313	0.516	<u>0.307</u>	0.752	0.492	0.552	0.352	0.683	0.424	0.640	0.350	0.645	0.394	0.626	0.382									
Avg		0.477	0.293	<b>0.428</b>	<b>0.282</b>	0.484	0.298	<u>0.466</u>	<u>0.287</u>	0.466	0.495	0.529	0.341	0.667	0.426	0.62	0.336	0.625	0.383	0.610	0.376									
1st Count		<b>20</b>	<b>24</b>	7	5	4	1	3	4	0	0	0	0	0	0	0	0	0	0	0	0									

Notably, DPWMixer still maintains its performance gap when horizon becomes longer. For instance, on Electricity dataset when horizon reaches to the longest ( $T = 720$ ), our method could obtain  $\text{MSE}=0.223$  while the state-of-the-art iTransformer achieves 0.225. It shows that our hierarchical multi-scale design greatly alleviates the error accumulation issue in long-term forecasting. Moreover, the orthogonal wavelet decomposition enables the model to learn long-term trend information in coarser scales independently from high-frequency noisy parts so that the model could be more stable when predicting further into the future.

In summary, the experimental results have shown that, due to the orthogonal wavelet decomposition used for anti-aliasing and dual-path mixer used for harmonizing rigid and flexible modeling, DPWMixer sets a new state-of-the-art accuracy and robustness benchmark for long-term time series forecasting.

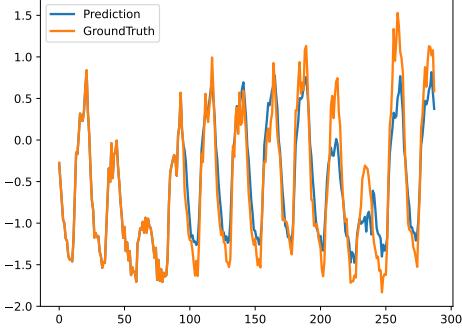
### 4.3 Visualized prediction results

To provide a more intuitive understanding of the model's behavior beyond aggregated metrics, we visualize the forecasting results of DPWMixer on Electricity dataset and other seven representative baselines. The Electricity dataset shows non-stationary dynamics and highly periodic behavior with both sharp cyclic changes and high-frequency fluctuations, which makes it an ideal dataset to evaluate the temporal fidelity of models. We visualize the forecasting results of DPWMixer on horizon  $T = 192$  and  $T = 336$  respectively with look-back window  $L = 96$ .

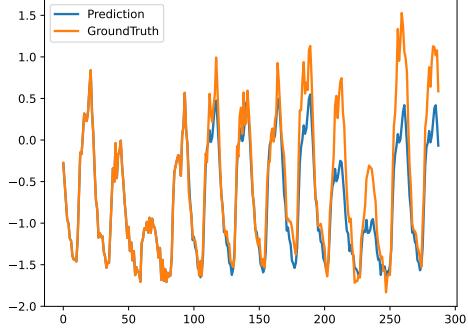
**Performance on Horizon  $T = 192$  (Figure 3):** In the medium-term prediction scenario, DPWMixer is almost aligned with the ground truth (Orange line). As shown in Figure 3 (a), our model could reconstruct the fine-grained temporal structure information, which is the sharp peaks and deep troughs of peak electricity load. Compared with other advanced models which could also capture the general periodicity of the time series, DPWMixer could further model the amplitude of extremum points more accurately. Due to the self-attention mechanism, the Transformer-based model would bring a smoothing effect on the original time series, which would lead to under-estimation in high volatility regions. While the Dual-Path design makes Local Evolution Path could focus on reconstructing these high-frequency information independently from global trend information, and thus it would not be smoothed down by other trend components.

**Robustness on Extended Horizon  $T = 336$  (Figure 4):** It is even more apparent that DPWMixer is superior to the other models when  $T = 336$ . It is a challenging task to forecast far into the future because the errors will accumulate over time. It is also a challenge to keep the phase shift stable over a long sequence. It is hard to keep structural consistency when forecasting over a long sequence as shown in Figure 4. DPWMixer (shown in Figure 4(a)) is very robust and can keep the tight phase and amplitude consistency over the long sequence and is able to accurately forecast the double-peak patterns.

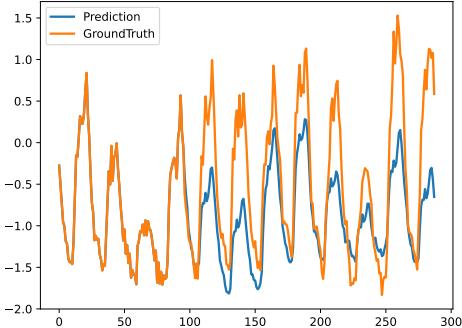
**Attribution to Architectural Innovations:** This visual superiority directly validates the efficacy of our proposed Lossless Haar Wavelet Pyramid. Unlike traditional average pooling, which may incur spectral aliasing and loss of high-frequency information, our orthogonal wavelet decomposition ensures that critical transient details



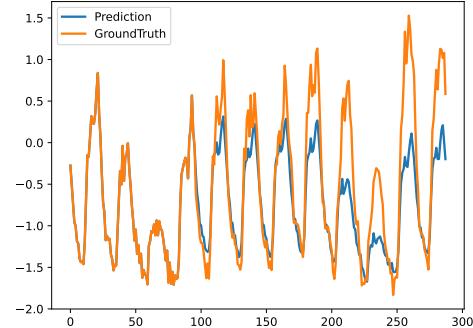
(a) Electricity-192-DPWMixer



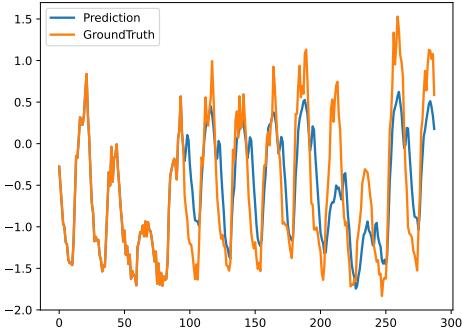
(b) Electricity-192-iTransformer



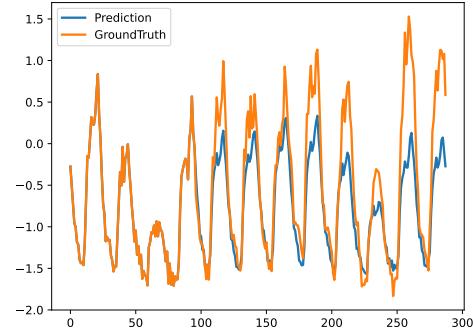
(c) Electricity-192-TimeMixer



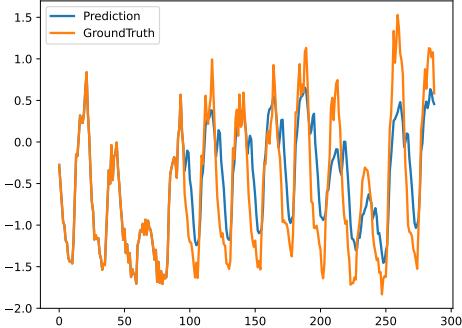
(d) Electricity-192-TimeXer



(e) Electricity-192-CrossFormer

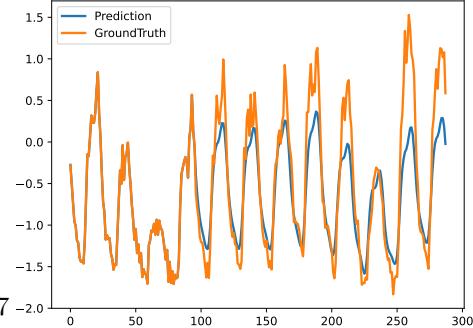


(f) Electricity-192-PatchTST



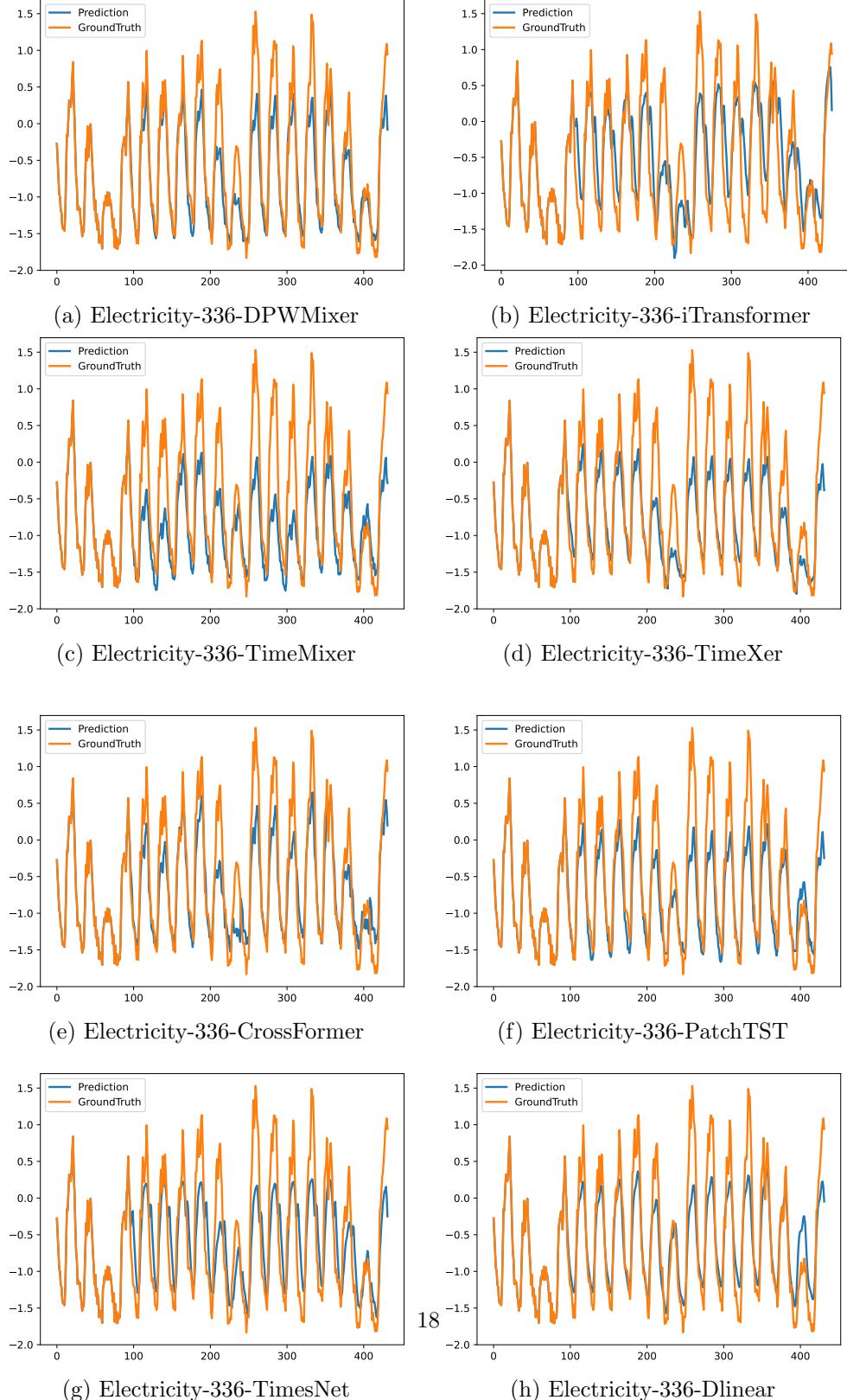
(g) Electricity-192-TimesNet

17



(h) Electricity-192-Dlinear

**Fig. 3** Visual comparison of forecasting performance on the Electricity dataset with a horizon of  $T = 192$ . DPWMixer (a) accurately captures cyclic patterns and maintains trend consistency over the extended horizon, exhibiting lower error accumulation than the baseline method.



**Fig. 4** Visual comparison of long-term forecasting results on the Electricity dataset ( $T = 336$ ). DPWMixer (a) accurately captures cyclic patterns and maintains trend consistency over the extended horizon, exhibiting lower error accumulation than the baseline methods.

are preserved and passed to the mixer layers. Furthermore, the Global Linear Path provides a stable anchor for the macroscopic trajectory, preventing the trend drift often observed in purely non-linear deep models over long horizons. Consequently, DPWMixer achieves a harmonized balance between trend stability and detail sensitivity, resulting in forecasts that are not only statistically accurate but also structurally faithful to the real-world signal dynamics.

#### 4.4 Ablation Studies

To rigorously verify the contribution of each component in DPWMixer, we conduct ablation studies on the Electricity and ETTm1 datasets. We analyze the impact of each component based on the results presented in Table 3.

1. Impact of Wavelet Decomposition (w/o Wavelet): Replacing the Haar wavelet with average pooling leads to a significant performance drop, particularly on the Electricity dataset (MSE +11.8%). This confirms that average pooling causes spectral aliasing, losing high-frequency load fluctuation details that are critical for accurate electricity forecasting.
2. Impact of Global Linear Path (w/o Global Path): Removing the linear path severely impacts long-term horizons. For ETTm1 at  $T = 720$ , the MSE degrades from 0.452 to 0.502. This validates the Global Path acts as a necessary anchor to prevent trend drifting in long sequence predictions.
3. Impact of Local Mixer Path (w/o Local Path): The removal of the mixer path results in the highest degradation in MAE for Electricity, indicating that the remaining linear component is too rigid to capture the complex, non-linear micro-dynamics of power consumption.
4. Impact of Adaptive Fusion: While the impact is smaller compared to other modules, the consistent degradation ( $\approx 2\text{--}3\%$ ) across all settings shows that statically aggregating multi-scale features is suboptimal compared to our channel-aware dynamic weighting strategy.

#### 4.5 Parameter Sensitivity Analysis

We investigate the impact of the wavelet decomposition depth, denoted as  $H$ , by varying it from 1 to 4 on the ETTh2 and Weather datasets. As illustrated in Figure 5, a consistent trend emerges where increasing the decomposition level initially leads to a significant reduction in Mean Squared Error (MSE) across all prediction horizons. The single-scale model ( $H = 1$ ) yields the poorest performance, attributed to its limited receptive field and inability to physically disentangle global trends from high-frequency noise. In contrast, deeper hierarchies enable the DPWMixer to capture multi-scale temporal dependencies more effectively, proving that the multi-resolution architecture provides a superior inductive bias for long-term forecasting.

A detailed analysis of individual datasets clarifies the source of these improvements. For ETTh2, the significant performance gain observed when increasing  $H$  from 1 to 3 suggests that coarse-grained coefficients at deeper levels effectively capture long-term trends, mitigating the drift often associated with single-stream models. Similarly, the Weather dataset benefits from multi-scale modeling, where separating high-frequency

**Table 3** Ablation study of DPWMixer components on Electricity and ETTm1 datasets. Ours represents the full model. The degradation row indicates the average percentage increase in MSE compared to the full model.

Dataset	Horizon	Ours (Full)		w/o Wavelet		w/o Global Path		w/o Local Path		w/o Fusion	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	<b>0.152</b>	<b>0.245</b>	0.169	0.262	0.159	0.252	0.174	0.269	0.156	0.248
	192	<b>0.160</b>	<b>0.251</b>	0.179	0.270	0.172	0.261	0.185	0.275	0.165	0.255
	336	<b>0.173</b>	<b>0.266</b>	0.194	0.288	0.184	0.278	0.199	0.291	0.178	0.272
	720	<b>0.223</b>	<b>0.306</b>	0.248	0.331	0.246	0.324	0.252	0.339	0.232	0.314
Avg. Degradation		-	-	+11.8%	+9.2%	+7.5%	+6.8%	+13.5%	+10.1%	+3.2%	+2.8%
ETTm1	96	<b>0.316</b>	<b>0.346</b>	0.335	0.362	0.329	0.355	0.342	0.375	0.321	0.351
	192	<b>0.356</b>	<b>0.373</b>	0.378	0.395	0.372	0.388	0.385	0.402	0.363	0.380
	336	<b>0.378</b>	<b>0.401</b>	0.405	0.422	0.415	0.435	0.410	0.431	0.386	0.409
	720	<b>0.452</b>	<b>0.437</b>	0.481	0.465	0.502	0.488	0.485	0.472	0.461	0.445
Avg. Degradation		-	-	+6.5%	+5.8%	+8.9%	+7.2%	+8.1%	+9.5%	+2.1%	+2.0%

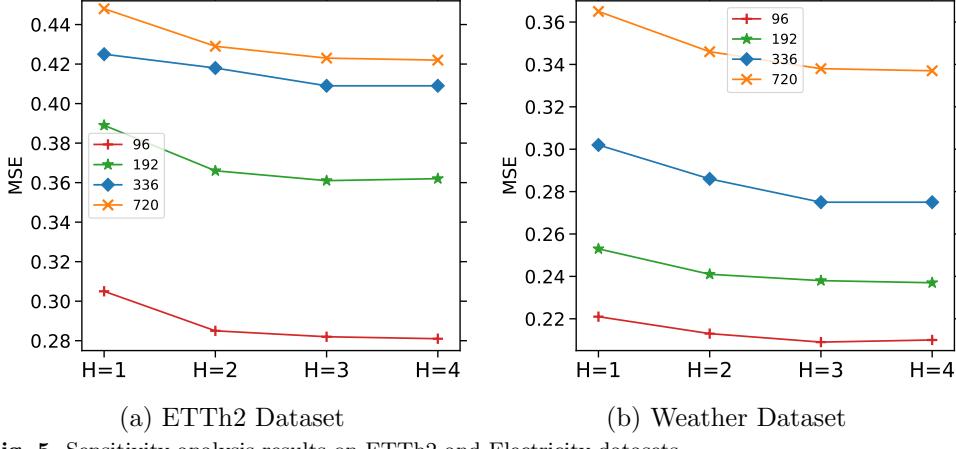


Fig. 5 Sensitivity analysis results on ETTh2 and Electricity datasets.

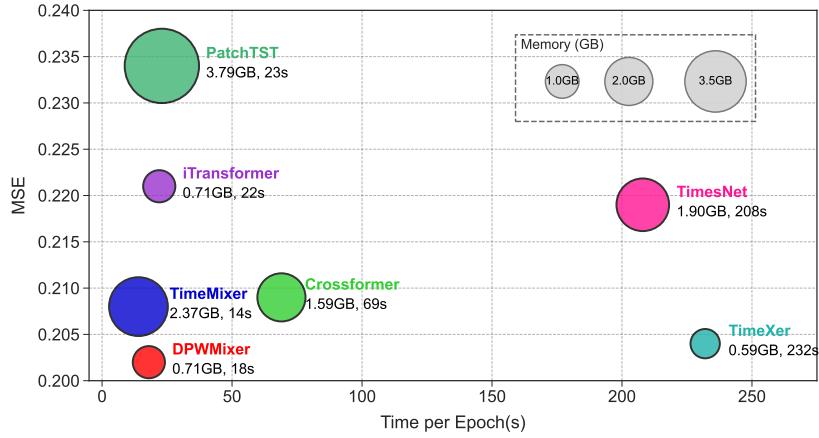
fluctuations from slow-varying components at  $H = 2$  and  $H = 3$  reduces prediction error. In both cases, the hierarchical structure utilizes wavelet orthogonality to preserve signal energy while expanding the temporal receptive field.

However, performance saturates or diminishes at  $H = 4$ . This suggests that excessive down-sampling causes information loss, as the sequence length at the coarsest level (e.g.,  $L/16$ ) becomes insufficient to represent complex patterns. Consequently, we select  $H = 3$  as the optimal setting, as it balances the need for a large receptive field with the retention of local details.

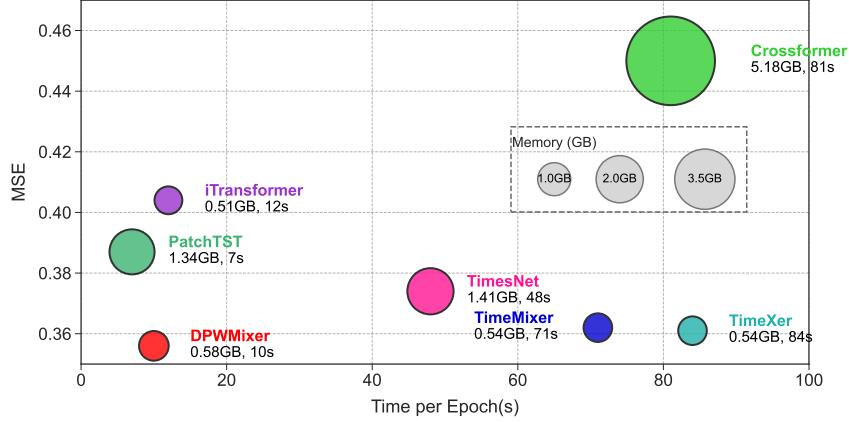
#### 4.6 Efficiency Analysis

For real-world deployment, the utility of a forecasting model is defined by the trade-off between predictive accuracy and computational cost. We analyze this balance in Figure 6 and 7, visualizing a multi-dimensional space where the x-axis represents training speed (time per epoch), the y-axis represents forecasting error (MSE), and the bubble size indicates the GPU memory footprint. Ideally, a model should occupy the bottom-left corner with a minimal bubble size, representing high speed, low error, and memory efficiency. We exclude ultra-lightweight models like DLinear and FITS, as their limited capacity to capture complex non-linear dynamics results in significant accuracy penalties on challenging datasets. Benchmarking high-capacity deep architectures against such simple mappings skews the visualization and obscures the critical trade-off between representation power and computational efficiency.

As illustrated, DPWMixer (represented by the red bubble) consistently achieves the optimal frontier. **Superiority over Transformers:** Due to the quadratic complexity  $\mathcal{O}(L^2)$  of attention mechanism, the deep architectures like Crossformer and PatchTST are very computationally expensive. While DPWMixer only consumes 0.58GB memory, it achieves 3.25 % MSE on ETTh1 dataset compared with Crossformer (5.18GB). The efficient Transformer decoding is also applied to our Dual-Path Mixers, which geometrically reduce sequence length to linear  $\mathcal{O}(L)$  by Haar Wavelet Pyramid. Our model decomposes the series into multi-scale representations and then



**Fig. 6** Efficiency comparison on the weather dataset ( $L = 96, T = 192$ ). The x-axis denotes training speed (s/epoch), the y-axis denotes MSE, and the bubble area represents GPU memory usage. DPWMixer (Red) achieves the optimal trade-off.



**Fig. 7** Efficiency comparison on the ETTm1 dataset ( $L = 96, T = 192$ ). The x-axis denotes training speed (s/epoch), the y-axis denotes MSE, and the bubble area represents GPU memory usage. DPWMixer (Red) achieves the optimal trade-off.

models them in the pyramid structure, which means the subsequent Dual-Path Mixers are actually working on much fewer tokens (at the deepest layers, the sequence length is  $L/2^N$  where  $N$  is the level of the pyramid).

**Superiority over Efficient Baselines:** Compared with TimesNet, which needs computationally expensive 2D convolutions on spatial and temporal modeling and variance modeling, DPWMixer is  $4\times$  faster. And TimeMixer is designed to be efficient using average pooling, while DPWMixer is even better in accuracy, which shows that

our orthogonal wavelet decomposition could preserve more high-frequency details than pooling, and achieves a better trade-off between speed and fidelity.

In conclusion, DPWMixer combines wavelets and dual-path mixing to capture both global trends and local details, achieving top-tier accuracy with high computational efficiency.

## 5 Conclusion

In this paper, we propose DPWMixer, a novel and computationally efficient framework for long-term time series forecasting. We identified the spectral aliasing issue in traditional pooling and proposed the Lossless Haar Wavelet Pyramid to achieve orthogonal decomposition of trends and details. Additionally, the proposed Dual-Path Trend Mixer effectively harmonizes rigid global trends and flexible local dynamics. Extensive experiments confirm that DPWMixer significantly outperforms Transformer-based baselines while maintaining linear complexity  $\mathcal{O}(L)$ , ensuring scalability for high-frequency forecasting tasks. Future work will focus on exploring learnable wavelet transforms to adaptively extract features and scaling this architecture for self-supervised pre-training.

**Acknowledgments** This work is supported by the National Natural Science Foundation of China (No. 62372366) .

**Data availability** Data is available on request.

## Declarations

**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- [1] Deb, C., Zhang, F., Yang, J., Lee, S.E., Shah, K.W.: A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews* **74**, 902–924 (2017)
- [2] Lago, J., Marcjasz, G., De Schutter, B., Weron, R.: Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy* **293**, 116983 (2021) <https://doi.org/10.1016/j.apenergy.2021.116983>
- [3] Abirami, S., Pethuraj, M., Uthayakumar, M., Chitra, P.: A systematic survey on big data and artificial intelligence algorithms for intelligent transportation system. *Case Studies on Transport Policy* **17**, 101247 (2024)

- [4] Cirstea, R.-G., Yang, B., Guo, C., Kieu, T., Pan, S.: Towards spatio-temporal aware traffic time series forecasting. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 2900–2913 (2022). IEEE
- [5] Wong, C.: How ai is improving climate forecasts. *Nature* **628**(8009), 710–712 (2024)
- [6] Qiu, X., Hu, J., Zhou, L., Wu, X., Du, J., Zhang, B., Guo, C., Zhou, A., Jensen, C.S., Sheng, Z., Yang, B.: Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proc. VLDB Endow.* **17**(9), 2363–2377 (2024)
- [7] Kim, J., Kim, H., Kim, H., Lee, D., Yoon, S.: A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges. *Artificial Intelligence Review* **58**(7), 1–95 (2025)
- [8] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L.: Transformers in time series: A survey. arXiv preprint arXiv:2202.07125 (2022)
- [9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
- [10] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 11106–11115 (2021)
- [11] Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems* **34**, 22419–22430 (2021)
- [12] Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. arXiv preprint arXiv:2211.14730 (2022)
- [13] Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 11121–11128 (2023)
- [14] Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., Yu, R.: Long-term forecasting with tide: Time-series dense encoder. *Trans. Mach. Learn. Res.* **2023** (2023)
- [15] Chan, F.K.-P., Fu, A.W.-C., Yu, C.: Haar wavelets for efficient similarity search of time-series: with and without time warping. *IEEE Transactions on Knowledge and Data Engineering* **15**(3), 686–705 (2003) <https://doi.org/10.1109/TKDE.2003.1198399>

- [16] Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: International Conference on Machine Learning, pp. 27268–27286 (2022). PMLR
- [17] Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A.X., Dustdar, S.: Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022 (2022). <https://openreview.net/forum?id=0EXmFzUn5I>
- [18] Zhang, Y., Yan, J.: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: The Eleventh International Conference on Learning Representations (2023)
- [19] Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M.: itransformer: Inverted transformers are effective for time series forecasting. arXiv preprint arXiv:2310.06625 (2023)
- [20] Li, Z., Qi, S., Li, Y., Xu, Z.: Revisiting Long-term Time Series Forecasting: An Investigation on Linear Mapping (2023). <https://arxiv.org/abs/2305.10721>
- [21] Das, A., Kong, W., Leach, A., Mathur, S.K., Sen, R., Yu, R.: Long-term forecasting with tiDE: Time-series dense encoder. Transactions on Machine Learning Research (2023)
- [22] Ekambaram, V., Jati, A., Nguyen, N., Sinthong, P., Kalagnanam, J.: Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In: Singh, A.K., Sun, Y., Akoglu, L., Gunopulos, D., Yan, X., Kumar, R., Ozcan, F., Ye, J. (eds.) Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023, pp. 459–469 (2023). <https://doi.org/10.1145/3580305.3599533>
- [23] Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., Xu, Q.: Scinet: Time series modeling and forecasting with sample convolution and interaction. Advances in Neural Information Processing Systems **35**, 5816–5828 (2022)
- [24] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: Timesnet: Temporal 2d-variation modeling for general time series analysis. In: The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023 (2023). [https://openreview.net/forum?id=ju\\_Uqw384Oq](https://openreview.net/forum?id=ju_Uqw384Oq)
- [25] Zhou, C., Jiang, K., Liu, Y., Che, C., Zhang, Q.: MSTF: enhancing long-term forecasting with multi-scale temporal fusion in time series forecasting. J. Supercomput. **81**(9), 1082 (2025) <https://doi.org/10.1007/S11227-025-07572-5>
- [26] Zhao, S., Lu, X., Zhao, S.: Scalemixnet: Adaptive multi-scale time-frequency fusion with hybrid loss for time series forecasting. J. Supercomput. **81**(15), 1382

- (2025) <https://doi.org/10.11007/S11227-025-07869-5>
- [27] Wang, S., Wu, H., Shi, X., Hu, T., Luo, H., Ma, L., Zhang, J.Y., ZHOU, J.: Timemixer: Decomposable multiscale mixing for time series forecasting. In: The Twelfth International Conference on Learning Representations (2024). <https://openreview.net/forum?id=7oLshfEIC2>
  - [28] Zhou, T., Ma, Z., Wen, Q., Sun, L., Yao, T., Yin, W., Jin, R., *et al.*: Film: Frequency improved legendre memory model for long-term time series forecasting. Advances in neural information processing systems **35**, 12677–12690 (2022)
  - [29] Woo, G., Liu, C., Sahoo, D., Kumar, A., Hoi, S.: Etsformer: Exponential smoothing transformers for time-series forecasting. arXiv preprint arXiv:2202.01381 (2022)
  - [30] Sundararajan, D.: Discrete Wavelet Transform: a Signal Processing Approach, (2016)
  - [31] Wang, Y., Wu, H., Dong, J., Qin, G., Zhang, H., Liu, Y., Qiu, Y., Wang, J., Long, M.: Timexer: Empowering transformers for time series forecasting with exogenous variables. Advances in Neural Information Processing Systems **37**, 469–498 (2024)
  - [32] Xu, Z., Zeng, A., Xu, Q.: FITS: modeling time series with 10k parameters. In: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024 (2024). <https://openreview.net/forum?id=bWcnvZ3qMb>