Wavelet-based Disentangled Adaptive Normalization for Non-stationary Time Series Forecasting

Junpeng Lin[1], Tian Lan[1], Bo Zhang[1], Ke Lin[2], Dandan Miao[2], Huiru He[3], Jiantao Ye[3], Yan-fu Li[1], Chen Zhang[1]*

[1]Department of Industrial Engineering, Tsinghua University
[2]Shanghai Research Center, Huawei Technologies
[3]Songshan Lake Research Center, Huawei Technologies
junpenglin2638@gmail.com, lant23@mails.tsinghua.edu.cn,
{bz16328,zhangchen01,liyanfu}@tsinghua.edu.cn, {linke2,miaodandan,hehuiru,yejiantao}@huawei.com

Abstract

Forecasting non-stationary time series is a challenging task because their statistical properties often change over time, making it hard for deep models to generalize well. Instance-level normalization techniques can help address shifts in temporal distribution. However, most existing methods overlook the multi-component nature of time series, where different components exhibit distinct non-stationary behaviors. In this paper, we propose Wavelet-based Disentangled Adaptive Normalization (WDAN), a model-agnostic framework designed to address non-stationarity in time series forecasting. WDAN uses discrete wavelet transforms to break down the input into low-frequency trends and high-frequency fluctuations. It then applies tailored normalization strategies to each part. For trend components that exhibit strong non-stationarity, we apply first-order differencing to extract stable features used for predicting normalization parameters. Extensive experiments on multiple benchmarks demonstrate that WDAN consistently improves forecasting accuracy across various backbone models. Code is available at this repository: https://github.com/MonBG/WDAN.

Introduction

Time series forecasting is an essential task in many fields, such as energy (Singh et al. 2013), economics (Ahmadi et al. 2019), transportation (Yu, Yin, and Zhu 2018), and healthcare (Kaushik et al. 2020). While deep learning models have made significant progress in capturing temporal patterns (Zhou et al. 2022; Zhang and Yan 2023; Kitaev, Kaiser, and Levskaya 2020), real-world time series remain difficult to predict due to their non-stationary characteristics. In non-stationary time series, statistical properties such as mean and variance change over time, which hinders the generalization of deep learning models (Lu et al. 2018; Li et al. 2022). While most current research focuses on capturing complex patterns through sophisticated architectures, the non-stationarity issue has not been fully addressed.

Our work is motivated by the observation that time series typically comprise multiple components exhibiting distinct characteristics. For example, real-world sequences often include slow-moving trends, periodic patterns, and high-frequency noise (RB 1990). Each of these components has its unique non-stationary behaviors and thus requires specialized handling. Traditional methods that treat the entire time series as a single entity often overlook these distinct behaviors. Time-frequency analysis provides an effective way to tackle this issue by breaking down a time series into separate components at different temporal scales, thereby revealing the underlying structure of non-stationarity.

Fig. 1 presents a forecasting example to illustrate our opinions. Decomposing the non-stationary series (Fig. 1(a)) via discrete wavelet transform (Chaovalit et al. 2011; Huang and Hsieh 2002) reveals that non-stationarity is primarily driven by the low-frequency trend (Fig. 1(b)), while the high-frequency residuals (Fig. 1(c)) remain relatively stable. Furthermore, applying first-order differencing to the trend component (Fig. 1(d)) significantly stabilizes its statistics compared to the raw trend, suggesting that differenced features can facilitate the modeling of non-stationary dynamics. Conventional normalization methods typically apply the same transformation to the entire sequence, thereby failing to capture the distinct behaviors of different components. Such limitations motivate the need for a more fine-grained normalization approach that leverages the inherent structure of time series.

Researchers have tried various approaches to handle non-stationarity, including differencing, seasonal decomposition and domain-specific feature engineering (Box et al. 2015; Liu et al. 2023a). Recently, normalization techniques (Kim et al. 2021; Fan et al. 2023; Liu et al. 2023b; Dai et al. 2024) have gained popularity due to their simplicity and compatibility with many deep learning models. Instance normalization methods help reduce distribution shifts by adjusting input statistics during training and testing. However, current normalization methods extract statistics directly from the raw sequence without considering the multi-component nature of time series. As a result, the statistics obtained blend information from components with different characteristics, which can lead to information loss during the normalization process.

To overcome these limitations, we propose Wavelet-based Disentangled Adaptive Normalization (WDAN), a model-agnostic framework for non-stationary time series forecast-

*Corresponding author

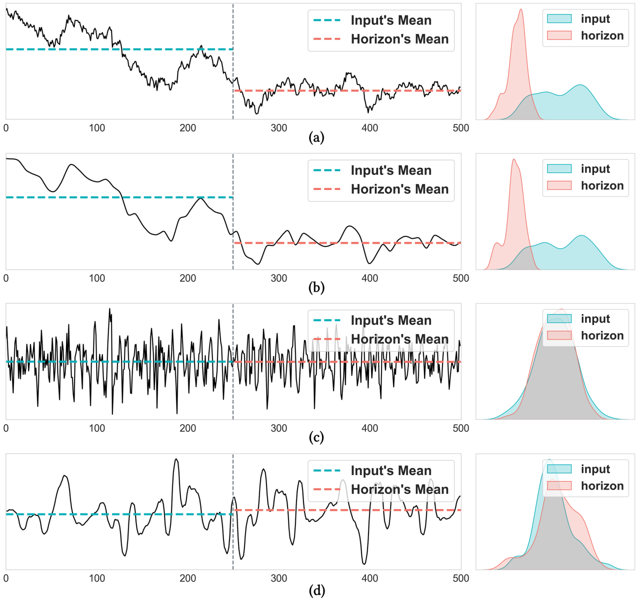


Figure 1: An illustration of a forecast sample indicating the changes of daily exchange rate (Lai et al. 2018). The vertical dotted line divides the **input** (historical data) and **horizon** (future series to be predicted). (a) The original series and corresponding distributions of input series and horizon series. (b) The low frequency trend component obtained by DWT and the corresponding distributions. (c) The residual component and the corresponding distributions. Note that the residual component is derived by subtracting the reconstructed low-frequency trend (b) from the original input series (a). (d) The first-order trend difference and the corresponding distributions.

ing. WDAN starts by applying discrete wavelet transforms to decompose each input sequence into trend and residual components. We then extract normalization statistics separately for each component to reflect their unique statistical properties. For the trend component, which usually shows stronger non-stationarity, we apply first-order differencing to obtain more stable features for statistics prediction. A lightweight prediction module estimates future normalization statistics, allowing us to adaptively de-normalize the model outputs.

Our main contributions are summarized as follows: (**i**) We introduce a novel perspective on non-stationarity by identifying and separately modeling the distinct statistical behaviors of different time series components through wavelet decomposition. (**ii**) We propose WDAN, a model-agnostic normalization framework that disentangles non-stationary components and applies specialized treatments to each, including first-order differencing for trend modeling. (**iii**) Extensive experiments on benchmark datasets demonstrate that WDAN consistently improves forecasting accuracy across multiple backbone models, with particularly significant gains in highly non-stationary settings.

Related Work

Time Series Forecasting

Time series forecasting is a critical task in fields like finance, energy, and transportation. Classical statistical models such as ARIMA (Box et al. 2015) use linear autoregressive structures to capture temporal dependencies but are limited by strong assumptions about ideal data properties.

Deep learning models have achieved significant progress by leveraging non-linear modeling capabilities. Architectures including Recurrent Neural Networks (RNNs) (Hochreiter and Schmidhuber 1997), Temporal Convolutional Networks (TCNs) (Bai, Kolter, and Koltun 2018), and Transformers (Vaswani et al. 2017; Zhou et al. 2021) have been widely adopted for capturing temporal patterns. Some works further improve performance by incorporating time series decomposition (Wu et al. 2021) or time-frequency representations (Zhou et al. 2022). While many recent models focus on cross-variable dependencies (Zhang and Yan 2023; Liu et al. 2024; Bai et al. 2020), our work takes a channel-independent approach, which is shown to be more robust to distribution shifts and non-stationary conditions (Wen et al. 2023; Han, Ye, and Zhan 2024).

Non-stationary Time Series Forecasting

Non-stationarity is a key challenge in time series forecasting, as distributional shifts over time can lead to poor generalization and unstable predictions. Pre-processing the input series with stationarization methods is a straightforward approach to tackle non-stationarity. Traditional statistical models such as ARIMA (Box et al. 2015) apply differencing to remove trends and achieve stationarity. In deep learning, normalization is the most popular technique to deal with non-stationarity. RevIN (Kim et al. 2021) proposes a symmetric normalization scheme that normalizes input series and then denormalizes model outputs using instance normalization. DAIN (Passalis et al. 2019) employs a nonlinear transformation on normalization statistics to adaptively stationarize inputs. However, the above methods overlook the distribution discrepancy between the input series and the horizon series. Dish-TS (Fan et al. 2023) proposes to address this issue by learning a mapping from the input distribution to the output distribution characterized by means and variances. SAN (Liu et al. 2023b) notices the distribution shifts across compact time slices and proposes to learn the distribution mapping on time slice level. While these methods improve robustness to distribution shifts, they typically apply the normalization scheme to the entire sequence, overlooking the fact that real-world time series often contain multiple components with different behaviors.

Other approaches model non-stationarity from a dynamical systems perspective. For example, Koopman-based methods (Liu et al. 2023a; Wang et al. 2023) treat non-stationary time series as a non-linear dynamical system and transform the system into a measurement function space, which can be described by a linear Koopman operator. While theoretically sound, these approaches often require specific domain knowledge to define appropriate functions or operators.

Our approach differs from existing methods by explicitly recognizing that different components of time series show different types and degrees of non-stationarity. By using wavelet decomposition to disentangle these components and applying tailored normalization to each, our approach addresses a fundamental limitation in current normalization techniques. Moreover, our framework incorporates differencing methods for handling strongly non-stationary trend components, while maintaining the model-agnostic advantages of normalization-based approaches. While concurrent works have proposed sophisticated architectures using pattern-specific experts (Sun et al. 2024) or adapting Large Language Models (Sun et al. 2025) to handle distribution shifts, WDAN distinctively focuses on a lightweight, plug-and-play normalization module applicable to various existing backbones.

Methodology

We propose the **W**avelet-based **D**isentangled **A**daptive **N**ormalization (WDAN) framework for non-stationary multivariate time series forecasting. Given the input series $\boldsymbol{X} \in \mathbb{R}^{N \times T} = (\boldsymbol{x}^1, ..., \boldsymbol{x}^N)$ with N variables and T input time steps, we aim to predict the future series $\boldsymbol{Y} \in \mathbb{R}^{N \times H}(\boldsymbol{y}^1, ..., \boldsymbol{y}^N)$ with prediction horizon length H. The core idea lies in disentangling the non-stationary components via wavelet transform and adaptively predict future non-stationary normalization statistics from the decomposed series. In this section, we outline the overall framework and detail our approach to addressing the non-stationarity problem in time series forecasting.

Overall Framework

As a model-agnostic framework, WDAN adopts a popular normalization framework, which first stationarizes the input sequences by normalization and later restores the non-stationarity information by denormalizing the output sequences. Unlike existing normalization methods, WDAN does not directly compute normalization statistics on the original input sequences. Instead, as shown in Fig. 2, WDAN begins by disentangling the input sequences into low-frequency and high-frequency components via discrete wavelet transform (DWT), which captures persistent trends and transient fluctuations, respectively. These components then undergo instance-specific normalization, where low-frequency signals guide the estimation of evolving mean trends, and high-frequency residuals inform time-varying variance calibration. The normalized sequence is input to a backbone forecasting model, i.e., RNNs or Transformers, to obtain the normalized predicted sequences. A statistics prediction module subsequently predicts future distribution statistics for denormalization by synthesizing multi-scale dynamics, including the differential trends. To ensure stable model training, the framework adopts a three-stage training strategy.

Wavelet-based Disentangled Normalization

In contrast to Fourier transforms, which presume global stationarity at the cost of local temporal information, or moving averages that introduce inevitable phase shifts, wavelet decomposition offers simultaneous time-frequency localization. This property is indispensable for detecting the transient statistical shifts inherent in non-stationary time series.

Formally, we utilize a K-level DWT to decompose the input series $\boldsymbol{x}^i$ into low-frequency approximation coefficients $\boldsymbol{c}_l^i$ and high-frequency detail coefficients $\{\boldsymbol{c}_{h_k}^i\}_{k=1}^K$. These coefficients are then mapped back to the time domain as interpretable components via the inverse DWT, formulated as:

$$\boldsymbol{c}_{l_k}^i, \boldsymbol{c}_{h_k}^i = \text{DWT}_{\phi_{l,h}}(\boldsymbol{c}_{l_{k-1}}^i), \forall k = 1, ..., K \quad (1)$$

$$\boldsymbol{c}_{l_0}^i = \boldsymbol{x}^i, \quad \boldsymbol{c}_l^i = \boldsymbol{c}_{l_K}^i, \quad (2)$$

$$\boldsymbol{x}_l^i = \text{IDWT}(\boldsymbol{c}_l^i, \boldsymbol{0}), \quad \boldsymbol{x}_h^i = \sum_{k=1}^{K} \text{IDWT}(\boldsymbol{0}, \boldsymbol{c}_{h_k}^i), \quad (3)$$

where $\phi_{l,h}$ is a pair of wavelet bases. $\boldsymbol{x}_l^i$ and $\boldsymbol{x}_h^i$ represent the low-frequency trend component and the high-frequency residual component of $\boldsymbol{x}^i$, respectively. Unlike Fourier-based methods that impose fixed frequency assumptions, wavelet decomposition adaptively isolates non-stationary patterns while preserving localized temporal features. $\boldsymbol{x}_l^i$ captures the continuously changing trends, which are the main source of non-stationarity. $\boldsymbol{x}_h^i$ aggregates transient fluctuations across different frequency bands, which reflects the inherent patterns of the input series.

Normalization parameters are dynamically derived from these components. Since the low-frequency signal provides the time-varying means due to non-stationarity, we define the mean series for input sequence normalization as $\boldsymbol{\mu}_x^i = (\mu_x^i[t - T + 1], \ldots, \mu_x^i[t]) \in \mathbb{R}^T$, where $\mu_x^i[t] = x_l^i[t]$ and $x_l^i[t]$ denotes the value of $\boldsymbol{x}_l^i$ at time step t. High-frequency residuals encompass both the localized features of the inherent patterns of the input series and the non-stationary variations caused by abrupt changes, noise, etc. Therefore, we derive the point-level standard deviation parameters from $\boldsymbol{x}_h^i$ using a sliding window along the temporal dimension. Replication padding is added to align the length of sliding statistics to the original series. This process is formulated as follows:

$$\mu_h^i[t] = \frac{1}{2w + 1} \sum_{j=-w}^{w} x_h^i[t + j], \quad (4)$$

$$\left(\sigma_x^i[t]\right)^2 = \frac{1}{2w + 1} \sum_{j=-w}^{w} \left(x_h^i[t + j] - \mu_h^i[t]\right)^2, \quad (5)$$

where $2w + 1$ is the size of the sliding window. Thus, the point-level instance normalization process is as follows:

$$\bar{x}^i[t] = \frac{1}{\sigma_x^i[t] + \epsilon}(x^i[t] - \mu_x^i[t]). \quad (6)$$

Here, $\bar{\boldsymbol{x}}^i$ is the stationary series and ϵ is a small positive number added for numerical stability. Through series decomposition, we provide a more detailed explanation of the sources of non-stationarity and then extract the normalization parameters from different components. The normalized series $\bar{\boldsymbol{X}} = (\bar{\boldsymbol{x}}^1, ..., \bar{\boldsymbol{x}}^N)$ is input to the downstream backbone forecasting model g_θ.

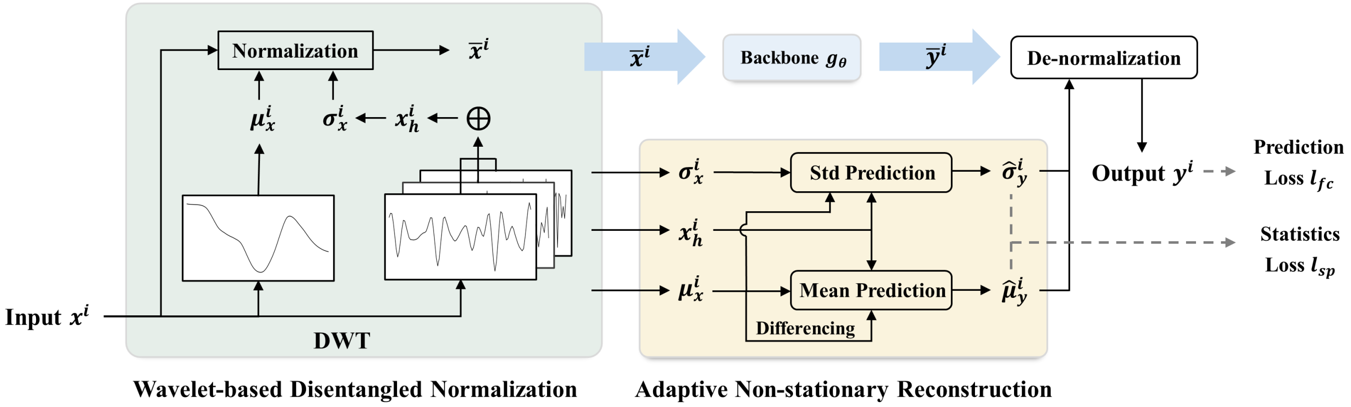


Figure 2: An illustration of the proposed WDAN framework

Adaptive Non-stationarity Reconstruction

To compensate for evolving distribution shifts, we predict future distribution parameters by fusing multi-scale dynamics. Then we denormalize the output of backbone forecasting models with the predicted statistics.

Statistics Prediction with Differencing Following previous distribution prediction works, we use a multi-layer perceptron network to predict future distributions efficiently. Instead of learning the statistics directly, we adopt a residual learning technique to predict the difference between future statistics and the overall mean of the current statistics series. We define

$$\bar{\mu}^i = \frac{1}{T}\sum_{t=1}^{T} \mu_x^i[t], \quad \bar{\sigma}^i = \frac{1}{T}\sum_{t=1}^{T} \sigma_x^i[t]. \tag{7}$$

Subsequently, the deviation of the statistics series from the overall mean is fed into the prediction module. Considering the potential interaction between high- and low-frequency signals, we take both high and low frequency signals as input. Furthermore, due to the non-stationarity of the mean sequence, we additionally perform first-order differencing on it and input the differenced series into the mean prediction module. The overall procedure is formulated as follows:

$$\tilde{\mu}_x^i[t] = \mu_x^i[t] - \bar{\mu}_x^i[t], \tag{8}$$

$$\tilde{\sigma}_x^i[t] = \sigma_x^i[t] - \bar{\sigma}_x^i[t], \tag{9}$$

$$\Delta\tilde{\mu}_x^i[t] = \tilde{\mu}_x^i[t] - \tilde{\mu}_x^i[t-1], \tag{10}$$

$$\hat{\boldsymbol{\mu}}_y^i = \mathrm{MLP}_\mu\left(\mathrm{MLP}_1(\tilde{\boldsymbol{\mu}}_x^i) \parallel \mathrm{MLP}_2(\Delta\tilde{\boldsymbol{\mu}}_x^i) \parallel \mathrm{MLP}_3(\boldsymbol{x}_h^i)\right) + \bar{\mu}^i, \tag{11}$$

$$\hat{\boldsymbol{\sigma}}_y^i = \mathrm{MLP}_\sigma\left(\mathrm{MLP}_4(\tilde{\boldsymbol{\sigma}}_x^i) \parallel \mathrm{MLP}_1(\tilde{\boldsymbol{\mu}}_x^i) \parallel \mathrm{MLP}_3(\boldsymbol{x}_h^i)\right) + \bar{\sigma}^i, \tag{12}$$

where $\parallel$ denotes concatenation. The statistics prediction module makes predictions at the point level, meaning that the parameter series for output sequence de-normalization $\hat{\boldsymbol{\mu}}_y^i \in \mathbb{R}^H$ and $\hat{\sigma}_y^i \in \mathbb{R}^H$. $\mathrm{MLP}_i : \mathbb{R}^T \rightarrow \mathbb{R}^D$ for $i = 1, 2, 3, 4$ initially maps the input sequence along the temporal dimension into a high-dimensional latent vector, where D is the hidden dimension. Then MLP_μ and MLP_σ project the concatenated feature vectors into predictions of the parameter series for de-normalization.

De-normalization After the aforementioned statistics predictions, WDAN denormalizes the output sequence of the backbone forecasting model to restore the non-stationary factors. Specifically, WDAN performs instance-specific denormalization as follows:

$$\bar{\boldsymbol{Y}} = g_\theta(\bar{\boldsymbol{X}}), \quad \bar{\boldsymbol{Y}} = (\bar{\boldsymbol{y}}^1, ..., \bar{\boldsymbol{y}}^N), \tag{13}$$

$$\hat{\boldsymbol{y}}^i = \bar{\boldsymbol{y}}^i \odot (\hat{\boldsymbol{\sigma}}_y^i + \epsilon) + \hat{\boldsymbol{\mu}}_y^i, \tag{14}$$

where $\odot$ is the elementwise product. Finally, we obtain the final prediction $\hat{\boldsymbol{Y}} = (\hat{\boldsymbol{y}}^1, ..., \hat{\boldsymbol{y}}^N)$ of the whole framework.

Three-stage Training Strategy

The training process constitutes a bi-level optimization problem where the results of statistical prediction greatly impact the prediction performance of the overall framework. We adopt a three-stage training strategy to decouple statistics prediction from temporal pattern learning of the backbone model. In the first training stage, the statistics prediction network undergoes dedicated pretraining using ground truth statistics series, i.e.,

$$\theta_d = \arg\min_{\theta_d} \sum_i l_{sp}\left((\hat{\boldsymbol{\mu}}_y^i, \hat{\boldsymbol{\sigma}}_y^i), (\boldsymbol{\mu}_y^i, \boldsymbol{\sigma}_y^i), \theta_d\right), \tag{15}$$

where θ_d represents the parameters of the statistics prediction module. l_{sp} is a loss function between predicted statistics and ground truth, e.g., mean absolute error (MAE) or mean squared error (MSE). The second training stage freezes the pretrained statistics prediction module and focuses on optimizing the backbone forecasting model g_θ, i.e.,

$$\theta_g = \arg\min_{\theta_g} \sum_i l_{fc}\left(\hat{\boldsymbol{y}}^i, \boldsymbol{y}^i, \theta_g\right), \tag{16}$$

where θ_g represents the parameters of g_θ. l_{fc} is a loss function to evaluate the overall forecasting results. In the third training stage, considering the mutual influence between the backbone model and the statistics prediction module, we perform joint fine-tuning with a relatively smaller learning rate, i.e.,

$$\{\theta_d, \theta_g\} = \arg\min_{\theta_g} \sum_i l_{fc}\left(\hat{\boldsymbol{y}}^i, \boldsymbol{y}^i, \{\theta_d, \theta_g\}\right). \tag{17}$$

Experiments

In this section, we evaluate the effectiveness of the proposed WDAN framework on multiple benchmark datasets for multivariate time series forecasting.

Experimental Setup

Datasets To demonstrate the effectiveness of WDAN in handling non-stationarity across various data scenarios, we conduct experiments on four widely-used benchmark datasets: (1) **Exchange** (Lai et al. 2018) dataset, which collects the daily exchange rate of 8 countries from 1990 to 2016. (2) **ETT** (Zhou et al. 2021) dataset, which records oil temperature and load features of electricity transformers from July 2016 to July 2018. It consists of four sub-datasets, where ETTh is sampled hourly and ETTm is sampled every 15 minutes. (3) **Weather**[1] dataset, which consists of 21 weather indicators including air temperature and humidity collected every 10 minutes in 2021. (4) **Electricity**[2], which contains the electricity consumption data of 321 clients from July 2016 to July 2019. Detailed information about these datasets is listed in Table 1, where we also report the results of the Augmented Dickey-Fuller (ADF) test that assesses time series stationarity. A smaller absolute value of the ADF statistic indicates stronger non-stationarity.

Following previous works, we split each dataset into training, validation, and test sets in chronological order. The split ratio is 6:2:2 for ETT datasets and 7:1:2 for other datasets. Additionally, we perform z-score normalization on the datasets based on the training data statistics as preprocessing, which allows measurements of different variables on the same scale. Note that z-score normalization uses global statistics for normalization, meaning it cannot handle non-stationarity since the statistics used in the normalization process are fixed across different samples.

Dataset	Vars	Sampling Frequency	Sequence Length	ADF Statistics
Exchange	8	1 day	7588	-1.90
ETTh1	7	1 hour	17420	-5.91
ETTh2	7	1 hour	17420	-4.13
ETTm1	7	15 mins	69680	-14.98
ETTm2	7	15 mins	69680	-5.66
Weather	21	10 mins	52696	-26.68
Electricity	321	1 hour	26304	-8.44

Table 1: Statistics of benchmark datasets

Baseline Methods WDAN is a model-agnostic approach that can be applied to any mainstream time series forecasting model. To demonstrate its versatility, we integrate WDAN with multiple representative models, including previously proposed FEDformer (Zhou et al. 2022), as well as recently introduced Crossformer (Zhang and Yan 2023), PatchTST (Nie et al. 2023), and iTransformer (Liu et al. 2024).

To highlight WDAN's effectiveness in handling non-stationarity, we also compare it with mainstream normalization frameworks, including: (1) SAN (Liu et al. 2023b): adopts segment-level instance normalization and predicts future statistics through segmented sequence prediction; (2) DDN (Dai et al. 2024): employs point-level instance normalization and normalizes the entire sequence in both time and frequency domains.

Implementation Details We use Mean Squared Error (MSE) as the loss function across all experiments. For evaluation metrics, we report both MSE and Mean Absolute Error (MAE) on the test set, where lower MSE/MAE indicates better predictive performance. Consistent with existing works, all models are trained and tested on four different prediction horizons $H \in \{96, 192, 336, 720\}$, with a fixed historical sequence length of $L = 720$. We report the average results of three runs for each model on each dataset to ensure stable performance. More implementation details of our experiments can be found in the technical appendix.

Main Results

Table 2 presents the multivariate time series forecasting performance of WDAN applied to four mainstream time series forecasting models. The results clearly show that WDAN significantly improves the predictive performance of all backbone models in most cases, demonstrating the effectiveness and universality of the proposed method in handling non-stationary time series data. WDAN delivers consistent performance improvements across all backbone models at different prediction horizons.

Notably, the performance gains yielded by WDAN become increasingly pronounced as the prediction horizon extends from $H = 96$ to $H = 720$, indicating that non-stationarity handling methods have greater advantages for long-term forecasting. This can be attributed to the fact that distributional shifts in sequences typically occur gradually, making non-stationarity more significant for long-term predictions. For example, when $H = 720$, WDAN reduces iTransformer's MSE and MAE on the ETTh1 dataset by 19.6% and 13.8% respectively, while at $H = 96$, the reduction is only 5.6% and 5.9%.

Furthermore, since iTransformer and PatchTST already incorporate RevIN, a basic instance normalization method for handling non-stationarity in their original models, the performance improvement provided by WDAN on these two models is relatively smaller compared to its impact on Crossformer and FEDformer. Nevertheless, when we replace RevIN with WDAN in iTransformer and PatchTST, our method still achieves better performance, indicating that WDAN is superior to RevIN in handling non-stationarity. Additionally, the application of WDAN to Crossformer and FEDformer compensates for the limitations of these relatively earlier models in non-stationary modeling, enabling them to perform comparably to state-of-the-art models like iTransformer, further highlighting the importance of non-stationarity handling in sequence forecasting.

[1] https://www.bgc-jena.mpg.de/wetter/

[2] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

Methods		iTransformer		+WDAN		PatchTST		+WDAN		Crossformer		+WDAN		FEDformer		+WDAN	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	96	0.118	0.250	**0.083**	**0.203**	0.101	0.230	**0.084**	**0.204**	0.561	0.547	**0.084**	**0.205**	0.744	0.668	**0.087**	**0.205**
	192	0.246	0.370	**0.176**	**0.300**	0.200	0.326	**0.176**	**0.302**	0.798	0.683	**0.174**	**0.300**	0.928	0.760	**0.180**	**0.305**
	336	0.429	0.493	**0.361**	**0.436**	0.405	0.470	**0.347**	**0.429**	1.455	0.975	**0.350**	**0.431**	1.180	0.857	**0.386**	**0.456**
	720	1.101	0.796	**1.009**	**0.767**	1.463	0.904	**1.081**	**0.781**	2.050	1.163	**1.019**	**0.762**	1.901	1.068	**1.110**	**0.789**
ETTh1	96	0.390	0.422	**0.368**	**0.397**	0.399	0.419	**0.369**	**0.398**	0.391	0.431	**0.366**	**0.396**	0.486	0.498	**0.381**	**0.415**
	192	0.427	0.448	**0.406**	**0.421**	0.447	0.450	**0.407**	**0.422**	0.490	0.493	**0.401**	**0.415**	0.522	0.512	**0.417**	**0.438**
	336	0.459	0.471	**0.428**	**0.444**	0.472	0.474	**0.443**	**0.449**	0.939	0.758	**0.420**	**0.429**	0.545	0.534	**0.436**	**0.452**
	720	0.562	0.545	**0.452**	**0.470**	0.519	0.511	**0.515**	**0.504**	1.080	0.796	**0.446**	**0.468**	0.678	0.620	**0.473**	**0.484**
ETTh2	96	0.301	0.360	**0.269**	**0.334**	0.322	0.376	**0.271**	**0.335**	1.225	0.764	**0.269**	**0.336**	0.405	0.459	**0.277**	**0.342**
	192	0.379	0.408	**0.331**	**0.376**	0.412	0.432	**0.336**	**0.376**	1.253	0.810	**0.333**	**0.378**	0.431	0.480	**0.347**	**0.386**
	336	0.415	0.436	**0.358**	**0.403**	0.450	0.455	**0.357**	**0.401**	1.500	0.920	**0.359**	**0.405**	0.436	0.481	**0.391**	**0.423**
	720	0.435	0.463	**0.377**	**0.429**	0.497	0.486	**0.394**	**0.440**	3.825	1.606	**0.384**	**0.428**	0.502	0.514	**0.483**	**0.478**
ETTm1	96	0.315	0.368	**0.291**	**0.350**	0.296	0.354	**0.293**	**0.348**	0.348	0.389	**0.293**	**0.349**	0.435	0.457	**0.295**	**0.354**
	192	0.345	0.386	**0.330**	**0.375**	0.346	0.388	**0.335**	**0.376**	0.360	0.395	**0.332**	**0.373**	0.432	0.455	**0.330**	**0.378**
	336	0.379	0.407	**0.362**	**0.396**	0.400	0.421	**0.362**	**0.397**	0.586	0.573	**0.360**	**0.394**	0.454	0.470	**0.370**	**0.403**
	720	0.443	0.444	**0.412**	**0.422**	0.477	0.464	**0.413**	**0.411**	0.977	0.787	**0.413**	**0.411**	0.499	0.489	**0.420**	**0.425**
ETTm2	96	0.182	0.276	**0.162**	**0.253**	0.179	0.272	**0.172**	**0.257**	0.310	0.388	**0.162**	**0.254**	0.317	0.379	**0.167**	**0.258**
	192	0.244	0.317	**0.217**	**0.292**	0.248	0.317	**0.240**	**0.304**	0.387	0.438	**0.216**	**0.290**	0.338	0.390	**0.237**	**0.299**
	336	0.298	0.351	**0.273**	**0.331**	0.311	0.357	**0.282**	**0.334**	0.614	0.558	**0.278**	**0.336**	0.366	0.404	**0.282**	**0.338**
	720	0.376	0.401	**0.353**	**0.382**	**0.418**	**0.426**	0.442	0.432	1.279	0.813	**0.390**	**0.405**	0.421	0.446	**0.395**	**0.417**
Weather	96	0.177	0.229	**0.147**	**0.199**	0.153	0.208	**0.148**	**0.202**	0.173	0.244	**0.147**	**0.198**	0.317	0.373	**0.153**	**0.207**
	192	0.226	0.268	**0.196**	**0.248**	0.206	0.261	**0.194**	**0.248**	0.218	0.288	**0.195**	**0.248**	0.347	0.397	**0.193**	**0.248**
	336	0.285	0.309	**0.242**	**0.282**	0.243	**0.286**	**0.242**	0.290	0.266	0.328	**0.244**	**0.287**	0.354	0.392	**0.247**	**0.293**
	720	0.359	0.361	**0.312**	**0.337**	0.315	0.336	**0.306**	**0.332**	0.334	0.380	**0.310**	**0.336**	0.400	0.421	**0.313**	**0.341**
Electricity	96	0.135	0.232	**0.128**	**0.225**	0.137	0.241	**0.128**	**0.224**	0.137	0.238	**0.128**	**0.223**	0.227	0.339	**0.142**	**0.245**
	192	0.157	0.253	**0.146**	**0.242**	0.153	0.256	**0.145**	**0.240**	0.157	0.258	**0.148**	**0.244**	0.231	0.343	**0.159**	**0.264**
	336	0.171	0.269	**0.156**	**0.256**	0.169	0.272	**0.160**	**0.258**	0.193	0.294	**0.164**	**0.262**	0.259	0.366	**0.174**	**0.279**
	720	0.196	0.290	**0.182**	**0.281**	0.205	0.302	**0.194**	**0.291**	0.281	0.358	**0.196**	**0.294**	0.297	0.393	**0.203**	**0.309**

Table 2: Multivariate time series forecasting performance of WDAN across different backbone models. The best results for each backbone model are highlighted in bold.

Comparison with Normalization Methods

To further validate WDAN's advantages in handling non-stationary time series data, Table 3 shows a comparison of WDAN with mainstream normalization methods across different backbone models for multivariate time series forecasting. Table 3 provides an overview of the average predictive performance across different prediction horizons, while more detailed results by prediction lengths are presented in the technical appendix.

From an overall performance perspective, WDAN achieves the best results on most datasets and model combinations. Regarding the datasets, WDAN demonstrates the most significant performance improvement on the Exchange and ETTh2 dataset, which exhibits the most pronounced non-stationarity (determined by the ADF test), achieving the best results in almost all prediction scenarios across all backbone models. On datasets with relatively weaker non-stationarity, such as ETTm1 and Weather, WDAN still shows stronger comprehensive predictive performance and competitive results in specific prediction scenarios compared to mainstream normalization methods. This not only demonstrates the versatility of non-stationarity handling methods across different data scenarios but also validates WDAN's advantages in handling non-stationarity.

Overall, the comparative experiments on normalization methods prove that WDAN's strategy of disentangling non-stationary components through wavelet decomposition and capturing dynamic changes through differencing enables more accurate handling of complex non-stationary patterns in time series. Compared to existing methods, WDAN significantly enhances the predictive capability for non-stationary time series data while maintaining the flexibility of the normalization framework.

Ablation Study

To further validate the effectiveness of WDAN, we conduct an ablation study focusing on its two key components: (1) the wavelet-based decomposition and (2) the use of first-order differencing for modeling trend dynamics. We compare our full WDAN framework with two variants:

Methods		Exchange	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Electricity
iTransformer	+WDAN	**0.407**	**0.413**	**0.334**	**0.349**	0.251	**0.224**	0.153
	+SAN	0.416	0.433	0.358	0.350	0.268	0.226	0.155
	+DDN	0.429	0.457	0.347	0.360	**0.251**	0.231	**0.153**
	IMP(%)	2.16	4.63	3.73	0.35	-0.12	0.95	-0.02
PatchTST	+WDAN	**0.422**	**0.434**	**0.340**	**0.351**	**0.284**	**0.222**	**0.157**
	+SAN	0.475	0.488	0.413	0.351	0.312	0.225	0.164
	+DDN	0.713	0.449	0.375	0.360	0.339	0.241	0.158
	IMP(%)	11.20	3.48	9.50	0.13	9.03	1.03	0.94
Crossformer	+WDAN	**0.407**	**0.408**	**0.336**	**0.349**	**0.261**	**0.224**	**0.159**
	+SAN	0.417	0.450	0.370	0.373	0.308	0.225	0.262
	+DDN	0.431	0.434	0.350	0.358	0.270	0.242	0.165
	IMP(%)	2.37	5.91	4.00	2.48	3.34	0.56	3.69
FEDformer	+WDAN	**0.441**	**0.427**	**0.374**	**0.354**	**0.270**	**0.227**	**0.170**
	+SAN	0.449	0.493	0.388	0.360	0.277	0.232	0.197
	+DDN	0.499	0.472	0.433	0.374	0.271	0.243	0.170
	IMP(%)	1.92	9.64	3.52	1.89	0.23	2.09	0.38

Table 3: Comparison of multivariate time series forecasting performance between WDAN and mainstream normalization methods across different backbone models. We report the average values of the MSE metric across different prediction horizons. The best results for each backbone model are highlighted in bold. IMP indicates the percentage improvement in MSE of WDAN.

Methods	iTransformer						Crossformer					
	WDAN		MovingAvg		No Differencing		WDAN		MovingAvg		No Differencing	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	**0.407**	**0.427**	0.408	0.427	0.438	0.441	**0.407**	**0.424**	0.418	0.432	0.437	0.441
ETTh2	**0.334**	**0.386**	0.360	0.405	0.338	0.389	**0.336**	**0.387**	0.371	0.410	0.342	0.390
Weather	**0.224**	**0.267**	0.231	0.272	0.227	0.270	**0.224**	**0.267**	0.226	0.269	0.228	0.270

Table 4: Results of ablation study for WDAN. We report the average values of the metrics across different prediction horizons.

(1) **MovingAvg**, which replaces DWT with a simple moving average method (Zeng et al. 2023) while retaining the differenced trend features; (2) **NoDiff**, which retains the DWT-based decomposition but excludes the first-order differencing feature from the trend statistics prediction module. We perform experiments using two representative backbone models, iTransformer and Crossformer, across three datasets with varying levels of non-stationarity: Exchange (high), ETTh2 (moderate), and Weather (low).

The results in Table 4 demonstrate that WDAN generally outperforms its variants across different settings. On the highly non-stationary Exchange dataset, WDAN consistently outperforms the NoDiff variant across all horizons for both backbones, highlighting the importance of differencing in modeling non-stationary trend dynamics. On the relatively stationary Weather dataset, all three variants perform similarly, suggesting that WDAN remains effective without introducing unnecessary complexity in more stationary settings.

Conclusion

In this paper, we present WDAN, a novel normalization framework designed to tackle non-stationarity in time series forecasting. By decomposing input sequences into low-frequency and high-frequency components using wavelet transforms, WDAN explicitly disentangles the sources of non-stationarity and applies component-wise normalization strategies. The proposed framework further enhances adaptability by leveraging differencing techniques and multi-scale residual dynamics to predict future normalization statistics. WDAN is model-agnostic and can be seamlessly integrated with a variety of backbone forecasting models. Extensive experiments across multiple datasets and forecasting models demonstrate the effectiveness and generalizability of WDAN. The success of WDAN underscores the importance of recognizing and separately modeling the diverse non-stationary behaviors exhibited by different time series components. This work opens promising directions for future research, including exploring adaptive wavelet basis selection, extending the approach to capture cross-variable dependencies in multivariate settings, and investigating theoretical connections between wavelet-based decomposition and formal stationarity properties.

Acknowledgements

This work was supported in part by NSFC Grant 72271138, Tsinghua-NUS Joint Funding 20243080039, and Huawei–Tsinghua Network Reliability Operations and Maintenance Innovation Research Long-term Cooperation Agreement (TC20210811013).

References

Ahmadi, M.; Jafarzadeh-Ghoushchi, S.; Taghizadeh, R.; and Sharifi, A. 2019. Presentation of a new hybrid approach for forecasting economic growth using artificial intelligence approaches. *Neural Computing and Applications*, 31: 8661–8680.

Bai, L.; Yao, L.; Li, C.; Wang, X.; and Wang, C. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems*, volume 33, 17804–17815. Red Hook, NY, USA: Curran Associates Inc.

Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv:1803.01271.

Box, G. E.; Jenkins, G. M.; Reinsel, G. C.; and Ljung, G. M. 2015. *Time series analysis: forecasting and control*. Hoboken, NJ: John Wiley & Sons.

Chaovalit, P.; Gangopadhyay, A.; Karabatis, G.; and Chen, Z. 2011. Discrete wavelet transform-based time series analysis and mining. *ACM Computing Surveys (CSUR)*, 43(2): 1–37.

Dai, T.; Wu, B.; Liu, P.; Li, N.; Yuerong, X.; Xia, S.-T.; and Zhu, Z. 2024. DDN: Dual-domain dynamic normalization for non-stationary time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, 108490–108517.

Fan, W.; Wang, P.; Wang, D.; Wang, D.; Zhou, Y.; and Fu, Y. 2023. Dish-TS: a general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 7522–7529. AAAI Press.

Han, L.; Ye, H.-J.; and Zhan, D.-C. 2024. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*, 9(8): 1735–1780.

Huang, S.-J.; and Hsieh, C.-T. 2002. Coiflet wavelet transform applied to inspect power system disturbance-generated signals. *IEEE Transactions on Aerospace and Electronic Systems*, 38(1): 204–210.

Kaushik, S.; Choudhury, A.; Sheron, P. K.; Dasgupta, N.; Natarajan, S.; Pickett, L. A.; and Dutt, V. 2020. AI in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, 3: 4.

Kim, T.; Kim, J.; Tae, Y.; Park, C.; Choi, J.-H.; and Choo, J. 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kitaev, N.; Kaiser, L.; and Levskaya, A. 2020. Reformer: The Efficient Transformer. In *International Conference on Learning Representations*.

Lai, G.; Chang, W.-C.; Yang, Y.; and Liu, H. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 95–104.

Li, W.; Yang, X.; Liu, W.; Xia, Y.; and Bian, J. 2022. Ddg-da: Data distribution generation for predictable concept drift adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 4092–4100.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations*.

Liu, Y.; Li, C.; Wang, J.; and Long, M. 2023a. Koopa: Learning non-stationary time series dynamics with koopman predictors. In *Advances in Neural Information Processing Systems*, 12271–12290.

Liu, Z.; Cheng, M.; Li, Z.; Huang, Z.; Liu, Q.; Xie, Y.; and Chen, E. 2023b. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. In *Advances in Neural Information Processing Systems*, volume 36, 14273–14292.

Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; and Zhang, G. 2018. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12): 2346–2363.

Nie, Y.; H. Nguyen, N.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations*.

Passalis, N.; Tefas, A.; Kanniainen, J.; Gabbouj, M.; and Iosifidis, A. 2019. Deep adaptive input normalization for time series forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9): 3760–3765.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

RB, C. 1990. STL: A seasonal-trend decomposition procedure based on loess. *J Off Stat*, 6: 3–73.

Singh, A. K.; Ibraheem, S. K.; Muazzam, M.; and Chaturvedi, D. 2013. An overview of electricity demand forecasting techniques. *Network and Complex Systems*, 3(3): 38–48.

Sun, Y.; Eldele, E.; Xie, Z.; Wang, Y.; Niu, W.; Hu, Q.; Kwoh, C. K.; and Wu, M. 2025. Adapting llms to time series forecasting via temporal heterogeneity modeling and semantic alignment. *arXiv preprint arXiv:2508.07195*.

Sun, Y.; Xie, Z.; Eldele, E.; Chen, D.; Hu, Q.; and Wu, M. 2024. Learning pattern-specific experts for time series forecasting under patch-level distribution shift. *arXiv preprint arXiv:2410.09836*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 6000–6010.

Wang, R.; Dong, Y.; Arik, S. O.; and Yu, R. 2023. Koopman Neural Operator Forecaster for Time-series with Temporal Distributional Shifts. In *International Conference on Learning Representations*.

Wen, Q.; Chen, W.; Sun, L.; Zhang, Z.; Wang, L.; Jin, R.; Tan, T.; et al. 2023. Onenet: Enhancing time series forecasting models under concept drift by online ensembling. In *Advances in Neural Information Processing Systems*, volume 36, 69949–69980.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, 22419–22430.

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 3634–3640. AAAI Press.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 11121–11128. AAAI Press.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11106–11115. AAAI Press.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 27268–27286. PMLR.

A Experiment Setting

A.1 Setting Details

All experiments are conducted on an NVIDIA A100 GPU using PyTorch (Paszke et al. 2019). We utilize ADAM (Kingma and Ba 2014) as the optimizer and employ Mean Squared Error (MSE) as the loss function. We choose Coiflet 3 as the default wavelet basis function. In the statistics prediction module, MLP$_i, i = 1, 2, 3, 4$ use single-layer perceptrons for feature mapping by default, while MLP$_\mu$ and MLP$_\sigma$ can use multi-layer perceptrons with the number of layers treated as a hyperparameter. We set the initial learning rate within $\{0.001, 0.0001\}$ and the fixed number of epochs for the second-stage training within $\{0, 1, 2\}$. The number of layers for the multi-layer perceptrons MLP$_\sigma$ and MLP$_\mu$ is configured within $\{0, 1, 2\}$. The sliding window size in Equations (4) and (5) is set within $\{5, 12, 24\}$. The decomposition level of DWT is selected from $\{2, 3\}$. We determine the final values of the aforementioned hyperparameters through grid search by minimizing the MSE on the validation set. For backbone networks, we use the default hyperparameter settings and perform grid search to optimize the additional hyperparameters introduced by WDAN.

A.2 Backbones

Our backbone models are described as follows:

- iTransformer (Liu et al. 2024) inverts the traditional Transformer architecture by embedding the whole time series of each variate independently into a token and treating each variate as a token for attention mechanism. The source code is available at https://github.com/thuml/iTransformer.
- PatchTST (Nie et al. 2023) is a Transformer-based model that segments time series into subseries-level patches as input tokens rather than using point-wise inputs. It employs channel-independence where each univariate series is processed separately while sharing the same parameters across all series. The source code is available at https://github.com/yuqinie98/PatchTST.
- Crossformer (Zhang and Yan 2023) introduces a two-stage attention mechanism that separately handles temporal and variate dimensions. It employs a unique cross-dimension attention design with a two-stage embedding architecture that decomposes the complex spatiotemporal modeling task, allowing it to efficiently capture both temporal patterns and inter-variable relationships in multivariate time series. The source code is available at https://github.com/Thinklab-SJTU/Crossformer.
- FEDformer (Zhou et al. 2022) combines seasonal-trend decomposition with frequency domain analysis for improved time series forecasting. It uses a mixture of experts to capture global properties while its frequency enhanced blocks extract patterns through Fourier or Wavelet transforms. The source code is available at https://github.com/MAZiqing/FEDformer.

B Supplement Experiments Results

B.1 Full Multivariate Forecasting Results

Due to space limitations in the main text, we provide complete multivariate time series prediction results in this subsection. Table 5 shows the prediction results of different backbone models under various normalization method frameworks. These results demonstrate that our proposed WDAN achieves leading performance across different datasets and backbone models.

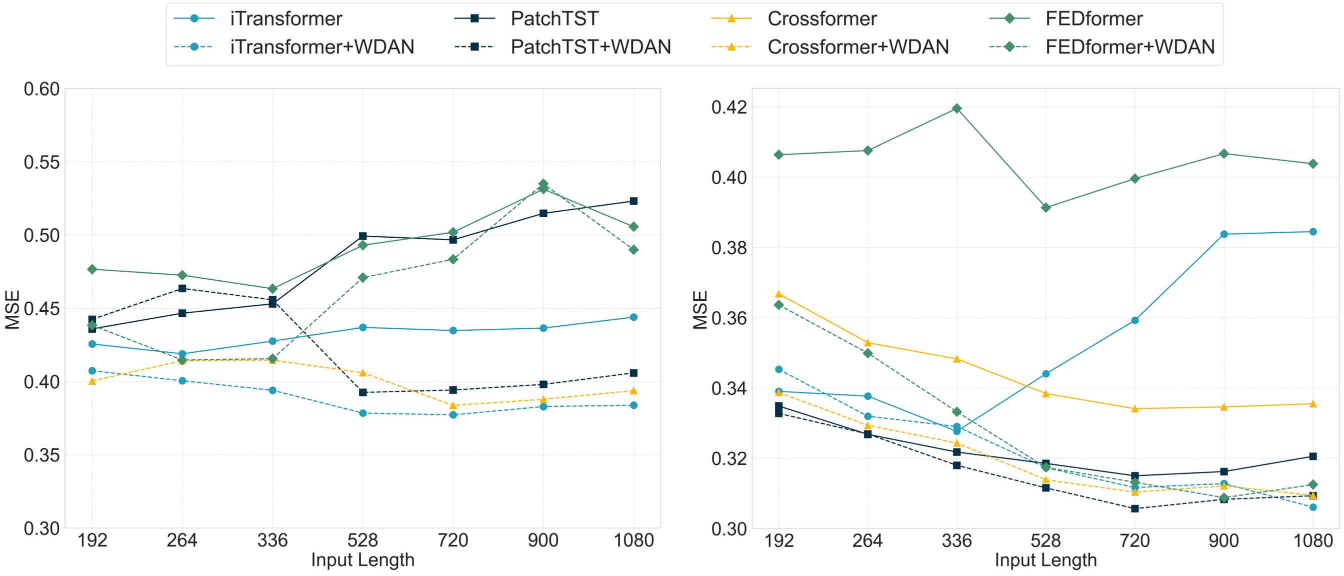
B.2 Analysis on Various Input Lengths

The length of input sequences is a crucial hyperparameter in time series forecasting tasks as it determines the

Methods		iTransformer		+SAN		+DDN		**+WDAN**		PatchTST		+SAN		+DDN		**+WDAN**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	96	0.118	0.250	0.091	0.215	0.092	0.215	**0.083**	**0.203**	0.101	0.230	0.093	0.217	0.087	0.207	**0.084**	**0.204**
	192	0.246	0.370	0.189	0.311	0.190	0.311	**0.176**	**0.300**	0.200	0.326	0.192	0.314	0.177	**0.300**	**0.176**	0.302
	336	0.429	0.493	**0.353**	**0.429**	0.365	0.437	0.361	0.436	0.405	0.470	0.378	0.447	0.449	0.487	**0.347**	**0.429**
	720	1.101	0.796	1.032	**0.766**	1.068	0.777	**1.009**	0.767	1.463	0.904	1.238	0.826	2.139	1.114	**1.081**	**0.781**
ETTh1	96	0.390	0.422	0.388	0.418	0.379	0.407	**0.368**	**0.397**	0.399	0.419	0.399	0.416	0.399	0.417	**0.369**	**0.398**
	192	0.427	0.448	0.422	0.441	0.417	0.432	**0.406**	**0.421**	0.447	0.450	0.449	0.447	0.419	0.429	**0.407**	**0.422**
	336	0.459	0.471	0.447	0.459	0.458	0.459	**0.428**	**0.444**	0.472	0.474	0.495	0.481	0.459	0.455	**0.443**	**0.449**
	720	0.562	0.545	0.477	0.492	0.574	0.539	**0.452**	**0.470**	0.519	0.511	0.611	0.545	0.521	**0.503**	**0.515**	0.504
ETTh2	96	0.301	0.360	0.288	0.350	0.277	0.341	**0.269**	**0.334**	0.322	0.376	0.341	0.386	0.281	0.341	**0.271**	**0.335**
	192	0.379	0.408	0.352	0.392	0.338	0.381	**0.331**	**0.376**	0.412	0.432	0.437	0.447	0.395	0.407	**0.336**	**0.376**
	336	0.415	0.436	0.379	0.417	0.365	0.407	**0.358**	**0.403**	0.450	0.455	0.434	0.456	0.374	0.414	**0.357**	**0.401**
	720	0.435	0.463	0.412	0.450	0.409	0.448	**0.377**	**0.429**	0.497	0.486	0.440	0.461	0.451	0.479	**0.394**	**0.440**
ETTm1	96	0.315	0.368	0.296	**0.348**	0.308	0.352	**0.291**	0.350	0.296	0.354	0.297	**0.347**	0.299	0.351	**0.293**	0.348
	192	0.345	0.386	0.330	**0.369**	0.340	0.371	**0.330**	0.375	0.346	0.388	**0.331**	**0.368**	0.336	0.374	0.335	0.376
	336	0.379	0.407	**0.362**	**0.388**	0.370	0.389	0.362	0.396	0.400	0.421	0.363	**0.388**	0.366	0.394	**0.362**	0.397
	720	0.443	0.444	**0.412**	0.418	0.422	**0.417**	0.412	0.422	0.477	0.464	0.415	0.418	0.437	0.439	**0.413**	**0.411**
ETTm2	96	0.182	0.276	0.176	0.268	0.163	0.254	**0.162**	**0.253**	0.179	0.272	0.184	0.274	0.174	0.258	**0.172**	**0.257**
	192	0.244	0.317	0.234	0.307	**0.217**	**0.291**	0.217	0.292	0.248	0.317	0.263	0.324	0.262	0.317	**0.240**	**0.304**
	336	0.298	0.351	0.295	0.347	**0.269**	**0.327**	0.273	0.331	0.311	0.357	0.317	0.361	0.332	0.351	**0.282**	**0.334**
	720	0.376	0.401	0.368	0.396	0.355	0.385	**0.353**	**0.382**	**0.418**	**0.426**	0.485	0.453	0.587	0.482	0.442	0.432
Weather	96	0.177	0.229	0.151	0.207	0.153	0.214	**0.147**	**0.199**	0.153	0.208	0.148	0.204	0.158	0.211	**0.148**	**0.202**
	192	0.226	0.268	**0.195**	0.249	0.197	0.258	0.196	**0.248**	0.206	0.261	0.194	0.251	0.222	0.266	**0.194**	**0.248**
	336	0.285	0.309	0.246	0.292	0.250	0.303	**0.242**	**0.282**	0.243	0.286	0.245	**0.286**	0.253	0.303	**0.242**	0.290
	720	0.359	0.361	**0.311**	0.341	0.326	0.370	0.312	**0.337**	0.315	0.336	0.312	0.335	0.330	0.354	**0.306**	**0.332**
Electricity	96	0.135	0.232	0.131	0.230	0.129	0.226	**0.128**	**0.225**	0.137	0.241	0.137	0.242	0.129	0.226	**0.128**	**0.224**
	192	0.157	0.253	0.147	0.246	0.146	0.244	**0.146**	**0.242**	0.153	0.256	0.152	0.256	0.146	0.242	**0.145**	**0.240**
	336	0.171	0.269	0.157	0.258	**0.155**	**0.255**	0.156	0.256	0.169	0.272	0.167	0.271	0.161	0.260	**0.160**	**0.258**
	720	0.196	0.290	0.185	0.285	**0.182**	0.281	0.182	**0.281**	0.205	0.302	0.200	0.300	0.196	0.293	**0.194**	**0.291**

Methods		Crossformer		+SAN		+DDN		**+WDAN**		FEDformer		+SAN		+DDN		**+WDAN**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	96	0.561	0.547	0.092	0.216	0.098	0.223	**0.084**	**0.205**	0.744	0.668	0.102	0.230	0.107	0.235	**0.087**	**0.205**
	192	0.798	0.683	0.188	0.310	0.197	0.317	**0.174**	**0.300**	0.928	0.760	0.226	0.342	0.280	0.377	**0.180**	**0.305**
	336	1.455	0.975	**0.342**	**0.422**	0.389	0.453	0.350	0.431	1.180	0.857	**0.370**	**0.447**	0.413	0.468	0.386	0.456
	720	2.050	1.163	1.045	0.773	1.038	0.773	**1.019**	**0.762**	1.901	1.068	**1.099**	0.795	1.197	0.819	1.110	**0.789**
ETTh1	96	0.391	0.431	0.386	0.407	0.377	0.403	**0.366**	**0.396**	0.486	0.498	0.448	0.458	0.392	0.418	**0.381**	**0.415**
	192	0.490	0.493	0.431	0.434	0.413	0.423	**0.401**	**0.415**	0.522	0.512	0.484	0.477	0.440	0.447	**0.417**	**0.438**
	336	0.939	0.758	0.483	0.470	0.461	0.456	**0.420**	**0.429**	0.545	0.534	0.502	0.490	0.479	0.469	**0.436**	**0.452**
	720	1.080	0.796	0.501	0.494	0.484	0.483	**0.446**	**0.468**	0.678	0.620	0.539	0.517	0.578	0.519	**0.473**	**0.484**
ETTh2	96	1.225	0.764	0.284	0.348	0.274	0.338	**0.269**	**0.336**	0.405	0.459	0.301	0.361	0.306	0.369	**0.277**	**0.342**
	192	1.253	0.810	0.343	0.384	0.339	0.380	**0.333**	**0.378**	0.431	0.480	0.370	0.407	0.385	0.419	**0.347**	**0.386**
	336	1.500	0.920	0.378	0.413	0.365	0.407	**0.359**	**0.405**	0.436	0.481	0.405	0.435	0.443	0.456	**0.391**	**0.423**
	720	3.825	1.606	0.476	0.481	0.424	0.455	**0.384**	**0.428**	0.502	0.514	**0.476**	**0.477**	0.599	0.527	0.483	0.478
ETTm1	96	0.348	0.389	0.295	**0.346**	**0.292**	0.349	0.293	0.349	0.435	0.457	0.306	0.358	0.314	0.370	**0.295**	**0.354**
	192	0.360	0.395	**0.332**	**0.368**	0.336	0.372	0.332	0.373	0.432	0.455	0.338	**0.378**	0.342	0.386	**0.330**	0.378
	336	0.586	0.573	0.362	**0.387**	0.366	0.392	**0.360**	0.394	0.454	0.470	0.373	**0.398**	0.380	0.405	**0.370**	0.403
	720	0.977	0.787	0.504	0.459	0.439	0.431	**0.413**	**0.411**	0.499	0.489	0.426	0.429	0.461	0.443	**0.420**	**0.425**
ETTm2	96	0.310	0.388	0.181	0.273	0.163	0.254	**0.162**	**0.254**	0.317	0.379	0.178	0.271	0.172	0.262	**0.167**	**0.258**
	192	0.387	0.438	0.286	0.343	0.224	0.298	**0.216**	**0.290**	0.338	0.390	0.239	0.314	**0.234**	0.305	0.237	**0.299**
	336	0.614	0.558	0.334	0.376	0.287	0.343	**0.278**	**0.336**	0.366	0.404	0.306	0.363	0.288	0.345	**0.282**	**0.338**
	720	1.279	0.813	0.429	0.436	0.409	0.431	**0.390**	**0.405**	0.421	0.446	**0.386**	0.418	0.389	**0.411**	0.395	0.417
Weather	96	0.173	0.244	0.150	0.207	0.153	0.210	**0.147**	**0.198**	0.317	0.373	0.154	0.212	0.154	0.213	**0.153**	**0.207**
	192	0.218	0.288	0.195	0.250	0.200	0.260	**0.195**	**0.248**	0.347	0.397	0.200	0.256	0.201	0.260	**0.193**	**0.248**
	336	0.266	0.328	0.244	0.291	0.260	0.308	**0.244**	**0.287**	0.354	0.392	0.252	0.296	0.256	0.305	**0.247**	**0.293**
	720	0.334	0.380	0.311	0.341	0.355	0.370	**0.310**	**0.336**	0.400	0.421	0.321	0.348	0.362	0.370	**0.313**	**0.341**
Electricity	96	0.137	0.238	0.138	0.237	0.137	0.237	**0.128**	**0.223**	0.227	0.339	0.170	0.278	0.147	0.250	**0.142**	**0.245**
	192	0.157	0.258	0.161	0.258	0.155	0.254	**0.148**	**0.244**	0.231	0.343	0.186	0.293	0.161	**0.263**	**0.159**	0.264
	336	0.193	0.294	0.177	0.275	0.169	0.270	**0.164**	**0.262**	0.259	0.366	0.202	0.309	**0.170**	**0.273**	0.174	0.279
	720	0.281	0.358	0.572	0.576	0.199	0.300	**0.196**	**0.294**	0.297	0.393	0.232	0.334	**0.203**	**0.303**	0.203	0.309

Table 5: Full multivariate time series prediction under different normalization frameworks. The best results for each backbone model are highlighted in bold.



(a) 720 steps-ETTh2

(b) 720 steps-Weather

Figure 3: The long-term forecasting MSE evaluations of different backbone models under various input lengths. Large values are discarded to illustrate the overall trend better.

amount of historical information available for deep models to mine. Ideally, a well-designed model should benefit from longer input sequences, leading to improved forecasting performance. However, recent studies have shown that many Transformer-based deep models struggle with long input sequences (Zeng et al. 2023). In fact, their forecasting accuracy tends to degrade as the input length increases. This phenomenon has been attributed to the non-stationarity of time series data (Liu et al. 2023b). As the input length grows, the distributional discrepancy across time points becomes more pronounced, making it harder for deep models to learn consistent temporal patterns. Therefore, as a model-agnostic framework for addressing non-stationarity, we expect that after processing non-stationarity with WDAN, the prediction errors of deep models would decrease or at least remain stable with increasing sequence length.

To validate our hypothesis, we evaluated the long-term ($H = 720$) forecasting performance of various backbone models under different input lengths $L \in \{192, 264, 336, 528, 720, 900, 1080\}$. The MSE evaluation results are presented in Fig. 3. We can observe that with the assistance of WDAN, the predictive performance of deep backbone models is significantly enhanced across different input lengths. Without WDAN, most models suffer from performance degradation as input length increases across both datasets. In contrast, with WDAN, this degradation is substantially mitigated, and the performance of most backbone models improves as the input length grows. To be specific, on the Weather dataset, iTransformer achieves a reduction on MSE of 11.36% when prolonging input from 192 steps to 1080 steps, and the average improvement on four backbones is 10.28%. These results align well with our expectations and also validate the effectiveness of WDAN on various input lengths.

B.3 Hyperparameter Sensitivity

We investigate the sensitivity of WDAN to three key hyperparameters: the number of MLP layers in the statistics prediction module, the hidden dimension and the level of the DWT. The results of the sensitivity analysis are shown in Fig. 4. We observe that in the highly non-stationary Exchange dataset, different hyperparameter settings significantly impact prediction results. It suggests that careful hyperparameter tuning is particularly important for non-stationary datasets. In contrast, on other relatively stationary datasets, the model performance remains relatively robust across different hyperparameter settings.

B.4 Training Strategy Ablation

In existing normalization methods, different training strategies have been adopted for the statistics prediction module and the backbone model. To validate the effectiveness of our three-stage training strategy, we conducted ablation experiments comparing it with other training approaches found in existing work. Specifically, we compared four training strategies: (1) Three-stage, the approach adopted in this paper and in DDN (Dai et al. 2024); (2) Two-stage (Alternative): alternately training the statistical prediction module and the backbone model, freezing the parameters of the statistical prediction module when training the backbone; (3) Two-stage (Co-train): training the statistical prediction module in the first stage, then simultaneously training both modules in the second stage; (4) Single-stage: directly training both modules simultaneously.

Table 6 show the results of these ablation experiments, showing that the three-stage strategy outperforms other

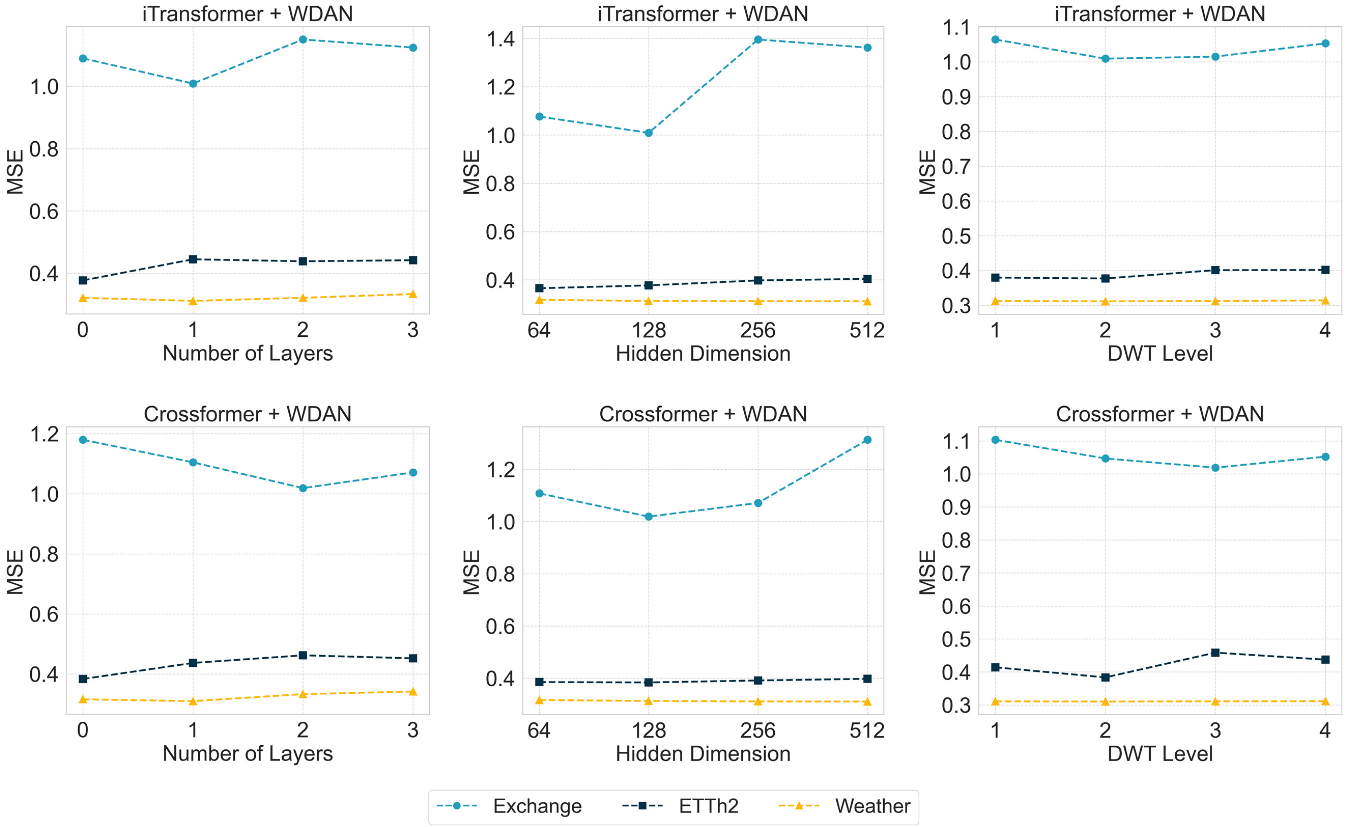


Figure 4: Hyperparameter sensitivity with respect to the number of MLP layers of statistics prediction module, the hidden dimension and the level of DWT. The results are recorded with the lookback window length $T = 720$ and the forecast window length $H = 720$.

training strategies in most settings. These results indicate that the three-stage training strategy offers greater training flexibility while maintaining a strong lower bound on predictive performance.

B.5 Error Bar

We report in Table 7 the means and standard deviations of the evaluation metrics for the backbone models and WDAN under three runs using different random seeds. The results demonstrate that the proposed framework exhibits robust performance across various backbone models and consistently performs well on all seven benchmark datasets.

Methods		iTransformer + WDAN							
		Three-stage		Two-stage (Alternative)		Two-stage (Co-train)		Single-stage	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	96	**0.083**	**0.203**	0.094	0.217	0.083	0.203	0.094	0.219
	192	**0.176**	**0.300**	0.188	0.312	0.177	0.302	0.201	0.324
	336	0.361	0.436	**0.361**	**0.436**	0.369	0.437	0.424	0.475
	720	**1.009**	**0.767**	1.115	0.811	1.009	0.767	1.139	0.819
ETTh2	96	**0.269**	**0.334**	0.276	0.340	**0.269**	**0.334**	0.274	0.339
	192	**0.331**	**0.376**	0.342	0.382	**0.331**	**0.376**	0.336	0.381
	336	**0.358**	**0.403**	0.359	0.405	0.365	0.407	0.370	0.411
	720	**0.377**	**0.429**	0.385	0.436	0.401	0.442	0.396	0.440
Weather	96	**0.147**	**0.199**	0.148	0.204	0.150	0.202	0.150	0.206
	192	0.200	**0.251**	0.198	0.254	0.209	0.259	**0.197**	0.257
	336	**0.242**	**0.282**	0.245	0.291	0.253	0.292	0.252	0.304
	720	**0.312**	**0.337**	0.313	0.342	0.319	0.340	0.322	0.350

Methods		Crossformer + WDAN							
		Three-stage		Two-stage (Alternative)		Two-stage (Co-train)		Single-stage	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	96	**0.084**	**0.205**	0.091	0.214	**0.084**	**0.205**	0.095	0.220
	192	**0.174**	**0.299**	0.191	0.314	**0.174**	**0.299**	0.204	0.326
	336	**0.358**	**0.437**	0.373	0.452	0.364	0.440	0.402	0.469
	720	**1.067**	**0.788**	1.176	0.838	1.172	0.817	1.123	0.816
ETTh2	96	**0.269**	**0.337**	0.272	0.337	0.276	0.337	0.276	0.344
	192	**0.333**	**0.377**	0.334	0.381	0.344	0.383	0.351	0.390
	336	**0.359**	**0.405**	0.366	0.411	0.371	0.408	0.379	0.418
	720	**0.379**	**0.425**	0.397	0.436	0.391	0.433	0.415	0.450
Weather	96	**0.147**	**0.198**	0.149	0.205	0.152	0.202	0.150	0.205
	192	0.207	0.259	0.200	0.258	0.221	0.269	**0.197**	**0.253**
	336	**0.243**	**0.285**	0.243	0.295	0.264	0.300	0.248	0.297
	720	**0.310**	**0.336**	0.313	0.341	0.321	0.341	0.322	0.352

Table 6: Results of training strategy ablation study with iTransformer and Crossformer as the backbone models. The best results are highlighted in bold.

Methods		iTransformer		+WDAN		PatchTST		+WDAN	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	96	0.118±0.006	0.250±0.008	**0.083±0.001**	**0.203±0.001**	0.101±0.007	0.230±0.010	**0.084±0.001**	**0.204±0.000**
	192	0.246±0.019	0.370±0.014	**0.176±0.002**	**0.300±0.002**	0.200±0.008	0.326±0.003	**0.176±0.004**	**0.302±0.005**
	336	0.429±0.002	0.493±0.002	**0.361±0.008**	**0.436±0.005**	0.405±0.062	0.470±0.041	**0.347±0.003**	**0.429±0.001**
	720	1.101±0.006	0.796±0.002	**1.009±0.015**	**0.767±0.004**	1.463±0.235	0.904±0.072	**1.081±0.009**	**0.781±0.008**
ETTh1	96	0.390±0.001	0.422±0.001	**0.368±0.000**	**0.397±0.000**	0.399±0.004	0.419±0.005	**0.369±0.002**	**0.398±0.002**
	192	0.427±0.001	0.448±0.001	**0.406±0.000**	**0.421±0.001**	0.447±0.004	0.450±0.004	**0.407±0.003**	**0.422±0.002**
	336	0.459±0.003	0.471±0.002	**0.428±0.001**	**0.444±0.000**	0.472±0.038	0.474±0.022	**0.443±0.011**	**0.449±0.006**
	720	0.562±0.005	0.545±0.005	**0.452±0.003**	**0.470±0.002**	0.519±0.028	0.511±0.017	**0.515±0.061**	**0.504±0.027**
ETTh2	96	0.301±0.004	0.360±0.004	**0.269±0.000**	**0.334±0.000**	0.322±0.005	0.376±0.002	**0.271±0.002**	**0.335±0.001**
	192	0.379±0.013	0.408±0.007	**0.331±0.000**	**0.376±0.000**	0.412±0.007	0.432±0.006	**0.336±0.001**	**0.376±0.001**
	336	0.415±0.014	0.436±0.005	**0.358±0.001**	**0.403±0.000**	0.450±0.007	0.455±0.001	**0.357±0.001**	**0.401±0.001**
	720	0.435±0.008	0.463±0.004	**0.377±0.001**	**0.429±0.001**	0.497±0.045	0.486±0.020	**0.394±0.002**	**0.440±0.002**
ETTm1	96	0.315±0.001	0.368±0.000	**0.291±0.001**	**0.350±0.001**	0.296±0.001	0.354±0.002	**0.293±0.002**	**0.348±0.002**
	192	0.345±0.000	0.386±0.000	**0.330±0.002**	**0.375±0.001**	0.346±0.009	0.388±0.005	**0.335±0.013**	**0.376±0.003**
	336	0.379±0.001	0.407±0.001	**0.362±0.004**	**0.396±0.002**	0.400±0.008	0.421±0.001	**0.362±0.002**	**0.397±0.001**
	720	0.443±0.006	0.444±0.006	**0.412±0.002**	**0.422±0.000**	0.477±0.004	0.464±0.005	**0.413±0.001**	**0.411±0.000**
ETTm2	96	0.182±0.002	0.276±0.002	**0.162±0.001**	**0.253±0.001**	0.179±0.005	0.272±0.005	**0.172±0.007**	**0.257±0.005**
	192	0.244±0.008	0.317±0.004	**0.217±0.002**	**0.292±0.002**	0.248±0.014	0.317±0.006	**0.240±0.008**	**0.304±0.004**
	336	0.298±0.004	0.351±0.002	**0.273±0.005**	**0.331±0.004**	0.311±0.008	0.357±0.005	**0.282±0.007**	**0.334±0.005**
	720	0.376±0.009	0.401±0.004	**0.353±0.001**	**0.382±0.001**	**0.418±0.019**	**0.426±0.009**	0.442±0.014	0.432±0.005
Weather	96	0.177±0.005	0.229±0.007	**0.147±0.000**	**0.199±0.001**	0.153±0.000	0.208±0.001	**0.148±0.003**	**0.202±0.005**
	192	0.226±0.007	0.268±0.004	**0.196±0.011**	**0.248±0.008**	0.206±0.003	0.261±0.004	**0.194±0.004**	**0.248±0.002**
	336	0.285±0.003	0.309±0.002	**0.242±0.000**	**0.282±0.001**	0.243±0.002	**0.286±0.002**	**0.242±0.001**	0.290±0.002
	720	0.359±0.013	0.361±0.006	**0.312±0.002**	**0.337±0.002**	0.315±0.002	0.336±0.003	**0.306±0.001**	**0.332±0.000**
Electricity	96	0.135±0.001	0.232±0.001	**0.128±0.001**	**0.225±0.002**	0.137±0.001	0.241±0.001	**0.128±0.000**	**0.224±0.000**
	192	0.157±0.001	0.253±0.001	**0.146±0.001**	**0.242±0.002**	0.153±0.000	0.256±0.000	**0.145±0.000**	**0.240±0.000**
	336	0.171±0.001	0.269±0.001	**0.156±0.002**	**0.256±0.002**	0.169±0.000	0.272±0.000	**0.160±0.000**	**0.258±0.000**
	720	0.196±0.002	0.290±0.002	**0.182±0.001**	**0.281±0.002**	0.205±0.001	0.302±0.001	**0.194±0.001**	**0.291±0.001**

Methods		Crossformer		+WDAN		FEDformer		+WDAN	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange	96	0.561±0.212	0.547±0.102	**0.084±0.003**	**0.205±0.003**	0.744±0.045	0.668±0.021	**0.087±0.001**	**0.205±0.001**
	192	0.798±0.334	0.683±0.152	**0.174±0.001**	**0.300±0.000**	0.928±0.021	0.760±0.013	**0.180±0.007**	**0.305±0.007**
	336	1.455±0.049	0.975±0.029	**0.350±0.008**	**0.431±0.006**	1.180±0.033	0.857±0.012	**0.386±0.050**	**0.456±0.031**
	720	2.050±0.172	1.163±0.049	**1.019±0.013**	**0.762±0.012**	1.901±0.024	1.068±0.010	**1.110±0.122**	**0.789±0.038**
ETTh1	96	0.391±0.008	0.431±0.010	**0.366±0.001**	**0.396±0.001**	0.486±0.014	0.498±0.009	**0.381±0.003**	**0.415±0.003**
	192	0.490±0.083	0.493±0.063	**0.401±0.001**	**0.415±0.001**	0.522±0.043	0.512±0.020	**0.417±0.003**	**0.438±0.002**
	336	0.939±0.081	0.758±0.039	**0.420±0.006**	**0.429±0.003**	0.545±0.026	0.534±0.017	**0.436±0.006**	**0.452±0.004**
	720	1.080±0.331	0.796±0.135	**0.446±0.010**	**0.468±0.005**	0.678±0.028	0.620±0.019	**0.473±0.007**	**0.484±0.006**
ETTh2	96	1.225±0.230	0.764±0.080	**0.269±0.002**	**0.336±0.003**	0.405±0.006	0.459±0.010	**0.277±0.002**	**0.342±0.002**
	192	1.253±0.308	0.810±0.084	**0.333±0.003**	**0.378±0.002**	0.431±0.017	0.480±0.014	**0.347±0.003**	**0.386±0.002**
	336	1.500±0.290	0.920±0.114	**0.359±0.001**	**0.405±0.000**	0.436±0.002	0.481±0.002	**0.391±0.010**	**0.423±0.006**
	720	3.825±1.747	1.606±0.431	**0.384±0.003**	**0.428±0.002**	0.502±0.024	0.514±0.012	**0.483±0.045**	**0.478±0.024**
ETTm1	96	0.348±0.013	0.389±0.010	**0.293±0.001**	**0.349±0.001**	0.435±0.026	0.457±0.009	**0.295±0.004**	**0.354±0.001**
	192	0.360±0.022	0.395±0.016	**0.332±0.004**	**0.373±0.003**	0.432±0.025	0.455±0.009	**0.330±0.003**	**0.378±0.001**
	336	0.586±0.061	0.573±0.044	**0.360±0.003**	**0.394±0.001**	0.454±0.005	0.470±0.002	**0.370±0.009**	**0.403±0.003**
	720	0.977±0.112	0.787±0.055	**0.413±0.001**	**0.411±0.000**	0.499±0.040	0.489±0.024	**0.420±0.011**	**0.425±0.006**
ETTm2	96	0.310±0.016	0.388±0.027	**0.162±0.001**	**0.254±0.003**	0.317±0.008	0.379±0.003	**0.167±0.003**	**0.258±0.002**
	192	0.387±0.040	0.438±0.020	**0.216±0.002**	**0.290±0.002**	0.338±0.004	0.390±0.004	**0.237±0.004**	**0.299±0.003**
	336	0.614±0.148	0.558±0.077	**0.278±0.011**	**0.336±0.011**	0.366±0.007	0.404±0.001	**0.282±0.004**	**0.338±0.003**
	720	1.279±0.223	0.813±0.079	**0.390±0.014**	**0.405±0.011**	0.421±0.009	0.446±0.010	**0.395±0.001**	**0.417±0.001**
Weather	96	0.173±0.002	0.244±0.003	**0.147±0.000**	**0.198±0.000**	0.317±0.029	0.373±0.028	**0.153±0.007**	**0.207±0.006**
	192	0.218±0.001	0.288±0.001	**0.195±0.009**	**0.248±0.007**	0.347±0.002	0.397±0.006	**0.193±0.002**	**0.248±0.001**
	336	0.266±0.002	0.328±0.001	**0.244±0.002**	**0.287±0.004**	0.354±0.010	0.392±0.008	**0.247±0.005**	**0.293±0.007**
	720	0.334±0.005	0.380±0.006	**0.310±0.001**	**0.336±0.001**	0.400±0.006	0.421±0.008	**0.313±0.003**	**0.341±0.004**
Electricity	96	0.137±0.002	0.238±0.002	**0.128±0.000**	**0.223±0.000**	0.227±0.012	0.339±0.010	**0.142±0.003**	**0.245±0.004**
	192	0.157±0.002	0.258±0.003	**0.148±0.001**	**0.244±0.000**	0.231±0.006	0.343±0.005	**0.159±0.002**	**0.264±0.003**
	336	0.193±0.013	0.294±0.009	**0.164±0.000**	**0.262±0.000**	0.259±0.016	0.366±0.013	**0.174±0.001**	**0.279±0.001**
	720	0.281±0.017	0.358±0.011	**0.196±0.001**	**0.294±0.000**	0.297±0.010	0.393±0.008	**0.203±0.009**	**0.309±0.009**

Table 7: Error bar of WDAN across different backbone models. The best results for each backbone model are highlighted in bold.