Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

FSHLNet: Frequency-Separated Dual-Hypergraph Learning for Long-Term Time Series Forecasting

Jingru Geng
Zhengzhou University
Yuxin Xie
Zhengzhou University
Bingqian Du
Zhengzhou University
Yanna Dong
Zhengzhou University
Hengheng Yang
Zhengzhou University
Mingliang Xu
iexieyuxin@163.com

Zhengzhou University

Research Article

Keywords: Long-term time series forecasting, Hypergraph learning, Frequency separation, Dual-hypergraph architecture, Cross-hypergraph attention

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11438610/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

FSHLNet: Frequency-Separated Dual-Hypergraph Learning for Long-Term Time Series Forecasting

Jingru Geng[iD][1,2,3], Yuxin Xie[iD][1], Bingqian Du[iD][1], Yanna Dong[iD][1], Hengheng Yang[iD][1], Mingliang Xu[iD][1,2,3*]

[1]School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou, 450001, China.
[2]Engineering Research Center of Intelligent Swarm Systems, Ministry of Education, Zhengzhou, 450001, China.
[3]National Supercomputing Center in Zhengzhou, Zhengzhou, 450001, China.

*Corresponding author(s). E-mail(s): iexumingliang@zzu.edu.cn;
Contributing authors: gengjingru@zzu.edu.cn; ieyxxie@gs.zzu.edu.cn; dubingqian@gs.zzu.edu.cn; dongyn@gs.zzu.edu.cn; henghengyang01@gs.zzu.edu.cn;

Abstract

Long-term time series forecasting is fundamental for supporting decisions in a wide range of fields. However, existing methods often underutilize frequency-domain information, exhibiting two key limitations: inadequate modeling of high-order dependencies among frequency components, and loss of structurally informative low-energy components due to coarse frequency-selection strategies. To overcome the above challenges, we introduce the Frequency-Separated Hypergraph Learning Network (FSHLNet). At its core, FSHLNet employs a frequency-domain dual-hypergraph architecture that separates the spectrum into principal and secondary components, constructs dedicated hypergraphs for each, and enables cross-group interaction via cross-hypergraph hyperedge attention. In parallel, a time-domain branch extracts multi-scale local patterns through temporal convolution. The outputs of the dual-hypergraph architecture, the temporal features, and the normalized raw series are then integrated through adaptive weighted fusion for prediction. Experiments on six real-world benchmark datasets across four prediction horizons show that FSHLNet yields consistently competitive results. Ablation experiments further verify the contribution of the dual-hypergraph architecture, cross-hypergraph attention, and adaptive fusion strategy.

Keywords: Long-term time series forecasting, Hypergraph learning, Frequency separation, Dual-hypergraph architecture, Cross-hypergraph attention

Acknowledgments This work was supported by the Science and Technology Research Project of Henan Province under Grant 252102210148.

1 Introduction

Time series prediction lies at the heart in numerous domains, aiming to infer future evolutionary trends based on historical observations (Benidis et al. 2022). Owing to the swift progress in IoT technologies, sensor networks, and information technology, massive time series data continuously emerge across various domains, providing a broad application space for forecasting technologies. In fields such as energy (Cheong et al. 2026; Yi et al. 2024), transportation (He et al. 2022; Chen et al. 2025), finance (Ariyo et al. 2014), meteorology (Wu et al. 2023), and healthcare (He et al. 2024), time series forecasting provides fundamental support for resource optimization, early warning of risks, and scientific decision-making. Accurate prediction of future trends in time series holds significant theoretical value and practical significance.

Existing time series forecasting methods primarily encompass traditional statistical approaches and deep learning architectures. Traditional statistical approaches (Zhang 2003) possess strong interpretability and low computational costs, but their reliance on linear assumptions renders them inadequate for capturing intricate nonlinear patterns and long-range dependencies. Among deep learning models, methods based on Multilayer Perceptrons (MLPs) (Zeng et al. 2023; Li et al. 2023) feature simple structures but lack explicit modeling of temporal order. Despite their efficiency in local pattern extraction, Convolutional Neural Networks (CNNs) (Bai et al. 2018; Wu et al. 2022; Cheng et al. 2025) face an inherent limitation: the restricted receptive field impedes their ability to model long-range dependencies. Transformer-based architectures (Vaswani et al. 2017; Wu et al. 2021; Zhou et al. 2022b; Zhang and Yan 2023; Liu et al. 2023) achieve long-range dependency capture realized by self-attention mechanisms and yield high prediction accuracy, yet they incur high computational complexity and exhibit limited capability in modeling periodic patterns. Approaches leveraging Large Language Models (LLMs) (Garza et al. 2023; Jin et al. 2024; Jiang et al. 2025; Liu et al. 2025; Ma et al. 2025) demonstrate strong zero-shot generalization abilities, but they demand exorbitant computational costs and lack explicit modeling of the unique structures inherent in temporal data. While these methods advance forecasting technologies from different perspectives, they share a common limitation. They primarily model data on time-domain, leaving frequency-domain features underexplored. Consequently, this paper focuses on the frequency domain analysis perspective and identifies two key challenges.

The modeling of High-order dependencies in the frequency domain remains inadequate. Different frequency components in frequency domain features do not exist independently. Instead, they exhibit high-order nonlinear associations, such as harmonic relationships and multi-frequency coupling, which constitute the intrinsic structure of frequency domain features (Yi et al. 2025) . However, existing research exhibits obvious limitations when processing frequency domain features. On one hand, current methods typically treat feature vectors of different frequency components separately, ignoring the correlations among frequencies. For instance, FreTS (Yi et al. 2023b) directly decomposes the frequency domain representation into independent real and imaginary vectors after the Discrete Fourier Transform before feeding them into an MLP. DeepTime (Woo et al. 2023) simply concatenates multi-scale random Fourier features. This processing paradigm severs the global context of the spectrum, making it difficult for the model to capture cross-frequency interaction patterns. On

the other hand, to model the dependencies among frequencies, researchers introduce Graph Neural Networks (GNNs) (StemGNN (Cao et al. 2020), FourierGNN (Yi et al. 2023a)). However, traditional GNNs only process pairwise relationships and struggle to capture the high-order group interaction patterns ubiquitously present in the real world, which still leads to information loss in frequency-domain representations.

Frequency-domain information remains insufficiently exploited. Existing research predominantly employs two specific methods for frequency selection in frequency domain processing. The first strategy is the Top-k frequency selection, which retains only the k frequency elements with the largest magnitudes after calculating the amplitude spectrum, while directly discarding the remaining components as noise or secondary information (DEPTS (Fan et al. 2022), TimesNet (Wu et al. 2022)). The second strategy utilizes low-pass filtering, retaining only components below a certain cutoff frequency and discarding all high-frequency components with low energy as noise (DEPTS (Fan et al. 2022), FiLM (Zhou et al. 2022a), TSLANet (Eldele et al. 2024)). Both strategies suffer from notable drawbacks. On the one hand, frequency components with smaller amplitudes often carry essential semantic information. On the other hand, even if certain frequency components contribute less energy, they may form harmonic relationships with dominant frequencies and jointly constitute complete periodic structures. By discarding such weak but critical components, these selection strategies break intrinsic spectral correlations and become a hidden bottleneck that constrains model performance.

To remedy the aforementioned issues, we introduce a Frequency-Separated Hypergraph Learning Network (FSHLNet) for time series forecasting. The model incorporates two core designs. First, it introduces hypergraph learning into frequency domain analysis to capture high-order group interaction patterns among multiple frequency components through hyperedges, thereby overcoming the limitations of traditional pairwise relationship modeling. Additionally, it features a full-coverage modeling strategy based on principal and secondary frequency separation. This strategy decouples frequency domain features into principal and secondary components, constructs hypergraph models for each, and facilitates collaborative modeling between them through cross-hypergraph message passing to achieve the complete utilization of frequency domain information.

The primary contributions of this study are summarized as follows.

1) We present FSHLNet, which attains deep exploration and complete utilization of information spanning both the temporal and spectral domains through dual-hypergraph high-order dependency modeling, a principal-secondary frequency full-coverage strategy, and time-frequency multi-branch feature fusion.

2) We construct a dual-hypergraph architecture for frequency domain high-order dependency modeling, introducing hypergraph learning into frequency analysis to capture group interaction patterns among frequency components via adaptive hypergraph construction and high-order message passing, thus solving the problem of traditional methods struggling to model high-order dependencies.

3) We design a full-coverage modeling strategy with principal and secondary frequency separation, which decouples frequency features into Top-k and non-Top-k components, builds hypergraphs for both, and enables information exchange through a

cross-hypergraph message passing mechanism, thoroughly mining auxiliary semantics without sacrificing dominant information.

4) We validate the effectiveness and generalization of our model on six publicly available datasets. The results confirm that FSHLNet clearly surpasses existing baselines under a range of metrics, verifying its performance advantages and robustness.

2 Related Work

2.1 Time Series Forecasting Methods

From early statistical approaches to contemporary deep learning paradigms, the field of time series forecasting has experienced nearly a century of development. Conventional approaches to time series forecasting are mainly grounded in classic statistical tools, including exponential smoothing methods (Holt 2004; Gardner Jr 1985) and autoregressive integrated moving average models (Zhang 2003). These methods demonstrate good performance when processing linear and stationary time series, but their modeling capability and prediction accuracy are significantly constrained when facing complex temporal data that feature non-linear dynamics, non-stationarity, and varying granularities.

To overcome the aforementioned limitations, researchers explore machine learning and deep learning methods with stronger nonlinear representation capabilities. MLP receives considerable interest in the time series prediction domain, owing to its concise structure and computational efficiency. DLinear (Zeng et al. 2023) separates input sequences into trend and seasonal constituent, appling a single-layer linear network to model each. RLinear (Li et al. 2023) further introduces residual connections and linear layer stacking strategies, achieving prediction accuracy comparable to complex models with an extremely simple structure. FreTS (Yi et al. 2023b) first introduces MLP into the frequency domain, applying linear layers to model associations among frequency components after spectral transformation, confirming the feasibility of frequency domain MLP architectures. However, existing MLP-based methods possess inherent limitations. Standard MLPs perform identical linear transformations on each time step of the sequence, lacking explicit modeling of temporal positional relationships. Although DLinear (Zeng et al. 2023) and RLinear (Li et al. 2023) partially compensate for this defect through sequence decomposition or residual structures, they do not fundamentally change the essence of MLPs encoding temporal points independently. Although FreTS (Yi et al. 2023b) explores the frequency domain modeling path, it treats spectral components as isolated feature vectors inputted to the MLP, ignoring the inherent harmonic structures and multi-frequency coupling relationships within the frequency domain. This simplified strategy of treating frequencies pairwise or independently restricts the model capacity to characterize frequency domain group interaction patterns. These limitations cause existing MLP-based methods to underperform when facing complex forecasting tasks that require delicate frequency domain deconstruction and reconstruction.

CNNs are also extensively studied in time series forecasting, extracting local features and hierarchical structures of time series through convolution and pooling operations. TCN (Bai et al. 2018) achieve parallel processing of sequences via dilated and causal

convolutions. TimesNet (Wu et al. 2022) reshapes 1D sequences into 2D tensors, utilizing multi-scale convolutions to capture complex temporal variation patterns in the two-dimensional space. ConvTimeNet (Cheng et al. 2025) synchronously solves the problems of local pattern extraction and multi-scale dependency capture through a deep hierarchical fully convolutional structure, showcasing the continuous potential of convolutional networks for temporal data analysis. However, CNN-based methods present two limitations. Their receptive fields are restricted by the convolution kernel size, and even when expanded through deep stacking or dilated convolutions, they remain insufficient for modeling global dependencies among ultra-long period frequencies. Moreover, these methods mostly focus on time domain operations, failing to deeply mine the inherent harmonic structures and multi-frequency coupling relationships in the frequency domain, resulting in insufficient utilization of frequency domain information. Taking TimesNet as an example, although it involves frequency domain processing, it only retains the k frequency components with the largest amplitudes, discarding massive frequency domain details and structural relationships in the process.

The self-attention mechanism inherent in Transformer architecture (Vaswani et al. 2017) effectively captures long-range dependencies, making it a highly favored approach in time series forecasting. Autoformer (Wu et al. 2021) achieves progressive separation of trend and periodic components through autocorrelation mechanisms and decomposition strategies. FEDformer (Zhou et al. 2022b) combines Fourier transforms with the Transformer architecture, utilizing the sparse characteristics of the frequency domain to reduce computational complexity. Crossformer (Zhang and Yan 2023) and iTransformer (Liu et al. 2023) enhance multivariate time-series modeling with respect to variable dependencies and inverted dimensions, respectively. However, the self-attention mechanism of Transformers fundamentally models pairwise relationships. It calculates the pairwise correlations between sequence points, making it difficult to directly capture high-order group interaction patterns among three or more frequency components. Furthermore, although some works introduce Fourier transforms (Autoformer (Wu et al. 2021), FEDformer (Zhou et al. 2022b)), they typically only use them for computational acceleration or treat them as independent features, ignoring the nonlinear association structures among frequency components.

Recently, time series foundation models such as TimeGPT (Garza et al. 2023), TimeLLM (Jin et al. 2024), and TimeXL (Jiang et al. 2025) draw inspiration from the pre-training paradigm of large language models to achieve zero-shot inference through large-scale pre-training. These models possess strong generalization capabilities. However, they incur exorbitant computational costs and generally lack explicit modeling of the unique structures of time series (such as periodicity and trends). They neither design specific mechanisms to capture high-order dependencies among frequency components nor distinguish and utilize the differential importance of various frequency components in forecasting tasks.

Non-pairwise high-order relationships, such as harmonic structures and multi-frequency coupling, naturally exist among frequency components in the frequency domain. Although existing methods have respective advantages in processing pairwise relationships or local patterns of temporal data, they generally struggle to model high-order group interactions among frequency components. This makes hypergraph

learning an ideal tool for modeling frequency domain interactions. The following section reviews hypergraph neural networks and their related works in various domains.

2.2 Hypergraph Neural Networks

GNNs model node interaction relationships through weighted edges and achieve significant breakthroughs in domains such as traffic flow forecasting (Yu et al. 2017), power load forecasting (Simeunović et al. 2021), and financial analysis (Xiang et al. 2022). STGCN (Yu et al. 2017) captures spatio-temporal dependencies through modeling traffic networks at multiple scales, yet they rely on predefined graph structures and struggle to process dynamic dependencies. MSGNet (Cai et al. 2024) capture multi-scale sequence correlations through frequency domain analysis and adaptive graph convolutions, capable of handling dynamic changes. Nevertheless, edges in GNNs can only associate two nodes, fundamentally remaining pairwise relationship modeling. This makes GNN-based methods inadequate for expressing the high-order group interactions ubiquitously present in the frequency domain, such as harmonic structures jointly formed by multiple frequency components, where the interaction of any two cannot fully characterize the overall structural information. FourierGNN (Yi et al. 2023a) and MagNet (Hong et al. 2024) combine graph neural networks with Fourier transforms and wavelet transforms, respectively, providing new paths for time series forecasting, but they similarly fail to break through the limitation of pairwise interactions.

Hypergraph Learning (HL), as an important advancement of graph learning, provides a rigorous mathematical instrument for modeling higher-order nonlinear interactions. Hyperedges in a hypergraph can connect multiple nodes simultaneously, more naturally representing many-body interactions and high-order dependencies in complex systems to capture group-level interaction patterns (Gao et al. 2020). HL achieves successful applications in computer vision (Nong et al. 2021; Hao et al. 2021), medical image analysis (Di et al. 2022; Li et al. 2025), and fault diagnosis (Yan et al. 2024; Luo et al. 2024). HGNN (Feng et al. 2019) extends graph convolutional networks to hypergraph structures, learning node representations through hypergraph convolutions. HyperGCN (Yadati et al. 2019) proposes a hyperedge approximation strategy, converting hyperedges into pairwise edges to train graph convolutional networks. HGAT (Bai et al. 2021) introduces attention mechanisms to learn hyperedge weights. HGNN+(Gao et al. 2022), acting as a high-order multimodal association framework, fuses different modal information through hyperedge groups. For time-series forecasting scenarios, hypergraphs are utilized to model high-order dependencies among multiple variables (such as ASHyper (Shang et al. 2024)), but existing research primarily focuses on time domain features.

3 Methodology

This study focuses on achieving comprehensive time-frequency fusion for time series forecasting. Accordingly, the overall architecture of the Frequency-Separated Hypergraph Learning Network (FSHLNet) is illustrated in Fig. 1 , comprising four synergistically designed modules: **(a) Data Preprocessing**, which standardizes the raw time series to provide a unified input; **(b) Time-Domain Feature Extraction**, which

captures local patterns and short-term dependencies through a 1D-CNN branch while retaining the normalized raw sequence to preserve global temporal structure; **(c) Frequency-Domain Dual-Hypergraph Feature Learning**, which decouples frequency components into principal and secondary groups, constructs dedicated hypergraphs for each, and enables their interaction through hypergraph convolution and cross-hypergraph attention, producing node-level and hyperedge-level frequency representations; and **(d) Time-Frequency Multi-Branch Fusion and Prediction**, which integrates the heterogeneous representations from both domains via an adaptive weighting mechanism for final prediction generation.

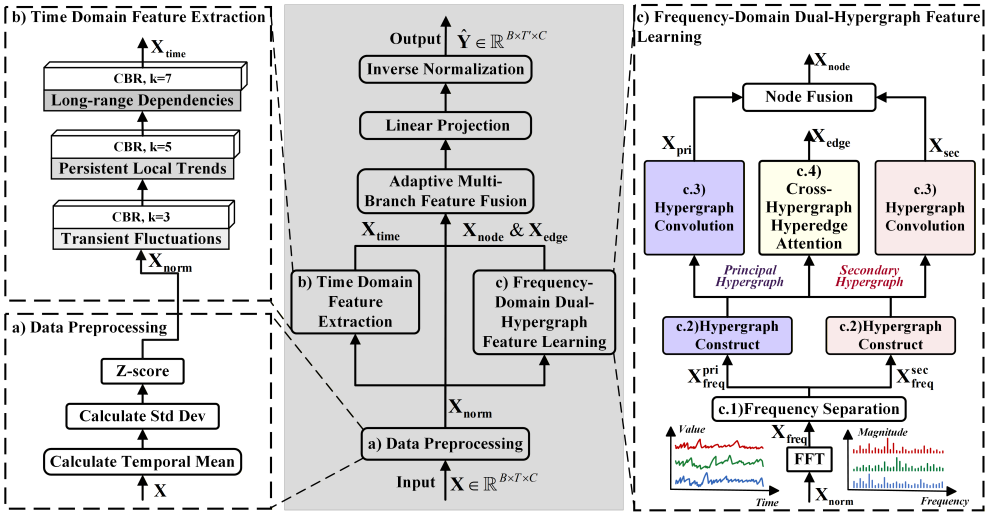


Fig. 1 Overall architecture of FSHLNet

3.1 Data Preprocessing

Given input time series $\mathbf{X} \in \mathbb{R}^{B\times T\times C}$, where B, T, and C represent the batch size, sequence length, and number of variates respectively, we apply Z-score normalization per variate across the temporal dimension:

$$\mu_{enc} = \frac{1}{T}\sum_{t=1}^{T} X_t \tag{1}$$

$$\sigma_{enc} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(X_t - \mu_{enc})^2 + \epsilon} \tag{2}$$

$$\mathbf{X}_{\text{norm},t} = \frac{\mathbf{X}_t - \mu_{\text{enc}}}{\sigma_{\text{enc}}}, \quad t = 1, \ldots, T \tag{3}$$

where $\mathbf{X}_t \in \mathbb{R}^{B\times C}$ corresponds to the data slice at time step t, $\mu_{\text{enc}} \in \mathbb{R}^{B\times 1\times C}$ and $\sigma_{\text{enc}} \in \mathbb{R}^{B\times 1\times C}$ are the temporal mean and standard deviation for each variate, $\epsilon = 10^{-5}$ is a stability constant, and $\mathbf{X}_{\text{norm}} \in \mathbb{R}^{B\times T\times C}$ is the normalized time series. $\mathbf{X}_{\text{norm}}$ serves as the unified input for feature extraction modules operating in the time and frequency domains.

3.2 Time-Domain Feature Extraction

To capture local patterns and multi-scale short-term dependencies within time-series data, we employ a one-dimensional convolutional neural network for time-domain feature extraction. Unlike frequency-domain analysis that reveals global periodic structures, this module focuses on local variation characteristics under different receptive fields, revealing fine-grained dynamic behaviors along the temporal dimension.

Specifically, this module consists of three cascaded one-dimensional convolutional layers, with batch normalization and ReLU activation applied after each layer. The three layers adopt progressively increasing kernel sizes to form a hierarchical multi-scale receptive field structure: the first layer ($k = 3$) captures transient fluctuations between adjacent time steps; the second layer ($k = 5$) extracts persistent local trends over a broader temporal window; the third layer ($k = 7$) models longer-range local dependencies with the widest receptive field. This design enables parallel feature extraction across multiple temporal scales, efficiently modeling hierarchical dependencies from fine-grained fluctuations to sustained local patterns without resorting to recurrent or attention mechanisms. Each convolutional layer is formalized as:

$$\mathbf{X}_{\text{time}}^{(l)} = \sigma\left(\text{BN}\left(\text{Conv1D}\left(\mathbf{X}_{\text{time}}^{(l-1)}, \mathbf{W}_{\text{time}}^{(l)}, \mathbf{b}_{\text{time}}^{(l)}\right)\right)\right) \tag{4}$$

where $l \in \{1, 2, 3\}$ denotes the layer index; $\mathbf{X}_{\text{time}}^{(0)} = \mathbf{X}_{\text{norm}}$ is the input; Conv1D$(\cdot)$ refers to the 1D convolution operation; $\mathbf{W}_{\text{time}}^{(l)} \in \mathbb{R}^{k_l\times C\times C}$ and $\mathbf{b}_{\text{time}}^{(l)} \in \mathbb{R}^{C}$ are the convolution kernel weights and bias; BN$(\cdot)$ corresponds to batch normalization; and $\sigma(\cdot)$ indicates the ReLU activation. The final output is denoted as $\mathbf{X}_{\text{time}} = \mathbf{X}_{\text{time}}^{(3)} \in \mathbb{R}^{B\times T\times C}$.

The extracted time-domain features $\mathbf{X}_{\text{time}}$, together with the normalized raw sequence $\mathbf{X}_{\text{norm}}$ (which preserves global trends and long-term structural information), constitute the complete time-domain representations that will be forwarded to the subsequent time-frequency fusion stage.

3.3 Frequency-Domain Dual-Hypergraph Feature Learning

Frequency-domain analysis reveals implicit periodic structures and frequency composition patterns that are cannot be effectively identified in the time domain, offering a complementary perspective for time series prediction. This section presents the frequency-domain feature learning method based on a dual-hypergraph architecture, as illustrated in Fig. 2 . The process involves three stages: first, the time series is mapped to the frequency domain through Fourier transformation; second, frequency components are separated into principal and secondary groups according to their energy contributions; finally, dedicated hypergraphs are constructed for each group, and collaborative

modeling between the two groups is realized through hypergraph convolution and a cross-hypergraph hyperedge attention mechanism. This module ultimately produces two complementary representations: node-level fused features and hyperedge-level interaction features, which jointly serve as the frequency-domain contribution to the subsequent multi-branch fusion.

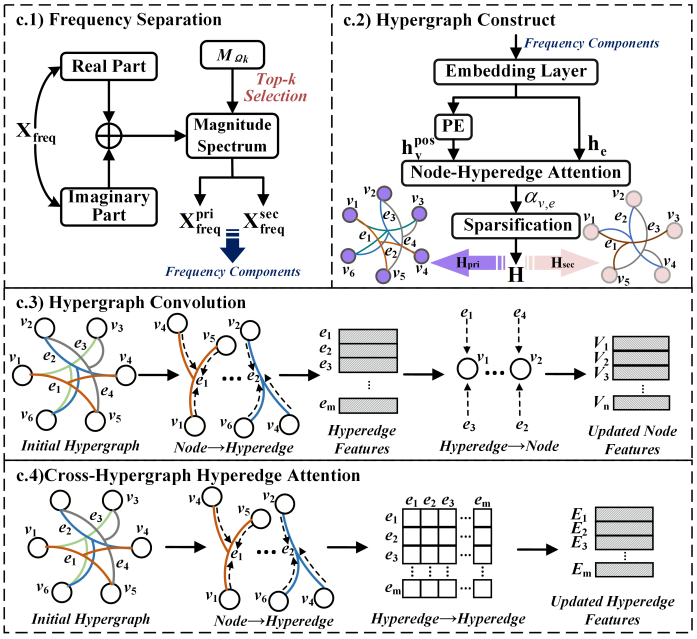


Fig. 2 Structure of the frequency-domain dual-hypergraph feature learning module

3.3.1 Time-Frequency Transformation

Frequency-domain analysis begins by converting the normalized time series into the frequency domain, where the data is represented as a superposition of sinusoidal constituents at distinct frequency values. This transformation reveals the periodic structures and spectral characteristics that are not directly observable in the original time domain. Therefore, the Fast Fourier Transform (FFT) is applied across the time axis for each variate of $\mathbf{X}_{\text{norm}}$:

$$\mathbf{X}_{\text{freq}} = \text{FFT}(\mathbf{X}_{\text{norm}}) \tag{5}$$

where $\mathbf{X}_{\text{freq}} \in \mathbb{C}^{B \times T \times C}$ is the complex-valued frequency-domain representation. Each element of $\mathbf{X}_{\text{freq}}$ encodes both the amplitude and phase of a specific frequency component, providing a complete description of the spectral composition within time-series data, and underpins the subsequent principal-secondary frequency separation and hypergraph-based dependency modeling.

3.3.2 Principal-Secondary Frequency Separation

The magnitude spectrum reflects the relative energy intensity of each frequency component in the time series and serves as a crucial basis for identifying dominant periodic patterns. To quantify the spectral energy distribution, the average magnitude across all variates is derived from $\mathbf{X}_{\text{freq}}$:

$$\mathbf{s}_{\text{freq}} = \frac{1}{C} \sum_{j=1}^{C} \sqrt{\text{Re}(\mathbf{X}_{\text{freq}}[:, :, j])^2 + \text{Im}(\mathbf{X}_{\text{freq}}[:, :, j])^2} \tag{6}$$

where j indexes the variates, and $\mathbf{s}_{\text{freq}} \in \mathbb{R}^{B \times T}$ denotes the average magnitude at each frequency index.

Based on $\mathbf{s}_{\text{freq}}$, we select the k frequency components with the most prominent energy as the principal frequency components, which carry the most primary periodic structures within time-series data; the remaining frequency components are categorized as secondary frequency components. Although their energy is relatively weak, they may contain critical auxiliary semantics, such as harmonic correlations and precursor signals of local anomalies. Departing from conventional approaches that directly discard the secondary components, this paper adopts a full-retention dual-branch strategy, constructing a frequency mask to achieve decoupling:

$$\mathcal{M}_{\Omega_k}[f] = \begin{cases} 1 & \text{if } f \in \Omega_k \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$\begin{aligned} \mathbf{X}_{\text{freq}}^{\text{pri}} &= \mathbf{X}_{\text{freq}} \odot \mathcal{M}_{\Omega_k} \\ \mathbf{X}_{\text{freq}}^{\text{sec}} &= \mathbf{X}_{\text{freq}} \odot (1 - \mathcal{M}_{\Omega_k}) \end{aligned} \tag{8}$$

where Ω_k denotes the index set of the $Top-k$ frequency components; $\mathcal{M}_{\Omega_k} \in \mathbb{R}^{T \times 1}$ is the frequency mask matrix; $\odot$ denotes element-wise multiplication; and $X_{\text{freq}}^{\text{pri}} \in \mathbb{R}^{B \times T \times C}$ and $X_{\text{freq}}^{\text{sec}} \in \mathbb{R}^{B \times T \times C}$ correspond to the principal and secondary frequency components, respectively. Through this separation strategy, both categories of frequency components are fully preserved and separately fed into two independent hypergraph networks for modeling, thereby achieving lossless coverage of the entire frequency-domain information. Through this separation strategy, both categories of frequency components are fully preserved and separately fed into two independent hypergraph networks for modeling, thereby achieving lossless coverage of the entire frequency-domain information.

3.3.3 Dual-Hypergraph High-Order Dependency Modeling

As a natural generalization of graph structures, hypergraphs allow hyperedges to simultaneously connect multiple nodes, thereby inherently expressing high-order group interaction patterns among frequency components. To fully exploit the distinct characteristics of the principal and secondary frequency components as well as their synergistic relationships, the Dual-Hypergraph High-Order Dependency Modeling architecture is

proposed. Two hypergraph networks sharing identical architectures are constructed, with $\mathbf{X}_{\text{freq}}^{\text{pri}}$ and $\mathbf{X}_{\text{freq}}^{\text{sec}}$ serving as their respective inputs. Through adaptive hypergraph construction, a sample-specific hypergraph structure is formed. After construction, each hypergraph sequentially undergoes hypergraph convolution and node fusion, and cross-hypergraph hyperedge attention, ultimately producing node-level fused features and hyperedge-level interaction features, which are jointly forwarded to the time-frequency fusion module.

Adaptive Hypergraph Construction

Adaptive hypergraph construction aims to obtain the initial node and hyperedge representations and establish the association mapping between them. Specifically, each frequency component serves as a node, representing an individual frequency component in the spectrum, while each hyperedge represents a potential high-order group interaction pattern among multiple frequency components. First, the frequency components are mapped to continuous vector representations through embedding layers, yielding node embeddings $\mathbf{h}_v$ and hyperedge embeddings $\mathbf{h}_e$. To preserve the positional information of frequency components in the spectrum, sinusoidal positional encoding is introduced and added element-wise to the node embeddings:

$$\mathbf{h}_v^{\text{pos}} = \mathbf{h}_v + \text{PE}(v) \tag{9}$$

where $\mathbf{h}_v^{\text{pos}}$ denotes the node embedding with positional encoding, $\text{PE}(v)$ is the sinusoidal positional encoding for frequency component v. Subsequently, the association strength between nodes and hyperedges is adaptively learned through an attention mechanism:

$$\alpha_{v,e} = \frac{\exp\left(\mathbf{h}_v^{\text{pos}} \cdot \mathbf{h}_e^{\top}/\sqrt{d}\right)}{\sum_{e'=1}^{E} \exp\left(\mathbf{h}_v^{\text{pos}} \cdot \mathbf{h}_{e'}^{\top}/\sqrt{d}\right)} \tag{10}$$

$$\mathbf{H}_{v,e} = \begin{cases} 1 & \text{if } \alpha_{v,e} > \theta \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where $\alpha_{v,e}$ quantifies the attentional relevance linking node v and hyperedge e, d functions as the scaling factor to prevent gradient vanishing, e' indexes all hyperedges for softmax normalization, θ is the attention weight threshold that controls the sparsity of the hypergraph. $\mathbf{H} \in \{0,1\}^{V \times E}$ stands for the learned hypergraph incidence matrix, with V denoting the number of node set and E denoting the number of hyperedge set.

The principal and secondary frequency components each independently undergo the above process to construct their respective hypergraph structures, denoted as $\mathbf{H}_{\text{pri}}$ and $\mathbf{H}_{\text{sec}}$.

Hypergraph Convolution and Node Fusion

Building upon the constructed hypergraph structures, hypergraph convolution performs a "node-hyperedge-node" feature transformation. For a hypergraph with incidence matrix $\mathbf{H}$, the node degree matrix $\mathbf{D}_v$ and hyperedge degree matrix $\mathbf{D}_e$ are first

computed:

$$\begin{aligned} \mathbf{D}_v &= \operatorname{diag}\left(\sum_e \mathbf{H}_{v,e}\right) \\ \mathbf{D}_e &= \operatorname{diag}\left(\sum_v \mathbf{H}_{v,e}\right) \end{aligned} \tag{12}$$

The initial node features are set to the corresponding frequency-domain features, denoted as $\mathbf{X}^{(0)} \in \mathbb{R}^{V \times C}$, where each frequency component contributes its C-dimensional feature vector as the initial representation of the corresponding node. A first-order hypergraph convolution is then formulated as:

$$\mathbf{X}^{(l+1)} = \sigma\left((D_v)^{-1/2}\mathbf{H}(D_e)^{-1}\mathbf{H}^\top(D_v)^{-1/2}\mathbf{X}^{(l)}\mathbf{W}^{(l)}\right) \tag{13}$$

where $\mathbf{X}^{(l)} \in \mathbb{R}^{V \times d}$ denotes the node features at the l-th layer with $d = C$ for $l = 0$, $\mathbf{W}^{(l)} \in \mathbb{R}^{d \times d'}$ represents the learnable weight matrix, and $\sigma(\cdot)$ refers to the activation function. This message-passing process updates each node by aggregating information from all related nodes within its incident hyperedges.

The above convolution is applied independently to the principal and secondary hypergraph, following L convolutional layers, the resulting node features $\mathbf{X}_{\text{pri}} \in \mathbb{R}^{V \times d}$ and $\mathbf{X}_{\text{sec}} \in \mathbb{R}^{V \times d}$ are obtained. These two sets of node features are concatenated across the feature dimension and fused via a multi-layer perceptron with a bottleneck structure:

$$\mathbf{X}_{\text{node}} = \text{MLP}_{\text{freq}}\left(\text{Concat}(\mathbf{X}_{\text{pri}}, \mathbf{X}_{\text{sec}})\right) \tag{14}$$

where MLP_{freq} first expands the feature dimension to four times the original size to promote information interaction, then projects it back to the original dimension after ReLU activation. The resulting unified frequency-domain node features are reshaped to $\mathbf{X}_{\text{node}} \in \mathbb{R}^{B \times T \times C}$, serving as the node-level representation for subsequent fusion.

Cross-Hypergraph Hyperedge Attention

While the node features have been updated through hypergraph convolution in the previous section, the hyperedge embeddings $\mathbf{h}_e$ remain at their initial values from the adaptive hypergraph construction stage. These embeddings encode distinct high-order group patterns: hyperedges in $\mathbf{H}_{\text{pri}}$ capture principal frequency group structures, while those in $\mathbf{H}_{\text{sec}}$ capture secondary frequency group structures. To enable information exchange across these two groups, a cross-hypergraph attention mechanism is deployed over all hyperedge embeddings.

The hyperedge embeddings from both hypergraphs are collected into a unified set of $E_p + E_s$ vectors, where E_p and E_s represent the hyperedge counts of the principal and secondary hypergraphs, and stacked into a matrix $\mathbf{S} \in \mathbb{R}^{B \times (E_p + E_s) \times d}$. Multi-head self-attention is then performed over this matrix, allowing each hyperedge to attend to every other hyperedge regardless of its origin:

$$\mathbf{S}' = \text{LayerNorm}\left(\mathbf{S} + \text{MultiHead}(\mathbf{S}, \mathbf{S}, \mathbf{S})\right) \tag{15}$$

where $\mathbf{S}'$ denotes the updated hyperedge features after multi-head attention. The attended hyperedge features are further processed through a feed-forward network with residual connection and layer normalization:

$$\mathbf{S}'' = \text{LayerNorm}\left(\mathbf{S}' + \text{FFN}(\mathbf{S}')\right) \tag{16}$$

Finally, the updated hyperedge features are aggregated and projected to the temporal dimension via a learnable linear transformation:

$$\mathbf{X}_{\text{edge}} = \text{Linear}\left(\frac{1}{E_p + E_s} \sum_{e=1}^{E_p+E_s} \mathbf{S}''_{:,e,:}\right) \tag{17}$$

where $\mathbf{S}''$ is the output of the feed-forward network, and the resulting vector is broadcast to $\mathbf{X}_{\text{edge}} \in \mathbb{R}^{B\times T\times C}$ by replicating along the temporal dimension. This output, together with the fused node features $\mathbf{X}_{\text{node}}$ from the previous section, constitutes the frequency-domain contribution to the final multi-branch fusion.

3.4 Time-Frequency Multi-Branch Fusion and Prediction

The time-domain and frequency-domain modules produce four complementary representations: the normalized raw sequence $\mathbf{X}_{\text{norm}}$, the multi-scale temporal features $\mathbf{X}_{\text{time}}$, the fused node-level frequency features $\mathbf{X}_{\text{node}}$, and the updated hyperedge-level frequency features $\mathbf{X}_{\text{edge}}$. These four branches are combined via an adaptive fusion scheme and mapped to the prediction results.

3.4.1 Adaptive Multi-Branch Feature Fusion

To dynamically integrate the four heterogeneous representations, we adopt a learnable scalar weighting strategy. A learnable weight parameter is assigned to each branch, initialized to 1, and normalized by the softmax function to yield a competitive weight distribution:

$$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]^\top = \text{softmax}(\mathbf{w}) \tag{18}$$

where $\mathbf{w} = [w_1, w_2, w_3, w_4]^\top$ denotes the four learnable scalar parameters. The fused feature is obtained through weighted summation:

$$\mathbf{X}_{\text{final}} = \alpha_1 \mathbf{X}_{\text{norm}} + \alpha_2 \mathbf{X}_{\text{time}} + \alpha_3 \mathbf{X}_{\text{node}} + \alpha_4 \mathbf{X}_{\text{edge}} \tag{19}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^4$ satisfies $\sum_{i=1}^{4} \alpha_i = 1$, and $\mathbf{X}_{\text{final}} \in \mathbb{R}^{B\times T\times C}$ is the fused representation. During training, the weight parameters are end-to-end optimized by gradient signals, enabling the model to adaptively modulate the contribution of each branch with respect to the forecasting task.

3.4.2 Prediction Output

The fused feature $\mathbf{X}_{\text{final}}$ is transformed into the prediction horizon via a linear mapping:

$$\hat{\mathbf{Y}}_{\text{norm}} = \mathbf{W}_{\text{pred}} \mathbf{X}_{\text{final}} + \mathbf{b}_{\text{pred}} \tag{20}$$

Since the input data underwent Z-score normalization, the predicted outputs are rescaled to the original data range via inverse normalization:

$$\hat{\mathbf{Y}} = \hat{\mathbf{Y}}_{\text{norm}} \odot \sigma_{\text{enc}} + \mu_{\text{enc}} \tag{21}$$

where $\mathbf{W}_{\text{pred}} \in \mathbb{R}^{T' \times T}$ and $\mathbf{b}_{\text{pred}} \in \mathbb{R}^{T'}$ are the prediction layer parameters, $\hat{\mathbf{Y}}_{\text{norm}} \in \mathbb{R}^{B \times T' \times C}$ is the normalized prediction, $\hat{\mathbf{Y}} \in \mathbb{R}^{B \times T' \times C}$ is the final forecast, and σ_{enc}, μ_{enc} are the standard deviation and mean computed in the preprocessing stage.

4 Experiment

4.1 Experimental Setup

1) **Datasets:** FSHLNet is evaluated against six publicly available benchmark datasets spanning diverse application domains. Their statistical details are listed in Table 1. Each dataset is splited into training, validation, and test sets following the standard chronological split, using ratios of 6:2:2 for ETT and 7:2:1 for Weather and Electricity. Each dataset is normalized using instance-wise Z-score normalization to mitigate distribution shift issues across training and testing sets.

Table 1 Statistics of datasets

Dataset	Metics	Samples	Frequency	Information
ETTh1	7	17420	1h	Temperature
ETTh2	7	17420	1h	Temperature
ETTm1	7	69680	15min	Temperature
ETTm2	7	69680	15min	Temperature
Weather	21	52696	10min	Weather
Electricity	321	26304	1h	Electricity

2) **Baselines:** FSHLNet is compared against nine representative methods spanning diverse technical paradigms:

- MLP-based model: DLinear.
- Transformer-based models: Crossformer, FEDformer.
- Graph/Hypergraph-based models: MSGNet, Ada-MSHyper.
- CNN-based model: TimesNet.
- LLM-based model: TimeCMA.
- State space models: Timepro (Mamba-based), FiLM.

3) **Metrics:** Following the standard practice, Mean Squared Error (MSE) and Mean Absolute Error (MAE) are adopted to quantify forecasting accuracy. MSE amplifies the impact of large prediction errors, making it sensitive to extreme values, while MAE directly quantifies the magnitude of average forecasting errors. Their formulations are

given below:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{22}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{23}$$

4.2 Main Results

Long-term multivariate forecasting experiments are conducted with a constant input sequence length $T = 96$ and prediction horizon $H \in \{96, 192, 336, 720\}$. As illustrated in Table 2, the top-ranked performances are highlighted in bold, while the runner-up results are underlined. All entries are reported in MSE/MAE format.

1) **Overall Performance:** From Table 2, FSHLNet ranks first or second across nearly all datasets and forecast lengths. Compared with the baselines, the average MSE improvement across all six datasets and four prediction horizons ranges from 1.6% to 63.5%, while the MAE improvement ranges from 1.1% to 48.1%.

Fig. 3 provides an intuitive comparison of overall forecasting performance across multiple datasets. The left radar chart reports the average MSE of each model on the six benchmarks, and the right chart reports the average MAE. Lower values indicate smaller prediction errors. FSHLNet consistently occupies the outermost layer across the vast majority of datasets, confirming its state-of-the-art performance.

This overall performance advantage stems from the three core designs of FSHLNet. First, the dual-hypergraph architecture captures group-level interactions among frequency components, overcoming the pairwise limitation of graph-based methods. Second, the principal-secondary frequency separation strategy retains the full spectrum rather than discarding low-energy components, eliminating the information degradation stemming from hard thresholding. Third, the cross-hypergraph attention mechanism facilitates synergy between principal and secondary frequency groups, allowing the model to exploit subtle yet predictive patterns overlooked by prior methods.

2) **Temporal Stability in Long-Range Forecasting:** A critical strength of FSHLNet lies in its remarkable stability with increasing prediction horizons. Fig. 4 illustrates the variation in MSE across different prediction horizons: the multi-step prediction error increases gradually with the prediction length, following the typical pattern of cumulative uncertainty in time series forecasting. The error growth slows significantly after the initial steps and remains stable without exploding even at distant future time steps, verifying the robustness of the model for long-term forecasting.

This temporal stability stems from the complementary design of the principal-secondary frequency separation and the time-frequency fusion mechanism. As the prediction horizon extends, the uncertainty associated with dominant periodic patterns progressively accumulates. Under such conditions, the secondary frequency components, which are fully preserved and independently modeled in FSHLNet, serve as supplementary constraints that mitigate error drift. Their harmonic relationships with the principal frequency components, captured through the cross-hypergraph hyperedge attention mechanism, provide structurally meaningful constraints when the predictive

Table 2 Comparative performance analysis of different models over multiple datasets. The look-back length L for all models is uniformly set to 96, and the forecast horizon $H \in \{96, 192, 336, 720\}$

Models		FSHLNet		Timepro		TimeCMA		Ada-MSHyper		MSGNet		Crossformer		TimesNet		DLinear		FiLM		FEDformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.315**	**0.351**	0.324	0.362	0.330	0.371	0.340	0.371	0.319	0.366	0.404	0.426	0.338	0.375	0.345	0.372	0.353	0.370	0.379	0.419
	192	**0.362**	**0.376**	0.367	0.384	0.370	0.391	0.388	0.392	0.376	0.397	0.450	0.451	0.374	0.387	0.380	0.389	0.389	0.387	0.426	0.441
	336	**0.388**	**0.396**	0.401	0.408	0.406	0.413	0.403	0.403	0.417	0.422	0.532	0.515	0.410	0.411	0.413	0.413	0.421	0.408	0.445	0.459
	720	**0.450**	**0.433**	0.463	0.445	0.469	0.449	0.471	0.436	0.481	0.458	0.666	0.589	0.478	0.450	0.474	0.453	0.481	0.441	0.543	0.490
	Avg	**0.379**	**0.389**	0.389	0.400	0.394	0.406	0.401	0.401	0.398	0.411	0.513	0.495	0.400	0.406	0.403	0.407	0.411	0.402	0.448	0.452
ETTm2	96	**0.172**	**0.251**	0.178	0.260	0.184	0.267	0.180	0.264	0.177	0.262	0.287	0.366	0.187	0.267	0.193	0.292	0.183	0.266	0.203	0.287
	192	**0.234**	**0.293**	0.243	0.303	0.256	0.315	0.246	0.306	0.247	0.307	0.414	0.492	0.249	0.309	0.284	0.362	0.248	0.305	0.269	0.328
	336	**0.301**	**0.334**	0.303	0.342	0.312	0.349	0.315	0.349	0.312	0.346	0.597	0.542	0.321	0.351	0.369	0.427	0.309	0.343	0.325	0.366
	720	**0.396**	**0.390**	0.404	0.401	0.411	0.405	0.408	0.401	0.414	0.403	1.730	1.042	0.408	0.403	0.554	0.522	0.410	0.400	0.421	0.415
	Avg	**0.276**	**0.317**	0.282	0.327	0.291	0.334	0.287	0.330	0.288	0.330	0.757	0.611	0.291	0.333	0.350	0.401	0.288	0.329	0.305	0.349
ETTh1	96	0.380	0.399	**0.375**	0.398	0.391	0.412	0.386	**0.395**	0.390	0.411	0.423	0.448	0.384	0.402	0.386	0.400	0.438	0.433	0.376	0.419
	192	**0.413**	0.424	0.432	0.430	0.432	0.434	0.437	**0.423**	0.442	0.442	0.471	0.474	0.436	0.429	0.437	0.432	0.493	0.466	0.420	0.448
	336	0.466	0.442	0.469	0.448	0.477	0.456	0.475	**0.441**	0.480	0.468	0.570	0.546	0.491	0.469	0.481	**0.459**	0.547	0.495	0.459	0.465
	720	**0.464**	**0.465**	0.477	0.474	0.469	0.470	0.568	0.518	0.494	0.488	0.653	0.621	0.521	0.500	0.519	0.516	0.586	0.538	0.506	0.507
	Avg	**0.431**	**0.433**	0.438	0.438	0.442	0.443	0.467	0.444	0.452	0.452	0.529	0.522	0.458	0.450	0.456	0.452	0.516	0.483	0.440	0.460
ETTh2	96	**0.284**	**0.338**	0.301	0.351	0.329	0.364	0.299	0.345	0.328	0.371	0.745	0.584	0.340	0.374	0.333	0.387	0.322	0.364	0.358	0.397
	192	**0.366**	**0.386**	**0.366**	0.393	0.408	0.413	0.387	0.400	0.402	0.414	0.877	0.656	0.402	0.414	0.477	0.476	0.404	0.414	0.429	0.439
	336	**0.410**	**0.419**	0.423	0.432	0.440	0.441	0.450	0.445	0.435	0.443	1.043	0.731	0.452	0.452	0.481	0.459	0.435	0.445	0.496	0.487
	720	0.419	**0.435**	0.428	0.446	0.445	0.455	0.449	0.457	**0.417**	0.441	1.104	0.763	0.462	0.468	0.831	0.657	0.447	0.458	0.463	0.474
	Avg	**0.370**	**0.395**	0.380	0.406	0.406	0.418	0.396	0.412	0.396	0.417	0.942	0.684	0.414	0.427	0.531	0.495	0.402	0.420	0.437	0.449
Weather	96	0.161	**0.207**	0.176	0.216	0.172	0.219	0.179	0.229	0.163	0.212	**0.158**	0.230	0.172	0.220	0.196	0.255	0.199	0.262	0.217	0.296
	192	0.210	**0.251**	0.225	0.257	0.222	0.261	0.229	0.270	0.212	0.254	**0.206**	0.277	0.219	0.261	0.237	0.296	0.228	0.288	0.276	0.336
	336	0.277	0.301	0.280	0.297	0.278	0.302	0.294	0.315	0.272	0.299	0.272	0.335	0.280	0.306	0.283	0.335	**0.267**	0.323	0.339	0.380
	720	0.362	0.353	0.358	**0.348**	0.351	0.350	0.376	0.363	0.350	**0.348**	0.398	0.418	0.365	0.359	0.345	0.381	**0.319**	0.361	0.403	0.428
	Avg	0.253	**0.278**	0.260	0.280	0.256	0.283	0.270	0.294	**0.249**	**0.278**	0.259	0.315	0.259	0.287	0.265	0.317	0.253	0.309	0.309	0.360
Electricity	96	0.157	0.254	0.162	**0.249**	**0.147**	0.250	0.181	0.274	0.165	0.274	0.219	0.314	0.168	0.272	0.197	0.282	0.198	0.274	0.193	0.308
	192	0.171	0.265	0.172	**0.259**	**0.168**	0.270	0.192	0.284	0.184	0.292	0.231	0.322	0.184	0.289	0.196	0.285	0.198	0.278	0.201	0.315
	336	**0.187**	0.282	0.188	**0.275**	0.215	0.315	0.212	0.301	0.195	0.302	0.246	0.337	0.198	0.300	0.209	0.301	0.217	0.300	0.214	0.329
	720	**0.218**	**0.306**	0.228	0.309	0.255	0.351	0.253	0.335	0.231	0.332	0.280	0.363	0.220	0.320	0.245	0.333	0.278	0.356	0.246	0.355
	Avg	**0.183**	0.277	0.188	**0.273**	0.196	0.297	0.210	0.299	0.194	0.300	0.244	0.334	0.193	0.295	0.212	0.300	0.223	0.302	0.214	0.327
Count		**31**		6		2		3		2		2		0		1		2		0	

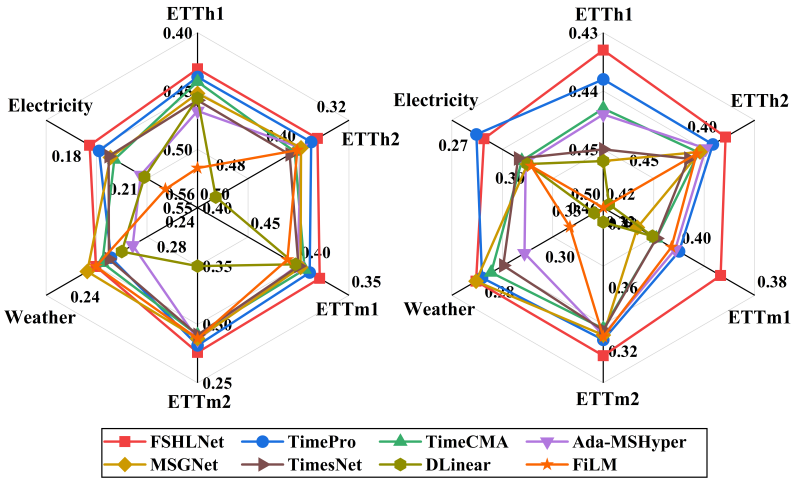


Fig. 3 Overall performance comparison via radar charts (Left: MSE, Right: MAE)

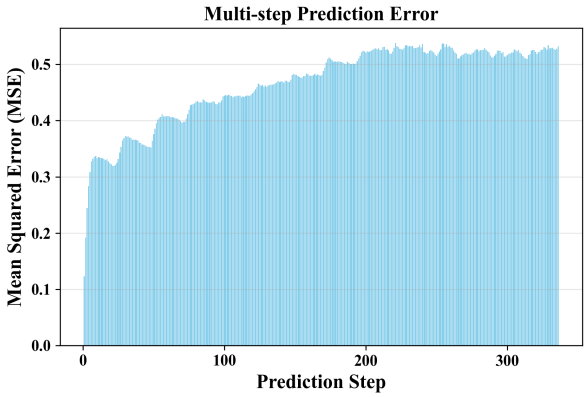


Fig. 4 Multi-step prediction error of FSHLNet in terms of MSE on the ETTh1 dataset across different prediction horizons

power of principal components alone begins to degrade. Meanwhile, the original temporal details preserved by the time-domain branch play a critical constraining role in long-range forecasting: when the frequency-domain branch accumulates deviations in characterizing local trends due to its focus on periodic structures, the adaptive weighting mechanism automatically increases the weight contribution of the time-domain branch, imposing implicit correction on the frequency-domain predictive tendency.

4.3 Ablation Studies

Ablation studies are conducted to quantify the contribution of each key module in FSHLNet. Optimal values are highlighted in bold.

1) **Effect of Time-Frequency Information Fusion:** The contribution of time-frequency fusion is examined by comparing FSHLNet against two variants.

- FSHLNet-w/o-time: Removes the time-domain feature extraction module, retaining only the frequency-domain processing path.
- FSHLNet-w/o-freq: Removes the frequency-domain module and directly feeds the original time-domain data into the hypergraph.

Fig. 5 reports the experimental outcomes obtained from ETTh2 and ETTm2. FSHLNet attains state-of-the-art results across all scenarios. The degraded performance of FSHLNet-w/o-freq verifies that frequency-domain analysis provides vital periodic priors, whose absence severely impairs long-term dependency modeling. Meanwhile, the performance drop of FSHLNet-w/o-time indicates that local time-domain patterns complement global frequency-domain features, and the multi-scale time-domain branch improves model sensitivity to short-term fluctuations and local patterns.

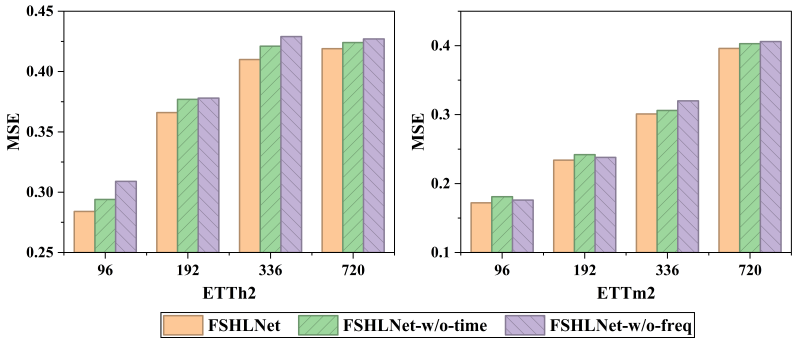


Fig. 5 Effect of time-frequency information fusion

2) **Effect of Dual-Hypergraph Architecture:** The impact of the dual-hypergraph architecture is assessed through comparison with three variants.

- FSHLNet-w-GCN: Replaces hypergraph convolution with GCN.
- FSHLNet-w-GAT: Replaces hypergraph convolution with GAT.
- FSHLNet-w-Shyper: Uses a single hypergraph to directly process the complete frequency-domain features.

Table 3 Effect of dual-hypergraph architecture

Models		FSHLNet		FSHLNet-w-GCN		FSHLNet-w-GAT		FSHLNet-w-Shyper	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.380**	**0.399**	0.415	0.423	0.391	0.402	0.412	0.416
	192	**0.413**	**0.424**	0.449	0.449	0.441	0.432	0.442	0.429
	336	**0.466**	**0.442**	0.479	0.446	0.474	0.447	0.487	0.453
	720	**0.464**	**0.465**	0.485	**0.465**	0.480	0.469	0.481	0.467
	Avg	**0.431**	**0.433**	0.457	0.446	0.447	0.438	0.456	0.441
ETTm1	96	**0.315**	**0.351**	0.327	0.359	0.325	0.358	0.326	0.358
	192	**0.362**	**0.376**	0.377	0.388	**0.362**	**0.376**	0.365	0.377
	336	**0.388**	**0.396**	0.405	0.409	0.393	0.397	0.397	0.399
	720	**0.450**	**0.433**	0.463	0.437	0.472	0.446	0.452	0.438
	Avg	**0.379**	**0.389**	0.393	0.398	0.388	0.394	0.385	0.393
Weather	96	**0.161**	**0.207**	**0.161**	**0.207**	0.164	0.209	0.167	0.213
	192	**0.210**	**0.251**	0.218	0.260	0.220	0.260	0.217	0.259
	336	**0.277**	**0.301**	0.309	0.320	0.317	0.323	0.284	0.303
	720	**0.362**	**0.353**	0.367	0.357	0.368	0.358	0.367	0.355
	Avg	**0.253**	**0.278**	0.264	0.286	0.267	0.288	0.259	0.283

The experimental results of these variants over the ETTh1, ETTm1, and Weather benchmarks are detailed in Table 3. Both FSHLNet-w-GCN and FSHLNet-w-GAT underperform FSHLNet. This indicates that dependencies among frequency components are not limited to pairwise interactions but involve more complex high-order group relationships. Hypergraphs naturally model high-order interactions via hyperedges, resulting in superior performance. FSHLNet-w-Shyper performs noticeably worse than the complete dual-hypergraph architecture of FSHLNet. This phenomenon arises because the dual-hypergraph architecture enables the model to learn distinct dependency structures and feature representations for each group of frequency components, avoiding feature ambiguity caused by information mixing and thus achieving more fine-grained and accurate frequency-domain modeling.

3) **Effect of Cross-Hypergraph Hyperedge Attention:** The role of cross-hypergraph hyperedge attention is evaluated against two variants.

- FSHLNet-w/o-hyperattn: Removes the cross-hypergraph hyperedge attention mechanism, directly using the initial hyperedge embeddings for subsequent computation.
- FSHLNet-w-avgpool: Replaces the cross-hypergraph hyperedge attention mechanism with average pooling over hyperedges.

The performance of these variants over the ETTh2 and ETTm1 benchmarks are presented in Fig. 6. FSHLNet attains the top-ranked performance across every test scenarios. The performance degradation of FSHLNet-w/o-hyperattn indicates that the cross-hypergraph hyperedge attention mechanism is able to adaptively identify the significance of different hyperedges for the current prediction task. The performance of FSHLNet-w-avgpool lies between that of FSHLNet and FSHLNet-w/o-hyperattn,

demonstrating that the attention-based weighting mechanism assigns hyperedge weights more flexibly than simple average pooling, enabling the model to prioritize the most informative high-order relationships targeted at the current prediction task. The conclusion is further supported by the heatmaps in Fig. 7.

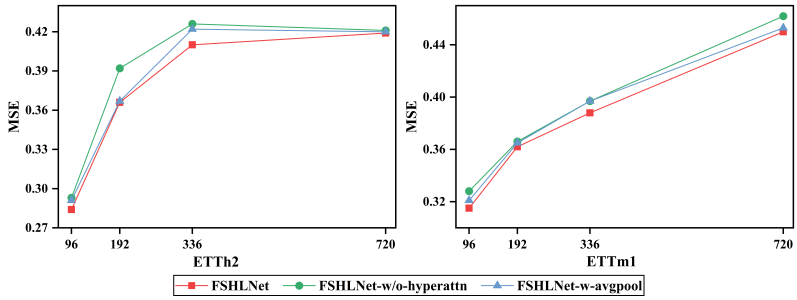


Fig. 6 Effect of cross-hypergraph hyperedge attention

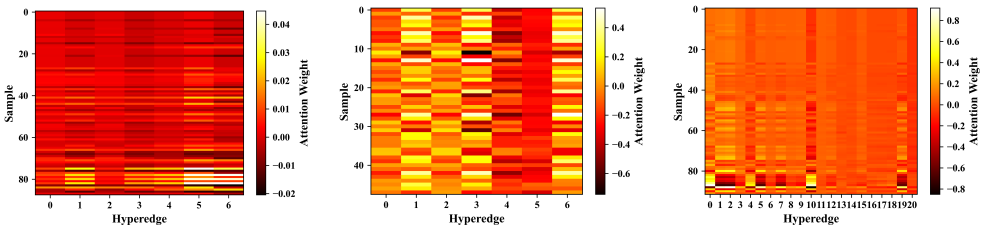


Fig. 7 Cross-hypergraph hyperedge attention weight heatmaps. Subfigures (a), (b), and (c) correspond to ETTm1 at horizon 720, ETTm2 at horizon 192, and Weather at horizon 96, respectively

The heatmaps reveal distinct sample-specific distributions of attention weights across different hyperedges. For different samples, certain hyperedges are assigned significantly high weights (bright regions), while others are suppressed (dark regions). This confirms that the cross-hypergraph hyperedge attention mechanism adaptively activates key high-order dependency channels and weakens irrelevant information in light of the inherent properties of input sequence, rather than adopting a fixed average pooling strategy. This dynamic and differentiated weight assignment is a key reason why FSHLNet can capture complex temporal dependencies and achieve superior prediction performance.

***4)* Effect of Adaptive Weighted Fusion Strategy:** One benefit of this adaptive weighted fusion strategy is measured by comparing with three variants.

- FSHLNet-w-add: Replaces adaptive weighted fusion with simple addition.

- FSHLNet-w-gate: Replaces adaptive weighted fusion with a gating mechanism.
- FSHLNet-w-attn: Replaces adaptive weighted fusion with an attention mechanism.

Table 4 Effect of adaptive weighted fusion strategy

Models		FSHLNet		FSHLNet-w-add		FSHLNet-w-gate		FSHLNet-w-attn	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.380**	**0.399**	0.406	0.389	0.400	0.383	0.400	0.400
	192	**0.413**	**0.424**	0.434	0.447	0.436	0.444	0.436	0.436
	336	**0.466**	**0.442**	0.445	0.464	0.475	0.450	0.475	0.450
	720	**0.464**	**0.465**	0.478	0.475	0.496	0.474	0.496	0.474
	Avg	**0.431**	**0.433**	0.441	0.444	0.452	0.438	0.452	0.440
ETTm1	96	**0.315**	**0.351**	0.357	0.321	0.321	0.357	0.321	0.357
	192	**0.362**	**0.376**	0.380	0.368	0.370	0.383	0.370	0.383
	336	**0.388**	**0.396**	0.403	0.390	0.397	0.400	0.397	0.400
	720	**0.450**	**0.433**	0.449	0.457	0.477	0.443	0.477	0.443
	Avg	**0.379**	**0.389**	0.397	0.384	0.391	0.396	0.391	0.396
Weather	96	**0.161**	**0.207**	0.211	0.166	0.164	0.212	0.164	0.212
	192	**0.210**	**0.251**	0.274	0.218	0.222	0.264	0.222	0.264
	336	**0.277**	**0.301**	0.315	0.287	0.291	0.311	0.291	0.311
	720	**0.362**	**0.353**	0.358	0.366	0.370	0.360	0.370	0.360
	Avg	**0.253**	**0.278**	0.289	0.259	0.262	0.282	0.262	0.282

Table 4 lists the experimental outcomes obtained by these variants on ETTh1, ETTm1 and Weather datasets. The adaptive weighted fusion strategy of FSHLNet outperforms other fusion methods across all prediction horizons. Among the three variants, FSHLNet-w-add yields the worst performance, as equal-weight summation fails to differentiate the importance of each feature branch and introduces noise into the fused representation. While the gating mechanism and attention mechanism enable selective feature integration, they still underperform the adaptive weighting scheme at longer prediction horizons.

4.4 Parameter Analysis

This study examines three critical hyperparameters: (1) the count of Top-k principal frequency components k; (2) the layer quantity of 1D-CNN in the time-domain feature extraction module L; and (3) the kernel size of these 1D-CNN layers K. In each experiment, we alter only the hyperparameter under study and keep the rest unchanged.

1) **Analysis of the Top-k selection threshold.** This threshold determines the boundary between principal and secondary frequency components, directly shaping the frequency ranges processed by the two hypergraphs. Fig. 8 depicts the experimental outcomes obtained by FSHLNet over the ETTm1 dataset, with k varying from 5 to 15.

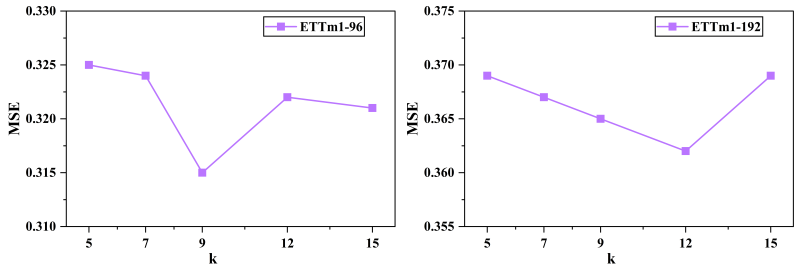


Fig. 8 Analysis of the Top-k selection threshold

The results reveal a U-shaped trend: model performance first improves and then degrades as k increases. When k is too small, many principal frequency components critical for capturing dominant periodic patterns are assigned to the secondary hypergraph, limiting the model's capacity to capture core spectral features. When k is too large, certain secondary frequency components are admitted into the principal hypergraph, introducing uninformative interference that degrades prediction accuracy. On ETTm1, the best-performing k is 9 for the 96-step horizon and 12 for the 192-step horizon.

2) **Analysis of the layer quantity of 1D-CNN.** We examine the impact of the layer quantity of 1D-CNN in the time-domain feature extraction module by evaluating configurations with $L \in \{1, 2, 3, 4\}$. The corresponding results are reported in Table 5. A consistent pattern emerges across all three datasets and prediction horizons: as the layer quantity grows from 1 to 3, the prediction error drops monotonically, while further deepening the network to 4 layers leads to a clear performance degradation. This finding indicates that a three-layer 1D-CNN strikes an appropriate balance between model expressiveness and generalization, allowing the time-domain branch to effectively capture local patterns at multiple scales while preventing underfitting and overfitting.

3) **Analysis of the 1D-CNN kernel size.** We further investigate the effect of kernel size configurations by evaluating six distinct settings: three homogeneous configurations $[3, 3, 3]$ (K1), $[5, 5, 5]$ (K2), $[7, 7, 7]$ (K3), and three heterogeneous ones $[3, 5, 7]$ (K4), $[7, 5, 3]$ (K5), $[3, 7, 5]$ (K6). Detailed quantitative outcomes are listed in Table 6.

The results reveal two key insights. First, kernel diversity across layers is more critical than the absolute kernel size: heterogeneous configurations (K4, K5, K6) consistently outperform homogeneous ones (K1, K2, K3), as varying kernel sizes equip different layers with complementary receptive fields for capturing multi-scale local patterns. Second, the direction of kernel size progression matters significantly: the increasing scheme K4 performs best, followed by K5 and K6, which exhibit comparable performance. This is because small kernels in shallow layers preserve fine-grained temporal details, while large kernels in deeper layers aggregate them into higher-level representations. Disrupting this order, whether by decreasing (K5) or non-monotonic (K6) progression, introduces coarse-grained aggregation prematurely, leading to irreversible loss of fine-grained information. We therefore adopt K4 as the default configuration.

Table 5 Analysis of the layer quantity of CNN

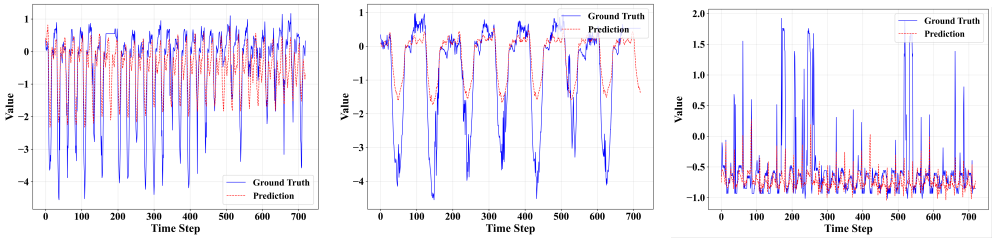
Methods		L1		L2		L3		L4	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.381	**0.392**	0.385	0.396	**0.380**	0.399	0.382	0.396
	192	0.557	0.484	0.431	0.426	**0.413**	**0.424**	0.448	0.435
	336	0.478	0.446	0.472	0.444	**0.466**	**0.442**	0.484	0.447
	720	0.512	0.476	0.523	0.503	**0.464**	**0.465**	0.484	0.469
	Avg	0.482	0.450	0.453	0.442	**0.431**	**0.433**	0.450	0.437
ETTm1	96	0.325	0.358	0.333	0.367	0.319	**0.354**	**0.312**	0.356
	192	0.367	0.377	0.365	0.377	**0.363**	**0.375**	0.366	0.379
	336	0.397	0.399	0.393	0.396	**0.389**	**0.396**	0.398	0.398
	720	0.460	0.431	0.459	0.439	**0.450**	**0.431**	0.490	0.452
	Avg	0.387	0.391	0.386	0.395	**0.380**	**0.389**	0.392	0.396
ETTm2	96	0.173	0.253	0.172	0.252	**0.172**	**0.251**	0.179	0.257
	192	0.237	0.294	0.243	0.299	**0.234**	**0.293**	0.235	0.294
	336	0.301	0.335	0.301	0.335	**0.300**	**0.334**	0.303	0.335
	720	0.399	0.392	0.403	0.392	**0.396**	**0.390**	0.404	0.393
	Avg	0.278	0.319	0.280	0.320	**0.276**	**0.317**	0.280	0.320

Table 6 Analysis of the CNN kernel size

Methods		K1		K2		K3		K4		K5		K6	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.387	0.399	0.384	**0.395**	0.383	0.396	**0.380**	0.399	0.391	0.399	0.386	0.398
	192	0.441	0.428	0.424	0.426	0.453	0.438	**0.413**	**0.424**	0.423	0.428	0.437	0.430
	336	0.478	0.446	0.471	0.443	**0.462**	0.446	0.466	**0.442**	0.498	0.456	0.473	0.444
	720	0.505	0.487	0.477	0.473	0.927	0.630	**0.464**	**0.465**	0.479	0.466	0.485	0.480
	Avg	0.453	0.439	0.439	0.434	0.556	0.479	**0.431**	**0.433**	0.448	0.437	0.445	0.439
ETTm1	96	0.331	0.361	0.323	0.358	0.329	0.361	**0.319**	**0.354**	0.319	0.355	0.324	0.358
	192	0.366	0.377	0.369	0.381	0.368	0.379	**0.363**	**0.375**	0.369	0.380	0.364	0.377
	336	0.396	0.398	0.405	0.403	0.401	0.402	**0.389**	**0.396**	0.390	0.396	0.394	0.402
	720	0.454	0.434	0.451	0.431	0.460	0.437	**0.450**	**0.431**	0.458	0.437	0.451	0.433
	Avg	0.387	0.393	0.387	0.393	0.390	0.395	**0.380**	**0.389**	0.384	0.392	0.383	0.393
ETTm2	96	0.180	0.259	0.175	0.254	0.175	0.254	**0.172**	**0.251**	0.176	0.255	0.171	0.252
	192	0.244	0.299	0.241	0.296	0.240	0.295	**0.234**	**0.293**	0.236	0.294	0.236	0.294
	336	0.303	0.335	0.302	0.335	0.302	0.335	**0.300**	**0.334**	0.301	0.334	0.302	0.335
	720	0.406	0.395	0.401	0.391	0.406	0.398	**0.396**	**0.390**	0.398	0.393	0.403	0.393
	Avg	0.283	0.322	0.280	0.319	0.281	0.321	**0.276**	**0.317**	0.278	0.319	0.278	0.319

4.5 Visualization Analysis

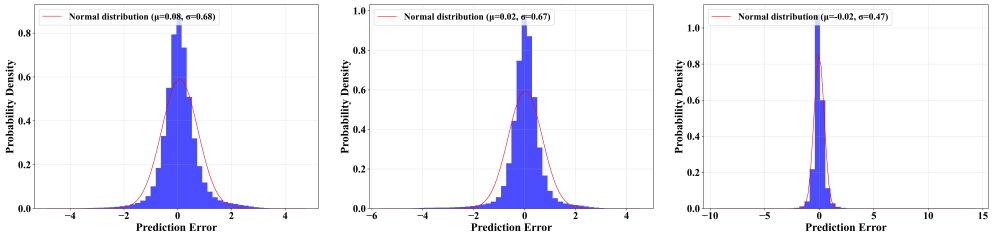
For an intuitive evaluation of FSHLNet's forecasting capability, the 720-step prediction results on ETTh1, ETTm1, and Electricity datasets are visualized in Fig. 9. These observations show that FSHLNet accurately captures the periodic patterns, fine-grained fluctuations, abrupt changes, and global trends of the time series. The predicted curves are highly consistent with the ground truth, with only minor deviations at extreme troughs and peak values, verifying the strong generalization capability of FSHLNet across diverse forecasting scenarios.



(a) (b) (c)

Fig. 9 Prediction vs. ground truth of FSHLNet for 720-step forecasting on the ETTh1, ETTm1, and Electricity datasets

To further evaluate the prediction reliability and error characteristics, the probability distribution histograms of prediction errors are plotted on the same datasets, with a fitted normal distribution curve shown alongside. As presented in Fig. 10, the prediction errors of FSHLNet exhibit small mean and standard deviation values, and the distribution closely approximates a normal distribution with gradually narrowing tails and a very low probability of large errors. These results indicate that the model produces unbiased and stable predictions with no systematic bias, confirming its reliability and robustness.



(a) (b) (c)

Fig. 10 Prediction error distribution of FSHLNet on the ETTh1, ETTm1, and Electricity datasets, with fitted normal curves

5 Conclusion

This work develops FSHLNet, a Frequency-Separated Hypergraph Learning Network to address multivariate long-term time series forecasting. FSHLNet separates the frequency spectrum into principal and secondary components, constructs dedicated hypergraphs for each to capture high-order group interactions, and enables cross-group collaborative modeling via cross-hypergraph hyperedge attention. A time-domain branch and adaptive weighted fusion module further complement the frequency-domain representations with local temporal patterns. Experiments on six real-world benchmark datasets yield two key findings. First, FSHLNet achieves superior forecasting accuracy, with particularly pronounced advantages in long-range prediction. Second, ablation studies demonstrate the indispensability of all core modules, with the dual-hypergraph architecture providing the largest performance gain. Future work may explore dynamic hypergraph construction for enhanced adaptability under distribution shifts and improved generalization under highly non-stationary temporal variations.

Statements and Declarations

Funding

This work was supported by the Science and Technology Research Project of Henan Province under Grant 252102210148.

Author contributions

J.G. conceptualized the study, developed the methodology, supervised the work, and reviewed and edited the manuscript. Y.X. developed the methodology and software, performed the formal analysis and validation, prepared the visualizations, and wrote the original manuscript draft. Y.D. conducted the investigation and contributed to the methodology. H.Y. performed formal analysis and validation. B.D. provided resources and curated the data. M.X. conceptualized the study, supervised the work, and administered the project. All authors reviewed and approved the final manuscript.

References

Ariyo AA, Adewumi AO, Ayo CK (2014) Stock price prediction using the arima model. In: 2014 UKSim-AMSS 16th international conference on computer modelling and simulation, IEEE, pp 106–112

Bai S, Kolter JZ, Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:180301271

Bai S, Zhang F, Torr PH (2021) Hypergraph convolution and hypergraph attention. Pattern Recognition 110:107637

Benidis K, Rangapuram SS, Flunkert V, et al (2022) Deep learning for time series forecasting: Tutorial and literature survey. ACM Computing Surveys 55(6):1–36

Cai W, Liang Y, Liu X, et al (2024) Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In: Proceedings of the AAAI conference on artificial intelligence, pp 11141–11149

Cao D, Wang Y, Duan J, et al (2020) Spectral temporal graph neural network for multivariate time-series forecasting. Advances in neural information processing systems 33:17766–17778

Chen J, Shao Q, Chen D, et al (2025) Decoupling spatio-temporal prediction: When lightweight large models meet adaptive hypergraphs. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2, pp 167–178

Cheng M, Yang J, Pan T, et al (2025) Convtimenet: A deep hierarchical fully convolutional model for multivariate time series analysis. In: Companion Proceedings of the ACM on Web Conference 2025, pp 171–180

Cheong WS, Jiang LL, Ling JNS (2026) Assessing electricity demand forecasting with exogenous data in time series foundation models. arXiv preprint arXiv:260205390

Di D, Zou C, Feng Y, et al (2022) Generating hypergraph-based high-order representations of whole-slide histopathological images for survival prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence 45(5):5800–5815

Eldele E, Ragab M, Chen Z, et al (2024) Tslanet: Rethinking transformers for time series representation learning. arXiv preprint arXiv:240408472

Fan W, Zheng S, Yi X, et al (2022) Depts: Deep expansion learning for periodic time series forecasting. arXiv preprint arXiv:220307681

Feng Y, You H, Zhang Z, et al (2019) Hypergraph neural networks. In: Proceedings of the AAAI conference on artificial intelligence, pp 3558–3565

Gao Y, Zhang Z, Lin H, et al (2020) Hypergraph learning: Methods and practices. IEEE transactions on pattern analysis and machine intelligence 44(5):2548–2566

Gao Y, Feng Y, Ji S, et al (2022) Hgnn+: General hypergraph neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 45(3):3181–3199

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

Garza A, Challu C, Mergenthaler-Canseco M (2023) Timegpt-1. arXiv preprint arXiv:231003589

Hao X, Li J, Guo Y, et al (2021) Hypergraph neural network for skeleton-based action recognition. IEEE Transactions on Image Processing 30:2263–2275

He H, Zhang Q, Bai S, et al (2022) Catn: Cross attentive tree-aware network for multivariate time series forecasting. In: Proceedings of the AAAI conference on artificial intelligence, pp 4030–4038

He H, Zhang Q, Yi K, et al (2024) Distributional drift adaptation with temporal conditional variational autoencoder for multivariate time series forecasting. IEEE Transactions on Neural Networks and Learning Systems 36(4):7287–7301

Holt CC (2004) Forecasting seasonals and trends by exponentially weighted moving averages. International journal of forecasting 20(1):5–10

Hong X, Hu J, Xu T, et al (2024) Magnet: Multilevel dynamic wavelet graph neural network for multivariate time series classification. ACM Transactions on Knowledge Discovery from Data 19(1):1–22

Jiang Y, Yu W, Lee G, et al (2025) Timexl: Explainable multi-modal time series prediction with llm-in-the-loop. arXiv preprint arXiv:250301013

Jin M, Wang S, Ma L, et al (2024) Time-llm: Time series forecasting by reprogramming large language models, 2024. arXiv preprint arXiv:231001728

Li J, Dong D, Zheng M, et al (2025) High-precision mixed feature fusion network using hypergraph computation for cervical abnormal cell detection. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, pp 250–259

Li Z, Qi S, Li Y, et al (2023) Revisiting long-term time series forecasting: An investigation on linear mapping. arXiv preprint arXiv:230510721

Liu C, Xu Q, Miao H, et al (2025) Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 18780–18788

Liu Y, Hu T, Zhang H, et al (2023) itransformer: Inverted transformers are effective for time series forecasting. arXiv preprint arXiv:231006625

Luo X, Chen Z, Huang D, et al (2024) An unsupervised domain adaption method for fault diagnosis via multichannel variational hypergraph autoencoder. IEEE Transactions on Instrumentation and Measurement 73:1–16

Ma X, Ni Z, Xiao S, et al (2025) Timepro: Efficient multivariate long-term time series forecasting with variable-and time-aware hyper-state. arXiv preprint arXiv:250520774

Nong L, Wang J, Lin J, et al (2021) Hypergraph wavelet neural networks for 3d object classification. Neurocomputing 463:580–595

Shang Z, Chen L, Wu B, et al (2024) Ada-mshyper: adaptive multi-scale hypergraph transformer for time series forecasting. Advances in Neural Information Processing Systems 37:33310–33337

Simeunović J, Schubnel B, Alet PJ, et al (2021) Spatio-temporal graph neural networks for multi-site pv power forecasting. IEEE Transactions on Sustainable Energy 13(2):1210–1220

Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Advances in neural information processing systems 30

Woo G, Liu C, Sahoo D, et al (2023) Learning deep time-index models for time series forecasting. In: International Conference on Machine Learning, PMLR, pp 37217–37237

Wu H, Xu J, Wang J, et al (2021) Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in neural information processing systems 34:22419–22430

Wu H, Hu T, Liu Y, et al (2022) Timesnet: Temporal 2d-variation modeling for general time series analysis. arXiv preprint arXiv:221002186

Wu H, Zhou H, Long M, et al (2023) Interpretable weather forecasting for worldwide stations with a unified deep model. Nature Machine Intelligence 5(6):602–611

Xiang S, Cheng D, Shang C, et al (2022) Temporal and heterogeneous graph neural network for financial time series prediction. In: Proceedings of the 31st ACM international conference on information & knowledge management, pp 3584–3593

Yadati N, Nimishakavi M, Yadav P, et al (2019) Hypergcn: A new method for training graph convolutional networks on hypergraphs. Advances in neural information processing systems 32

Yan X, Shi Z, Sun Z, et al (2024) Multisensor fusion on hypergraph for fault diagnosis. IEEE Transactions on Industrial Informatics 20(8):10008–10018

Yi K, Zhang Q, Fan W, et al (2023a) Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. Advances in neural information processing systems 36:69638–69660

Yi K, Zhang Q, Fan W, et al (2023b) Frequency-domain mlps are more effective learners in time series forecasting. Advances in Neural Information Processing Systems 36:76656–76679

Yi K, Zhang Q, He H, et al (2024) Deep coupling network for multivariate time series forecasting. ACM Transactions on Information Systems 42(5):1–28

Yi K, Zhang Q, Fan W, et al (2025) A survey on deep learning based time series analysis with frequency transformation. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2, pp 6206–6215

Yu B, Yin H, Zhu Z (2017) Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:170904875

Zeng A, Chen M, Zhang L, et al (2023) Are transformers effective for time series forecasting? In: Proceedings of the AAAI conference on artificial intelligence, pp 11121–11128

Zhang GP (2003) Time series forecasting using a hybrid arima and neural network model. Neurocomputing 50:159–175

Zhang Y, Yan J (2023) Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: The eleventh international conference on learning representations

Zhou T, Ma Z, Wen Q, et al (2022a) Film: Frequency improved legendre memory model for long-term time series forecasting. Advances in neural information processing

systems 35:12677–12690

Zhou T, Ma Z, Wen Q, et al (2022b) Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: International conference on machine learning, PMLR, pp 27268–27286