FreDC: A Frequency-Enhanced Dynamic Clustering Network for Multivariate Time Series Forecasting

Mengna Hu[1], Jinghua Wang[1,2], Zenglin Xu[3,*]

[1]College of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

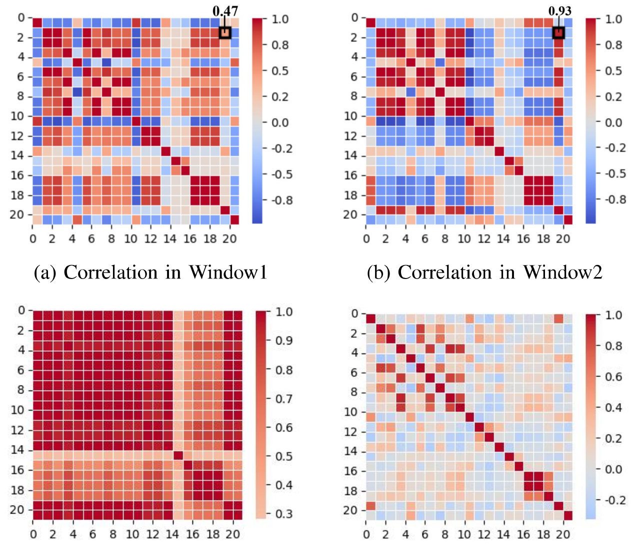
[2]State Key Laboratory of Smart Farm Technologies and Systems

[3]Artificial Intelligence Innovation and Incubation (AI³) Institute, Fudan University, Shanghai, China

23s151133@stu.hit.edu.cn, wangjh2012@foxmail.com, Zenglinxu@fudan.edu.cn

***Abstract*—Multivariate time series forecasting plays a critical role in various real-world applications. However, most existing approaches for modeling inter-variable dependencies overlook the temporal dynamics and structural sparsity. Meanwhile, existing frequency-domain methods primarily focus on amplitude information and neglect phase information, which is crucial for preserving temporal alignment. To address these limitations, we propose FreDC, a lightweight yet effective model that integrates Frequency-domain enhancement with variable Dynamic Clustering. Our FreDC exploits complementary amplitude and phase information in the frequency domain to construct informative prior representations. To capture time-varying and sparse inter-variable correlations, we introduce a Mixture-of-Experts (MoE)–based variable dynamic clustering module. The MoE consists of multiple private experts for modeling cluster-specific variable dynamics and a shared expert for capturing global dependency patterns. Furthermore, we design an inter-variable association consistency loss to explicitly align the dependency structure of predictions with ground-truth correlations, providing effective structural supervision. Extensive experiments on multiple real-world datasets demonstrate that FreDC consistently achieves state-of-the-art performance, while maintaining competitive computational efficiency.**

***Index Terms*—Multivariate time series forecasting, Fourier transform, Mixture-of-Experts, Lightweight network**



(a) Correlation in Window1 (b) Correlation in Window2

(c) Amplitude Corr. in Window1 (d) Phase Corr. in Window1

Fig. 1. Comparison of inter-variable correlation patterns across different lookback windows on the Weather dataset.

I. Introduction

Multivariate time series forecasting (MTSF) plays a critical role in many real-world applications, including economics [1], weather [2] and energy [3]. A core challenge in MTSF lies in modeling the complex and dynamically coupled dependencies among variables, which are essential for accurate long-term forecasting.

Most existing approaches [4], [5] model inter-variable dependencies via attention, linear projections, or convolution operations. However, these methods often overlook two key properties of real-world multivariate time series: (1) temporal dynamics, where inter-variable correlations vary across time windows; and (2) structural sparsity, where only a small subset of variables exhibits strong correlations at any given time window. As illustrated in Fig.1 (a) and (b), the correlation between variables (e.g., T and $Tlog$) can shift significantly across different lookback windows, while strong dependencies remain sparse.

Recent studies [6] have demonstrated that incorporating frequency-domain information can enhance forecasting performance. Some methods [7] adopt filtering-based strategies to suppress noise. Alternatively, other approaches [6] leverage dominant periodic patterns derived from amplitude spectra. However, most existing frequency-domain methods primarily rely on amplitude information and overlook phase information, which is critical for preserving temporal alignment. As shown in Fig. 1(a), (c) and (d), the phase-based correlation matrix exhibits structural patterns more consistent with time-domain correlations, whereas amplitude-only representations introduce many spurious and irrelevant associations.

To address these limitations, we propose **FreDC**, a lightweight network that integrates **Fre**quency-domain enhancement with variable **D**ynamic **C**lustering. FreDC exploits complementary amplitude and phase information to construct informative frequency-domain priors. To model time-varying and sparse inter-variable dependencies, we introduce a MoE [8]–based dynamic clustering module. The MoE con-

*Corresponding author

sists of multiple private experts for modeling cluster-specific variable dynamics and a shared expert for capturing global dependency patterns. The fusion of private and shared representations enables effective modeling of evolving variable relationships. Extensive experiments demonstrate that FreDC achieves superior forecasting performance. The main contributions are as follows:

- We propose FreDC, a lightweight multivariate time series forecasting model that jointly integrates frequency-domain enhancement and dynamic variable clustering.
- We design a frequency-domain enhancement module that adaptively fuses complementary amplitude and phase information, and a MoE-based variable dynamic clustering module with top-k soft routing to capture time-varying and sparse inter-variable correlations. An auxiliary inter-variable association consistency loss further provides structural supervision.
- Extensive experiments demonstrate that FreDC consistently outperforms state-of-the-art forecasting methods.

II. Related Work

A. Time-Domain Based Time Series Prediction Models

Transformer-based models [9], [10] have become a dominant paradigm in time series forecasting due to their ability to capture long-range temporal dependencies via self-attention. Informer [11] reduces the quadratic complexity of attention to $O(L \log L)$ using ProbSparse attention and adopts direct multi-step forecasting. PatchTST [12] segments time series into patches to better model local temporal patterns, while Pyraformer [13] employs a pyramidal architecture to capture multi-resolution dependencies with linear complexity.

MLP-based models offer a lightweight yet effective alternative. DLinear [14] shows that simple linear architectures can outperform Transformers and RLinear [15] further demonstrates the competitiveness of carefully designed linear models. TimeMixer [16] combines multi-scale downsampling with trend–seasonality decomposition, and CycleNet [17] explicitly models periodic patterns via learnable cycles.

B. Frequency-Domain Based Time Series Prediction Models

Frequency-domain methods exploit the underlying spectral characteristics. Fedformer [18] integrates frequency-enhanced blocks and attention mechanisms to model global dependencies. TimesNet [6] identifies dominant periods and reshapes sequences to enable joint intra- and inter-period modeling. FITS [19] performs interpolation in the frequency domain using complex-valued linear layers.

C. Variable-Dependency-Based Multivariate Time Series Prediction Models

Modeling inter-variable dependencies is crucial for multivariate time series forecasting. MTGNN [20] constructs a data-driven graph to capture latent variable relations. iTransformer [4] treats variables as tokens and applies self-attention to directly model inter-variable correlations. SOFTS [21] introduces an aggregate–redistribute mechanism to facilitate global patterns across all variables. TQNet [22] integrates attention mechanism with periodically shifted learnable query vectors to model global dependencies.

III. Methodology

A. Preliminaries

a) Problem Definition: The multivariate time series forecasting (MTSF) task aims to model and predict time series data composed of multiple variables. Let $X = [x_0, \ldots, x_{L-1}] \in \mathbb{R}^{C \times L}$ be the input multivariate time series, where C denotes the number of variables and L represents the lookback window length. Specifically, for each variable $i \in \{1, \ldots, C\}$, we represent its corresponding sequence as $X^i = [x_0^i, \ldots, x_{L-1}^i] \in \mathbb{R}^L$. Given the historical observations, the objective of MTSF is to predict the future sequence over the next H time steps, *i.e.*, $Y = [x_L, \ldots, x_{L+H-1}] \in \mathbb{R}^{C \times H}$.

b) Discrete Fourier transform: The Discrete Fourier Transform (DFT) maps a time-domain signal into the frequency domain and reveals the underlying spectral components. Given the input multivariate time series $X \in \mathbb{R}^{C \times L}$, the DFT transforms X into a set of complex-valued frequency components:

$$F_k = \sum_{t=0}^{L-1} x_t \cdot e^{-j2\pi kt/L}, \quad k = 0, 1, \ldots, L-1 \tag{1}$$

where F_k denotes the frequency-domain representation at frequency index k, x_t is the time-domain value at time step t, and $j^2 = -1$. We can express each frequency-domain component in polar form as:

$$F_k = A_k \cdot e^{j\phi_k}, \tag{2}$$

where A_k denotes the magnitude, indicating the strength of the k-th frequency component, and ϕ_k represents the phase, encoding temporal alignment information. By applying the DFT and decomposing the frequency representations into amplitude and phase components, we obtain complementary frequency-domain priors.

B. Overall Architecture

As shown in Fig.2, the proposed model FreDC comprises the following five main components: 1) **Reversible Instance Normalization** [23] (RevIN) to alleviate the non-stationarity by applying *Instance Normalization* and *Instance De-Normalization*; 2) **Frequency-Domain Enhancement** to construct informative frequency-domain priors by exploiting complementary amplitude and phase information; 3) **Temporal Embedding** to model temporal dependencies along the time dimension; 4) **Variable Dynamic Clustering** to capture time-varying inter-variable correlations via top-k soft routing; and 5) **Linear Predictor** to generate the final forecasting results. Detailed descriptions of each module are provided in the following subsections.

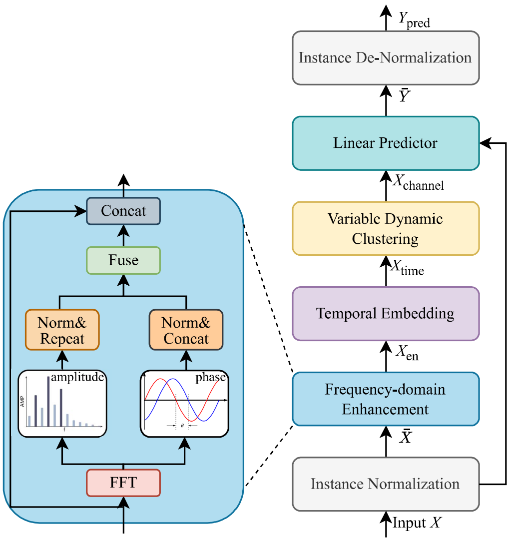


Fig. 2. Overall structure of the proposed FreDC.

C. Reversible Instance Normalization

RevIN is a simple yet effective technique for alleviating the non-stationarity. RevIN consists of two stages: instance normalization before model processing and instance denormalization after prediction. In the instance normalization, RevIN normalizes each lookback window to stabilize the input distribution. Formally, RevIN computes the normalized time series $\bar{X}$ as:

$$\bar{X} = \frac{X - \mu(X)}{\sigma(X) + \epsilon}, \tag{3}$$

where $\mu(X)$ and $\sigma(X)$ denote the mean and standard deviation of X, and ϵ is a small constant added for numerical stability.

D. Frequency-Domain Enhancement

As shown in Fig. 2, we first transform the normalized time-domain sequence into the frequency domain via the Fourier transform. The representation is decomposed into amplitude and phase components, which encode complementary spectral information. After normalization and alignment, we adaptively fuse the amplitude and phase representations to obtain a unified frequency-domain embedding. This embedding is then concatenated with the normalized time-domain sequence to form the frequency-enhanced representation.

a) Amplitude Representation: We can obtain the amplitude A_k by applying the modulus operation to the frequency-domain components F_k. In practice, we adopt the real-valued Fast Fourier Transform (rFFT) to accelerate the transform. Since rFFT retains only half of the frequency spectrum, we apply energy compensation and normalizations as:

$$amp = \frac{2A_k}{L}. \tag{4}$$

We discard $amp[0]$ because it represents the mean value of the signal and does not convey periodic information. We then repeat the remaining amplitude components along the time dimension to align them with the original sequence length, yielding the amplitude representation:

$$X_{\text{amp}} = \text{Repeat}(amp[1:], 2), \tag{5}$$

where $\text{Repeat}(\cdot)$ denotes the repetition operation.

b) Phase Representation: We compute the phase angles ϕ_k using the function $\text{atan2}(\cdot)$. Similarly, we discard the direct current component $\phi_k[0]$. Directly modeling raw phase values may cause numerical instability, as the phase angles lie in $(-\pi, \pi]$ and exhibit inherent periodicity. To obtain a smooth and bounded representation, we apply sine and cosine transformations to project the phase information. We then concatenate the transformed components to construct the phase representation:

$$X_{\text{pha}} = \text{Concat}\left(\sin(\phi_k[1:]), \cos(\phi_k[1:])\right), \tag{6}$$

where $\text{Concat}(\cdot)$ denotes the concatenation operation.

c) Adaptive Frequency-Domain Fusion: The relative importance of amplitude and phase information may vary across input windows and datasets. We adaptively fuse these two representations to construct the frequency-domain representation:

$$X_f = \alpha \cdot X_{\text{amp}} + (1 - \alpha) \cdot X_{\text{pha}}, \tag{7}$$

where $\alpha \in (0, 1)$ is a data-driven and dynamically determined coefficient that adjusts the relative contribution of amplitude and phase representations. We compute α based on how well the amplitude and phase representations preserve the inter-variable dependency structure of the normalized time-domain representation. Concretely, we construct Pearson correlation coefficient matrices for the normalized time-domain, amplitude and phase representation, yielding R_{time}, R_{amp} and $R_{\text{pha}} \in \mathbb{R}^{C \times C}$, respectively. We then quantify the structural discrepancy using the Frobenius norm:

$$d_{\text{amp}} = ||R_{\text{amp}} - R_{\text{time}}||_F. \tag{8}$$

$$d_{\text{pha}} = ||R_{\text{pha}} - R_{\text{time}}||_F. \tag{9}$$

A smaller distance indicates a higher degree of structural similarity. We further map these discrepancies to contribution weights through monotonically decreasing functions:

$$w_{\text{amp}} = \frac{1}{1 + d_{\text{amp}}}, \quad w_{\text{pha}} = \frac{1}{1 + d_{\text{pha}}}. \tag{10}$$

After normalization, we compute the fusion coefficient α as:

$$\alpha = \frac{w_{\text{amp}}}{w_{\text{amp}} + w_{\text{pha}}}, \tag{11}$$

where α depends on the input-specific amplitude and phase representation, enabling the model to dynamically emphasize the most informative frequency-domain cues. Finally, we concatenate the fused frequency-domain representation with the normalized time-domain representation to obtain the frequency-enhanced representation:

$$X_{\text{en}} = \text{Concat}(\bar{X}, X_f). \tag{12}$$

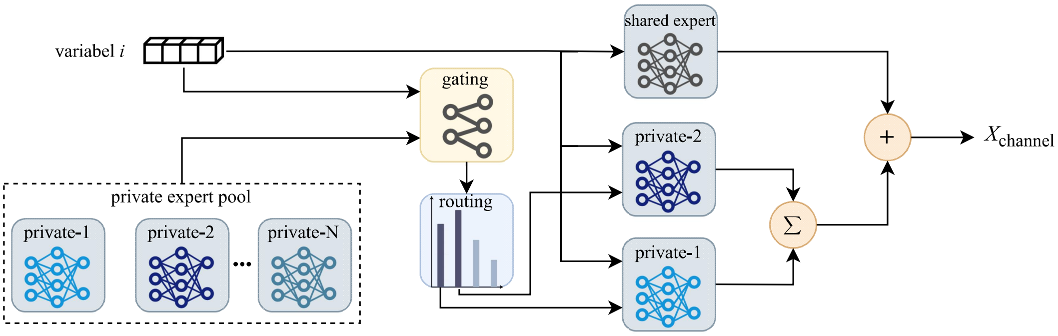


Fig. 3. Variable Dynamic Clustering. The gating network computes matching scores for each variable and selects the top-k relevant private experts via soft routing. The variable is processed by the selected private experts to learn cluster-specific representations, while a shared expert simultaneously captures global dependency patterns to learn shared representations. The private and shared representations are then fused to produce the final channel-wise representation.

E. Temporal Embedding

Recent studies [14], [15] have demonstrated that pure-MLP models can achieve performance comparable to more complex architectures, while reducing computational cost. Motivated by these observations, we adopt a lightweight two-layer MLP to capture temporal dependencies along the time dimension:

$$X_{\text{time}} = \text{MLP}(X_{\text{en}}), \tag{13}$$

where $\text{MLP}(\mathbb{R}^{C\times 2L} \mapsto \mathbb{R}^{C\times d_{model}} \mapsto \mathbb{R}^{C\times L})$ fuses the frequency-enhanced representation and captures temporal dependencies in a parameter-efficient manner.

F. Variable Dynamic Clustering

To capture time-varying inter-variable correlations while maintaining stable and efficient training, we adopt a Mixture-of-Experts (MoE) architecture with top-k soft routing for variable dynamic clustering. As shown in Fig. 3, the gating network computes expert relevance scores for each variable and dynamically combines private and shared representations to produce channel-wise enhanced features.

a) Variable Clustering via Top-k Expert Routing: For each variable $i \in \{1, \dots, C\}$, the gating network computes matching scores over all N private experts:

$$g^i = W_g X^i_{\text{time}} + B_g, \tag{14}$$

where X^i_{time} denotes the temporal representation of variable i and W_g, B_g are learnable parameters. To reduce routing noise and computational overhead, we retain only the top-k private experts with the highest scores:

$$\mathcal{E}^i_k = \text{TopK}(g^i, k), \tag{15}$$

where $\mathcal{E}^i_k \subset \{1, ..., N\}$ denotes the selected private expert indices. We then construct a binary routing mask $m^i \in \{0, 1\}^N$ to suppress non-selected experts:

$$m^i_e = \begin{cases} 1, & e \in \mathcal{E}^i_k \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

where $e \in \{1, \dots, N\}$ is the index of private experts. We apply a masked softmax to obtain the normalized routing weights:

$$w^i = \text{Softmax}(g^i \odot m^i). \tag{17}$$

where only the selected experts receive non-zero weights. This top-k soft routing mechanism enables each variable to be adaptively assigned to a small subset of private experts based on its temporal characteristics.

b) Private and Shared Representation Learning: We construct N private experts with identical architectures but independent parameters. To capture cluster-level information, we compute an expert context by aggregating weighted variable representations:

$$C_e = \sum_{i=1}^{C} w^i_e \cdot X^i_{\text{time}}, \tag{18}$$

where w^i_e denotes the routing weight of variable i to expert e. Each expert applies a two-stage nonlinear transformation with feature fusion to capture similar patterns among variables:

$$h^i_e = \text{MLP}^1_e(X^i_{\text{time}}), \tag{19}$$

$$X^i_e = \text{MLP}^2_e\left(\text{Concat}(X^i_{\text{time}}, h^i_e)\right), \tag{20}$$

where $\text{MLP}^1_e(\mathbb{R}^L \mapsto \mathbb{R}^{d_{model}})$ projects variable embeddings into an expert-specific feature space. $\text{MLP}^2_e(\mathbb{R}^{L+d_{model}} \mapsto \mathbb{R}^L)$ fuses cluster-specific information and projects the representation back to the original temporal dimension. We then aggregate expert outputs using routing weights and inject the cluster context to obtain private representation of variable i:

$$X^i_{\text{private}} = \sum_{e=1}^{N} (w^i_e \cdot X^i_e + C_e). \tag{21}$$

To capture global correlations and shared patterns across all variables, we further introduce a shared expert following the SOFTS [21] architecture:

$$\text{core} = \text{Stoch\_Pool}\left(\text{MLP}^1_s(X_{\text{time}})\right), \tag{22}$$

$$X_{\text{shared}} = \text{MLP}^2_s(\text{Concat}(X_{\text{time}}, \text{core})), \tag{23}$$

where $\text{MLP}^1_s(\mathbb{R}^{C\times L} \mapsto \mathbb{R}^{C\times d_{core}})$ extracts shared latent features into a low dimension space. Stoch_Pool [24] combines the advantages of mean and max pooling to aggregate core representations. And $\text{MLP}^2_s(\mathbb{R}^{C\times(L+d_{core})} \mapsto \mathbb{R}^{C\times L})$ fuses the shared core features with the original representations.

c) Private and Shared Representation Fusion: Finally, we combine cluster-specific representations from private experts with the global representations from the shared expert to form enhanced variable representations:

$$X_{\text{channel}} = X_{\text{private}} + X_{\text{shared}}. \tag{24}$$

This fusion allows the model to jointly capture both variable-specific dynamics and global dependency patterns.

G. Linear predictor

After capturing temporal dependencies and dynamic inter-variable correlations, we employ a linear prediction head to generate the final forecasting results. To improve training stability and facilitate gradient propagation, we introduce a shortcut connection that concatenates the normalized original input $\bar{X}$ with the learned variable representations X_{channel}. The prediction is computed as:

$$\bar{Y} = \text{Linear}\left(\text{Concat}(X_{\text{channel}}, \bar{X})\right). \tag{25}$$

Finally, we apply the instance de-normalization applied to restore the predictions to the original sequence scale:

$$Y = \bar{Y} \times (\sigma(X) + \epsilon) + \mu(X). \tag{26}$$

This design allows the predictor to jointly leverage the learned representations and the original input statistics, leading to more stable and accurate forecasts.

H. Inter-Variable Association Consistency Loss

To further strengthen the modeling of dynamic inter-variable correlations, we introduce an auxiliary inter-variable association consistency loss, which aligns the dependency structure of the predictions with that of the ground truth.

We quantify pairwise inter-variable dependencies using Pearson correlation coefficients. Given the predicted sequence and the corresponding ground-truth sequence, we compute their Pearson correlation matrices P_{pred} and P_{true}, respectively. To emphasize dominant correlation patterns while suppressing weak and noisy dependencies, we apply a softmax normalization to each correlation matrix:

$$R_{\text{true}} = \text{Softmax}(P_{\text{true}}), \quad R_{\text{pred}} = \text{Softmax}(P_{\text{pred}}). \tag{27}$$

Based on the normalized correlation matrices, we define the inter-variable association consistency loss as:

$$\mathcal{L}_{\text{corr}} = \text{MSE}(R_{\text{pred}}, R_{\text{true}}), \tag{28}$$

where $\text{MSE}(\cdot)$ denotes the mean squared error. This auxiliary objective guides the model to preserve the underlying inter-variable dependency structure during forecasting.

IV. Experiments

A. Experimental Settings

a) Datasets: We evaluate FreDC on datasets from various domain, including ETTh1, ETTh2, ETTm1, ETTm2 [11], Electricity, Traffic, Solar-Energy and Weather datasets [25].

b) Baselines and setup: We compare FreDC with representative state-of-the-art models, including: CycleNet [17], FEDformer [18], TimesNet [6], iTransformer [4], SOFTS [21] and TQNet [22]. We fix the lookback window length (L) to 96, and the prediction horizon (H) to 96, 192, 336, 720 for all datasets. Performance is evaluated using two widely adopted metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE). Partial baseline results are collected from CycleNet [17]. The experiments are conducted on an NVIDIA A100-SXM4-40GB GPU.

B. Main Results

Table I lists the multivariate long-term forecasting results of different models. Lower MSE and MAE values indicate better performance. Overall, FreDC consistently achieves top-tier performance, ranking within the top 2 in 72 out of 80 evaluations. Notably, FreDC exhibits particularly superior performance on high-dimensional multivariate datasets, such as Weather, Solar-Energy and Electricity, highlighting its effectiveness in modeling complex inter-variable dependencies. Moreover, compared with the recent state-of-the-art multivariate forecasting model TQNet, FreDC achieves an average MSE improvement of 1.9% and an average MAE improvement of 2%. These results validate the effectiveness of FreDC and demonstrate the benefits of jointly incorporating frequency-domain enhancement and variable dynamic clustering.

C. Ablation Study

To systematically validate the effectiveness of each components in FreDC, we conduct ablation experiments on multiple datasets. **W/O-F** removes the frequency-domain enhancement module. **W/O-C** removes the variable dynamic clustering module. **W/O-L** removes the inter-variable association consistency loss. As reported in Table II, both W/O-F and W/O-C consistently underperform across all settings. Moreover, W/O-L fails to achieve optimal performance in most cases.

D. Effect of Hyper-parameters

In FreDC, the top-k parameter in the soft routing mechanism controls the number of private experts each variable can interact with and thus plays a crucial role in balancing modeling flexibility and training stability. In practice, we set k to a small constant relative to N (*e.g.,* $k \in \{1, 2, 3\}$). As shown in Fig. 4, on the ETTh2 dataset with seven variables, FreDC achieves the best performance when $k = 3$ with the number of private experts fixed at $N = 7$.

E. Efficiency Analysis

Training efficiency and memory consumption are also important for evaluating models. We compare FreDC with baselines on the Weather dataset with a lookback window $L = 96$, prediction horizon $H = 96$, and batch size of 16. As shown in Fig. 5, FreDC achieves superior forecasting performance while maintaining lower memory usage. The training time of FreDC is mainly constrained by the variable dynamic clustering module, where the top-k routing and softmax normalization introduce additional computational.

TABLE I
FULL RESULTS OF DIFFERENT MODELS WITH THE LOOK-BACK LENGTH $L = 96$. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND THE SECOND BEST ARE UNDERLINED.

Dataset	Model	FEDformer		TimesNet		iTransformer		CycleNet		SOFTS		TQNet		FreDC(ours)	
	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.376	0.419	0.384	0.402	0.386	0.405	0.378	0.391	0.381	0.399	**0.371**	0.393	0.376	**0.388**
	192	**0.420**	0.448	0.436	0.429	0.441	0.436	0.426	0.419	0.435	0.431	0.428	0.426	0.430	**0.417**
	336	**0.459**	0.465	0.491	0.469	0.487	0.458	0.464	0.439	0.480	0.452	0.476	0.446	0.469	**0.437**
	720	0.506	0.507	0.521	0.500	0.503	0.491	**0.461**	**0.460**	0.499	0.433	0.487	0.470	0.471	0.463
	Avg	0.440	0.460	0.458	0.450	0.454	0.448	**0.432**	0.427	0.449	0.442	0.441	0.434	0.437	**0.426**
ETTh2	96	0.358	0.397	0.340	0.374	0.297	0.349	0.285	0.335	0.297	0.347	0.295	0.343	**0.278**	**0.332**
	192	0.429	0.439	0.402	0.414	0.380	0.400	0.373	0.391	0.373	0.394	0.367	0.393	**0.351**	**0.379**
	336	0.496	0.487	0.452	0.452	0.428	0.432	0.421	0.433	0.410	0.426	0.417	0.427	**0.396**	**0.415**
	720	0.463	0.474	0.462	0.468	0.427	0.445	0.453	0.458	**0.411**	**0.433**	0.433	0.446	0.414	0.435
	Avg	0.437	0.449	0.414	0.427	0.383	0.407	0.383	0.404	0.373	0.400	0.378	0.402	**0.360**	**0.390**
ETTm1	96	0.379	0.419	0.338	0.375	0.334	0.368	0.325	0.363	0.325	0.361	0.311	0.353	**0.310**	**0.348**
	192	0.426	0.441	0.374	0.387	0.377	0.391	0.366	0.382	0.375	0.389	**0.356**	0.378	**0.356**	**0.374**
	336	0.445	0.459	0.410	0.411	0.426	0.420	0.396	0.401	0.405	0.412	0.390	0.401	**0.386**	**0.396**
	720	0.543	0.490	0.478	0.450	0.491	0.459	0.457	**0.433**	0.466	0.447	0.452	0.440	**0.445**	**0.432**
	Avg	0.448	0.452	0.400	0.406	0.407	0.410	0.386	0.395	0.393	0.403	0.377	0.393	**0.374**	**0.388**
ETTm2	96	0.203	0.287	0.187	0.267	0.180	0.264	**0.166**	**0.248**	0.180	0.261	0.173	0.256	0.168	**0.248**
	192	0.269	0.328	0.249	0.309	0.250	0.309	**0.233**	**0.291**	0.246	0.306	0.238	0.298	**0.233**	**0.291**
	336	0.325	0.366	0.321	0.351	0.311	0.348	0.293	0.330	0.319	0.352	0.301	0.340	**0.292**	**0.329**
	720	0.421	0.415	0.408	0.403	0.412	0.407	0.395	0.389	0.405	0.401	0.397	0.396	**0.390**	**0.388**
	Avg	0.305	0.349	0.291	0.333	0.288	0.332	0.272	0.315	0.287	0.330	0.277	0.323	**0.271**	**0.314**
Electricity	96	0.193	0.308	0.168	0.272	0.148	0.240	0.141	0.234	0.143	0.233	**0.134**	**0.229**	0.137	0.230
	192	0.201	0.315	0.184	0.289	0.162	0.253	0.155	0.247	0.158	0.248	**0.154**	0.247	0.155	**0.246**
	336	0.214	0.329	0.198	0.300	0.178	0.269	0.172	**0.264**	0.178	0.269	**0.169**	**0.264**	0.175	0.265
	720	0.246	0.355	0.220	0.320	0.225	0.317	0.210	0.296	0.218	0.305	**0.201**	**0.294**	0.209	0.295
	Avg	0.214	0.327	0.193	0.295	0.178	0.270	0.170	0.260	0.174	0.264	**0.164**	**0.259**	0.169	**0.259**
Traffic	96	0.587	0.366	0.593	0.321	0.395	0.268	0.480	0.314	**0.376**	0.251	0.413	0.261	0.402	**0.248**
	192	0.604	0.373	0.617	0.336	0.417	0.276	0.482	0.313	**0.398**	0.261	0.432	0.271	0.427	**0.260**
	336	0.621	0.383	0.629	0.336	0.433	0.283	0.476	0.303	**0.415**	0.269	0.450	0.277	0.441	**0.267**
	720	0.626	0.382	0.640	0.350	0.467	0.302	0.503	0.320	**0.447**	**0.287**	0.486	0.295	0.474	**0.287**
	Avg	0.610	0.376	0.620	0.336	0.428	0.282	0.485	0.313	**0.409**	0.267	0.445	0.276	0.436	**0.266**
Solar-Energy	96	0.242	0.342	0.250	0.292	0.203	0.237	0.209	0.260	0.200	0.230	0.173	0.233	**0.171**	**0.228**
	192	0.285	0.380	0.296	0.318	0.233	0.261	0.231	0.269	0.229	**0.253**	0.199	0.257	**0.182**	**0.250**
	336	0.282	0.376	0.319	0.330	0.248	0.273	0.246	0.275	0.243	0.269	0.211	0.263	**0.199**	**0.259**
	720	0.357	0.427	0.338	0.337	0.249	0.275	0.255	0.274	0.245	0.272	0.209	0.270	**0.200**	**0.261**
	Avg	0.292	0.381	0.301	0.319	0.233	0.262	0.235	0.270	0.229	0.256	0.198	0.256	**0.188**	**0.250**
Weather	96	0.217	0.296	0.172	0.220	0.174	0.214	0.170	0.216	0.166	0.208	0.157	0.200	**0.150**	**0.194**
	192	0.276	0.336	0.219	0.261	0.221	0.254	0.222	0.259	0.217	0.253	0.206	0.245	**0.200**	**0.241**
	336	0.339	0.380	0.280	0.306	0.278	0.296	0.275	0.296	0.282	0.300	0.262	0.287	**0.257**	**0.284**
	720	0.403	0.428	0.365	0.359	0.358	0.349	0.349	0.345	0.356	0.351	0.344	0.342	**0.337**	**0.339**
	Avg	0.309	0.360	0.259	0.287	0.258	0.278	0.254	0.279	0.255	0.278	0.242	0.269	**0.236**	**0.265**

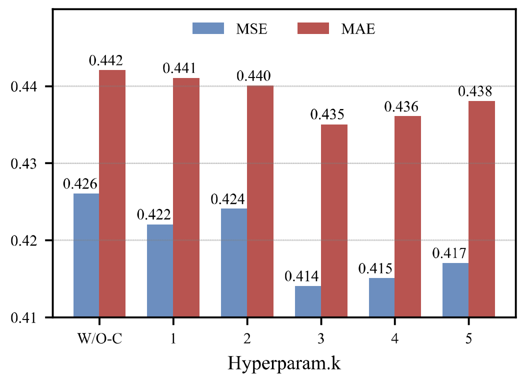


Fig. 4. Performance of FreDC different values of k.

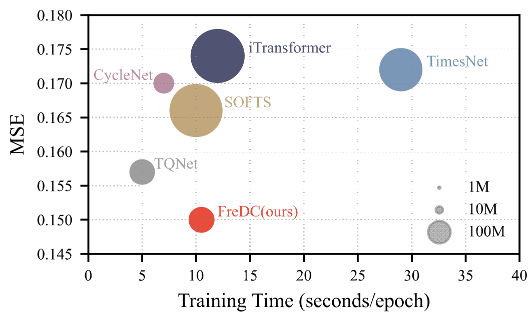


Fig. 5. Computational efficiency of FreDC. Comparison of MSE, parameter size, and training time on the Weather dataset with prediction horizon of 96.

V. Conclusion

In this paper, we propose FreDC, a lightweight yet effective multivariate time series forecasting model that integrates

TABLE II
ABLATION OF DIFFERENT COMPONENTS IN OUR METHOD.

Model	Metric	ETTh2				Weather				Solar-Energy			
		96	192	336	720	96	192	336	720	96	192	336	720
FreDC	MSE	**0.278**	**0.351**	**0.396**	**0.414**	**0.150**	**0.200**	**0.257**	**0.337**	**0.171**	**0.182**	**0.199**	**0.200**
	MAE	**0.332**	**0.379**	**0.415**	**0.435**	0.194	**0.241**	**0.284**	**0.339**	**0.228**	**0.250**	**0.259**	**0.261**
W/O-F	MSE	0.287	0.360	0.410	0.428	0.163	0.208	0.264	0.343	0.213	0.241	0.247	0.243
	MAE	0.336	0.385	0.422	0.443	0.206	0.249	0.289	0.342	0.266	0.287	0.290	0.284
W/O-C	MSE	0.289	0.361	0.411	0.426	0.163	0.209	0.265	0.344	0.218	0.238	0.243	0.239
	MAE	0.340	0.384	0.423	0.442	0.206	0.249	0.291	0.343	0.273	0.287	0.289	0.284
W/O-L	MSE	0.279	0.359	0.397	0.416	**0.150**	0.201	0.258	0.338	0.172	0.183	0.212	0.201
	MAE	**0.332**	0.386	0.416	0.438	**0.193**	**0.241**	0.287	0.343	**0.228**	**0.250**	0.265	0.262

frequency-domain enhancement with variable dynamic clustering. FreDC explicitly exploits complementary amplitude and phase information to construct informative prior representations. To capture time-varying and sparse inter-variable dependencies, we introduce a MoE–based dynamic clustering mechanism composed of multiple private experts for learning cluster-specific variable dynamics and a shared expert for capturing global dependency. By jointly learning cluster-specific private representations and global shared patterns, FreDC effectively captures inter-variable dependencies. Furthermore, we design an inter-variable association consistency loss to provide structural supervision. Extensive experiments on multiple real-world datasets demonstrate that FreDC consistently achieves state-of-the-art forecasting performance, particularly on high-dimensional multivariate datasets.

Acknowledgment

This work was supported in part by NSFC under Grants (No.62572149), Shenzhen Science and Technology Program (Project No.GXWD 20231130125451001), and Guangdong Major Project of Basic and Applied Basic Research (Grant No. 2023B0303000010).

References

[1] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," *Pattern Recognit.*, vol. 121, p. 108218, 2022.

[2] P. Hewage, *et al.*, "Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station," *Soft Comput.*, vol. 24, no. 21, pp. 16453–16482, 2020.

[3] Y. Hu, X. Cheng, S. Wang, J. Chen, T. Zhao, and E. Dai, "Times series forecasting for urban building energy consumption based on graph convolutional network," *Appl. Energy*, vol. 307, p. 118231, 2022.

[4] Y. Liu, *et al.*, "iTransformer: Inverted transformers are effective for time series forecasting," in *Proceedings of the International Conference on Learning Representations*, 2024.

[5] Z. Li, X. Li, Z. Rao, L. Pan, and Z. Xu, "MTS-Mixers: Multivariate time series forecasting via factorized temporal and channel mixing," in *International Joint Conference on Neural Networks*, pp. 1–8, Jun. 2025.

[6] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis," in *The Eleventh International Conference on Learning Representations*, 2023.

[7] K. Yi, et al., "Filternet: Harnessing frequency filters for time series forecasting," *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 55115–55140, 2024.

[8] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, Mar. 1991.

[9] A. Vaswani, *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[10] Y. Li, S. Qi, Z. Li, Z. Rao, L. Pan, and Z. Xu, "SMARTformer: Semi-Autoregressive Transformer with Efficient Integrated Window Attention for Long Time Series Forecasting," in *International Joint Conference on Artificial Intelligence*, pp. 2169–2177, Aug. 2023.

[11] H. Zhou, *et al.*, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, pp. 11106–11115, May 2021.

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

[13] S. Liu, *et al.*, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *Proceedings of the International Conference on Learning Representations*, 2021.

[14] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, pp. 11121–11128, Jun. 2023.

[15] Z. Li, S. Qi, Y. Li, and Z. Xu, "Revisiting long-term time series forecasting: An investigation on linear mapping," *Academia AI and Applications*, vol. 2, no. 2, 2026.

[16] S. Wang, *et al.*, "TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting," in *The Twelfth International Conference on Learning Representations*, 2024.

[17] S. Lin, W. Lin, X. Hu, W. Wu, R. Mo, and H. Zhong, "Cyclenet: Enhancing time series forecasting through modeling periodic patterns," *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 106315–106345, 2024.

[18] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International Conference on Machine Learning*, pp. 27268–27286, Jun. 2022.

[19] Z. Xu, A. Zeng, and Q. Xu, "FITS: Modeling Time Series with $10k$ Parameters," in *The Twelfth International Conference on Learning Representations*, 2024.

[20] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 753–763, Aug. 2020.

[21] L. Han, X. Chen, H. Ye, and D. Zhan, "Softs: Efficient multivariate time series forecasting with series-core fusion," *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 64145–64175, 2024.

[22] S. Lin, H. Chen, H. Wu, C. Qiu, W. Lin, "Temporal Query Network for Efficient Multivariate Time Series Forecasting," in *Proceedings of the International Conference on Machine Learning*, 2025.

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

[24] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," in *International Conference on Learning Representations*, 2013.

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