FeTS: A Feature-Aware Framework for Time Series Forecasting

Le Wang, Jianyong Chen*, Songbai Liu

School of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
lllucky.086@gmail.com, jychen@szu.edu.cn, songbai@szu.edu.cn

Abstract

Time series forecasting faces a fundamental challenge: the uneven distribution of predictive importance in time series data, where some specific time points and feature combinations carry disproportionately predictive power. As a result, uniform processing methods that treat all data alike inevitably fall short of optimal performance. To address this problem, we propose FeTS, a feature-aware framework that comprehensively learns temporal features through two key components: (i) Adaptive Feature Extraction (AdaFE), which dynamically discovers the most important features within each temporal patch and extracts them on the fly, yielding sharper and more focused local representations; and (ii) Dual-Scale Feed-Forward Network (DSFFN), which strategically integrates fine-grained local features with global long-term dependencies to achieve richer dual-scale representation learning. Extensive experiments on eight benchmark datasets demonstrate that FeTS achieves state-of-the-art performance in time series forecasting tasks, offering a novel solution to the challenge of uneven predictive importance in forecasting.

Code — https://github.com/lllucky111/FeTS

Introduction

Time series forecasting plays a crucial role across numerous domains, including financial analysis (Cao, Li, and Li 2019), weather prediction (Angryk et al. 2020), energy consumption management (Kardakos et al. 2013), and traffic flow optimization (Kadiyala and Kumar 2014). The field has witnessed significant evolution from traditional statistical methods to modern deep learning approaches. Early forecasting primarily relied on statistical techniques like Autoregressive Integrated Moving Average (ARIMA) (Box et al. 2015) and exponential smoothing (Gardner Jr 1985). With the advent of machine learning, methods such as Support Vector Machines (SVM) (Tay and Cao 2001) and Random Forest (Mei et al. 2014) were introduced to capture nonlinear relationships. As time series data grows in scale and complexity, deep learning has emerged as the dominant paradigm for temporal prediction, with architectures including Multi-Layer Perceptrons (MLP) (Das et al. 2023; Liu et al. 2023; Chen et al. 2023; Xu, Zeng, and Xu 2024; Han et al. 2024), Convolutional Neural Networks (CNN) (Luo and Wang 2024; Wang et al. 2022; Liu et al. 2022a), and Transformer (Liu et al. 2021; Zhou et al. 2022b; Liu et al. 2022b; Zhou et al. 2021; Wu et al. 2021; Liu et al. 2024; Nie et al. 2023). These deep learning approaches have demonstrated superior capability in modeling long-range dependencies and capturing complex spatiotemporal patterns, driving groundbreaking advances in time series forecasting.

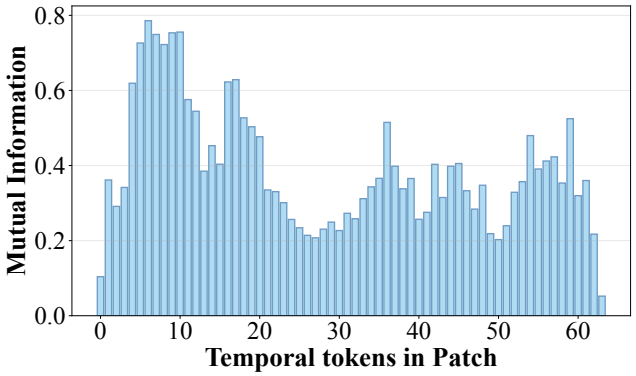


Figure 1: Mutual Information in Patch

Although these models have made progress, they still struggle to address the inherent problem of imbalanced importance in real-world time series data (Nie et al. 2023; Wu et al. 2023; Lin et al. 2023; Zhou et al. 2022a). What does this uneven importance mean? Taking the Weather dataset as an example, observations captured at pivotal transitions, such as shifting seasons or abrupt temperature swings, carry disproportionately greater predictive power for future forecasting than data from stable periods. Our mutual information analysis (Batina et al. 2011) of patch-processed weather data provides quantitative validation of this phenomenon in Figure 1, where the fluctuating strip heights reveal the sharply uneven predictive power of each temporal token in the current patch when forecasting the next. It can be seen that the values of mutual information exhibit a significantly nonuniform distribution, with a clear gap between the predictive power at peak moments and that in low periods. The disparity in predictive power reflects the non-uniform importance across the

*Corresponding Author.

temporal dimension, where for prediction tasks, feature combinations with higher predictive power should be identified as important features and given enhanced attention, representing the core challenge that existing uniform processing approaches fail to address effectively.

Motivated by this insight, we develop FeTS, a feature-aware framework for time series forecasting, designed to precisely capture the time-varying salience of sequential features. Our solution introduces two key innovations: the novel Adaptive Feature Extraction (AdaFE) mechanism and the Dual-Scale Feed-Forward Network (DSFFN). Technically, AdaFE dynamically identifies important features within each patch through a feature scoring mechanism based on a hybrid basis space, enabling more precise local pattern capture. This adaptive approach fundamentally shifts away from traditional uniform processing, effectively mitigating the pronounced imbalance in feature importance across the sequence. Meanwhile, DSFFN integrates global contextual guidance with localized patterns, leveraging dual-scale fusion for richer feature representation. Together, these components allow FeTS to consistently achieve state-of-the-art performance across diverse domains, opening new possibilities in time series analysis. In summary, our contributions are reflected in three key aspects:

- We demonstrate that predictive salience is intrinsically uneven across time series and introduce FeTS, a dynamic and feature-aware framework that explicitly redistributes temporal importance to boost forecasting accuracy.
- We propose AdaFE that crafts fine-grained local features via adaptive and patch-wise extraction, while the proposed DSFFN fuses these representations across different scales.
- FeTS consistently achieves state-of-the-art performance across diverse domains while maintaining superior computational efficiency, establishing a new paradigm for time series forecasting.

Related Work

Developments of Time Series Forecasting

Time series forecasting approaches fall into three main paradigms (Jia et al. 2024): MLP-based, CNN-based, and Transformer-based models. MLP-based approaches like DLinear (Zeng et al. 2023), FITS (Xu, Zeng, and Xu 2024), and FreTS (Yi et al. 2023) employ simple network architectures to comprehensively analyze time series characteristics, including trends and periodicity in both frequency and time domains. CNN-based forecasting models showcase the prowess of convolutional architectures: MICN (Wang et al. 2022) marries causal convolutions with a multi-scale structure, while ModernTCN (Luo and Wang 2024) revamps the convolutional stack with contemporary design principles. Transformer-based solutions combine attention mechanisms with signal processing techniques, as seen in Autoformer (Wu et al. 2021) and FEDformer (Zhou et al. 2022b), while Crossformer (Zhang and Yan 2023) and iTransformer (Liu et al. 2024) enhance performance by examining either temporal dependencies among points or cross-variable relationships.

Development of Patch-Based Models

Since TimesNet (Wu et al. 2023), PatchTST (Nie et al. 2023), and TSMixer (Ekambaram et al. 2023) demonstrate that segmenting time into patches unlocks richer local structure than point-wise modeling, patch-based methods have rapidly become the prevailing paradigm. PatchMixer (Gong, Tang, and Liang 2023) elevates this paradigm by leveraging depthwise convolutions to extract sharper patch-level representations. Other developments include Transformer-based architectures like Pathformer (Chen et al. 2024) and MTST (Zhang et al. 2024), as well as MLP-based frameworks such as MSD-Mixer (Zhong et al. 2023), WPMixer (Murad, Aktukmak, and Yilmaz 2025), and PatchMLP (Tang and Zhang 2025), employing different patch structures to capture multi-scale information. Meanwhile, by incorporating RNN components, SegRNN (Lin et al. 2023) achieves segmental information iteration.

In this paper, we identify the non-uniform importance problem in time series, where certain temporal points and feature combinations contain information that is more critical for prediction. Based on this observation, we propose the FeTS model. Unlike earlier patch-based models, which process patch information uniformly, FeTS incorporates an innovative adaptive patch-level feature-aware system. This system dynamically identifies and extracts features with higher predictive value within patches, thereby facilitating more effective learning and enhancing prediction performance.

Methodology

Time series forecasting involves predicting output sequences of length H from multivariate input sequences of length L (Lin et al. 2024a). However, real-world time series data inherently feature an uneven distribution of importance. To address this limitation, we propose FeTS, a novel and flexible framework capable of effectively extracting and integrating important features, thereby achieving robust forecasting performance, as illustrated in Figure 2. Technically, given input $X \in \mathbb{R}^{M \times L}$ with M variables and length L, we use PatchTST's patching approach (Nie et al. 2023) to produce embeddings $X_{\text{emb}} \in \mathbb{R}^{M \times N \times D}$, where D is patch embedding dimension and N denotes the number of patches. Using X_{emb}, we perform both AdaFE-based extraction and DSFFN's dual-scale fusion. The following sections detail these core modules' operations.

AdaFE

The **AdaFE** block executes two key operations: (i) importance scoring and activation via the **Fourier-Poly Mask**, and (ii) sparse convolution through **Mask-Controlled Einsum**. Through these two stages, the model can specifically focus on positions that hold significant predictive power within each patch.

Fourier-Poly Mask: importance scoring and activation At this stage, we identify and select important features within each patch. Specifically, we evaluate the importance of each temporal point to the periodicity and trend of the sequence, score each point for its importance, and use these scores to filter out the more critical features.

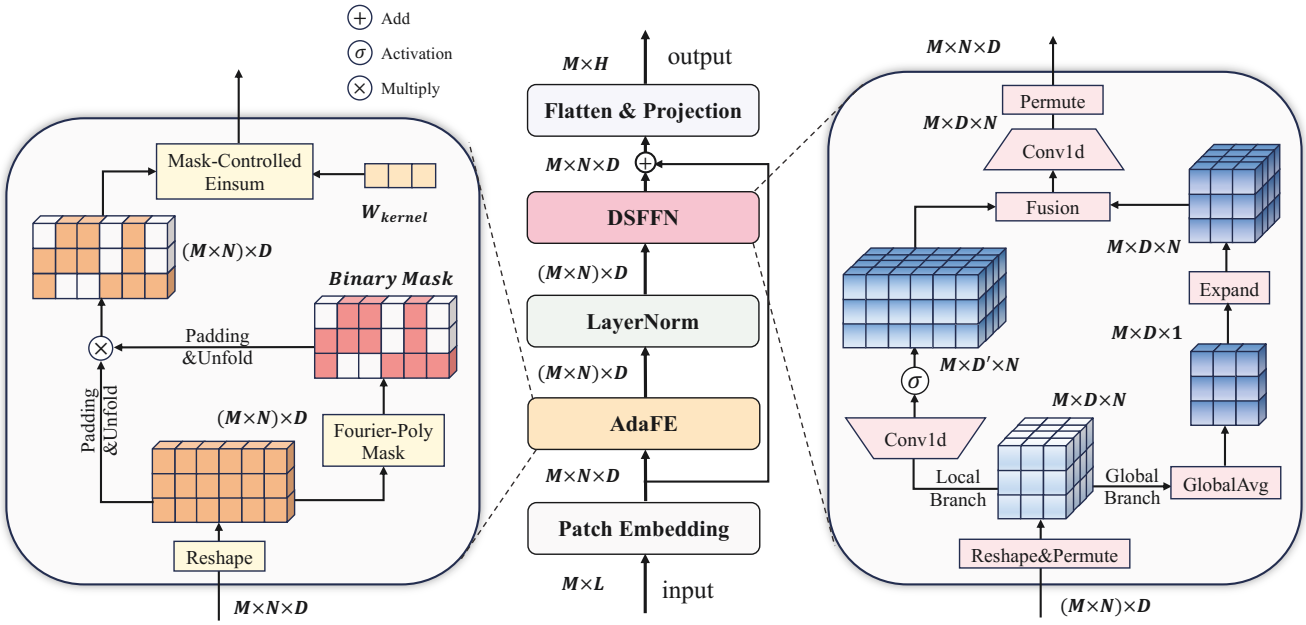


Figure 2: Structure overview of FeTS.

Theorem 1. *Let $f(x)$ be a periodic function with period 2π that satisfies the Dirichlet conditions (i.e., f has a finite number of discontinuities and extrema in any bounded interval). Then $f(x)$ can be represented as an infinite sum of sine and cosine functions (Butzer and Nessel 1971):*

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cos(nx) + b_n \sin(nx)\right]. \quad (1)$$

Theorem 2. *If $g(x)$ is a continuous real-valued function on a closed interval $[a, b]$ and if $\epsilon > 0$ is given, then there exists a polynomial $P(x)$ such that $|g(x) - P(x)| < \epsilon$ for all $x \in [a, b]$ (Stone 1948).*

Based on the approximation theories outlined in Theorem 1 and Theorem 2, we construct a hybrid transformation space integrating spectral and polynomial representations: $s(x) = f(x) + g(x)$. The Fourier component $f(x)$, which utilizes sine/cosine basis functions, models periodic patterns—effectively capturing oscillatory modes and periodic behaviors within time series. Additionally, the polynomial component $g(x)$ captures trend elements to model aperiodic trends and gradual changes, where higher-order terms adapt to increasingly complex trend patterns.

This integrated approach learns a nonlinear importance representation function that simultaneously evaluates spectral characteristics and trend correlations. Through global interaction, we then derive the final importance scores. The scoring mechanism selectively focuses on critical temporal positions, identifying significant periodic events or key trend shifts, thereby enabling the model to dynamically activate key points with higher predictive power.

We reshape the input tensor $X_{\text{emb}} \in \mathbb{R}^{M\times N\times D}$ into $X_{\text{flat}} \in \mathbb{R}^{(M\times N)\times D}$ by flattening both variable and temporal dimensions, enabling efficient importance computation across feature dimension D. In the forward pass, X_{flat} is transformed using Fourier and polynomial basis functions as follows, forming the basis of our nonlinear transformation space:

$$X_{\cos} = \cos(X_{\text{flat}} \otimes \mathbf{k}_{\cos} \cdot \pi) \in \mathbb{R}^{(M\times N)\times D\times(F+1)}, \quad (2)$$

$$X_{\sin} = \sin(X_{\text{flat}} \otimes \mathbf{k}_{\sin} \cdot \pi) \in \mathbb{R}^{(M\times N)\times D\times F}, \quad (3)$$

$$X_{\text{poly}} = X_{\text{flat}}^{\mathbf{k}_{\text{poly}}} \in \mathbb{R}^{(M\times N)\times D\times(P+1)}, \quad (4)$$

where F is the maximum frequency index for Fourier terms, P is the highest degree for polynomial terms, $\otimes$ denotes the tensor outer product transforming inputs to the frequency domain, and $X_{\text{flat}}^{\mathbf{k}_{\text{poly}}}$ performs element-wise exponentiation of X_{flat} to the powers specified in $\mathbf{k}_{\text{poly}}$, capturing nonlinear relationships through polynomial terms. The frequency indices $\mathbf{k}_{\cos}$ and $\mathbf{k}_{\sin}$ range from 0 to F and 1 to F respectively, serving as characteristic frequencies for our Fourier basis and determining detectable periodic components, while $\mathbf{k}_{\text{poly}}$ ranges from 0 to P, controlling the complexity of representable nonlinear patterns through polynomial degree selection.

Next, we introduce three groups of learnable coefficients $a_f \in \mathbb{R}^{D\times D\times(F+1)}$, $b_f \in \mathbb{R}^{D\times D\times F}$, and $c_p \in \mathbb{R}^{D\times D\times(P+1)}$ to weight their contributions across frequency

bands and polynomial terms, resulting in an importance representation $O \in \mathbb{R}^{(M\times N)\times D}$ that comprehensively models nonlinear data patterns:

$$\begin{aligned} O[r,d] = &\sum_{i=1}^{D}\sum_{j=0}^{F} X_{\cos}[r,i,j]\cdot a_f[i,d,j] \\ &+ \sum_{i=1}^{D}\sum_{j=1}^{F} X_{\sin}[r,i,j]\cdot b_f[i,d,j] \\ &+ \sum_{i=1}^{D}\sum_{j=0}^{P} X_{\text{poly}}[r,i,j]\cdot c_p[i,d,j] + \beta, \end{aligned} \tag{5}$$

where $r \in \{1,\ldots,M\times N\}$, $d \in \{1,\ldots,D\}$, and β is a learnable bias term. Then this intermediate representation undergoes global interaction to generate the final importance score Z:

$$Z = Linear(O) \in \mathbb{R}^{(M\times N)\times D}. \tag{6}$$

The score Z quantifies the relative importance of each position. To enable selective focus on critical positions, we employ a threshold-based activation method that uses the average score $\mu_Z[r]$ as the threshold to filter key features. This generates a binary mask matrix for subsequent key feature activation:

$$\mu_Z[r] = \frac{1}{D}\sum_{d=1}^{D} Z[r,d], \tag{7}$$

$$\text{mask}[r,d] = \begin{cases} 1 & \text{if } Z[r,d] \geq \mu_Z[r], \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Mask-Controlled Einsum: sparse convolution In the second stage, we perform sparse convolution guided by the activation mask to concentrate computation on activated positions. First, we apply symmetric padding to both the input X_{emb} and the mask to maintain consistent operation at boundaries, yielding X_{padding} and $\text{mask}_{\text{padding}}$. Using an unfold operation, we generate sliding window representations $X_{\text{un}} \in \mathbb{R}^{(M\times N)\times D\times k}$ and $\text{mask}_{\text{un}} \in \mathbb{R}^{(M\times N)\times D\times k}$, where k is the convolution kernel size. This extracts patches of size k centered at each position. Finally, we implement the masked convolution via an optimized Einsum operation that efficiently computes the weighted sum of input patches under mask control:

$$Y[r,d] = \sum_{j=1}^{k} W[j]\cdot X_{\text{un}}[r,d,j]\cdot \text{mask}_{\text{un}}[r,d,j], \tag{9}$$

where $Y \in \mathbb{R}^{(M\times N)\times D}$ denotes the output sequence with activated important features, $W \in \mathbb{R}^k$ is the learnable weight. The binary mask selectively enables computation only at activated positions ($\text{mask}_{\text{un}}[r,d,j] = 1$), enabling more flexible and precise extraction of critical features from different patches.

DSFFN

After extracting finer features from patches, we enhance the model's comprehension by introducing different temporal granularities and performing dual-scale structure fusion. To achieve this, we propose the **DSFFN**, comprising two parallel branches: a **Local Branch** for local pattern extraction and a **Global Branch** for long-range dependency modeling. Their outputs are fused to generate unified feature representations. To ensure a clear separation of local and global perspectives, we first reshape the layer-normalized output from $Y \in \mathbb{R}^{(M\times N)\times D}$ to $Y' \in \mathbb{R}^{M\times D\times N}$.

In the Local Branch, we use a standard 1D convolutional network (Liu et al. 2022c) to learn local patterns from the reshaped tensor $Y' \in \mathbb{R}^{M\times D\times N}$:

$$F_{\text{local}} = \phi(W_{\text{local}} * Y' + b_{\text{local}}), \tag{10}$$

where $*$ denotes convolution operation, $W_{\text{local}} \in \mathbb{R}^{D'\times D\times K}$ is a learnable convolution kernel with D' as the output dimension, $b_{\text{local}} \in \mathbb{R}^{D'}$ is the bias term, and $\phi(\cdot)$ is the GELU activation (Hendrycks and Gimpel 2016), introducing nonlinearity to enhance information expressiveness. The output $F_{\text{local}} \in \mathbb{R}^{M\times D'\times N}$ captures localized temporal patterns.

In the Global Branch, we apply a global average pooling operation along the temporal dimension N to aggregate sequence information into the global representation:

$$F_{\text{global}} = \text{Expand}\left(\frac{1}{N}\sum_{i=1}^{N} Y'[:,:,i]\right), \tag{11}$$

where $Y'[:,:,i] \in \mathbb{R}^{M\times D}$ represents the feature slice at temporal position i. The global average pooling operation produces a tensor of shape $\mathbb{R}^{M\times D}$, which is then expanded to match the temporal dimension of F_{local}, yielding $F_{\text{global}} \in \mathbb{R}^{M\times D\times N}$.

After acquiring dual-scale features from both branches, we integrate local and global representations through three sequential processes. First, we concatenate both data along the temporal dimension to generate a combined representation. Then, fusion convolution is applied to enhance interactions between local and global features. Finally, the integrated representation undergoes processing via a regularized projection layer with dropout to improve model generalization and yield the final output. The specific formulation is as follows:

$$F_{\text{combined}} = [F_{\text{local}}, F_{\text{global}}] \in \mathbb{R}^{M\times(D'+D)\times N}, \tag{12}$$

$$F_{\text{fused}} = W_{\text{fused}} * F_{\text{combined}} + b_{\text{fused}}, \tag{13}$$

$$Out = \text{Dropout}(W_{\text{out}} * F_{\text{fused}} + b_{\text{out}}), \tag{14}$$

where $[\cdot,\cdot]$ denotes concatenation, $\text{Dropout}(\cdot)$ denotes dropout process. $W_{\text{fused}} \in \mathbb{R}^{D_{\text{fused}}\times(D'+D)\times K'}$ and $W_{\text{out}} \in \mathbb{R}^{D_{\text{out}}\times D_{\text{fused}}\times K''}$ are the fusion convolution kernel (kernel size K') and the output convolution kernel (kernel size K'') respectively, with corresponding bias terms $b_{\text{fused}} \in \mathbb{R}^{D_{\text{fused}}}$ and $b_{\text{out}} \in \mathbb{R}^{D_{\text{out}}}$. The fusion feature $F_{\text{fused}} \in \mathbb{R}^{M\times D_{\text{fused}}\times N}$ captures dual-scale information combining both local and global features, generating the richer feature representation $Out \in \mathbb{R}^{M\times D_{\text{out}}\times N}$ for downstream prediction tasks, thereby providing adaptability to the inconsistent predictive importance existing across various scales in time series data.

Dataset	ETTh1		ETTh2		ETTm1		ETTm2		Weather		Electricity		Traffic		Solar-Energy	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PatchTST [2023]	0.484	0.476	0.410	0.431	0.356	0.388	0.265	0.328	0.233	0.271	0.168	0.268	0.404	0.281	0.226	0.305
Crossformer [2023]	0.760	0.647	1.617	0.955	0.453	0.466	1.025	0.693	0.254	0.315	0.185	0.279	0.560	0.312	0.218	0.258
DLinear [2023]	0.444	0.454	0.468	0.463	0.358	0.381	0.282	0.345	0.246	0.299	0.167	0.264	0.435	0.297	0.252	0.313
TimesNet [2023]	0.502	0.487	0.415	0.443	0.422	0.423	0.282	0.333	0.248	0.285	0.199	0.300	0.616	0.333	0.238	0.285
iTransformer [2024]	0.450	0.457	0.390	0.416	0.367	0.395	0.272	0.329	0.238	0.272	0.162	0.257	**0.395**	0.278	0.287	0.340
TimeXer [2024]	0.492	0.482	0.375	0.410	0.371	0.395	0.262	0.317	0.226	0.265	0.171	0.270	0.420	0.281	0.303	0.346
SparseTSF [2024]	0.418	0.423	0.371	0.406	0.359	0.385	0.271	0.325	0.235	0.280	0.165	0.256	0.412	**0.270**	0.196	0.246
TimeMixer [2024]	0.434	0.439	0.396	0.425	0.382	0.397	0.273	0.325	0.226	**0.263**	0.168	0.260	0.408	0.274	0.197	0.266
ModernTCN [2024]	0.404	0.421	0.322	0.378	0.355	0.383	0.260	0.318	0.228	0.268	0.163	0.257	0.410	0.279	0.193	0.269
Amplifier [2025]	0.417	0.426	0.341	0.390	0.358	0.385	0.266	0.324	0.231	0.271	0.169	0.261	0.424	0.295	0.200	0.272
ConvTimeNet [2025]	0.409	0.424	0.325	0.377	0.357	0.382	0.259	0.318	0.233	0.269	0.162	0.256	0.404	0.277	0.203	0.251
FeTS	**0.398**	**0.420**	**0.314**	**0.369**	**0.349**	**0.380**	**0.251**	**0.316**	**0.225**	**0.263**	**0.161**	**0.254**	0.401	0.272	**0.183**	**0.240**

Table 1: Multivariate long-term time series forecasting results. The best results are highlighted in **bold** and the second best are underlined.

Experiments

Setup

Datasets We conduct experiments on widely used benchmark datasets, including ETT series (4 subsets), Weather, Traffic, Electricity, and Solar-Energy (Lai et al. 2018; Lin et al. 2024a). The preprocessing operations on all datasets, including data splitting and normalization methods, are kept consistent with established practices in prior works such as Autoformer (Wu et al. 2021) and Informer (Zhou et al. 2021).

Baselines We choose recent state-of-the-art models as our benchmark, including Transformer-based model: PatchTST (Nie et al. 2023), iTransformer (Liu et al. 2024), Crossformer (Zhang and Yan 2023), TimeXer (Wang et al. 2024b); MLP-based model: DLinear (Zeng et al. 2023), SparseTSF (Lin et al. 2024b), TimeMixer (Wang et al. 2024a), Amplifier (Fei et al. 2025); CNN-based model: TimesNet (Wu et al. 2023), ModernTCN (Luo and Wang 2024), ConvTimeNet (Cheng et al. 2025). The lower MSE or MAE indicates a more accurate prediction result.

Main Results

Table 1 presents the performance comparison of FeTS against baseline models across 8 datasets. For fair evaluation, the lookback window L is fixed at 336, with average performance computed across prediction horizons of $H \in \{96, 192, 336, 720\}$ as the reference metric. Overall, FeTS achieves superior performance on seven datasets, outperforming Transformer-based models (PatchTST, iTransformer, Crossformer, TimeXer), MLP-based models (DLinear, SparseTSF, TimeMixer, Amplifier), and CNN-based models (TimesNet, ModernTCN, ConvTimeNet). Notably, on the highly volatile Solar-Energy dataset, FeTS significantly outperforms models that exclusively learn point-level sequential relationships (such as iTransformer and DLinear). Compared to patch-based uniform local processing methods (such as PatchTST, ModernTCN, and ConvTimeNet), FeTS also delivers substantial performance gains. This underscores the intrinsic shortcomings of both (i) scattering attention across globally uneven sequences and (ii) imposing uniform indiscriminate local processing. FeTS effectively tackles these challenges by integrating dynamic mechanisms into a patch-based dual-scale modeling framework. This design enables the model to pinpoint salient features and capture more nuanced local characteristics, thereby pioneering a novel approach to time series modeling.

We also observe that on the Traffic dataset, which exhibits unique properties including numerous extreme values and strong spatio-temporal correlations (Lin et al. 2024a), FeTS shows slightly higher MSE than iTransformer, an architecture focused on variable relationships (Liu et al. 2024). For MAE, it performs marginally worse than SparseTSF, which prioritizes overall pattern recognition (Lin et al. 2024b). However, in comprehensive evaluations, FeTS demonstrates balanced superiority by outperforming iTransformer in MAE and surpassing SparseTSF in MSE. These results confirm the broad applicability of FeTS's dual-scale feature-aware approach across diverse datasets.

Ablation Study

To evaluate the contributions of AdaFE and DSFFN in FeTS, we perform comprehensive ablation studies across five datasets, evaluating both component replacement (Replace) and removal (w/o), where the Replace sub-experiment indicates we replace the AdaFE and DSFFN modules with ordinary CNN and FFN module, respectively. The comparative results, averaged over four lengths of prediction (96, 192, 336, and 720), are presented in Table 2.

Experimental results show that FeTS, incorporating both AdaFE and DSFFN components, achieves the best performance. Compared to replacing the AdaFE module with a standard CNN or directly removing it, FeTS outperforms in MSE and MAE with an average reduction of 3.6% and 3.19%, respectively. These results confirm the necessity of the adaptive feature extraction mechanism and indicate that, compared with the uniform processing method of CNN, FeTS constructs a hybrid basis space to adaptively identify key patterns containing rich periodic information and nonlinear

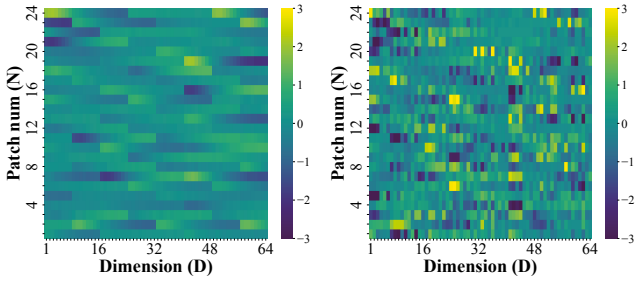
Model	FeTS		Replace AdaFE		w/o AdaFE		Replace DSFFN		w/o DSFFN	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.349**	**0.380**	0.359	0.392	0.355	0.382	0.356	0.385	0.358	0.388
Weather	**0.225**	**0.263**	0.240	0.270	0.237	0.271	0.235	0.271	0.231	0.268
Electricity	**0.161**	**0.254**	0.168	0.262	0.165	0.260	0.169	0.255	0.166	0.259
Traffic	**0.401**	**0.272**	0.413	0.284	0.414	0.282	0.411	0.283	0.412	0.287
Solar-Energy	**0.183**	**0.240**	0.199	0.259	0.186	0.248	0.197	0.257	0.192	0.250
Avg	**0.264**	**0.282**	0.276	0.293	0.271	0.289	0.274	0.290	0.272	0.290

Table 2: Ablation study of AdaFE and DSFFN in FeTS. The best results are highlighted in **bold**. Avg denotes the average performance across the five datasets.

patterns—thereby capturing precise local features with higher predictive value and improving overall prediction performance. Similarly, compared to replacing DSFFN with FFN or completely removing this module, FeTS maintains a lead of 3.41% and 2.84% in MSE and MAE, respectively. This demonstrates that dual-perspective features are indispensable in FeTS, where incorporating global feature guidance enables the model to learn interactions between local patterns and the global context, thereby avoiding issues of global consistency loss that can arise from local optimization in traditional approaches. AdaFE and DSFFN's synergistic combination creates an effective extraction-fusion pipeline that progressively enhances temporal representations, ultimately forming a comprehensive and accurate prediction framework.

Model Analysis

Visualizing the effects of AdaFE In FeTS, we use an adaptive feature extraction method, as opposed to the uniform processing of local features. Specifically, AdaFE leverages the Fourier-polynomial hybrid basis space to automatically identify key moments that contain abundant periodic regularities and key nonlinear dynamics, thereby extracting sparse feature representations with high discriminative power from data. To better demonstrate AdaFE's identification performance, we visualize its input and output data with the Weather dataset (consisting of 24 patches), with results shown in Figure 3.



(a) Data at input of AdaFE. (b) Data at output of AdaFE.

Figure 3: Heatmap comparison of data at input and output of AdaFE.

Specifically, Figure 3 (a) presents the initial data after patch embedding, with a uniform color distribution indicating that important features have not yet been emphasized and the data lack marked discriminative power. In contrast, Figure 3 (b) shows the results extracted by AdaFE, displaying a clearer structure, with blue (low-value) regions expanding significantly while yellow (high-value) regions are sparsely distributed. This indicates that AdaFE successfully detects key features, captures various important details within the patches, and boosts feature discriminative power while increasing information density, thereby providing more accurate local features for subsequent predictions.

AdaFE generality To further validate AdaFE's effectiveness, we examine its general applicability using the classic PatchTST and TimesNet models as test cases. Specifically, we incorporate the AdaFE module after PatchTST's patch embedding operation and following TimesNet's 2D time series transformation to enhance local feature accuracy. Table 3 presents a performance comparison between models before and after integrating this module across four datasets. The results show that AdaFE significantly improves the predictive performance of these two classical models, reducing MSE by 7.73% and 8.68%, respectively. By introducing a dynamic salient feature identification mechanism, these patch-based models demonstrate marked improvements in precise local feature extraction, resulting in enhanced overall prediction accuracy.

Model	PatchTST		PatchTST + AdaFE		TimesNet		TimesNet +AdaFE	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.484	0.476	**0.436**	**0.442**	0.502	0.487	**0.455**	**0.454**
ETTh2	0.410	0.431	**0.369**	**0.407**	0.415	0.443	**0.372**	**0.406**
Weather	0.233	0.271	**0.230**	**0.267**	0.248	0.285	**0.240**	**0.271**
Solar-Energy	0.226	0.305	**0.204**	**0.269**	0.238	0.285	**0.210**	**0.257**
Improved(Avg)	-	-	**7.73%**	**6.50%**	-	-	**8.68%**	**7.47%**

Table 3: AdaFE generality for the classic patch-based models. The results are calculated as the average over four prediction lengths (96, 192, 336, and 720). **Improved(Avg)** is the average improvement accuracy across the four datasets. The better results are highlighted in **bold**.

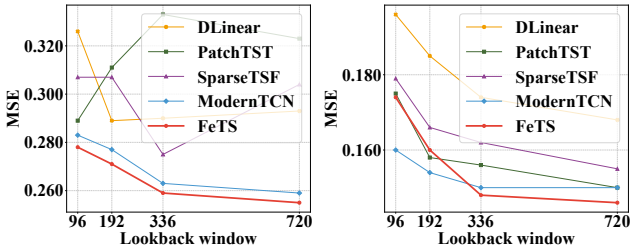
Impact of hyperparameter D We evaluate the impact of the hyperparameter D (the feature dimension after patch embedding) on FeTS, as shown in Table 4. Experiments are conducted on the four datasets with prediction horizons of $H \in \{96, 192, 336, 720\}$. FeTS is tested at $D \in \{32, 64, 128, 256, 512\}$, using ModernTCN, which similarly utilizes patch segmentation processing within a CNN framework, as the baseline for performance comparison.

The model architecture demonstrates that varying D values represent distinct needs for recognizing and extracting feature combinations. As shown in Table 4, the model exhibits consistent performance across datasets of different sizes, with minimal fluctuation among different D values. Moreover, even with sub-optimal D configurations, the results predominantly outperform the ModernTCN model without the adaptive feature extraction mechanism. This indicates that FeTS is robust to dimensional scaling of D, where various sizes of D can all adaptively identify more predictive feature combinations

across local feature spaces. Furthermore, these results suggest that the superior performance of FeTS primarily stems from its feature extraction and adaptive learning mechanisms, rather than dependence on specific embedding dimensions.

Dataset		ETTh2		ETTm1		Weather		Solar-Energy	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
FeTS	$D = 32$	0.316	0.373	0.355	0.381	0.230	0.264	0.188	0.247
	$D = 64$	**0.314**	0.371	0.350	0.382	**0.225**	**0.263**	0.187	0.245
	$D = 128$	**0.314**	**0.369**	**0.349**	**0.380**	**0.225**	**0.263**	**0.183**	**0.240**
	$D = 256$	0.317	0.373	**0.349**	0.381	0.226	0.264	0.187	0.243
	$D = 512$	0.320	0.377	0.350	0.382	0.228	0.265	0.193	0.249
ModernTCN		0.322	0.378	0.355	0.383	0.228	0.268	0.193	0.269

Table 4: Performance comparison of FeTS with different D values and ModernTCN model. The best results are highlighted in **bold**.



(a) ETTh2 (b) Weather

(c) Electricity (d) Solar-Energy

Figure 4: Performance of FeTS and other models with different lookback windows. The forecast horizon is set as 96.

Varying lookback windows To further evaluate FeTS's performance under different lookback windows, we conduct additional experiments with other lookback window settings and select models from Table 1 for comparison, including MLP-based models (DLinear, SparseTSF), Transformer-based models (PatchTST), and CNN-based models (ModernTCN). As illustrated in Figure 4, the MSE of FeTS exhibits a consistent downward trend with the increase in window size. This phenomenon indicates that FeTS is capable of adaptively identifying critical features across diverse temporal characteristics under varying lookback windows. Moreover, it can derive more comprehensive and enriched representations from extended lookback windows, ultimately enabling more accurate predictive performance.

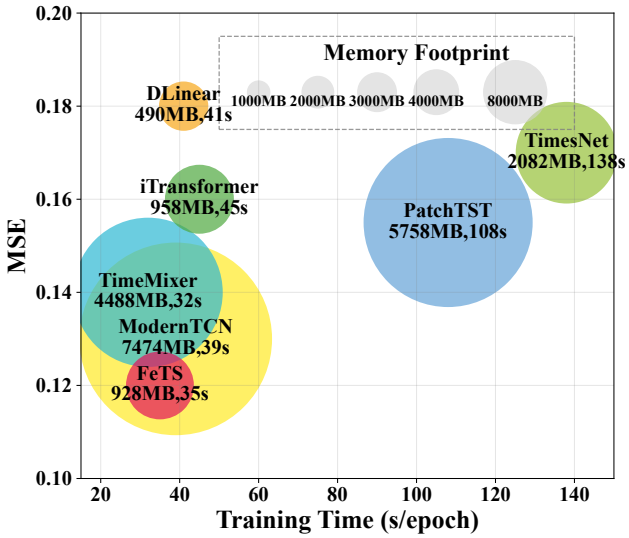


Figure 5: Model efficiency comparison. The running efficiency of seven models on the Weather dataset with the prediction length $H = 96$.

Eiffcient analysis To validate the efficiency performance of FeTS, we conduct comparative experiments on the Weather dataset with a fixed input length L as 336 and prediction length H as 96 against baseline models. Figure 5 demonstrates that FeTS shows superior performance in memory efficiency, training speed, and prediction accuracy. While maintaining lower memory usage and faster training times, FeTS achieves the best predictive results among all models. Notably, even with just a single-layer network architecture, FeTS surpasses multi-layer stacked models in capturing deep sequential patterns, delivering higher predictive accuracy while reducing memory usage and accelerating training speed. When compared to the structurally simple DLinear model, FeTS demonstrates clear advantages in both training efficiency and predictive performance, confirming its overall effectiveness as a comprehensive solution.

Conclusion

In this paper, we demonstrate that time series exhibit unevenly distributed importance, a phenomenon overlooked by existing unified mapping methods. To address this, we propose FeTS, a feature-aware framework that incorporates two core modules, AdaFE and DSFFN, to enhance the model's ability to focus on critical features, acquire more refined local features, and effectively learn diverse patterns. Experiments show that FeTS consistently delivers state-of-the-art performance across diverse time series forecasting tasks, outperforming other patch-based models. Our findings highlight the importance of effectively capturing important features to mitigate the impact of unevenly distributed importance on prediction performance, inspiring exploration into deeper connections between future deep learning architectures and time series characteristics.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grant 62306180, the Natural Science Foundation of Guangdong Province under Grant 2023A1515011238, the Shenzhen Science and Technology Program under Grant JCYJ20250604181503004.

References

Angryk, R. A.; Martens, P. C.; Aydin, B.; Kempton, D.; Mahajan, S. S.; Basodi, S.; Ahmadzadeh, A.; Cai, X.; Filali Boubrahimi, S.; Hamdi, S. M.; et al. 2020. Multivariate time series dataset for space weather data analytics. *Scientific data*, 7(1): 227.

Batina, L.; Gierlichs, B.; Prouff, E.; Rivain, M.; Standaert, F.-X.; and Veyrat-Charvillon, N. 2011. Mutual information analysis: a comprehensive study. *Journal of Cryptology*, 24(2): 269–291.

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

Butzer, P. L.; and Nessel, R. J. 1971. Fourier analysis and approximation, Vol. 1. *Reviews in Group Representation Theory, Part A (Pure and Applied Mathematics Series, Vol. 7)*.

Cao, J.; Li, Z.; and Li, J. 2019. Financial time series forecasting model based on CEEMDAN and LSTM. *Physica A: Statistical mechanics and its applications*, 519: 127–139.

Chen, P.; Zhang, Y.; Cheng, Y.; Shu, Y.; Wang, Y.; Wen, Q.; Yang, B.; and Guo, C. 2024. Pathformer: Multi-scale Transformers with Adaptive Pathways for Time Series Forecasting. In *International Conference on Learning Representations*.

Chen, S.-A.; Li, C.-L.; Yoder, N.; Arik, S. O.; and Pfister, T. 2023. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*.

Cheng, M.; Yang, J.; Pan, T.; Liu, Q.; Li, Z.; and Wang, S. 2025. Convtimenet: A deep hierarchical fully convolutional model for multivariate time series analysis. In *Companion Proceedings of the ACM on Web Conference 2025*, 171–180.

Das, A.; Kong, W.; Leach, A.; Mathur, S.; Sen, R.; and Yu, R. 2023. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*.

Ekambaram, V.; Jati, A.; Nguyen, N.; Sinthong, P.; and Kalagnanam, J. 2023. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 459–469.

Fei, J.; Yi, K.; Fan, W.; Zhang, Q.; and Niu, Z. 2025. Amplifier: Bringing Attention to Neglected Low-Energy Components in Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 11645–11653.

Gardner Jr, E. S. 1985. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1): 1–28.

Gong, Z.; Tang, Y.; and Liang, J. 2023. Patchmixer: A patch-mixing architecture for long-term time series forecasting. *arXiv preprint arXiv:2310.00655*.

Han, L.; Chen, X.-Y.; Ye, H.-J.; and Zhan, D.-C. 2024. SOFTS: Efficient Multivariate Time Series Forecasting with Series-Core Fusion. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Hendrycks, D.; and Gimpel, K. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Jia, Y.; Lin, Y.; Yu, J.; Wang, S.; Liu, T.; and Wan, H. 2024. PGN: The RNN's New Successor is Effective for Long-Range Time Series Forecasting. *Advances in Neural Information Processing Systems*, 37: 84139–84168.

Kadiyala, A.; and Kumar, A. 2014. Multivariate time series models for prediction of air quality inside a public transportation bus using available software. *Environmental Progress & Sustainable Energy*, 33(2): 337–341.

Kardakos, E. G.; Alexiadis, M. C.; Vagropoulos, S. I.; Simoglou, C. K.; Biskas, P. N.; and Bakirtzis, A. G. 2013. Application of time series and artificial neural network models in short-term forecasting of PV power generation. In *2013 48th International Universities' Power Engineering Conference (UPEC)*, 1–6. IEEE.

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.

Lin, S.; Lin, W.; Hu, X.; Wu, W.; Mo, R.; and Zhong, H. 2024a. Cyclenet: enhancing time series forecasting through modeling periodic patterns. *Advances in Neural Information Processing Systems*, 37: 106315–106345.

Lin, S.; Lin, W.; Wu, W.; Chen, H.; and Yang, J. 2024b. SparseTSF: Modeling Long-term Time Series Forecasting with 1k Parameters. In *Forty-first International Conference on Machine Learning*.

Lin, S.; Lin, W.; Wu, W.; Zhao, F.; Mo, R.; and Zhang, H. 2023. Segrnn: Segment recurrent neural network for long-term time series forecasting. *arXiv preprint arXiv:2308.11200*.

Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; and Xu, Q. 2022a. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35: 5816–5828.

Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A. X.; and Dustdar, S. 2021. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*.

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 *The Twelfth International Conference on Learning Representations*.

Liu, Y.; Li, C.; Wang, J.; and Long, M. 2023. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in neural information processing systems*, 36: 12271–12290.

Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2022b. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35: 9881–9893.

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022c. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11976–11986.

Luo, D.; and Wang, X. 2024. ModernTCN: A modern pure convolution structure for general time series analysis. In *The Twelfth International Conference on Learning Representations*.

Mei, J.; He, D.; Harley, R.; Habetler, T.; and Qu, G. 2014. A random forest method for real-time price forecasting in New York electricity market. In *2014 IEEE PES general meeting—conference & exposition*, 1–5. IEEE.

Murad, M. M. N.; Aktukmak, M.; and Yilmaz, Y. 2025. WPMixer: Efficient multi-resolution mixing for long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 19581–19588.

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*.

Stone, M. H. 1948. The generalized Weierstrass approximation theorem. *Mathematics Magazine*, 21(5): 237–254.

Tang, P.; and Zhang, W. 2025. Unlocking the Power of Patch: Patch-Based MLP for Long-Term Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 12640–12648.

Tay, F. E.; and Cao, L. 2001. Application of support vector machines in financial time series forecasting. *omega*, 29(4): 309–317.

Wang, H.; Peng, J.; Huang, F.; Wang, J.; Chen, J.; and Xiao, Y. 2022. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The Eleventh International Conference on Learning Representations*.

Wang, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J. Y.; and ZHOU, J. 2024a. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

Wang, Y.; Wu, H.; Dong, J.; Qin, G.; Zhang, H.; Liu, Y.; Qiu, Y.; Wang, J.; and Long, M. 2024b. TimeXer: Empowering Transformers for Time Series Forecasting with Exogenous Variables. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34: 22419–22430.

Xu, Z.; Zeng, A.; and Xu, Q. 2024. FITS: Modeling Time Series with $10k$ Parameters. In *The Twelfth International Conference on Learning Representations*.

Yi, K.; Zhang, Q.; Fan, W.; Wang, S.; Wang, P.; He, H.; An, N.; Lian, D.; Cao, L.; and Niu, Z. 2023. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems*, 36: 76656–76679.

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.

Zhang, Y.; Ma, L.; Pal, S.; Zhang, Y.; and Coates, M. 2024. Multi-resolution time-series transformer for long-term forecasting. In *International conference on artificial intelligence and statistics*, 4222–4230. PMLR.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *International Conference on Learning Representations*.

Zhong, S.; Song, S.; Zhuo, W.; Li, G.; Liu, Y.; and Chan, S.-H. G. 2023. A multi-scale decomposition mlp-mixer for time series analysis. *arXiv preprint arXiv:2310.11959*.

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.

Zhou, T.; Ma, Z.; Wen, Q.; Sun, L.; Yao, T.; Yin, W.; Jin, R.; et al. 2022a. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in neural information processing systems*, 35: 12677–12690.

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