

# DUET: Dual Clustering Enhanced Multivariate Time Series Forecasting

Xiangfei Qiu  
East China Normal University  
Shanghai, China  
xfqiu@stu.ecnu.edu.cn

Xingjian Wu  
East China Normal University  
Shanghai, China  
xjwu@stu.ecnu.edu.cn

Yan Lin  
Aalborg University  
Aalborg, Denmark  
lyan@cs.aau.dk

Chenjuan Guo  
East China Normal University  
Shanghai, China  
cjguo@dase.ecnu.edu.cn

Jilin Hu ✉  
East China Normal University  
Shanghai, China  
jlhu@dase.ecnu.edu.cn

Bin Yang  
East China Normal University  
Shanghai, China  
byang@dase.ecnu.edu.cn

## Abstract

Multivariate time series forecasting is crucial for various applications, such as financial investment, energy management, weather forecasting, and traffic optimization. However, accurate forecasting is challenging due to two main factors. First, real-world time series often show heterogeneous temporal patterns caused by distribution shifts over time. Second, correlations among channels are complex and intertwined, making it hard to model the interactions among channels precisely and flexibly.

In this study, we address these challenges by proposing a general framework called **DUET**, which introduces **D**Ual clustering on the temporal and channel dimensions to **E**nhance multivariate **T**ime series forecasting. First, we design a Temporal Clustering Module (TCM) that clusters time series into fine-grained distributions to handle heterogeneous temporal patterns. For different distribution clusters, we design various pattern extractors to capture their intrinsic temporal patterns, thus modeling the heterogeneity. Second, we introduce a novel Channel-Soft-Clustering strategy and design a Channel Clustering Module (CCM), which captures the relationships among channels in the frequency domain through metric learning and applies sparsification to mitigate the adverse effects of noisy channels. Finally, DUET combines TCM and CCM to incorporate both the temporal and channel dimensions. Extensive experiments on 25 real-world datasets from 10 application domains, demonstrate the state-of-the-art performance of DUET.

## ACM Reference Format:

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu ✉, and Bin Yang. 2018. **DUET: Dual Clustering Enhanced Multivariate Time Series Forecasting**. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 14 pages. <https://doi.org/XXXXXXX.XXXXXXX>

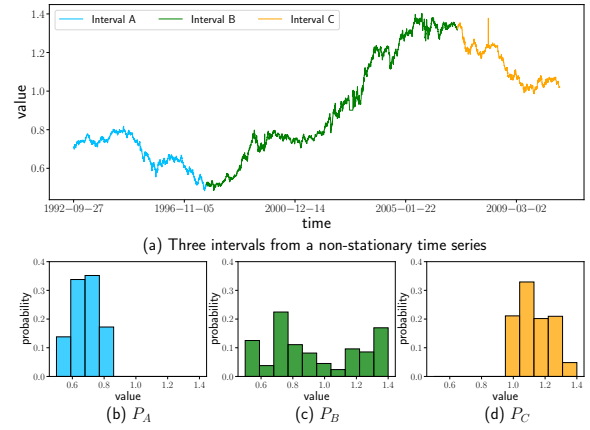
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/18/06

<https://doi.org/XXXXXXX.XXXXXXX>



**Figure 1: A Non-stationary time series with three intervals A, B, and C, exhibiting varying value distributions ( $P_A \neq P_B \neq P_C$ ) and temporal patterns.**

## 1 Introduction

Multivariate time series is a type of time series that organizes timestamps chronologically and involves multiple channels (a.k.a., variables) at each timestamp [21, 22, 28, 58, 61, 63]. In recent years, multivariate time series analysis has seen remarkable progress, with key tasks such as anomaly detection [24, 42, 44, 62, 72, 78], classification [6, 80], and imputation [17, 64–66], among others [23, 27, 79, 81], gaining attention. Among these, multivariate time series forecasting (MTSF) [9, 26, 69, 83, 84, 88, 90] stands out as a critical and widely studied task. It has been extensively applied in diverse domains, including economics [29, 56], traffic [11, 31, 73, 74, 76, 77], energy [19, 59, 67], and AIOPs [5, 8, 36, 39, 52], highlighting its importance and impact.

Building an MTSF method typically involves modeling correlations on the temporal and channel dimensions. However, real-world time series often exhibit heterogeneous temporal patterns caused by the shifting of distribution over time, a phenomenon called *Temporal Distribution Shift (TDS)* [4, 15]. Additionally, the correlation among multiple channels is complex and intertwined [55]. Therefore, developing a method that can effectively extract heterogeneous temporal patterns and channel dependencies is essential yet challenging. Specifically, achieving these goals is hindered by two major challenges.

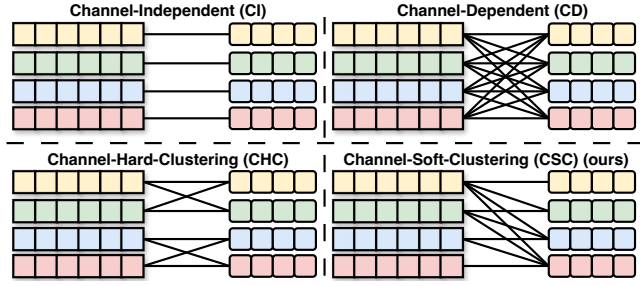


Figure 2: Channel strategies. Different colors represent different channels, with squares representing features before processing with various channel strategies, and squares with rounded corners representing features after processing.

First, **heterogeneous temporal patterns caused by TDS are difficult to model**. In real applications, time series that describe unstable systems are easily influenced by external factors [50, 82]. Such non-stationarity of time series implies the data distribution changes over time, a phenomenon called TDS [4, 15, 16]. TDS causes time series to have different temporal patterns, formally known as heterogeneity of temporal patterns [15, 16, 47, 48]. For example, Figure 1(a) illustrates a time series in economics, which shows the changes with the international circumstances. We can observe that the three intervals A, B, and C follow different temporal distributions, as evidenced by the value histograms shown in Figures 1(b), 1(c), and 1(d). This shift in distribution also comes with varying temporal patterns. As shown in Figure 1(a), the blue interval A shows a descending trend, the green interval B shows an increasing trend, and the yellow interval C shows a steeper descending trend. It is crucial to incorporate these patterns considering their common presence in time series. However, recent studies [37, 45, 68, 75] primarily address heterogeneous temporal patterns in an implicit manner, which ultimately undermines prediction accuracy.

Second, **complex channel interrelations are difficult to model flexibly**. For an MTSF task, it is crucial to model the correlations among different channels, as the predictive accuracy for a particular channel can often be enhanced by leveraging information from other related channels. For example, in weather forecasting, temperature predictions can be improved by incorporating data on humidity, wind speed, and pressure, as these factors are interrelated and provide a more comprehensive view of weather conditions.

Researchers have explored various strategies to manage multiple channels, including 1) treating each channel independently [45, 51] (*Channel-Independent, CI*), 2) assuming each channel correlates with all other channels [87] (*Channel-Dependent, CD*), and 3) grouping channels into clusters [43] (*Channel-Hard-Clustering, CHC*). Figure 2 illustrates these three strategies. CI imposes the constraint of using the same model across different channels. While it offers robustness [51], it overlooks potential interactions among channels and can be limited in generalizability and capacity for unseen channels [20, 55]. CD, on the other hand, considers all channels simultaneously and generates joint representations for decoding [87], but may be susceptible to noise from irrelevant channels, reducing the model’s robustness. CHC partitions multivariate time series into disjoint clusters through hard clustering, applying CD modeling

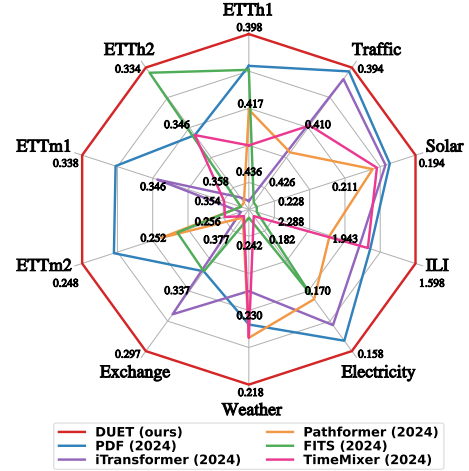


Figure 3: Performance of DUET. Results (MSE) are averaged from all forecasting horizons. DUET outperforms strong baselines in 10 commonly used datasets.

methods within each cluster and CI methods among clusters [43]. However, this approach only considers relationships within the same cluster, limiting flexibility and versatility. In conclusion, there is yet an approach to model the complex interactions among channels precisely and flexibly.

In this study, we address the above two challenges by proposing a general framework, **DUET**, which introduces *D*ual clustering on the temporal and channel dimensions to *E*nhance multivariate *T*ime series forecasting. First, to model heterogeneous temporal patterns caused by TDS, we design a *Temporal Clustering Module* (TCM). This module clusters time series into fine-grained distributions, allowing us to use various pattern extractors to capture their intrinsic temporal patterns, thereby modeling the heterogeneity of temporal patterns. This method effectively handles both stationary and non-stationary data, demonstrating strong generality. Second, to flexibly model relationships among channels, we propose a *Channel Clustering Module* (CCM). Using a channel-soft-clustering strategy, this module captures relationships among channels in the frequency domain through metric learning and applies sparsification. This approach enables each channel to focus on those beneficial for downstream prediction tasks, while mitigating the impact of noisy or irrelevant channels, thereby achieving effective channel soft clustering. Finally, the *Fusion Module* (FM), based on a masked attention mechanism, efficiently combines the temporal features extracted by the TCM with the channel mask matrix generated by the CCM. Experimental results show that the proposed DUET achieves state-of-the-art performance on real-world forecasting datasets—see Figure 3.

Our contributions are summarized as follows.

- To address MTSF, we propose a general framework called DUET. It learns an accurate and adaptive forecasting model through dual clustering on both temporal and channel dimensions.
- We design the TCM that clusters time series into fine-grained distributions. Various pattern extractors are then designed for

different distribution clusters to capture their unique temporal patterns, modeling the heterogeneity of temporal patterns.

- We design the CCM that flexibly captures the relationships among channels in the frequency domain through metric learning and applies sparsification.
- We conduct extensive experiments on 25 datasets. The results show that DUET outperforms state-of-the-art baselines. Additionally, all datasets and code are available at <https://github.com/decisionintelligence/DUET>.

## 2 Related Works

### 2.1 Temporal Distribution Shift in MTSF

Time series forecasting suffers from *Temporal Distribution Shift* (TDS), as the distribution of real-world series changes over time [1, 13, 40]. In recent years, various methods have been proposed to address this issue. Some works tackle TDS from a **normalization perspective**. DAIN [53] adaptively normalizes the series with non-linear neural networks. RevIN [32] utilizes instance normalization on input and output sequences by normalizing the input sequences and then denormalizing the model output sequences. Dish-TS [16] identifies intra- and inter-space distribution shifts in time series and mitigates these issues by learning distribution coefficients. Non-stationary Transformer [46] presents de-stationary attention that incorporates non-stationary factors in self-attention, significantly improving transformer-based models. Some works address TDS from a **distribution perspective**. DDG-DA [34] predicts evolving data distribution in a domain adaptation fashion. AdaRNN [15] proposes an adaptive RNN to alleviate the impact of non-stationary factors by characterizing and matching distributions. Other works address TDS from a **time-varying model parameters perspective**. Triformer [10] proposes a light-weight approach to enable variable-specific model parameters, making it possible to capture distinct temporal patterns from different variables. ST-WA [11] use distinct sets of model parameters for different time period.

Despite the effectiveness of existing methods, our work explicitly models heterogeneous temporal patterns separately under different distributions, which can further improve the performance.

### 2.2 Channel Strategies in MTSF

It is essential to consider the correlations among channels in MTSF. Most existing methods adopt either a *Channel-Independent (CI)* or *Channel-Dependent (CD)* strategy to utilize the spectrum of information in channels. CI strategy approaches [12, 37, 51, 75] share the same weights across all channels and make forecasts independently. Conversely, CD strategy approaches [7, 10, 25, 41, 45, 87] consider all channels simultaneously and generates joint representations for decoding. The CI strategy is characterized by low model capacity but high robustness, whereas the CD strategy exhibits the opposite characteristics. DGCformer [43] proposes relatively balanced channel strategies called *Channel-Hard-Clustering (CHC)*, trying to mitigate this polarization effect and improve predictive capabilities. Specifically, DGCformer designs a graph clustering module to assign channels with significant similarities into the same cluster, utilizing the CD strategy inside each cluster and the CI strategy across them. This approach adopts the CHC strategy, focusing solely on channel correlations within the same cluster. As

a result, this method suffers from the limitations of rigidly adhering to channel-similarity rules defined by human experience.

Different from the above methods, we adopt a Channel-Soft-Clustering (CSC) strategy and devise a fully adaptive sparsity module to dynamically build group for each channel, which is a more comprehensive design covering the CHC strategy.

## 3 Preliminaries

### 3.1 Definitions

*Definition 3.1 (Time series).* A time series  $X \in \mathbb{R}^{N \times T}$  is a time-oriented sequence of  $N$ -dimensional time points, where  $T$  is the number of timestamps, and  $N$  is the number of channels. If  $N = 1$ , a time series is called univariate, and multivariate if  $N > 1$ .

For convenience, we separate dimensions with commas. Specifically, we denote  $X_{i,j} \in \mathbb{R}$  as the  $i$ -th channel at the  $j$ -th timestamp,  $X_{n,:} \in \mathbb{R}^T$  as the time series of  $n$ -th channel, where  $n = 1, \dots, N$ . We also introduce some definitions used in our methodology:

*Definition 3.2 (Temporal Distribution Shift [15]).* Given a time series  $X \in \mathbb{R}^{N \times L}$ , by sliding the window, we get a set of time series with the length of  $T$ , denoted as  $\mathcal{D} = \{X_{n,i:i+T} | n \in [1, N] \& i \in [1, L - T]\}$ , where each  $X_{n,i:i+T}$  equals to such  $X_{n,:}$ . Then, temporal distribution shift is referred to the case that  $\mathcal{D}$  can be clustered into  $K$  sets, i.e.,  $\mathcal{D} = \bigcup_{i=1}^K \mathcal{D}_i$ , where each  $\mathcal{D}_i$  denotes the set with data distribution  $P_{\mathcal{D}_i}(x)$ , where  $P_{\mathcal{D}_i}(x) \neq P_{\mathcal{D}_j}(x), \forall i \neq j$  and  $1 \leq i, j \leq K$ .

*Definition 3.3 (Frequency Metric Space).* The Frequency Metric Space is defined by the frequency space and the distance metric. The frequency space consists of functions from  $L^2$  space represented by infinite fourier basis, of which the coordinates are calculated by Fourier Transform. And the metric is used to measure the distances among functions in the frequency space. For discrete univariate time series with length  $T$ , we use rFFT [3] to represent them under  $\frac{T}{2}$  fourier basis and design proper metric to measure the distances.

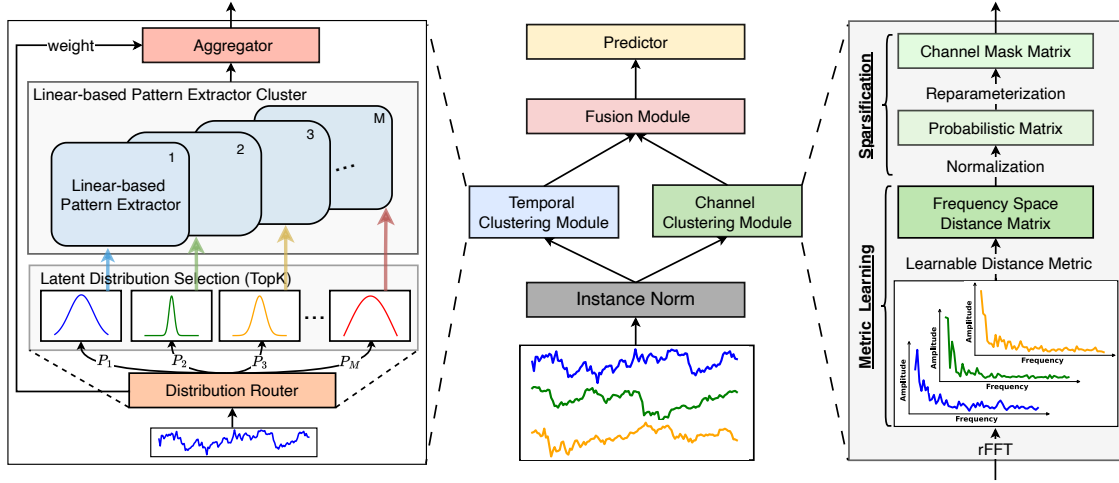
### 3.2 Problem Statement

**Multivariate Time Series Forecasting** aims to predict the next  $F$  future timestamps, formulated as  $Y = \langle X_{:,T+1}, \dots, X_{:,T+F} \rangle \in \mathbb{R}^{N \times F}$  based on the historical time series  $X = \langle X_{:,1}, \dots, X_{:,T} \rangle \in \mathbb{R}^{N \times T}$  with  $N$  channels and  $T$  timestamps.

## 4 Methodology

### 4.1 Structure Overview

Figure 4 shows the architecture of DUET, which adopts a dual clustering on both temporal and channel dimensions, simultaneously mining intrinsic temporal patterns and dynamic channel correlations. Specifically, we first use the Instance Norm [32] to unify the distribution of training and testing data. Then, the Temporal Clustering Module (TCM) utilizes a specially designed Distribution Router (Figure 5a) to capture the potential latent distributions of each time series  $X_{n,:} \in \mathbb{R}^T$  in a channel-independent way, and then clusters time series with similar latent distributions by assigning them to the same group of Linear-based Pattern Extractors (Figure 5b). In this way, we can mitigate the issue that single structure



**Figure 4: The architecture of DUET. Temporal Clustering Module clusters time series into fine-grained distribution. For different distribution clusters, various pattern extractors are designed to capture their intrinsic temporal patterns. Channel Clustering Module flexibly captures the relationships among channels in the frequency domain space through Metric Learning and applies Sparsification. Fusion Module combines the temporal features and the channel mask matrix.**

cannot fully extract temporal features due to heterogeneity of temporal patterns, even with millions of parameters. Meanwhile, the Channel Clustering Module (CCM) captures the correlations among channels in the frequency space in a channel soft clustering way. By leveraging an adaptive metric learning technique and applying sparsification, the CCM outputs a learned channel-mask matrix, so each channel can focus on those beneficial for downstream prediction and isolate the adverse effects of irrelevant channels in a sparse connection way. Finally, the Fusion Module (FM), based on a masked attention mechanism (Figure 5d), effectively combines the temporal features extracted by the TCM and the channel-mask matrix generated by the CCM. A linear predictor is then used to forecast the future values at the end of our framework.

According to the above-mentioned description, the process of DUET can be formulated as follows:

$$X^{norm} = \text{InstanceNorm}(X) \quad (1)$$

$$X^{temp} = \text{TCM}(X^{norm}) \quad (2)$$

$$\mathcal{M} = \text{CCM}(X^{temp}) \quad (3)$$

$$X^{mix} = \text{FM}(X^{temp}, \mathcal{M}), \hat{Y} = \text{Predictor}(X^{mix}), \quad (4)$$

where  $X^{temp}, X^{mix} \in \mathbb{R}^{N \times d}$ ,  $\mathcal{M} \in \mathbb{R}^{N \times N}$ . Steps 2 and 3 can be computed simultaneously for higher efficiency. Below, we will introduce the details of each module in our framework.

## 4.2 Temporal Clustering Module (TCM)

To model heterogeneous temporal patterns caused by TDS, we design a Linear-based Pattern Extractor Cluster, where each Linear-based Pattern Extractor extracts temporal features for time series that has the same latent distribution. We adopt a linear model as the basic structure because it has been proven to efficiently extract temporal information [37, 38, 68, 75, 86], which also helps to keep

the cluster lightweight. Moreover, we design a simple yet effective Distribution Router—see Figure 5a, which can extract the potential latent distributions of the current time series and determine the corresponding Linear-based Pattern Extractor. Both the routing and extracting processes are conducted in a channel-independent way, focusing on clustering univariate time series  $X_{n,:} \in \mathbb{R}^T$  based on their distributions and fully extracting their temporal patterns. **Distribution Router:** Inspired by VAE [33], we first design two fully connected layer-based encoders to adaptively capture  $M$  candidate latent distributions for each time series  $X_{n,:} \in \mathbb{R}^T$  (see Figure 5a), where  $M$  denotes the size of the Linear-based Pattern Extractor Cluster. Empirically, we assume each time series follows a latent normal distribution and the process can be formulated as follows.

$$\text{Encoder}_\mu(X_{n,:}) = \text{ReLU}(X_{n,:} \cdot W_0^\mu) \cdot W_1^\mu \quad (5)$$

$$\text{Encoder}_\sigma(X_{n,:}) = \text{ReLU}(X_{n,:} \cdot W_0^\sigma) \cdot W_1^\sigma, \quad (6)$$

where  $W_0^\mu, W_0^\sigma \in \mathbb{R}^{T \times d_0}$ ,  $W_1^\mu, W_1^\sigma \in \mathbb{R}^{d_0 \times M}$ . Then, we utilize the Noisy Gating technique [57] to choose  $k$  most possible distributions which  $X_{n,:}$  potentially belongs to and calculate the corresponding weights. We can observe that the reparameterization trick used to measure normal distributions shares similarities with the noise addition technique in Noisy Gating, so we elegantly combine them in a unified form:

$$Z_n = \text{Encoder}_\mu(X_{n,:}) + \epsilon \odot \text{Softplus}(\text{Encoder}_\sigma(X_{n,:})) \quad (7)$$

$$H(X_{n,:}) = W^H \cdot Z_n, \quad (8)$$

where  $H(X_{n,:}), \epsilon \in \mathbb{R}^M$ ,  $\epsilon_i \sim \mathcal{N}(0, 1)$ ,  $W^H \in \mathbb{R}^{M \times M}$ . The introduction of  $\epsilon$  not only facilitates resampling from a normal distribution but also stabilizes the training process of Noisy Gating. The activation function *Softplus* helps keep the variance to be positive.  $H(X_{n,:})$  denotes the projected weights of distributions. We demonstrate the



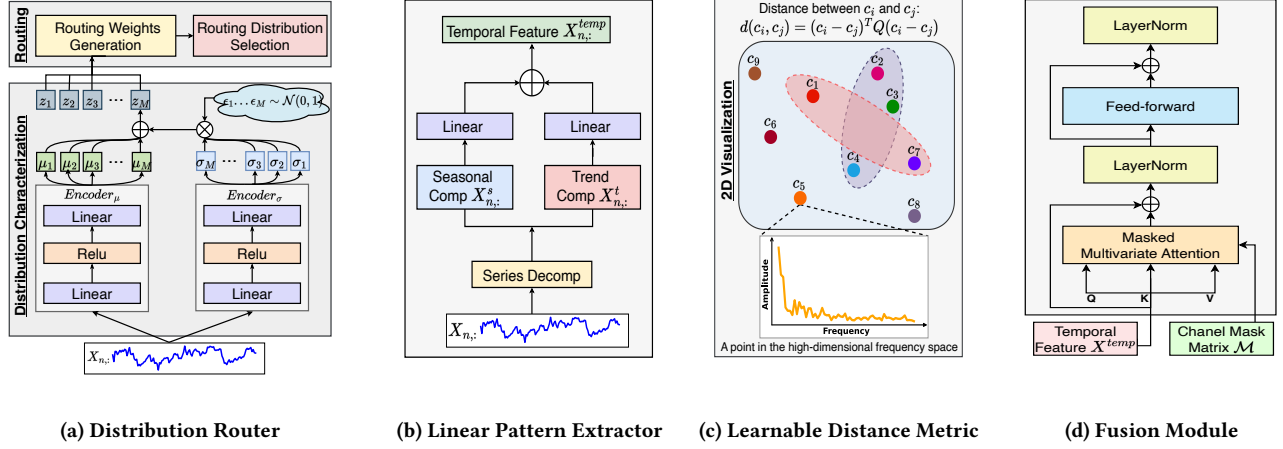


Figure 5: (a) The structure of the Distribution Router, which consists of Distribution Characterization and Routing. (b) The structure of the Linear-based Pattern Extractor, which decomposes the series into seasonal and trend parts, separately extracts temporal features with linear models and reads them. (c) The Learnable Distance Metric is to capture the relationships among channels. (d) The Fusion Module is to combine the temporal features and the channel mask matrix.

computational equivalence to the original Noisy Gating [57] in Appendix A.2. Subsequently, we select  $k$  ( $k \leq M$ ) most possible latent distributions and calculate the weights:

$$G(X_{n,:}) = \text{Softmax}(\text{KeepTopK}(H(X_{n,:}), k)) \quad (9)$$

$$\text{KeepTopK}(H(X_{n,:}), k)_i = \begin{cases} H(X_{n,:})_i & \text{if } i \in \text{ArgTopk}(H(X_{n,:})) \\ -\infty & \text{otherwise} \end{cases}, \quad (10)$$

where  $G(X_{n,:}) \in \mathbb{R}^k$  denotes the probabilities of each candidate distributions. From the perspective of clustering, univariate time series  $X_{n,:} \in \mathbb{R}^T$ ,  $n = 1, \dots, N$  belonging to the same  $k$  most possible latent distributions tend to be processed by the same group of  $k$  Linear-based Pattern Extractors.

**Linear-based Pattern Extractor:** The above-mentioned distribution router determines the latent distributions. Then,  $X_{n,:}$  is passed to corresponding  $k$  selected Linear-based Pattern Extractors for temporal feature extraction. For the  $i$ -th Linear-based Pattern Extractor—see Figure 5b, we first use the moving average technique to decompose  $X_{n,:}$  into seasonal and trend parts [71, 86], then separately extract features and finally fuse them to better capture the patterns of time series from the same distribution:

$$X_{n,:}^t = \text{AvgPool}(\text{padding}(X_{n,:})) \quad (11)$$

$$X_{n,:}^s = X_{n,:} - X_{n,:}^t \quad (12)$$

$$X_{n,:}^{temp^i} = X_{n,:}^t \cdot W_t^i + X_{n,:}^s \cdot W_s^i, \quad (13)$$

where  $X_{n,:}^t, X_{n,:}^s \in \mathbb{R}^T$  are the decomposed trend and seasonal parts,  $W_t^i, W_s^i \in \mathbb{R}^{T \times d}$  are the learnable parameters in linear transformations.  $X_{n,:}^{temp^i} \in \mathbb{R}^d$  is the temporal feature generated by the  $i$ -th extractor, where  $d$  is the hidden dimension.

**Aggregator:** After the selected  $k$  Linear-based Pattern Extractors output the corresponding features, the Aggregator gathers the temporal features based on the previously calculated weighted gates:

$$X_{n,:}^{temp} = \sum_{i=1}^k G(X_{n,:})_i \cdot X_{n,:}^{temp^i}, \quad (14)$$

where  $G(X_{n,:})_i$  represents the weights of the  $i$ -th Linear-based Pattern Extractor, and  $X_{n,:}^{temp} \in \mathbb{R}^d$  represents the temporal patterns of  $X_{n,:}$  extracted by the Linear-based Pattern Extractor Cluster. After obtaining temporal features of each  $X_{n,:}$  ( $n = 1, \dots, N$ ) in a channel-independent way, they are finally gathered into  $X^{temp} \in \mathbb{R}^{N \times d}$ .

### 4.3 Channel Clustering Module (CCM)

To mitigate the adverse effects caused by improper consideration of cross-channel relationships during prediction, we devise an efficient metric learning method to softly cluster channels in the frequency space, and generate a corresponding learned channel-mask matrix to achieve sparse connections.

**Learnable Distance Metric.** To fully utilize the cross-channel enhancement in prediction, we manage to leverage more useful information by modeling the correlations among channels in the perspective of frequency metric space. Theoretically, the “frequency space” describes functions under the Fourier basis, and their coordinates come from Fourier Transform by computing integrals in  $L^2$  space (see Appendix A.3). Specifically, given a discrete time series at the  $n$ -th channel,  $X_{n,:} \in \mathbb{R}^T$ , we first use real Fast Fourier Transform (rFFT [3]) to project it into a finite  $\frac{T}{2}$ -dimension frequency space with the same  $\frac{T}{2}$  Fourier basis (in complex form). Then, we need to find a proper distance metric that precisely evaluates channel relations in the frequency space.

We propose such a distance metric that precisely evaluates channel relations in the frequency space and make each channel obtain maximum neighbour gain in the prediction task. Specifically, we

take the norm (or amplitude) to obtain  $X_{n,:}^{chan} \in \mathbb{R}^{\frac{T}{2}}$  and adopt a learnable Mahalanobis distance metric [18] to adaptively discover the interrelationships among channels:

$$d(X_{i,:}, X_{j,:}) = (X_{i,:}^{chan} - X_{j,:}^{chan})^T \cdot Q \cdot (X_{i,:}^{chan} - X_{j,:}^{chan}) \quad (15)$$

$$X_{i,:}^{chan} = \text{norm}(\text{rFFT}(X_{i,:})), X_{j,:}^{chan} = \text{norm}(\text{rFFT}(X_{j,:})), \quad (16)$$

where  $Q \in \mathbb{R}^{\frac{T}{2} \times \frac{T}{2}}$  is a learnable semi-positive definite matrix. It can be practically constructed by  $Q = A^T \cdot A$ , where  $A$  is also a learnable matrix. This process introduces a more general and light-weight method from the perspective of metric space and adaptively explores a distance metric to measure the channel correlations for better prediction accuracy.

**Normalization.** With the learnable channel distance metric, we first compute the relationship matrix of channels and normalize it to the range of  $[0, 1]$ :

$$D_{ij} = d(X_{i,:}, X_{j,:}) \quad (17)$$

$$C_{ij} = \begin{cases} \frac{1}{D_{ij}} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}, P_{ij} = \begin{cases} \frac{C_{ij} \cdot \gamma}{\max_j(C_{ij})} & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}, \quad (18)$$

where  $D, C, P \in \mathbb{R}^{N \times N}$  are distance, relationship, and probability matrices,  $\gamma \in (0, 1)$  is a discount factor to avoid the absolute connection. Above processes probabilize the relationships among channels, where  $P_{ij}$  represents the probability that channel  $j$  is useful for channel  $i$  in the prediction task.

**Reparameterization.** Since our goal is to filter out the adverse effects of irrelevant channels and retain the beneficial effects of relevant channels, we further perform Bernoulli resampling on the probability matrix to obtain a binary channel mask matrix  $\mathcal{M} \in \mathbb{R}^{N \times N}$ , where  $\mathcal{M}_{ij} \approx \text{Bernoulli}(P_{ij})$ . Higher probability  $P_{ij}$  results in  $\mathcal{M}_{ij}$  closer to 1, indicating a relationship between channel  $i$  and channel  $j$ . Since  $P_{ij}$  contains learnable parameters, we use the Gumbel Softmax reparameterization trick [30] during Bernoulli resampling to ensure the propagation of gradients.

#### 4.4 Fusion Module (FM)

Through the dual clustering on both temporal and channel dimensions, our framework extracts the temporal feature  $X^{temp} \in \mathbb{R}^{N \times d}$  and a channel mask matrix  $\mathcal{M} \in \mathbb{R}^{N \times N}$  for each time series  $X \in \mathbb{R}^{N \times T}$ . Subsequently, we utilize a masked attention mechanism to further fuse them. This process can be formalized as follows.

$$Q = X^{temp} \cdot W^Q, K = X^{temp} \cdot W^K, V = X^{temp} \cdot W^V \quad (19)$$

$$\text{MaskedScores} = \frac{Q \cdot K^T}{\sqrt{d}} \odot \mathcal{M} + (1 - \mathcal{M}) \odot (-\infty) \quad (20)$$

$$X^{mix} = \text{Softmax}(\text{MaskedScores}) \cdot V, \quad (21)$$

where  $W^Q, W^K, W^V \in \mathbb{R}^{d \times d}$  are projection matrices in the attention block,  $\text{MaskedScores} \in \mathbb{R}^{N \times N}$  is the attention score matrix,  $X^{mix} \in \mathbb{R}^{N \times d}$  is the fused feature. With the application of the masked attention mechanism [60], the Fusion Module effectively fuses the temporal features extracted by TCM based on the sparse correlations captured by CCM. We also adopt LayerNorm, Feed-Forward and skip-connection in the Fusion Module (FM) like a classic transformer block. Specifically, LayerNorm [2] and Skip-Connection [85] ensure FM's stability and robustness, while the

Feed-Forward layer [14] empowers the FM to capture complex features. Their synergistic operation optimizes the representational capacity and training efficiency of the Fusion Module.

Finally, we adopt a linear projection to predict the future values, which is formulated as follows.

$$\hat{Y} = X^{mix} \cdot W^O, \quad (22)$$

where  $W^O \in \mathbb{R}^{d \times F}$ ,  $\hat{Y} \in \mathbb{R}^{N \times F}$ .

## 5 Experiments

### 5.1 Experimental Settings

**5.1.1 Datasets.** We use 25 real-world datasets from 10 different domains in time series forecasting benchmark (TFB) [55] to comprehensively evaluate the performance of DUET, more details of the benchmark datasets are included in Table 1. Due to space constraints, we report the results on 10 well-acknowledged forecasting datasets which include ETT (ETTh1, ETTh2, ETTm1, ETTm2), Exchange, Weather, Electricity, ILL, Traffic, and Solar in the main text. The results for the remaining 15 datasets are available in our code repository at <https://github.com/decisionintelligence/DUET>.

**5.1.2 Baselines.** We choose the latest state-of-the-art models to serve as baselines, including CNN-based models (TimesNet [70]), MLP-based models (FITS [75], TimeMixer [68], and DLinear [86]), and Transformer-based models (PDF [12], iTransformer [89], Pathformer [7], PatchTST [51], Crossformer [87], and Non-stationary Transformer (Stationary) [46]).

**5.1.3 Implementation Details.** To keep consistent with previous works, we adopt Mean Squared Error (mse) and Mean Absolute Error (mae) as evaluation metrics. We consider four forecasting horizon  $F$ : 24, 36, 48, and 60 for FredMd, NASDAQ, NYSE, NN5, ILL, Covid-19, and Wike2000, and we use another four forecasting horizon, 96, 192, 336, and 720, for all other datasets which have longer lengths. Since the size of the look-back window can affect the performance of different models, we choose the look-back window size in 36 and 104 for FredMd, NASDAQ, NYSE, NN5, ILL, Covid-19, and Wike2000, and 96, 336, and 512 for all other datasets and report each method's best results for fair comparisons.

We utilize the TFB code repository for unified evaluation, with all baseline results also derived from TFB. Following the settings in TFB [55] and FoundTS [35], we do not apply the "Drop Last" trick to ensure a fair comparison. All experiments of DUET are conducted using PyTorch [54] in Python 3.8 and executed on an NVIDIA Tesla-A800 GPU. The training process is guided by the L1 loss function and employs the ADAM optimizer. The initial batch size is set to 64, with the flexibility to halve it (down to a minimum of 8) in case of an Out-Of-Memory (OOM) issue.

### 5.2 Main Results

Comprehensive forecasting results are listed in Table 3 with the best in bold and the second underlined. We have the following observations:

1) Compared with forecasters of different structures, DUET achieves an excellent predictive performance. From the perspective

**Table 1: Statistics of multivariate datasets.**

Dataset	Domain	Frequency	Lengths	Dim	Split	Description
METR-LA	Traffic	5 mins	34,272	207	7:1:2	Traffic speed dataset collected from loopdetectors in the LA County road network
PEMS-BAY	Traffic	5 mins	52,116	325	7:1:2	Traffic speed dataset collected from the CalTrans PeMS
PEMS04	Traffic	5 mins	16,992	307	6:2:2	Traffic flow time series collected from the CalTrans PeMS
PEMS08	Traffic	5 mins	17,856	170	6:2:2	Traffic flow time series collected from the CalTrans PeMS
Traffic	Traffic	1 hour	17,544	862	7:1:2	Road occupancy rates measured by 862 sensors on San Francisco Bay area freeways
ETTh1	Electricity	1 hour	14,400	7	6:2:2	Power transformer 1, comprising seven indicators such as oil temperature and useful load
ETTh2	Electricity	1 hour	14,400	7	6:2:2	Power transformer 2, comprising seven indicators such as oil temperature and useful load
ETTh1	Electricity	15 mins	57,600	7	6:2:2	Power transformer 1, comprising seven indicators such as oil temperature and useful load
ETTh2	Electricity	15 mins	57,600	7	6:2:2	Power transformer 2, comprising seven indicators such as oil temperature and useful load
Electricity	Electricity	1 hour	26,304	321	7:1:2	Electricity records the electricity consumption in kWh every 1 hour from 2012 to 2014
Solar	Energy	10 mins	52,560	137	6:2:2	Solar production records collected from 137 PV plants in Alabama
Wind	Energy	15 mins	48,673	7	7:1:2	Wind power records from 2020-2021 at 15-minute intervals
Weather	Environment	10 mins	52,696	21	7:1:2	Recorded every for the whole year 2020, which contains 21 meteorological indicators
AQShunyi	Environment	1 hour	35,064	11	6:2:2	Air quality datasets from a measurement station, over a period of 4 years
AQWan	Environment	1 hour	35,064	11	6:2:2	Air quality datasets from a measurement station, over a period of 4 years
ZafNoo	Nature	30 mins	19,225	11	7:1:2	From the Sapflux data project includes sap flow measurements and nvironmental variables
CzeLan	Nature	30 mins	19,934	11	7:1:2	From the Sapflux data project includes sap flow measurements and nvironmental variables
FRED-MD	Economic	1 month	728	107	7:1:2	Time series showing a set of macroeconomic indicators from the Federal Reserve Bank
Exchange	Economic	1 day	7,588	8	7:1:2	ExchangeRate collects the daily exchange rates of eight countries
NASDAQ	Stock	1 day	1,244	5	7:1:2	Records opening price, closing price, trading volume, lowest price, and highest price
NYSE	Stock	1 day	1,243	5	7:1:2	Records opening price, closing price, trading volume, lowest price, and highest price
NN5	Banking	1 day	791	111	7:1:2	NN5 is from banking, records the daily cash withdrawals from ATMs in UK
ILI	Health	1 week	966	7	7:1:2	Recorded indicators of patients data from Centers for Disease Control and Prevention
Covid-19	Health	1 day	1,392	948	7:1:2	Provide opportunities for researchers to investigate the dynamics of COVID-19
Wike2000	Web	1 day	792	2,000	7:1:2	Wike2000 is daily page views of 2000 Wikipedia pages

**Table 2: Ablation study. Results are averaged from all forecasting horizons.**

Model Metrics	DUET		w/o TCM		w/o CCM		Full Attention		Tem Info	
	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae
ETTh2	<b>0.334</b>	<b>0.383</b>	0.344	0.391	0.343	0.391	0.344	0.389	0.345	0.390
ETTh2	<b>0.248</b>	<b>0.307</b>	0.256	0.310	0.256	0.312	0.254	0.310	0.254	0.311
Weather	<b>0.218</b>	<b>0.252</b>	0.225	0.255	0.232	0.262	0.223	0.255	0.222	0.254
Traffic	<b>0.394</b>	<b>0.257</b>	0.398	0.263	0.439	0.277	0.396	0.260	0.401	0.263

of absolute performance, DUET demonstrates a significant improvement against the second-best baseline PDF, with an impressive 6.9% reduction in MSE and a 6.2% reduction in MAE.

2) Considering different channel strategies, DUET also shows the advantage of CSC strategy. Compared with CI models such as FITS and PatchTST, DUET outperforms them comprehensively, especially on large datasets with abundant channel correlations such as Solar and Traffic. Besides, as the previous state-of-the-art, iTransformer and Crossformer with the CD strategy fail in many cases of ETT (4 subsets). Due to the weak correlations among channels of ETT, they may be affected by noise channels (uncorrelated channels), leading to a decrease in performance. In contrast, DUET with CSC strategy implements a soft clustering among channels (where each channel only focuses on the channels related to it), can better address this situation.

3) DUET also demonstrates strong performance in tackling temporal heterogeneity caused by Temporal Distribution Shift. Compared to the state-of-the-art Non-stationary Transformer model for non-stationary time series modeling. DUET achieves a significant reduction of 32.3% in MSE and 21.5% in MAE. This result indicates that the proposed Temporal Clustering Module is more effective in modeling heterogeneous temporal patterns.

## 5.3 Model Analyses

**5.3.1 Ablation Studies.** To ascertain the impact of different modules within DUET, we perform ablation studies focusing on the following components.

- 1) *w/o TCM*: Remove the Temporal Cluster Module.
- 2) *w/o CCM*: Remove the Channel Cluster Module.
- 3) *Full Attention*: Replace the Mask Attention with Full Attention.
- 4) *Tem-Info*: Change the channel distance calculation from the frequency domain to the temporal domain.

Table 2 illustrates the unique impact of each module. We have the following observations: 1) When the temporal clustering module is removed, the performance decreases relatively slightly for datasets with fewer distribution changes such as Traffic. While evaluating datasets with substantial temporal distribution changes, such as ETTh2, the performance drops more significantly. This indicates the effectiveness of the temporal clustering module. 2) When the channel clustering module is removed and our model works in CI strategy, the performance on Traffic, which has strong inter-channel correlations, drops significantly. This highlights the necessity of soft clustering among channels. 3) When a full attention mechanism (where each variable calculates attention scores with all other variables) is used among channels instead of a masked matrix, our model works in CD strategy and the performance on the ETTh2 dataset, which has less noticeable inter-channel correlations, also drops significantly. This demonstrates the effectiveness of our proposed CSC strategy with the masked matrix applying to channels. 4) When the distances among channels are calculated in the temporal domain instead of the frequency domain, the algorithm's performance also decreases. This suggests that clustering channels in the frequency domain introduces more semantic information and helps learn a robust distance metric to evaluate the channel relations.

**Table 3: Multivariate forecasting results with forecasting horizons  $F \in \{12, 24, 36, 48\}$  for ILI and  $F \in \{96, 192, 336, 720\}$  for others.**

Models	DUET (ours)		PDF (2024)		iTransformer (2024)		Pathformer (2024)		FiTS (2024)		TimeMixer (2024)		PatchTST (2023)		Crossformer (2023)		TimesNet (2023)		DLinear (2023)		Stationary (2022)		
Metrics	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae	
ETTh1	96	<b>0.352</b>	<b>0.385</b>	<u>0.360</u>	<u>0.391</u>	0.386	0.405	0.372	0.392	0.376	0.396	0.372	0.401	0.377	0.397	0.411	0.435	0.389	0.412	0.379	0.403	0.591	0.524
	192	<u>0.398</u>	<b>0.409</b>	<b>0.392</b>	<u>0.414</u>	0.430	0.435	0.408	0.415	0.400	0.418	0.413	0.430	0.409	0.425	0.409	0.438	0.440	0.443	0.431	0.429	0.615	0.540
	336	<b>0.414</b>	<b>0.426</b>	<u>0.418</u>	0.435	0.450	0.452	0.438	<u>0.434</u>	0.419	0.435	0.438	0.450	0.431	0.444	0.433	0.457	0.523	0.487	0.440	0.440	0.632	0.551
	720	<b>0.430</b>	<b>0.457</b>	0.456	0.462	0.495	0.487	0.450	0.463	<u>0.435</u>	<u>0.458</u>	0.483	0.483	0.457	0.477	0.501	0.514	0.521	0.495	0.445	0.462	0.828	0.658
ETTh2	96	<b>0.270</b>	<b>0.336</b>	0.276	0.341	0.292	0.347	0.279	<u>0.336</u>	0.277	0.345	<u>0.270</u>	0.342	0.274	0.337	0.728	0.603	0.334	0.370	0.275	0.336	0.347	0.387
	192	<b>0.331</b>	<b>0.375</b>	0.339	0.382	0.348	0.384	0.345	0.380	<u>0.331</u>	0.379	0.349	0.387	0.348	0.384	0.723	0.607	0.404	0.413	0.337	<u>0.376</u>	0.379	0.418
	336	<u>0.354</u>	<u>0.397</u>	0.374	0.406	0.372	0.407	0.378	0.408	<b>0.350</b>	<b>0.396</b>	0.367	0.410	0.377	0.416	0.740	0.628	0.389	0.435	0.365	0.401	0.358	0.413
	720	<b>0.382</b>	<b>0.424</b>	0.398	0.433	0.424	0.444	0.437	0.455	<u>0.382</u>	<u>0.425</u>	0.401	0.436	0.406	0.441	1.386	0.882	0.434	0.448	0.429	0.461	0.422	0.457
ETTm1	96	<b>0.279</b>	<b>0.333</b>	<u>0.286</u>	0.340	0.287	0.344	0.290	<u>0.335</u>	0.303	0.345	0.293	0.345	0.289	0.343	0.314	0.367	0.340	0.378	0.287	0.340	0.415	0.410
	192	<b>0.320</b>	<b>0.358</b>	<u>0.321</u>	0.364	0.331	0.371	0.337	<u>0.363</u>	0.337	0.365	0.335	0.372	0.329	0.368	0.374	0.410	0.392	0.404	0.324	0.363	0.494	0.451
	336	<b>0.347</b>	<b>0.376</b>	<u>0.354</u>	0.383	0.358	0.384	0.374	0.384	0.368	0.384	0.368	0.386	0.362	0.390	0.413	0.432	0.423	0.426	0.358	<u>0.382</u>	0.577	0.490
	720	<b>0.405</b>	<b>0.409</b>	<u>0.408</u>	0.415	0.412	0.416	0.428	0.416	0.420	<u>0.413</u>	0.426	0.417	0.416	0.423	0.753	0.613	0.475	0.453	0.410	0.413	0.636	0.535
ETTm2	96	<b>0.161</b>	<b>0.248</b>	<u>0.163</u>	0.251	0.168	0.262	0.164	<u>0.250</u>	0.165	0.254	0.165	0.256	0.165	0.255	0.296	0.391	0.189	0.265	0.164	0.255	0.210	0.294
	192	<b>0.215</b>	<b>0.286</b>	<u>0.219</u>	0.290	0.224	0.295	0.219	<u>0.288</u>	0.225	0.291	0.225	0.298	0.221	0.293	0.369	0.416	0.254	0.310	0.223	0.301	0.338	0.373
	336	<u>0.268</u>	<u>0.323</u>	0.269	0.330	0.274	0.330	<b>0.267</b>	<b>0.319</b>	0.272	0.326	0.277	0.332	0.276	0.327	0.588	0.600	0.313	0.345	0.273	0.328	0.432	0.416
	720	<b>0.348</b>	<b>0.373</b>	<u>0.349</u>	0.382	0.367	0.385	0.361	<u>0.377</u>	0.359	0.381	0.360	0.387	0.362	0.381	0.750	0.612	0.413	0.402	0.366	0.384	0.554	0.476
Exchange	96	<u>0.080</u>	<b>0.198</b>	0.083	0.200	0.080	0.201	0.088	0.208	0.082	<u>0.199</u>	0.087	0.209	<b>0.079</b>	0.200	0.088	0.213	0.112	0.242	0.080	0.202	0.083	0.203
	192	0.162	<b>0.287</b>	0.172	0.294	<b>0.155</b>	<u>0.287</u>	0.183	0.304	0.173	0.295	0.178	0.300	0.159	0.289	<u>0.157</u>	0.288	0.209	0.334	0.161	0.291	0.159	0.293
	336	<b>0.293</b>	<b>0.392</b>	0.323	0.411	0.298	<u>0.399</u>	0.354	0.429	0.317	0.406	0.376	0.451	<u>0.297</u>	0.399	0.332	0.429	0.358	0.435	0.327	0.434	0.317	0.412
	720	<b>0.652</b>	<b>0.611</b>	0.820	0.682	0.749	<u>0.650</u>	0.909	0.716	0.825	0.684	0.897	0.711	<u>0.751</u>	0.650	0.980	0.762	0.944	0.736	0.758	0.654	<u>0.725</u>	0.656
Weather	96	<u>0.147</u>	<b>0.191</b>	0.147	0.196	0.157	0.207	0.148	<u>0.195</u>	0.172	0.225	0.147	0.198	0.149	0.196	<b>0.143</b>	0.210	0.168	0.214	0.154	0.210	0.188	0.242
	192	<b>0.189</b>	<b>0.231</b>	0.193	0.240	0.200	0.248	<u>0.191</u>	<u>0.235</u>	0.215	0.261	0.191	0.242	0.191	0.239	0.198	0.260	0.219	0.262	0.198	0.252	0.241	0.290
	336	<b>0.234</b>	<b>0.268</b>	0.245	0.280	0.252	0.287	0.243	<u>0.274</u>	0.261	0.295	0.244	0.280	<u>0.242</u>	0.279	0.258	0.314	0.278	0.302	0.248	0.292	0.341	0.341
	720	<b>0.303</b>	<b>0.318</b>	0.323	0.334	0.320	0.336	0.318	<u>0.326</u>	0.326	0.341	0.316	0.331	<u>0.312</u>	0.330	0.335	0.385	0.353	0.351	0.318	0.344	0.403	0.388
Electricity	96	<b>0.128</b>	<b>0.219</b>	<u>0.128</u>	<u>0.222</u>	0.134	0.230	0.135	0.222	0.139	0.237	0.153	0.256	0.143	0.247	0.135	0.237	0.169	0.271	0.135	0.233	0.171	0.274
	192	<b>0.146</b>	<b>0.236</b>	<u>0.147</u>	<u>0.242</u>	0.154	0.250	0.157	0.253	0.154	0.250	0.168	0.269	0.158	0.260	0.160	0.262	0.204	0.304	0.151	0.248	0.180	0.283
	336	<b>0.163</b>	<b>0.254</b>	<u>0.165</u>	<u>0.260</u>	0.169	0.265	0.170	0.267	0.170	0.268	0.189	0.291	0.168	0.267	0.182	0.282	0.180	0.280	0.169	0.269	0.204	0.305
	720	<u>0.195</u>	<b>0.282</b>	0.199	0.289	<b>0.194</b>	<u>0.288</u>	0.211	0.302	0.212	0.304	0.228	0.320	0.214	0.307	0.246	0.337	0.245	0.333	0.203	0.301	0.221	0.319
ILI	24	<b>1.584</b>	<b>0.775</b>	1.801	0.874	<u>1.783</u>	0.846	2.086	0.922	2.182	1.002	1.804	<u>0.820</u>	1.932	0.872	2.981	1.096	2.131	0.958	2.208	1.031	2.394	1.066
	36	<b>1.587</b>	<b>0.789</b>	<u>1.743</u>	0.867	1.746	<u>0.860</u>	1.912	0.882	2.241	1.029	1.891	0.926	1.869	0.866	3.549	1.196	2.612	0.974	2.032	0.981	2.227	1.031
	48	<b>1.597</b>	<b>0.795</b>	1.843	0.926	<u>1.716</u>	0.898	1.985	0.905	2.272	1.036	1.752	<u>0.866</u>	1.891	0.883	3.851	1.288	1.916	0.897	2.209	1.063	2.525	1.003
	60	<b>1.623</b>	<b>0.805</b>	1.845	0.925	2.183	0.963	1.999	0.929	2.642	1.142	<u>1.831</u>	0.930	1.914	<u>0.896</u>	4.692	1.450	1.995	0.905	2.292	1.086	2.410	1.010
Solar	96	<b>0.165</b>	<b>0.191</b>	0.181	0.247	0.174	0.229	0.218	0.235	0.208	0.255	0.180	0.233	<u>0.170</u>	0.234	0.183	<u>0.208</u>	0.198	0.270	0.199	0.251	0.381	0.398
	192	<u>0.199</u>	<b>0.210</b>	0.200	0.259	0.205	0.270	<b>0.196</b>	<u>0.220</u>	0.229	0.267	0.201	0.259	<u>0.204</u>	0.302	0.208	0.226	0.206	0.276	0.220	0.282	0.395	0.386
	336	<u>0.206</u>	<b>0.212</b>	0.208	0.269	0.216	0.282	<b>0.195</b>	<u>0.220</u>	0.241	0.273	0.214	0.272	0.212	0.293	0.212	0.239	0.208	0.284	0.234	0.271	0.410	0.394
	720	<b>0.204</b>	<b>0.218</b>	0.212	0.275	0.211	0.260	0.208	<u>0.237</u>	0.248	0.277	0.218	0.278	0.215	0.307	0.215	0.256	0.232	0.294	<u>0.205</u>	0.260	0.377	0.376
Traffic	96	<b>0.360</b>	<b>0.238</b>	0.368	0.252	<u>0.363</u>	0.265	0.384	<u>0.250</u>	0.400	0.280	0.369	0.256	0.370	0.262	0.526	0.288	0.595	0.312	0.395	0.275	0.604	0.330
	192	<u>0.384</u>	<b>0.250</b>	<b>0.382</b>	0.261	0.384	0.273	0.405	<u>0.257</u>	0.412	0.288	0.400	0.271	0.386	0.269	0.503	0.263	0.613	0.322	0.407	0.280	0.610	0.338
	336	<u>0.395</u>	<b>0.260</b>	<b>0.393</b>	0.268	0.396	0.277	0.424	<u>0.265</u>	0.426	0.301	0.407	0.272	0.396	0.275	0.505	0.276	0.626	0.332	0.417	0.286	0.626	0.341
	720	<b>0.435</b>	<b>0.278</b>	0.438	0.297	0.445	0.308	0.452	<u>0.283</u>	0.478	0.339	0.462	0.316	<u>0.435</u>	0.295	0.552	0.301	0.635	0.340	0.454	0.308	0.643	0.347
1 <sup>st</sup> Count	29	38	<u>3</u>	0	2	0	<u>3</u>	<u>1</u>	1	<u>1</u>	0	0	1	0	1	0	0	0	0	0	0	0	

**Table 4: The comparison among different distance metrics. Results are averaged from all forecasting horizons.**

Models	DUET		Euclid		Cosine		DTW		Random Mask	
Metrics	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae
ETTh2	<b>0.334</b>	<b>0.383</b>	0.343	0.390	0.343	0.391	0.344	0.391	0.341	0.389
ETTm2	<b>0.248</b>	<b>0.307</b>	0.255	0.313	0.256	0.312	0.256	0.314	0.257	0.315
Weather	<b>0.218</b>	<b>0.252</b>	0.224	0.257	0.225	0.257	0.228	0.262	0.227	0.259
Traffic	<b>0.394</b>	<b>0.257</b>	0.397	0.260	0.398	0.262	0.396	0.259	0.397	0.261

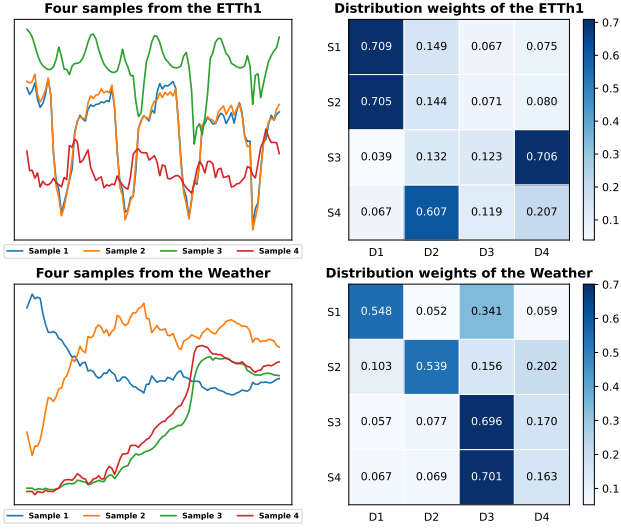
**5.3.2 Comparison Among Different Distance Metrics.** To obtain the mask matrix in the channel view, we adopt a learnable Mahalanobis distance metric to adaptively evaluate the correlations among channels in the frequency domain. We have demonstrated in 5.3.1 that

frequency domain actually introduces more useful information for channel clustering. Then, we demonstrate that the learnable Mahalanobis distance metric is better than other similarity measurement methods. We choose to replace the learnable metric with non-learnable similarity measurement methods: Euclidean distance, Cosine similarity, and DTW [49]. As shown in Table 4, using these methods results in a decrease in model performance, which fully demonstrates the effectiveness of the learnable metric. Furthermore, we also replace the learned mask matrix with a random mask matrix and prove it performs far worse than the learned one.



**Table 5: Parameter sensitivity study. The prediction accuracy varies with  $M$  which is the number of extractors. Results are averaged from all forecasting horizons.**

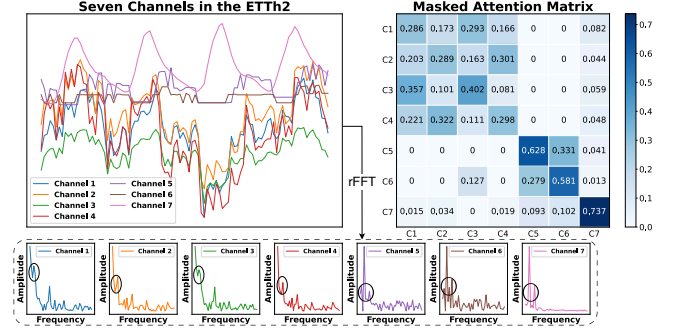
Extractors Metrics	M=1		M=2		M=3		M=4		M=5	
	mse	mae	mse	mae	mse	mae	mse	mae	mse	mae
ETTh1	0.402	0.421	0.408	0.426	0.403	0.424	<b>0.399</b>	<b>0.420</b>	0.411	0.430
ETTh2	0.346	0.389	0.343	0.388	0.342	0.387	<b>0.334</b>	<b>0.383</b>	0.341	0.387
ILI	1.666	0.798	<b>1.598</b>	<b>0.791</b>	1.701	0.816	1.708	0.818	1.679	0.810
Exchange	0.365	0.404	0.348	0.397	0.366	0.405	0.322	0.383	<b>0.297</b>	<b>0.372</b>



**Figure 6: The distribution weights of different time series samples from the ETTh1 and Weather. S1-S4 denote distinct samples, while D1-D4 represent different distributions.**

The performance of  $M=1$  is inferior to that of  $M$  not equal to 1. 2) Datasets from the same domain, such as the electricity domain datasets ETTh1 and ETTh2, their best values of  $M$  are the same, which are 4. 3) While for datasets from different domains, such as ILI (health) and Exchange (economic), their best values of  $M$  are 2 and 5. 4) With the most proper  $M$ , the performance leads other variants remarkably in most cases. This highlights the effectiveness of clustering in the temporal view, and implies datasets from the same domain often have similar temporal distributions, vice versa.

**5.3.4 Visualization of Distributions Weights:** The distribution router in TCM is designed to cluster samples by extracting the mean and variance of the implicit distributions. During clustering, we do not consider the temporal alignment of the samples; instead, we focus on the distributional characteristics of each sample to determine its distribution category. We show four time series samples with the same length of 96 from the ETTh1 and Weather datasets respectively, and described their unique distribution weights in Figure 6. We have the following observations: 1) Samples 1 and 2 in ETTh1 exhibit similar seasonal patterns, show similar distribution weights. However, for samples 1, 3, and 4, their distributions are vastly different. 2) Sample 3 and Sample 4 in Weather have similar trends, show



**Figure 7: The attention scores of different channels from the ETTh2. C1–C7 denote distinct channels.**

similar distribution weights. However, for samples 1, 2, and 3, their distributions are vastly different. These observations emphasize the adaptability of DUET and its ability to distinguish and cluster samples with different distributions.

**5.3.5 Visualization of Channel Weights:** We show the mutual attention weights of the seven channels in ETTh2 to exhibit the effectiveness of our Channel Clustering Module (CCM). For channels in the same time interval, the CCM measures their correlations for the downstream prediction task in the frequency space and generates corresponding mask matrices to obtain a clustering effect. Figure 7 shows an example of this flexible clustering paradigm by visualizing the masked attention weights in the Fusion Module. We can observe that channels with similar frequency component combinations may be clustered into a soft group while the cross-group correlations are also partially kept—(see 0.127 between C6 and C3) to maximize the neighbor information gain for the prediction task.

## 6 Conclusions

In this paper, we propose a general framework, DUET, which introduces a dual clustering on the temporal and channel dimensions to enhance multivariate time series forecasting. It integrates a Temporal Clustering Module (TCM) which clusters time series into fine-grained distributions. Various pattern extractors are then designed for different distribution clusters to capture their unique temporal patterns, modeling the heterogeneity of temporal patterns. Furthermore, we introduce the Channel Clustering Module (CCM) using a channel-soft-clustering strategy. This captures the relationships among channels in the frequency domain through metric learning and applies sparsification. Finally, the Fusion Module (FM), based on a masked attention mechanism, efficiently combines the temporal features extracted by the TCM with the channel mask matrix generated by the CCM. These innovative mechanisms collectively empower DUET to achieve outstanding prediction performance.

## References

- [1] Diyar Akay and Mehmet Atak. 2007. Grey prediction with rolling mechanism for electricity demand forecasting of Turkey. *energy* 32, 9 (2007), 1670–1675.
- [2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).
- [3] E Oran Brigham and RE Morrow. 1967. The fast Fourier transform. *IEEE spectrum* 4, 12 (1967), 63–70.
- [4] Peter J Brockwell and Richard A Davis. 1991. *Time series: theory and methods*. Springer science & business media.
- [5] David Campos, Tung Kieu, Chenjuan Guo, Feiteng Huang, Kai Zheng, Bin Yang, and Christian S. Jensen. 2022. Unsupervised Time Series Outlier Detection with Diversity-Driven Convolutional Ensembles. *Proc. VLDB Endow.* 15, 3 (2022), 611–623.
- [6] David Campos, Miao Zhang, Bin Yang, Tung Kieu, Chenjuan Guo, and Christian S. Jensen. 2023. LightTS: Lightweight Time Series Classification with Adaptive Ensemble Distillation. *Proc. ACM Manag. Data* 1, 2 (2023), 171:1–171:27.
- [7] Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. 2024. Pathformer: Multi-scale Transformers with Adaptive Pathways for Time Series Forecasting. In *ICLR*.
- [8] Zhichao Chen, Leilei Ding, Zhixuan Chu, Yucheng Qi, Jianmin Huang, and Hao Wang. 2023. Monotonic Neural Ordinary Differential Equation: Time-series Forecasting for Cumulative Data. In *CIKM*. 4523–4529.
- [9] Yunyao Cheng, Peng Chen, Chenjuan Guo, Kai Zhao, Qingsong Wen, Bin Yang, and Christian S. Jensen. 2023. Weakly Guided Adaptation for Robust Time Series Forecasting. *Proc. VLDB Endow.* 17, 4 (2023), 766–779.
- [10] Razvan-Gabriel Cirstea, Chenjuan Guo, Bin Yang, Tung Kieu, Xuanyi Dong, and Shirui Pan. 2022. Triformer: Triangular, Variable-Specific Attentions for Long Sequence Multivariate Time Series Forecasting. In *IJCAI*. 1994–2001.
- [11] Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. 2022. Towards spatio-temporal aware traffic time series forecasting. In *ICDE*. 2900–2913.
- [12] Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. 2024. Periodicity Decoupling Framework for Long-term Series Forecasting. In *ICLR*.
- [13] Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Xue Yuerong, Shu-Tao Xia, and Zexuan Zhu. 2024. DDN: Dual-domain Dynamic Normalization for Non-stationary Time Series Forecasting. In *NeurIPS*.
- [14] Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. 2023. Long-term Forecasting with TiDE: Time-series Dense Encoder. *Trans. Mach. Learn. Res.* 2023 (2023).
- [15] Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Jialin Pan, Tao Qin, Renjun Xu, and Chongjun Wang. 2021. AdaRNN: Adaptive Learning and Forecasting of Time Series. In *CIKM*. 402–411.
- [16] Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. 2023. Dish-TS: A General Paradigm for Alleviating Distribution Shift in Time Series Forecasting. In *AAAI*. 7522–7529.
- [17] Hongfan Gao, Wangmeng Shen, Xiangfei Qiu, Ronghui Xu, Jilin Hu, and Bin Yang. 2024. DiffImp: Efficient Diffusion Model for Probabilistic Time Series Imputation with Bidirectional Mamba Backbone. *arXiv preprint arXiv:2410.13338* (2024).
- [18] Jacob Goldberger, Sam T. Roweis, Geoffrey E. Hinton, and Ruslan Salakhutdinov. 2004. Neighbourhood Components Analysis. In *NeurIPS*. 513–520.
- [19] Chenjuan Guo, Bin Yang, Ove Andersen, Christian S Jensen, and Kristian Torp. 2015. Ecomark 2.0: empowering eco-routing with vehicular environmental models and actual vehicle fuel consumption data. *Geoinformatica* 19 (2015), 567–599.
- [20] Lu Han, Han-Jia Ye, and De-Chuan Zhan. 2024. The Capacity and Robustness Trade-Off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting. *IEEE Trans. Knowl. Data Eng.* 36, 11 (2024), 7129–7142.
- [21] Jiayi Hu, Yuehong Hu, Wei Chen, Ming Jin, Shirui Pan, Qingsong Wen, and Yuxuan Liang. 2024. Attractor memory for long-term time series forecasting: A chaos perspective. In *NeurIPS*.
- [22] Jiayi Hu, Disen Lan, Ziyu Zhou, Qingsong Wen, and Yuxuan Liang. 2024. Time-SSM: Simplifying and Unifying State Space Models for Time Series Forecasting. *arXiv preprint arXiv:2405.16312* (2024).
- [23] Jiayi Hu, Bowen Zhang, Qingsong Wen, Fugee Tsung, and Yuxuan Liang. 2024. Toward Physics-guided Time Series Embedding. *arXiv preprint arXiv:2410.06651* (2024).
- [24] Shiyuan Hu, Kai Zhao, Xiangfei Qiu, Yang Shu, Jilin Hu, Bin Yang, and Chenjuan Guo. 2024. MultiRC: Joint Learning for Time Series Anomaly Prediction and Detection with Multi-scale Reconstructive Contrast. *arXiv preprint arXiv:2410.15997* (2024).
- [25] Yifan Hu, Peiyuan Liu, Peng Zhu, Dawei Cheng, and Tao Dai. 2024. Adaptive Multi-Scale Decomposition Framework for Time Series Forecasting. *arXiv preprint arXiv:2406.03751* (2024).
- [26] Qihe Huang, Lei Shen, Ruixin Zhang, Jiahuan Cheng, Shouhong Ding, Zhengyang Zhou, and Yang Wang. 2024. HDMixer: Hierarchical Dependency with Extendable Patch for Multivariate Time Series Forecasting. In *AAAI*. 12608–12616.
- [27] Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. 2023. CrossGNN: Confronting Noisy Multivariate Time Series Via Cross Interaction Refinement. In *NeurIPS*.
- [28] Qihe Huang, Zhengyang Zhou, Kuo Yang, Gengyu Lin, Zhongchao Yi, and Yang Wang. 2024. LeRet: Language-Empowered Retentive Network for Time Series Forecasting. In *IJCAI*. 4165–4173.
- [29] Xuanwen Huang, Yang Yang, Yang Wang, Chunping Wang, Zhisheng Zhang, Jiarong Xu, Lei Chen, and Michalis Vazirgiannis. 2022. DGraph: A Large-Scale Financial Dataset for Graph Anomaly Detection. In *NeurIPS*.
- [30] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *ICLR*.
- [31] Duc Kieu, Tung Kieu, Peng Han, Bin Yang, Christian S. Jensen, and Bac Le. 2024. TEAM: Topological Evolution-aware Framework for Traffic Forecasting. *Proc. VLDB Endow.* 18 (2024).
- [32] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *ICLR*.
- [33] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *ICLR*.
- [34] Wendi Li, Xiao Yang, Weiqing Liu, Yingce Xia, and Jiang Bian. 2022. Ddg-da: Data distribution generation for predictable concept drift adaptation. In *AAAI*, Vol. 36. 4092–4100.
- [35] Zhe Li, Xiangfei Qiu, Peng Chen, Yihang Wang, Yanyin Cheng, Yang Shu, Jilin Hu, Chenjuan Guo, Aoying Zhou, Qingsong Wen, et al. 2024. FoundTS: Comprehensive and Unified Benchmarking of Foundation Models for Time Series Forecasting. *arXiv preprint arXiv:2410.11802* (2024).
- [36] Shengsheng Lin, Weiwei Lin, Keyi Wu, Songbo Wang, Minxian Xu, and James Z. Wang. 2024. Ccov: A compression algorithm for time-series data with continuous constant values in IoT-based monitoring systems. *Internet of Things* 25 (2024), 101049.
- [37] Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. 2024. SparseTSF: Modeling Long-term Time Series Forecasting with 1k Parameters. In *ICML*.
- [38] Shengsheng Lin, Weiwei Lin, HU Xinyi, Wentai Wu, Ruichao Mo, and Haocheng Zhong. 2024. CycleNet: Enhancing Time Series Forecasting through Modeling Periodic Patterns. In *NeurIPS*.
- [39] Shengsheng Lin, Weiwei Lin, Feiyu Zhao, and Haojun Chen. 2025. Benchmarking and revisiting time series forecasting methods in cloud workload prediction. *Cluster Computing* 28, 1 (2025), 71.
- [40] Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-tao Xia. 2024. TimeBridge: Non-Stationarity Matters for Long-term Time Series Forecasting. *arXiv preprint arXiv:2410.04442* (2024).
- [41] Peiyuan Liu, Beiliang Wu, Naiqi Li, Tao Dai, Fengmao Lei, Jigang Bao, Yong Jiang, and Shu-Tao Xia. 2024. WFTNet: Exploiting Global and Local Periodicity in Long-Term Time Series Forecasting. In *ICASSP*. 5960–5964.
- [42] Qinghua Liu, Paul Boniol, Themis Palpanas, and John Paparrizos. 2024. Time-Series Anomaly Detection: Overview and New Trends. *Proc. VLDB Endow.* 17, 12 (2024), 4229–4232.
- [43] Qinsuo Liu, Yanwen Fang, Pengtao Jiang, and Guodong Li. 2024. DGCformer: Deep Graph Clustering Transformer for Multivariate Time Series Forecasting. *arXiv preprint arXiv:2405.08440* (2024).
- [44] Qinghua Liu and John Paparrizos. 2024. The Elephant in the Room: Towards A Reliable Time-Series Anomaly Detection Benchmark. In *NeurIPS*.
- [45] Yong Liu, Tenge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *ICLR*.
- [46] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. In *NeurIPS*.
- [47] Wang Lu, Jindong Wang, Xinwei Sun, Yiqiang Chen, Xiangyang Ji, Qiang Yang, and Xing Xie. 2024. Diversify: A General Framework for Time Series Out-of-Distribution Detection and Generalization. *IEEE Trans. Pattern Anal. Mach. Intell.* 46, 6 (2024), 4534–4550.
- [48] Wang Lu, Jindong Wang, Xinwei Sun, Yiqiang Chen, and Xing Xie. 2023. Out-of-distribution Representation Learning for Time Series Classification. In *ICLR*.
- [49] Meinard Müller. 2007. Dynamic time warping. *Information retrieval for music and motion* (2007), 69–84.
- [50] Guy P Nason. 2006. Stationary and non-stationary time series. (2006).
- [51] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *ICLR*.
- [52] Zhicheng Pan, Yihang Wang, Yingying Zhang, Sean Bin Yang, Yunyao Cheng, Peng Chen, Chenjuan Guo, Qingsong Wen, Xiduo Tian, Yunliang Dou, et al. 2023. Magicscaler: Uncertainty-aware, predictive autoscaling. *Proc. VLDB Endow.* 16, 12 (2023), 3808–3821.
- [53] Nikolaos Passalis, Anastasios Tefas, Juho Kanninen, Moncef Gabbouj, and Alexandros Iosifidis. 2020. Deep Adaptive Input Normalization for Time Series Forecasting. *IEEE Trans. Neural Networks Learn. Syst.* 31, 9 (2020), 3760–3765.

- [54] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*. 8024–8035.
- [55] Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. 2024. TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods. *Proc. VLDB Endow.* 17, 9 (2024), 2363–2377.
- [56] Omer Berat Sezer, Mehmet Uğur Güdelek, and Ahmet Murat Özbayoglu. 2020. Financial time series forecasting with deep learning : A systematic literature review: 2005-2019. *Appl. Soft Comput.* 90 (2020), 106181.
- [57] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarczyk, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *ICLR*.
- [58] Yanru Sun, Zongxia Xie, Dongyue Chen, Emadeldeen Eldele, and Qinghua Hu. 2024. Hierarchical Classification Auxiliary Network for Time Series Forecasting. *arXiv preprint arXiv:2405.18975* (2024).
- [59] Yanru Sun, Zongxia Xie, Haocheng Wang, Xin Huang, and Qinghua Hu. 2022. Solar wind speed prediction via graph attention network. *Space Weather* 20, 7 (2022).
- [60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NeurIPS*. 5998–6008.
- [61] Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, and Jianxin Liao. 2024. Rethinking the Power of Timestamps for Robust Time Series Forecasting: A Global-Local Fusion Perspective. In *NeurIPS*.
- [62] Chengsen Wang, Zirui Zhuang, Qi Qi, Jingyu Wang, Xingyu Wang, Haifeng Sun, and Jianxin Liao. 2023. Drift doesn't Matter: Dynamic Decomposition with Diffusion Reconstruction for Unstable Multivariate Time Series Anomaly Detection. In *NeurIPS*.
- [63] Hao Wang. 2024. Improving Neural Network Generalization on Data-Limited Regression with Doubly-Robust Boosting. In *AAAI*. 20821–20829.
- [64] Hao Wang, Zhichao Chen, Zhaoran Liu, Haozhe Li, Degui Yang, Xinggao Liu, and Haoxuan Li. 2024. Entire Space Counterfactual Learning for Reliable Content Recommendations. *IEEE Trans. Autom. Sci. Eng.* (2024), 1–12.
- [65] Hao Wang, Zhichao Chen, Zhaoran Liu, Licheng Pan, Hu Xu, Yilin Liao, Haozhe Li, and Xinggao Liu. 2024. SPOT-I: Similarity Preserved Optimal Transport for Industrial IoT Data Imputation. *IEEE Transactions on Industrial Informatics* (2024).
- [66] Hao Wang, Xinggao Liu, Zhaoran Liu, Haozhe Li, Yilin Liao, Yuxin Huang, and Zhichao Chen. 2024. LSPT-D: Local Similarity Preserved Transport for Direct Industrial Data Imputation. *IEEE Trans. Inf. Forensics Security* (2024).
- [67] Hao Wang, Zhiyu Wang, Yunlong Niu, Zhaoran Liu, Haozhe Li, Yilin Liao, Yuxin Huang, and Xinggao Liu. 2024. An Accurate and Interpretable Framework for Trustworthy Process Monitoring. *IEEE Trans. Artif. Intell.* 5, 5 (2024), 2241–2252.
- [68] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. 2024. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *ICLR*.
- [69] Yihang Wang, Yuying Qiu, Peng Chen, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. 2024. ROSE: Register Assisted General Time Series Forecasting with Decomposed Frequency Learning. *arXiv preprint arXiv:2405.17478* (2024).
- [70] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *ICLR*.
- [71] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *NeurIPS*. 22419–22430.
- [72] Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. 2024. CATCH: Channel-Aware multivariate Time Series Anomaly Detection via Frequency Patching. *arXiv preprint arXiv:2410.12261* (2024).
- [73] Xinle Wu, Xingjian Wu, Bin Yang, Lekui Zhou, Chenjuan Guo, Xiangfei Qiu, Jilin Hu, Zhenli Sheng, and Christian S Jensen. 2024. AutoCTS++: zero-shot joint neural architecture and hyperparameter search for correlated time series forecasting. *The VLDB Journal* 33, 5 (2024), 1743–1770.
- [74] Xinle Wu, Xingjian Wu, Dalin Zhang, Miao Zhang, Chenjuan Guo, Bin Yang, and Christian S Jensen. 2024. Fully Automated Correlated Time Series Forecasting in Minutes. *arXiv preprint arXiv:2411.05833* (2024).
- [75] Zhijian Xu, Ailing Zeng, and Qiang Xu. 2024. FITS: Modeling Time Series with 10k Parameters. In *ICLR*.
- [76] Sean Bin Yang, Chenjuan Guo, Jilin Hu, Jian Tang, and Bin Yang. 2021. Unsupervised Path Representation Learning with Curriculum Negative Sampling. In *IJCAI*. 3286–3292.
- [77] Sean Bin Yang, Chenjuan Guo, and Bin Yang. 2022. Context-Aware Path Ranking in Road Networks. *IEEE Trans. Knowl. Data Eng.* 34, 7 (2022), 3153–3168.
- [78] Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. 2023. DCdetector: Dual Attention Contrastive Representation Learning for Time Series Anomaly Detection. In *SIGKDD*. 3033–3045.
- [79] Yuanyuan Yao, Shenjia Dai, Yilin Li, Lu Chen, Dimeng Li, Yunjun Gao, and Tianyi Li. 2024. A Demonstration of TENDS: Time Series Management System based on Model Selection. *Proc. VLDB Endow.* 17, 12 (2024), 4357–4360.
- [80] Yuanyuan Yao, Hailiang Jie, Lu Chen, Tianyi Li, Yunjun Gao, and Shiting Wen. 2024. TSec: An Efficient and Effective Framework for Time Series Classification. In *ICDE*. 1394–1406.
- [81] Yuanyuan Yao, Dimeng Li, Hailiang Jie, Lu Chen, Tianyi Li, Jie Chen, Jiaqi Wang, Feifei Li, and Yunjun Gao. 2023. SimpleTS: An Efficient and Universal Model Selection Framework for Time Series Forecasting. *Proc. VLDB Endow.* 16, 12 (2023), 3741–3753.
- [82] Peter C Young. 1999. Nonstationary time series analysis and forecasting. *Progress in Environmental Science* 1 (1999), 3–48.
- [83] Chengqing Yu, Fei Wang, Zezhi Shao, Tangwen Qian, Zhao Zhang, Wei Wei, and Yongjun Xu. 2024. Ginar: An end-to-end multivariate time series forecasting model suitable for variable missing. In *SIGKDD*. 3989–4000.
- [84] Chengqing Yu, Fei Wang, Zezhi Shao, Tao Sun, Lin Wu, and Yongjun Xu. 2023. DSformer: A Double Sampling Transformer for Multivariate Time Series Long-term Prediction. In *CIKM*. 3062–3072.
- [85] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide Residual Networks. In *BMVC*.
- [86] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting?. In *AAAI*, Vol. 37. 11121–11128.
- [87] Yunhao Zhang and Junchi Yan. 2022. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *ICLR*.
- [88] Wendong Zheng, Putian Zhao, Gang Chen, Huihui Zhou, and Yonghong Tian. 2023. A Hybrid Spiking Neurons Embedded LSTM Network for Multivariate Time Series Learning Under Concept-Drift Environment. *IEEE Trans. Knowl. Data Eng.* 35, 7 (2023), 6561–6574.
- [89] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, Vol. 35. 11106–11115.
- [90] Xuanbing Zhu, Dunbin Shen, Hongyu Wang, and Yingguang Hao. 2024. FCNet: Fully Complex Network for Time Series Forecasting. *IEEE Internet of Things Journal* (2024).

## A THEORETICAL ANALYSIS

### A.1 Computational Complexity Analysis

We compare the theoretical complexity across different Transformer-based models in Table 6 because the main Computational complexity of DUET comes from the Transformer-based Fusion Module. Subsequent works mainly utilize attention mechanism to capture temporal dependencies and channel correlations. Benefiting from our dual clustering design, DUET efficiently extracts temporal patterns so that we only need to capture the correlations among channels with attention mechanism, which can be paralleled on the look-back window size. This ensures low computational costs even when the look-back window size is extremely large.

### A.2 Noisy Gating

We combine the reparameterization trick and original Noisy Gating technique in following formulas of our paper:

$$Z_n = \text{Encoder}_\mu(X_{n,:}) + \epsilon \odot \text{Softplus}(\text{Encoder}_\sigma(X_{n,:}))$$

$$H(X_{n,:}) = W^H \cdot Z_n$$

We provide evidence that the combination is computational equivalent to the original Noisy Gating. First, we simplify the symbolic representation for legibility. For our formulas, we simplify them as follows:

$$H(X_{n,:}) = W^H \cdot (\mu + \epsilon \odot \sigma),$$

where  $\mu, \sigma, \epsilon \in \mathbb{R}^M$  and  $\epsilon_i \sim \mathcal{N}(0, 1)$ . Then we formalize the original Noisy Gating technique in our context:

$$H'(X_{n,:}) = W^{H_1} \cdot (\mu + \epsilon \odot \sigma) + \epsilon' \odot \text{Softplus}(W^{H_2} \cdot (\mu + \epsilon \odot \sigma)),$$

where  $\epsilon_i, \epsilon'_i \sim \mathcal{N}(0, 1)$  are all **i.i.d** and independent of each other. Next, we show the equivalency between the two formulas. We first simplify the part  $\epsilon' \odot \text{Softplus}(W^{H_2} \cdot (\mu + \epsilon \odot \sigma))$  as  $\epsilon' \odot V'$ , where  $V' \in \mathbb{R}^M$  is a learnable vector. Then we consider the every vector component in this formula:

$$H'(X_{n,:}) = W^{H_1} \cdot (\mu + \epsilon \odot \sigma) + \epsilon' \odot V'$$

$$H'(X_{n,:})_i = (W^{H_1} \cdot \mu)_i + (W^{H_1} \cdot (\epsilon \odot \sigma))_i + V'_i \cdot \epsilon'_i$$

$$H'(X_{n,:})_i = (W^{H_1} \cdot \mu)_i + \sum_{j=1}^M W_{ij}^{H_1} \cdot \sigma_j \cdot \epsilon_j + \epsilon'_i \cdot V'_i$$

the  $H'(X_{n,:})_i$  obeys normal distribution:

$$H'(X_{n,:})_i \sim \mathcal{N}((W^{H_1} \cdot \mu)_i, \sum_{j=1}^M (W_{ij}^{H_1} \cdot \sigma_j)^2 + (V'_i)^2)$$

for learnable components  $V'$  and  $W^{H_1}$ , they can be adaptively tuned and approximately equivalent to:

$$H'(X_{n,:})_i \sim \mathcal{N}((\hat{W}^{H_1} \cdot \mu)_i, \sum_{j=1}^M (\hat{W}_{ij}^{H_1} \cdot \sigma_j)^2)$$

$$e.g. (\hat{W}_{ij}^{H_1})^2 = (W_{ij}^{H_1})^2 + \frac{(V'_i)^2}{(M \cdot \sigma_j)^2}$$

**Table 6: Theoretical computational complexity per layer in Transformer-based models.**  $H$  denotes the look-back window size,  $N$  denotes the number of channels,  $p$  denotes the patch size in patch-based methods, and  $\max(p_i)$  denotes the maximum decoupled periodic length of PDF.

Method	Time complexity
Autoformer	$O(H \log H)$
PatchTST	$O((\frac{H}{p})^2)$
Crossformer	$O((\frac{H}{p})^2 + N^2)$
Pathformer	$O(H \log H)$
iTransformer	$O(N^2)$
PDF	$O((\frac{\max(p_i)}{p})^2)$
DUET (Ours)	$O(N^2)$

while  $H(X_{n,:})_i$  also obeys the same form of distribution:

$$H(X_{n,:})_i \sim \mathcal{N}((W^H \cdot \mu)_i, \sum_{j=1}^M (W_{ij}^H \cdot \sigma_j)^2)$$

Q.E.D.

### A.3 Fourier Transform in $L^2$ Space

We introduce the Fourier transform from the perspective of functional analysis: Taking  $L^2(-\pi, \pi)$  as an example, where the inner product is defined as  $(f, g) = \int_E f g d\mu$  for  $f, g \in L^2(-\pi, \pi)$ , induced  $L^2$ -norm  $(\int_E |f|^2 d\mu)^{\frac{1}{2}}$  (conforming to the inner product definition norm), this space constitutes a Hilbert space. Since functions eligible for Fourier expansion must satisfy conditions of Riemann integrability, considering only such functions reduces the inner product to the Riemann integral  $(f, g) = \int_{-\pi}^{\pi} f(x)g(x) dx$ . Direct computation confirms that the trigonometric function series:

$$[e_n] = \left[ \frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}} \cos(nx), \frac{1}{\sqrt{\pi}} \sin(nx), \dots, \frac{1}{\sqrt{\pi}} \cos(nx), \dots \right]$$

forms a standard orthonormal system, and  $\text{span}[e_n]$  is dense in  $L^2(-\pi, \pi)$ , hence serving as a standard orthonormal basis. It is evident that Fourier coefficients can be expressed as:

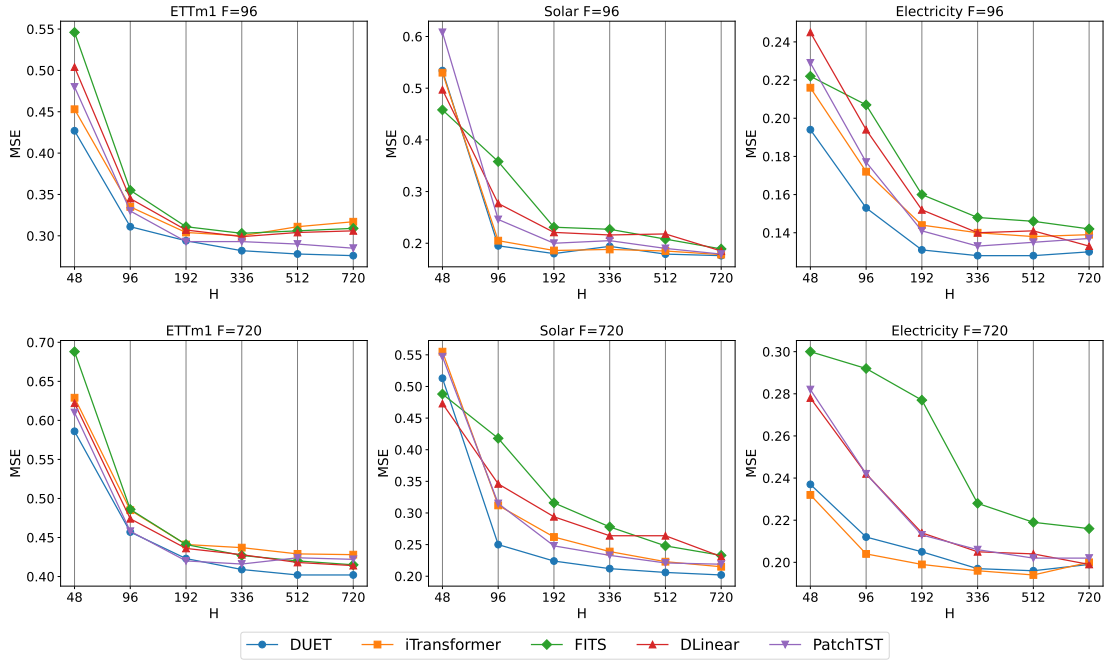
$$\begin{cases} \frac{a_0}{2} = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) dx = \frac{1}{\sqrt{2\pi}} (f, \frac{1}{\sqrt{2\pi}}) \\ a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} \cos(nx) f(x) dx = \frac{1}{\sqrt{\pi}} (f, \frac{1}{\sqrt{\pi}} \cos(nx)) \\ b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} \sin(nx) f(x) dx = \frac{1}{\sqrt{\pi}} (f, \frac{1}{\sqrt{\pi}} \sin(nx)) \end{cases}$$

$$(n = 1, 2, 3, \dots)$$

thus yielding:

$$f(x) = (f, \frac{1}{\sqrt{2\pi}}) \frac{1}{\sqrt{2\pi}} + \sum_{n=1}^{\infty} \left[ (f, \frac{1}{\sqrt{\pi}} \cos(nx)) \frac{1}{\sqrt{\pi}} \cos(nx) + (f, \frac{1}{\sqrt{\pi}} \sin(nx)) \frac{1}{\sqrt{\pi}} \sin(nx) \right]$$

From this point, Fourier Transform actually plays as an orthogonal transformation in  $L^2$  space, thus  $a_n$  and  $b_n$  are coordinates under the trigonometric function basis and can be seen as a point in the space.



**Figure 8: Forecasting performance (MSE) with varying look-back windows on 3 datasets: ETTm1, Solar, and Electricity. The look-back windows are selected to be  $H = 48, 96, 192, 336, 512$ , and  $720$ , and the forecasting horizons are  $F = 96$ , and  $720$ .**

## B HYPERPARAMETER SENSITIVITY

### B.1 Varying Look-back Window

In time series forecasting tasks, the size of the look-back window determines how much historical information the model receives. We select models with better predictive performance from the main experiments as baselines. We configure different look-back window to evaluate the effectiveness of DUET and visualize the prediction results for look-back window  $H$  of  $48, 96, 192, 336, 512, 720$ , and the forecasting horizons are  $F = 96, 720$ . From Figure 8, DUET consistently outperforms the baselines on the ETTm1, Solar, and Electricity. As the look-back window increases, the prediction metrics of DUET continue to decrease, indicating that it is capable of modeling longer sequences.

### B.2 The Advantages of Dual Clustering

As aforementioned, recent approaches assume the time series is sampled from a single distribution and use a single backbone to model the heterogeneous temporal patterns. While DUET clusters time series into different distributional categories and use parameter independent backbones to model heterogeneous temporal patterns. Figure 9 and Figure 11 show that other baselines perform insufficient sensibility to distributional changing while DUET fits well to such cases. For datasets like Electricity with strong seasonality, CI strategy fails to take advantage of correlations among channels while CD strategy suffers from the adverse effects of delays among channels. Both of them capture a fuzzy seasonality while DUET

with the CSC strategy precisely leverages pros and cons from other channels and shows a more accurate performance.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009



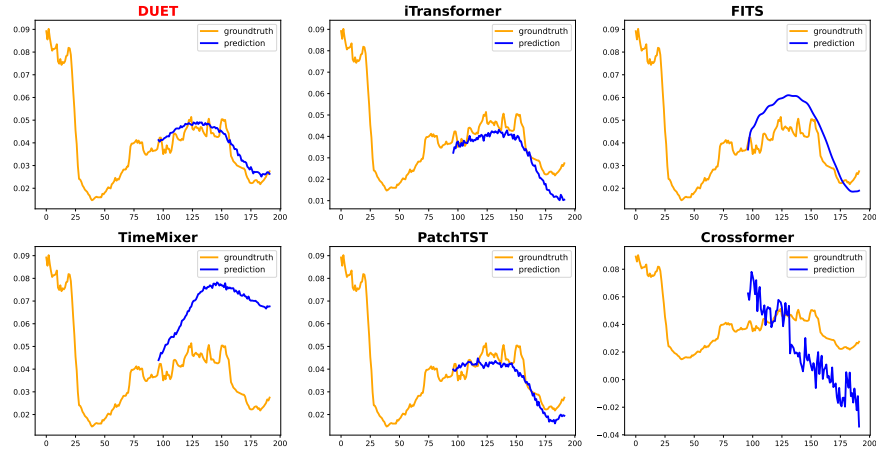


Figure 9: Visualization of input-96-predict-96 results on the Weather dataset.

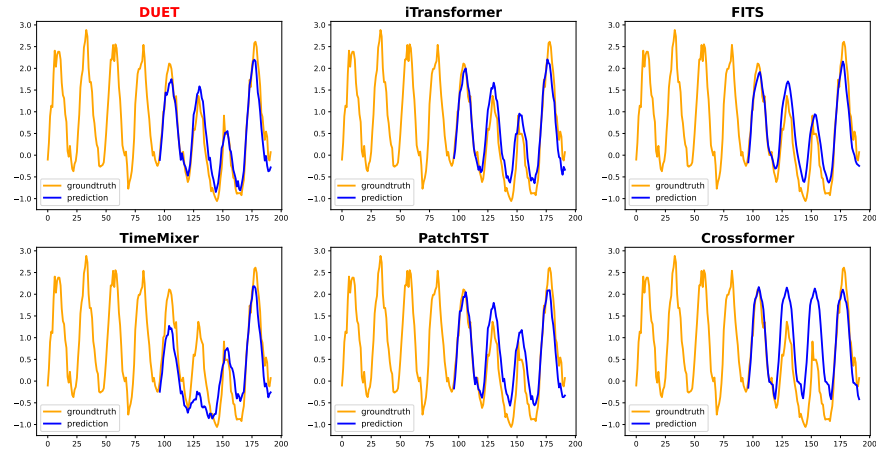


Figure 10: Visualization of input-96-predict-96 results on the Electricity dataset.

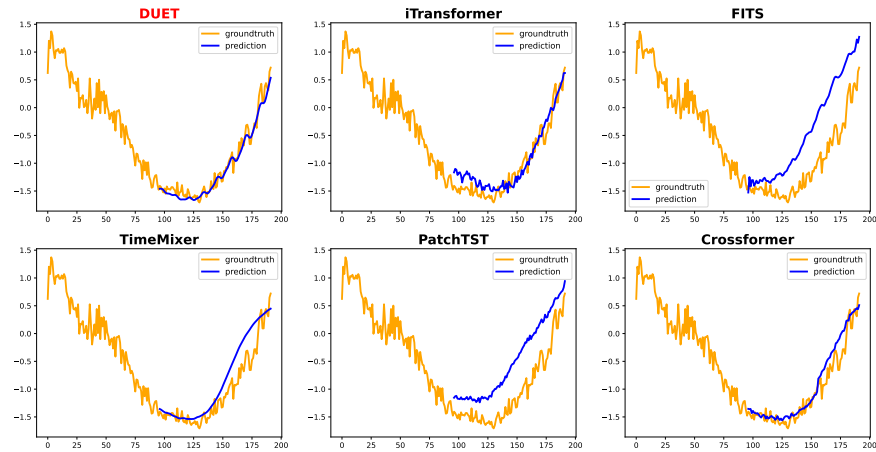


Figure 11: Visualization of input-96-predict-96 results on the PEMS04 dataset.