Information Fusion 117 (2025) 102836

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

AutoGRN: An adaptive multi-channel graph recurrent joint optimization network with Copula-based dependency modeling for spatio-temporal fusion in electrical power systems

Haoyu Wang [a], Xihe Qiu [a,*], Yujie Xiong [a], Xiaoyu Tan [b,c]

[a] School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, China
[b] INF Technology (Shanghai) Co., Ltd., Shanghai, China
[c] National University of Singapore, Singapore

ARTICLE INFO

Keywords:
Spatio-temporal information fusion
Adaptive graph neural networks
Multivariate prediction
Automated spatial feature learning

ABSTRACT

Multi-sensor, multi-source information fusion presents significant challenges in complex real-world applications such as power consumption prediction, where existing methods often have limitations in capturing both spatio-temporal features and fully exploit complex relationships among multi-variate features simultaneously. In real-world scenarios, such as complex electrical power system settings, capturing both correlations is important, as spatio-temporal contains vital geographical information and complex inter-series relationships between features. To address these limitations, we propose **AutoGRN** for enhancing prediction accuracy and efficiency in multi-source spatio-temporal data fusion, with a focus on complex electrical power system settings. **AutoGRN** integrates a novel adaptive multi-channel attentive framework with copula-based dependency modeling, combining graph neural diffusion convolution and recurrent optimization. The framework automatically learns spatial features, capturing complex correlations among regions, while a sequence encoder extracts temporal patterns, ensuring the acquisition of time series characteristics such as seasonality and trends. High-dimensional spatio-temporal features are then fused through a specially designed multi-channel recurrent graph neural network, incorporating copula functions to model complex dependencies between variables. Extensive experiments on multiple real-world electricity consumption datasets demonstrate that **AutoGRN** achieves substantial advantages over state-of-the-art benchmarks in multi-variate prediction tasks, showcasing its potential for applications in various multi-sensor, multi-source fusion scenarios, particularly in complex systems requiring simultaneous analysis of spatial and temporal dynamics with intricate inter-variable dependencies. Code is available at https://github.com/AmbitYuki/AutoGRN.

1. Introduction

Multivariate time series prediction is an essential research domain with wide applications in fields such as meteorological forecasting [1], stock market analysis [2], traffic flow prediction [3], and medication [4,5]. However, accurately forecasting trends and cycles in multiple time series remains a challenging problem due to the complexity and high dimensionality of time series data [6]. Thus, various techniques have been explored to enhance predictive performance.

In real-world applications such as complex electrical power systems, a power station's performance heavily depends on its location and its relationship with other stations. Due to the interconnected nature of these systems, a station's electrical functionality and efficiency are influenced by its spatial context. Factors like energy demand, distribution capacity, and transmission losses hinge on the geographical and operational network of stations. Modeling the power system as a graph is well-suited due to its inherent topological connections. Hence, a thorough understanding of both spatial and temporal dynamics is essential for enhancing the power grid's overall performance and reliability, underscoring the importance of simultaneous spatio-temporal feature analysis.

Despite recent advancements in multivariate time series prediction, several critical challenges persist in the field, particularly when applied to complex systems like electrical power grids [7]. Current methodologies often struggle to simultaneously address the multifaceted nature of spatio-temporal data and the intricate relationships among multiple variables. Many existing approaches treat spatial and temporal features in isolation [8], failing to capture their interdependencies effectively. This separation leads to a loss of crucial information embedded in the

* Corresponding author.
E-mail address: qiuxihe1993@gmail.com (X. Qiu).

https://doi.org/10.1016/j.inffus.2024.102836
Received 30 September 2024; Received in revised form 13 November 2024; Accepted 25 November 2024
Available online 30 November 2024

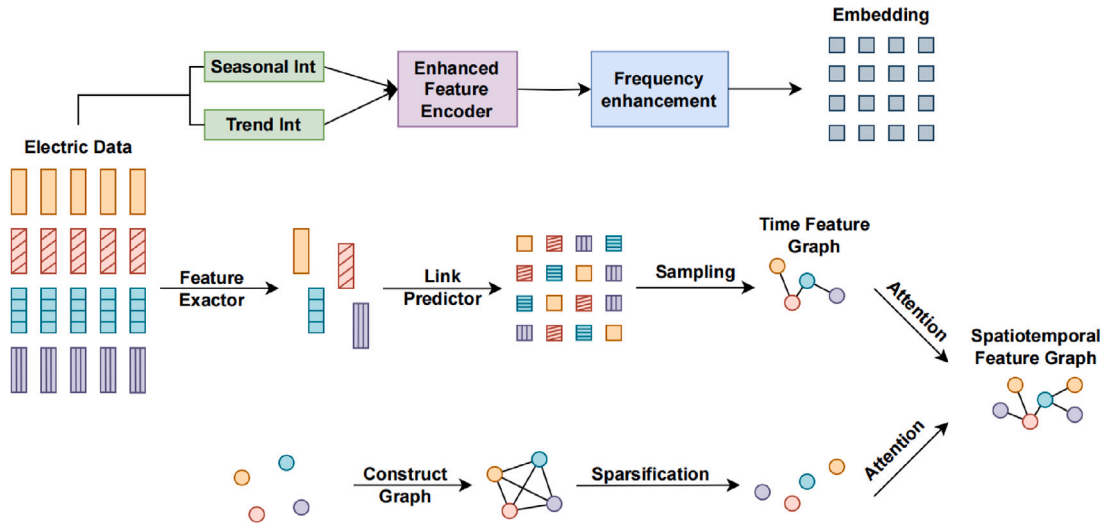


Fig. 1. In real-world power systems, a station's performance is heavily influenced by its spatial context and relationships with other interconnected stations. Our approach retains the original location data while learning detailed weights and adjacent node information to construct superior spatial datasets compared to manually defined graph structures.

spatio-temporal dynamics of power systems. Additionally, the high dimensionality and non-linear relationships inherent in multi-source data from power grids pose significant challenges for traditional models [9], often resulting in suboptimal predictions and decreased reliability [10]. Another major limitation is the inability of current methods to adapt to the dynamic nature of power systems, where the relationships between variables and the importance of different features can change over time [11]. This lack of adaptability can lead to model degradation and reduced accuracy in long-term forecasting. Furthermore, many existing approaches rely heavily on domain expertise for feature engineering [12], which can be both time-consuming and potentially biased, limiting the model's ability to discover novel patterns and relationships in the data [13]. The integration of external factors, such as weather conditions or economic indicators, which can significantly impact power consumption patterns [14], is often handled inadequately or omitted entirely in current models. Lastly, the computational efficiency of existing methods is frequently insufficient to handle the massive scales of data generated by modern power systems in real-time [15], hindering their practical application in operational settings.

Conventional neural network architectures exhibit certain limitations in handling multi-variate features [16]. They often struggle to fully capture complex nonlinear interdependencies and relationships among different variables; especially for multi-variate forecasting tasks, sophisticated nonlinear correlations, and dependencies may exist across features [17]. However, traditional networks are typically designed for single-variable inputs [18], thus showing inadequate multi-variate relationship learning.

Based on the aforementioned issues, we present a multi-channel spatio-temporal graph encoding structure learning framework (*i.e.*, **AutoGRN**), to improve prediction accuracy and efficiency in multivariate time-series forecasting in Fig. 2. AutoGRN extracts spatial subgraphs automatically and employs an improved sequence encoder to capture intricate data relationships. Through explicit multi-channel encoding of spatial, temporal, and inter-variable interactions, AutoGRN enables complex and complete modeling of relationships between variables and captures subtle dependencies. The main contributions can be concluded as follows:

1. Our model proficiently extracts spatial feature subgraphs, modeling geographical interrelationships in electrical power systems and constructing superior spatial datasets compared to manually defined graph structures. It retains the original location data while capturing station interconnections and learning detailed weights and adjacent node information in Fig. 1.
2. To address the challenge of extracting features from long time series with complex seasonal and periodic variations, we introduce a highly automated encoder. It efficiently extracts extended temporal features by adaptively learning time patterns and trends.
3. Our proposed multi-channel recurrent graph study network, integrating spatio-temporal features and leveraging a squeeze-and-excitation attention mechanism for channel recalibration, effectively combines feature extraction with deep neural networks in an end-to-end manner. The efficacy of our design is proven by comprehensive experiments and comparisons with state-of-the-art models, consistently demonstrating superior performance across various real-world electricity consumption datasets.

The remainder of the paper is organized as follows: Section 2 provides an overview of related work, while Section 3 presents the preliminaries and the proposed framework. In Section 4, we present the experimental findings, and we discuss the results of the ablation investigations. Finally, we conclude our work in Section 6.

2. Related work

2.1. Time series forecasting models

Time-series forecasting is a highly prominent research area. Traditional statistical models retain significance in time-series forecasting. Methods such as autoregressive moving average (ARMA) [19] and seasonal autoregressive moving average (SARMA) [20] are widely employed for modeling and forecasting time series data. Data-driven techniques, such as recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) [21], are also widely applied in time series forecasting. Researchers introduce an interactive memory learning approach for stereoscopic image super-resolution [22], enhancing the quality of 3D images by leveraging inter-view correlations. These models, leveraging memory units and gating mechanisms, adeptly capture long-term dependencies in time-series data. Recently, graph neural network-based methods [23,24] have emerged, utilizing graph structures to model relationships in time series data. These approaches represent time series data as graphs, employing graph neural networks for feature learning and prediction, thereby enhancing the capture of complex correlations in time series data.

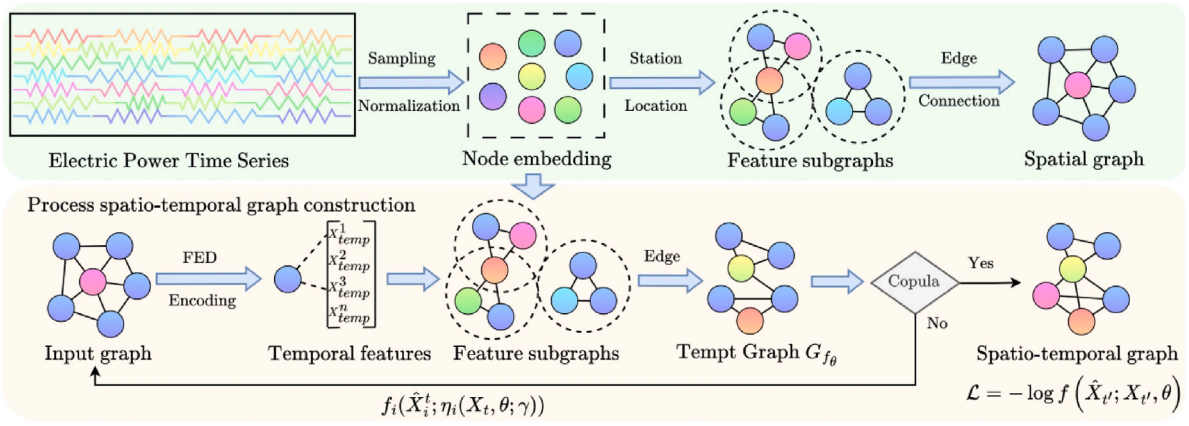


Fig. 2. The AutoGRN framework for multivariate time series forecasting in electrical power systems: (a) Spatial feature extraction: $G = (V, E, F)$ where V is the node set, E is the edge set, and F is the feature set. (b) Temporal sequence encoding: $X'_t = \text{Encoder}(X_t)$ where X_t is the input time series and X'_t is the encoded representation. (c) Multi-channel spatio-temporal network: $H_{t'} = U_{t'} \odot H_{t'-1} + (1 - U_{t'}) \odot C_{t'}$, where $U_{t'}$ and $H_{t'-1}$ are spatial and temporal hidden state, $\odot$ denotes element-wise multiplication. (d) Copula-based optimization: $\mathcal{L} = -\log f\left(\hat{X}_{t'};X_{t'},\theta\right)$.

2.2. Attention mechanism in time series forecasting

The Transformer has become popular for effectively capturing long-term dependencies in time series data [25]. Numerous studies have demonstrated the superiority of attention-based models in time series prediction tasks. Zeng et al. [26] encoded historical stock price sequences and used self-attention mechanisms to learn complex relationships between different time points for stock price prediction. Zhou et al. [27] applied a transformer model to weather forecasting, encoding historical weather data to learn spatio-temporal relationships between meteorological variables [28]. Zhu et al. [29] propose a lightweight image super-resolution method using an expectation–maximization attention mechanism, achieving improved performance with reduced computational complexity. Additionally, studies have explored transformer applications in diverse time series prediction tasks such as traffic flow [30] and energy load forecasting [31]. These findings collectively affirm the effectiveness of attention-based models in various domains of time series prediction.

2.3. Temporal sequence prediction with spatio-temporal networks

Graph neural networks (GNNs) are gaining attention in time series forecasting [32,33]. By modeling graph-structured relationships among time series data, GNNs effectively capture complex node dependencies, enhancing forecasting accuracy [34,35]. GNNs find application in traffic flow prediction, representing road networks as graphs and using traffic data as node features to learn spatio-temporal relationships [36]. In social media event prediction, GNNs construct networks of users and events as nodes to model interactions [37,38].

Spatio-temporal graph neural networks (ST-GNNs) have been widely adopted for modeling spatio-temporal data graphs [39] and capturing both spatial and temporal dependencies for enhanced time series prediction. By incorporating multidimensional spatial and temporal features, ST-GNNs demonstrate strong potential for forecasting applications, which have been applied to various domains [40,41]. Current research also aims to enhance representation learning in spatial, temporal, and spatio-temporal graphs through the development of graph convolution and attention mechanisms.

3. Methods

3.1. Transforming time series with spatial data into graphs

Converting time series data into graph structures can be achieved by constructing an adjacency matrix to represent connections between nodes, where each time step is treated as a node in the graph in Table 1. The adjacency matrix defines the relationships between the nodes, forming the time series into combinations of nodes and edges.

We collected large-scale datasets containing spatial location information, including geographic coordinates (*e.g.* latitudes and longitudes) as well as other position representations (*e.g.* grid coordinates) as shown in Fig. 3.

Distance matrices are computed to capture the distance or similarity between each pair of elements representing the nodes. Specifically, we adopt Euclidean distance matrices defined as:

$$d_{ij} = \| x_i - x_j \| \tag{1}$$

where x_i and x_j are the coordinates of the ith and jth power plants respectively, d_{ij} is their Euclidean distance, and $\| \cdot \|$ denotes the Euclidean norm.

Algorithm 1 Construction of spatial feature subgraphs

Require: Input spatial location data $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, where x_i represents the coordinates of the i-th power plant.
Ensure: Spatial feature subgraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ with node set $\mathcal{V}$, edge set $\mathcal{E}$, and feature set $\mathcal{F}$.
1: Compute pairwise Euclidean distances $d_{ij} = \|x_i - x_j\|$ between all pairs of locations (x_i, x_j).
2: **for** each pair (i, j) **do**
3: Construct adjacency matrix entry:
4: $A_{ij} = \begin{cases} 1, & \text{if } d_{ij} < \text{threshold} \\ 0, & \text{otherwise} \end{cases}$
5: **end for**
6: Compute weight matrix $W_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right)$ using a distance-based kernel.
7: Learn attention scores α_{qv} between distance graph g_q and discrete graph g_v using Eq. (5).
8: Obtain final integrated graph structure with weighted edges by multiplying attention scores with discrete graph.
9: **return** $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ representing the spatial feature subgraph.

$$A_{ij} = \begin{cases} 1, & \text{if } d_{ij} < \text{threshold} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

A is the adjacency matrix, with thresholded d_{ij} determining the connections in A_{ij}.

$$g_q, g_v \leftarrow W_{ij} = \exp\left(-\frac{\text{dist}\left(x_i, x_j\right)^2}{\sigma^2}\right) \tag{3}$$

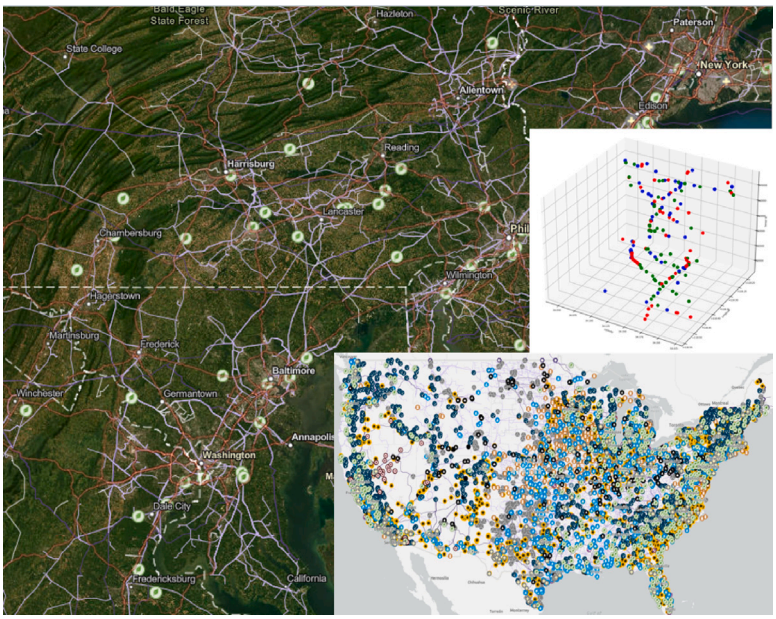


Fig. 3. Geospatial distribution of power plants showing connectivity based on proximity. Nodes represent power plants and edges indicate connections between plants based on distance thresholds, forming spatial graphs.

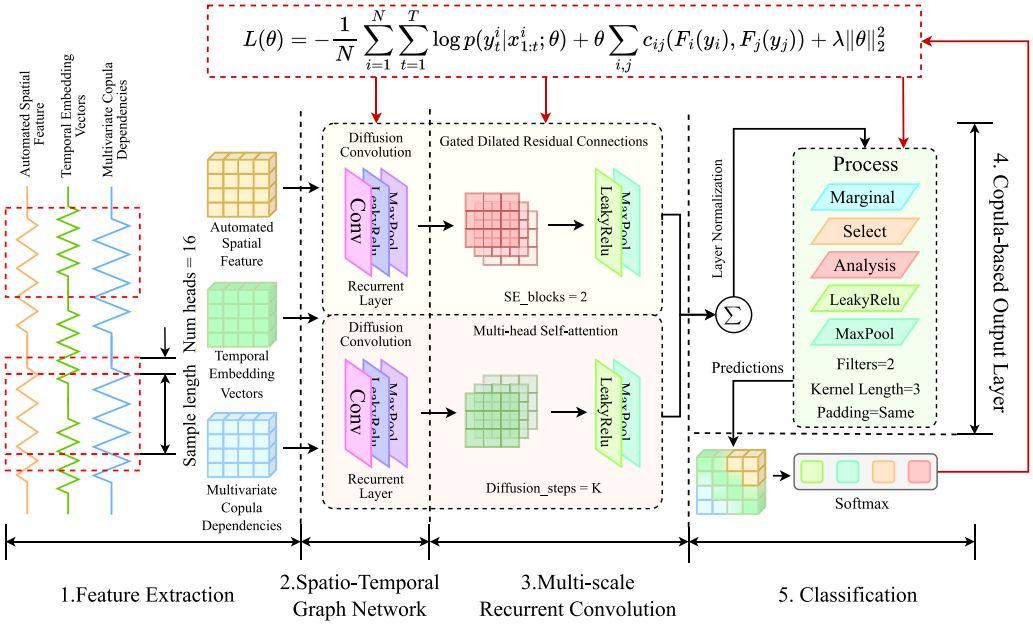


Fig. 4. Architectural overview of our Multi-Channel Graph Recurrent Joint Optimization Network with Copula-based Dependency. (a) Input data streams are processed through multiple parallel channels, each specialized for different aspects of the input. (b) Graph recurrent layers capture spatial–temporal dependencies across nodes. (c) The joint optimization module integrates information from all channels. (d) Copula-based dependency modeling enhances the capture of complex, non-linear relationships between variables. (e) The output layer provides optimized predictions or classifications.

We define a weight matrix W_{ij} used for constructing the distance graph. It computes the weight between power plants i and j as an exponential function of the squared distance. Here, σ is a parameter that influences the decay of the weight with increasing distance. Among them, $\leftarrow$ represents the assignment operation, while g_q and g_v represent the embedding of distance and discrete graphs, respectively.

The spatial location data is then combined with the adjacency matrices to generate feature subgraphs, where each location is regarded as a node, and the adjacency matrix specifies relationships between nodes [42]. This represents the spatial data as combinations of nodes and edges in graph structures, which as shown in Fig. 4.

$$score\left(g_q, g_v\right) = v_a^{\top}\tanh\left(W_1 g_q + W_2 g_v\right) \tag{4}$$

Among them, W_1 and W_2 are learnable weight matrices used for transforming graph embeddings. In this work, we learn a spatial-aware graph structure via attention mechanisms. The core idea is to assign a set of weight values that positively correlate with dependencies between elements dependency leads to larger weight. In particular, based on the spatial graph and learnable discrete graph, we calculate attention weights between them and multiply them with the discrete graph to obtain the final integrated structure.

$$\alpha_{qv} = \frac{\exp\left(\text{score}\left(g_q, g_v\right)\right)}{\sum_{v'=1}^{V}\exp\left(\text{score}\left(g_q, g_{v'}\right)\right)} \tag{5}$$

The computation of an attention score, denoted as a_{qv}, is achieved by comparing embeddings from two distinct sources—the distance graph g_q and the discrete graph g_v. We gain insights into the significance of connection weights by a crucial normalization step, ultimately resulting in the creation of a final embedding.

3.2. Long-term trend feature encoder

The automated encoder in AutoGRN is a sophisticated component designed to capture complex temporal patterns across multiple scales, combining a multi-scale decomposition approach with an attention mechanism to adaptively focus on the most relevant temporal features. The architecture begins with a multi-scale decomposition using Discrete Wavelet Transform (DWT) to decompose the input time series $X = \{x_1, \ldots, x_T\}$ into multiple frequency bands, resulting in wavelet coefficients $\{D_1, \ldots, D_L, A_L\}$, where D_i represents detail coefficients at level i, and A_L is the approximation coefficient at the deepest level L. Following this, scale-specific encoding is performed using separate LSTM encoders for each scale i, such that $h_i = \text{LSTM}_i(D_i)$. The multi-scale attention mechanism then computes attention weights for each scale using $\alpha_i = \text{softmax}(v^T \tanh(W_h h_i + b_h))$, allowing the model to combine scale-specific features as $h_{\text{multi}} = \sum_i \alpha_i \cdot h_i$.

To capture long-term trends, we apply approximation coefficients to yielding $h_{\text{trend}} = \text{LSTM}_{\text{trend}}(A_L)$. The final temporal representation is obtained by concatenating the multi-scale and trend features: $h_{\text{temporal}} = [h_{\text{multi}}; h_{\text{trend}}]$. This automated encoder structure enables AutoGRN to adaptively learn and focus on the most relevant temporal scales and long-term trends, thereby capturing complex seasonality and periodic patterns in power consumption data with remarkable efficiency.

According to FEDformer [28], we use discrete Fourier transform (DFT) to convert the input signal from the time domain to the frequency domain. Then, only M randomly selected frequency components are retained before transforming the signal back to the time domain. This approach reduces computational complexity and extracts key frequency domain information from the signal.

$$\widetilde{Q} = Select(F(q))FEB - f(q) = F^{-1}(Padding(\widetilde{Q} \odot R)), \tag{6}$$

where F represents the Fourier transform, F^{-1} represents the inverse transform, $\widetilde{Q} \in \mathbb{C}^{M \times D}$ denotes the randomly selected M frequency components that are retained, $R \in \mathbb{C}^{D \times D \times M}$ represents the learnable parameters, and $\odot$ as the Hadamard product (element-wise multiplication).

We apply the standard Transformer attention mechanism in the frequency domain. Specifically, the queries, keys, and values are transformed using DFT in the frequency domain to perform attention calculations. The results are then transformed back to the time domain. This approach also helps in reducing computational complexity.

$$FEA - f(q, k, v) = F^{-1}(Padding(\sigma(\widetilde{Q} \cdot \widetilde{K}^{\top}) \cdot \widetilde{V})) \tag{7}$$

In the given equations, $\widetilde{Q}$, $\widetilde{K}$, and $\widetilde{V}$ are derived by selecting the desired frequency components from the Fourier-transformed signals $F(q)$, $F(k)$, and $F(v)$, respectively, using the function Select(). The function FEA-f(q, k, v) represents the computation of frequency domain attention, where the frequency domain queries $\widetilde{Q}$, keys $\widetilde{K}$, and values $\widetilde{V}$ are utilized.

Lemma 1 (*Frequency-Enhanced Attention (FEA) Module*). *Let $X = \{x_1, x_2, \ldots, x_T\} \in \mathbb{R}^{T \times d}$ be a time series of length T with d features. The Frequency-Enhanced Attention (FEA) module operates as follows:*

(1) First, apply the Discrete Fourier Transform (DFT) to X:

$$\mathcal{F}(X) = \hat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_T\} \in \mathbb{C}^{T \times d}, \tag{8}$$

where $\hat{x}_k = \sum_{t=1}^{T} x_t e^{-i2\pi kt/T}$ for $k = 1, 2, \ldots, T$.

(2) Select M frequency components:

$$\tilde{X} = Select(\hat{X}) \in \mathbb{C}^{M \times d} \tag{9}$$

(3) Apply attention mechanism in the frequency domain:

$$Attention(\tilde{Q}, \tilde{K}, \tilde{V}) = softmax\left(\frac{\tilde{Q}\tilde{K}^T}{\sqrt{d_k}}\right)\tilde{V}, \tag{10}$$

where $\tilde{Q}, \tilde{K}, \tilde{V}$ are frequency domain representations of queries, keys, and values respectively, and d_k is the dimension of the keys.

(4) Transform back to time domain:

$$FEA(X) = \mathcal{F}^{-1}(Padding(Attention(\tilde{Q}, \tilde{K}, \tilde{V}))), \tag{11}$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform and Padding restores the original dimensions. The FEA module enhances traditional attention mechanisms by operating in the frequency domain, allowing it to capture both local and global temporal dependencies more effectively.

FEA inherently favors low-frequency components, which often correspond to long-term trends in time series data. By selecting only M frequency components, FEA reduces computational complexity from $O(T^2)$ to $O(M^2)$, where typically $M \ll T$. The frequency domain operation allows FEA to model dependencies across large time gaps more easily than time-domain attention.

Let C_{std} and C_{FEA} be the computational costs of standard attention and FEA respectively.

$$C_{\text{std}} = O(T^2 d) \tag{12}$$

$$C_{\text{FEA}} = O(T \log T + M^2 d + T \log T) = O(M^2 d + T \log T), \tag{13}$$

When $M^2 d < T \log T$, FEA is more efficient than standard attention. Furthermore, the error introduced by frequency component selection is bounded:

Lemma 2 (*Trade-off between Efficiency and Accuracy*). *Let ϵ be the error introduced by selecting M out of T frequency components. Then,*

$$\epsilon \leq C\sqrt{\frac{T - M}{T}} \tag{14}$$

where C is a constant depending on the spectral properties of the input signal. This theorem ensures that the approximation error decreases as M approaches T, allowing for a trade-off between computational efficiency and accuracy.

The attention scores are computed as the dot product between $\widetilde{Q}$ and the transposed $\widetilde{K}$, followed by the application of a sigmoid activation function $\sigma()$. Subsequently, the resulting attention scores are element-wise multiplied with $\widetilde{V}$. Finally, as depicted in Fig. 5, the inverse Fourier transform F^{-1} is applied to the result, after dimension padding using the Padding() function, to yield the output of the frequency domain attention mechanism.

$$X_l^{en} = Encoder(X_{l-1}^{en}) \tag{15}$$

Algorithm 2 Long-term trend feature encoder

Require: Input multivariate time series $\mathbf{q} = \{q_1, q_2, \ldots, q_T\}$ of length T.
Ensure: Encoded temporal features capturing long-term trends $\mathbf{H} = \{h_1, h_2, \ldots, h_T\}$.

```
Apply discrete Fourier transform: F(q) = {q̃_1, q̃_2, ..., q̃_T}
Randomly select M frequency components: Q̃ = Select(F(q))
Transform back to time domain with learnable parameters R:
    H^FEB = F^{-1}(Padding(Q̃ ⊙ R))
for l = 1 to L (number of encoder layers) do
    X_l^en = Encoder(X_{l-1}^en) {Update encoder representations}
    Q^l, K^l, V^l = Select(F(X_l^en)) {Select frequency components}
    H_l^FEA = FEA-f(Q^l, K^l, V^l) {Frequency-enhanced attention, Eq. (7)}
    X_l^de, T_l^de = Decoder(X_{l-1}^de, T_{l-1}^de) {Update decoder representations}
end for
return H = T_L^de {Final encoded temporal features}
```

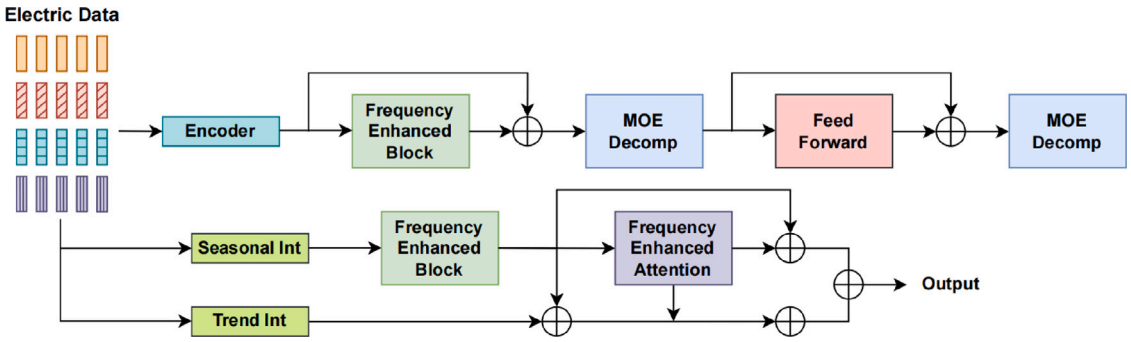


Fig. 5. Overview of the temporal sequence extraction module in the time-spatial feature composition framework. The input time series is transformed into the frequency domain using discrete Fourier transform and key components are selected. Attention operations are performed in the frequency domain before transforming back to the time domain.

The mixture of experts decomposition (MOEDecomp) module is used to extract trend and seasonal components from time series data. It consists of a set of average pooling layers with different sizes and a set of data-dependent weights to mix the extracted trend components.

$$X_l^{de}, T_l^{de} = Decoder(X_{l-1}^{de}, T_{l-1}^{de}), \tag{16}$$

where X_l^{de} and T_l^{de} are the decoder input and trend output at layer l respectively, and $Decoder(\cdot)$ represents the decoder function. We incorporate an encoder–decoder architecture into our framework. The encoder comprises several layers of frequency-enhanced block (FEB) and MOEDecomp. Meanwhile, the decoder consists of multiple layers of FEB, frequency-enhanced attention (FEA), and $MOEDecomp$.

By randomly selecting fixed frequency components, it reduces the computational complexity to linear complexity $O(L)$, where L represents the number of selected frequency components. This random selection helps in achieving computational efficiency while preserving important frequency information.

$$\begin{aligned} S_{\text{de}}^{l,1}, \mathcal{T}_{\text{de}}^{l,1} &= MOEDecomp\left(FEB\left(\mathcal{X}_{\text{de}}^{l-1}\right) + \mathcal{X}_{\text{de}}^{l-1}\right) \\ S_{\text{de}}^{l,2}, \mathcal{T}_{\text{de}}^{l,2} &= MOEDecomp\left(FEA\left(S_{\text{de}}^{l,1}, \mathcal{X}_{\text{en}}^{N}\right) + S_{\text{de}}^{l,1}\right) \\ S_{\text{de}}^{l,3}, \mathcal{T}_{\text{de}}^{l,3} &= MOEDecomp\left(Feed\,Forward\left(S_{\text{de}}^{l,2}\right) + S_{\text{de}}^{l,2}\right), \end{aligned} \tag{17}$$

where $MOEDecomp(\cdot)$ is the mixture of experts decomposition, $FEB(\cdot)$ is the frequency-enhanced block, $FEA(\cdot)$ is the frequency-enhanced attention, and $Feed\,Forward(\cdot)$ is a feed-forward neural network.

3.3. Multi-scale modeling with spatio-temporal encoders and multi-view convolution

Spatio-temporal encoders have strong multi-scale modeling capabilities for capturing dependencies across different time steps and spatial locations, and extracting spatio-temporal features at various granularities [43]. The encoded representations can better capture important information from the raw spatio-temporal data.

$$X_{:,p} \star_{\mathcal{G}} f_\theta = \sum_{k=0}^{K-1} \left(\theta_{k,1} \left(D_O^{-1} W\right)^k + \theta_{k,2} \left(D_I^{-1} W^\top\right)^k\right) \tag{18}$$

In this equation, $X_{:,p}$ is a feature or node vector in the input data. $\star_{\mathcal{G}}$ is the graph convolution operation. f_θ is a parameterized convolutional filter, learned during network training. D_O and D_I are output and input degree matrices used for graph data normalization. K represents the recurrent convolution order, determining operation depth. $\theta_{k,1}$ and $\theta_{k,2}$ are learned parameters for calibrating convolution on output and input graph data.

For integrating with the spatial–temporal network, we add the squeeze-and-excitation Block after the spatio-temporal convolutions to implement channel-wise attention.

$$R_{t'} = \text{sigmoid}\left(W_R \star_A \left[X_{t'} \parallel H_{t'-1}\right] + b_R\right) \tag{19}$$

SE Block is a self-attention mechanism focused on channel-wise recalibration, often referred to as a channel re-calibration technique [44].

$$C_{t'} = \tanh\left(W_C \star_A \left[X_{t'} \parallel \left(R_{t'} \odot H_{t'-1}\right) + b_C\right]\right) \tag{20}$$

In essence, it allows each channel's magnitude to be adjusted based on the information from other channels as shown in Fig. 6.

SE initially employs global average pooling (GAP) to gather global information for each channel. Subsequently, it utilizes two fully connected networks for channel compression and recovery.

$$U_{t'} = \text{sigmoid}\left(W_U \star_A \left[X_{t'} \parallel H_{t'-1}\right] + b_U\right) \tag{21}$$

After compression, the number of channels is reduced to a ratio of r times the input channels, and then it is expanded back to the original number of input channels. The final step in the SE block is to apply the sigmoid function to the last feature vector and multiply it with the input.

$$\begin{aligned} f_{ch} &= GAP(F) \\ \alpha_{ch} &= \sigma\left(FC_{\frac{s}{r}\to c}\left(\text{Re}\,LU\left(FC_{c\to\frac{c}{r}}\left(f_{ch}\right)\right)\right)\right) \\ SA(F) &= F \times \alpha_{ch} \end{aligned} \tag{22}$$

Here, σ represents the sigmoid function, GAP denotes the global average pooling function, r signifies the compression ratio.

$$H_{t'} = U_{t'} \odot H_{t'-1} + \left(1 - U_{t'}\right) \odot C_{t'}, \tag{23}$$

where $U_{t'}$ and $H_{t'-1}$ are spatial and temporal hidden states respectively, $\odot$ denotes element-wise multiplication, and α_s, α_t, and α_{st} are learnable parameters determining the contribution of each component. The attention weights are computed using a softmax operation $[\alpha_s, \alpha_t, \alpha_{st}] = \text{softmax}(W[H_s \| H_t \| (H_s \odot H_t)])$, where W is a learnable weight matrix and $\|$ denotes concatenation.

The recurrent process, which simultaneously processes input and updates hidden states across multiple time series, employs graph convolution to substitute traditional weight matrix multiplication. While various existing architectural solutions exist for this purpose, we opt for the diffusion convolutional GRU as defined in DCRNN [45] due to its specialized design for directed graphs.

$$W_Q \star_A Y = \sum_{k=0}^{K} \left(w_{k,1}^Q \left(D_O^{-1} A\right)^k + w_{k,2}^Q \left(D_I^{-1} A^T\right)^k\right) \tag{24}$$

By integrating lightweight but effective attention blocks, the model can achieve improved feature learning and discrimination capabilities. Then, we incorporate the encoders into the spatial–temporal network by applying the transformer as the encoder separately in the temporal and spatial domains.

$$\mathcal{T}_{\text{de}}^{l} = \mathcal{T}_{\text{de}}^{l-1} + \mathcal{W}_{l,1} \cdot \mathcal{T}_{\text{de}}^{l,1} + \mathcal{W}_{l,2} \cdot \mathcal{T}_{\text{de}}^{l,2} + \mathcal{W}_{l,3} \cdot \mathcal{T}_{\text{de}}^{l,3} \tag{25}$$

Table 1
Summary of key variables and symbols.

Variable/Symbol	Description	Equation
x_i, x_j	Coordinates of the ith and jth power plants	(1)
d_{ij}	Euclidean distance between x_i and x_j	(1)
A_{ij}	Adjacency matrix element	(2)
g_q, g_v	Discrete and distance graph embeddings	(3)
σ	Parameter influencing the weight decay with increasing distance	(3)
α_{qv}	Attention scores	(5)
$\mathbf{q}$	Input multivariate time series	(6)
$\mathbf{F}(\mathbf{q})$	Discrete Fourier transform of $\mathbf{q}$	(6)
$\mathbf{H}^{FEB}$	Output of Frequency-Enhanced Block	(6)
$\mathbf{H}_l^{FEA}$	Output of Frequency-Enhanced Attention at layer l	(7)
$\tilde{Q}, \tilde{K}, \tilde{V}$	Frequency components of queries, keys and values	(7)
$X^{\text{en}}l$	Encoded input at layer l	(15)
$X^{\text{de}}l, T^{\text{de}}l$	Decoder input and trend output at layer l	(16)
$\theta k, i$	Learned convolution parameters	(18)
$f\theta$	Convolutional filter	(18)
$\mathbf{X}_{:,p}$	Feature or node vector in the input data	(18)
D_O, D_I	Output and input degree matrices	(18)
R_t'	Reset gate	(19)
C_t'	Candidate hidden state	(20)
U_t'	Update gate	(21)
$\mathbf{f}_{ch}$	Global average pooled feature vector	(22)
α_{ch}	Channel attention weights	(22)
$\mathbf{SA}(\mathbf{F})$	Output of Squeeze-and-Excitation block	(22)
H_t'	Hidden state	(23)
wk, i^Q	Learned parameters	(24)
ℓ	Negative log-likelihood loss	(28)

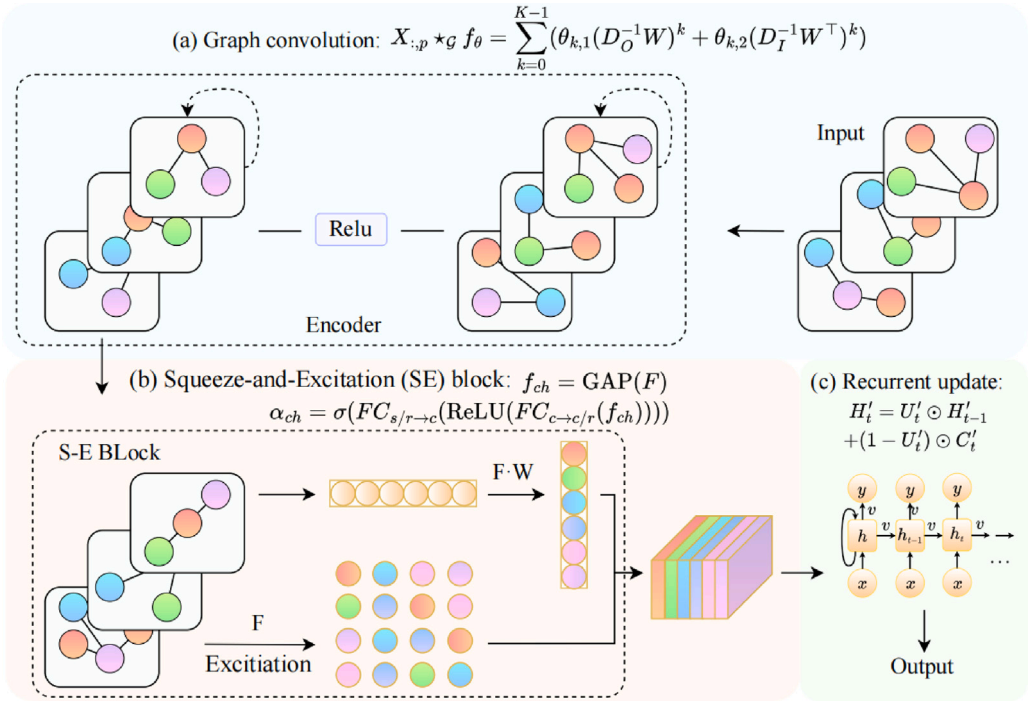


Fig. 6. The proposed diffusion recurrent convolution network integrating graph convolution and attention mechanisms for modeling complex spatio-temporal data interactions. Key components include: (a) Graph convolution: $X_{:,p} \star_G f_\theta = \sum_{k=0}^{K-1}(\theta_{k,1}(D_O^{-1}W)^k + \theta_{k,2}(D_I^{-1}W^\top)^k)$ (b) Squeeze-and-Excitation (SE) block: $f_{ch} = \text{GAP}(F)$, $\alpha_{ch} = \sigma(FC_{s/r\to c}(\text{ReLU}(FC_{c\to c/r}(f_{ch}))))$ (c) Recurrent update: $H_t' = U_t' \odot H_{t-1}' + (1 - U_t') \odot C_t'$.

where T_l^{de} is the final trend output at layer l, and $\alpha_{l,1}$, $\alpha_{l,2}$, $\alpha_{l,3}$ are learnable parameters that control the contribution of each decomposition step. The squeeze-and-excitation attention mechanism allows AutoGRN to adaptively recalibrate channel-wise feature responses, emphasizing informative features and suppressing less useful ones. This enhances the model's ability to capture complex interactions between spatial and temporal features in the power consumption data.

Lemma 3 (*Lipschitz Continuity of Graph Convolution*). *Let f_θ be a graph convolutional filter with parameters θ. Then, f_θ is Lipschitz continuous with respect to its input, i.e., there exists a constant $L > 0$ such that for any two input signals x and y on the graph:*

$$\| f_\theta(x) - f_\theta(y) \| \leq L \| x - y \| \tag{26}$$

Algorithm 3 Spatio-temporal graph convolution

Require: Input node features $\mathbf{X} \in \mathbb{R}^{N \times F}$, adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, diffusion order K.
Ensure: Convolved node representations $\mathbf{X}' \in \mathbb{R}^{N \times F'}$.
1: Compute degree matrices $\mathbf{D}O = \text{diag}(\mathbf{A}\mathbf{1}N)$ and $\mathbf{D}I = \text{diag}(\mathbf{A}^T\mathbf{1}N)$.
2: **for** $p = 1$ to F' **do**
3: $\mathbf{X}:, p' = \mathbf{0}N \times 1$ {Initialize output feature}
4: **for** $k = 0$ to $K - 1$ **do**
5: $\mathbf{X}:, p' \; += \; \theta k, 1 \left((\mathbf{D}O^{-1}\mathbf{A})^k \mathbf{X}\right) + \theta k, 2 \left((\mathbf{D}_I^{-1}\mathbf{A}^T)^k \mathbf{X}\right)$ {Diffusion convolution, Eq. (18)}
6: **end for**
7: **end for**
8: Obtain spatial features $\mathbf{X}_s = \mathbf{X}'$.
9: Compute temporal features $\mathbf{X}_t$ using long-term trend encoder (Section 3.2).
10: Feed $\mathbf{X}_s$ and $\mathbf{X}_t$ into multi-channel spatio-temporal network (Section 3.4).
11: **return** Convolved node representations $\mathbf{X}'$ encoding spatio-temporal information.

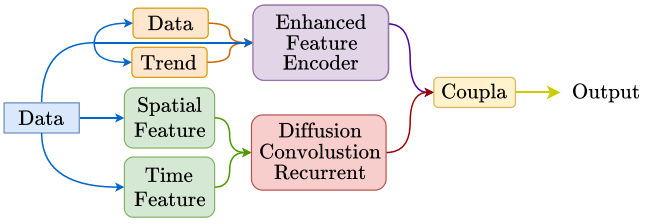


Fig. 7. Framework overview of the Copula-based optimization module for modeling nonlinear dependencies between variables. The Copula density term captures multivariate distributions, while the edge density models relationships between predicted and actual values.

This lemma ensures the stability of our graph convolution operations, which is crucial for the robustness of our spatial feature extraction process.

3.4. Modeling nonlinear dependencies with copula

Using linear correlation assumptions may fail to accurately capture nonlinear relationships in time series data. Copula functions allow more flexible modeling of dependencies between multivariate variables, permitting nonlinear relationships [46]. For extreme events or tail risks in time series, which can significantly impact forecasting, Copulas better captures tail dependencies between variables for improved accuracy under extremes as shown in Fig. 7.

$$\ell = \sum_{ij} -A_{ij}^a \log \theta_{ij} - \left(1 - A_{ij}^a\right) \log\left(1 - \theta_{ij}\right), \tag{27}$$

where is the cross-entropy loss, A_{ij}^a represents the actual adjacency matrix, and θ_{ij} is the predicted edge probability between nodes i and j.

In certain scenarios, temporal series may include pre-defined graphs. However, when the explicit structure is unknown, neighborhood graphs can serve as reasonable surrogates for this information. This approach effectively addresses the limitations of directly imposing the ℓ constraint on the graph, particularly since the graph is not the primary variable optimized.

Consequently, the use of cross-entropy between θ and the neighborhood graph acts as a regularization technique aimed at enhancing the quality of the graph.

$$\begin{aligned} \mathcal{L} &= -\log f\left(\hat{X}_{t'}; X_{t'}, \theta\right) \\ &= -\log c(u;\theta) - \sum_{i=1}^{m} \log f_i\left(\hat{X}_{t'}^i; \eta_i\left(X_{t'}, \theta; \gamma\right)\right), \end{aligned} \tag{28}$$

where $\mathcal{L}$ is the negative log-likelihood loss, $f(\cdot)$ is the joint probability density function, $c(u;\theta)$ is the copula density, $f_i(\cdot)$ is the marginal density for the ith variable, $\hat{X}_t'$ and X_t' are the predicted and actual values at time t respectively, θ and γ are model parameters, and $\eta_i(\cdot)$ is a function that maps the input to the parameters of the ith marginal distribution.The loss function, denoted as $\mathcal{L}$, which is the negative log-likelihood, can be expressed as the product of the Copula density, represented as c(u; θ), and the edge density, which encapsulates the relationship between the predicted values $\hat{X}_{t'}$ and the actual values $X_{t'}$ with respect to the parameter θ.

In this equation, the first term, $-\log c(u;\theta)$, quantifies the negative log-likelihood of the Copula density, while the second term, $-\sum_{i=1}^{m} \log f_i$ represents the negative log-likelihood of the edge density, where m denotes the number of variables considered in the joint density estimation. This loss function $\left(\hat{X}_t^i; \eta_i\left(X_{t'}, \theta; \gamma\right)\right)$ aims to capture the dissimilarity between the observed and predicted values, taking into account the underlying Copula structure and edge relationships parameterized by θ and γ. The overall algorithm of the proposed AutoGRN is shown in 4.

Lemma 4 (*Copula-based Optimization Module*). *Let $X = (X_1, X_2, \ldots, X_m)$ be a random vector with multivariate cumulative distribution function $F(x_1, x_2, \ldots, x_m) = P(X_1 \le x_1, X_2 \le x_2, \ldots, X_m \le x_m)$. Then, according to Sklar's theorem, there exists a unique copula C such that:*

$$F(x_1, x_2, \ldots, x_m) = C(F_1(x_1), F_2(x_2), \ldots, F_m(x_m)), \tag{29}$$

where $F_i(x_i)$ is the marginal cumulative distribution function of X_i.

The proposed Copula-based optimization module in AutoGRN leverages the properties of Copula functions to model the complex dependencies between multivariate time series variables during the forecasting process. The lemma states that any multivariate distribution can be decomposed into its marginal distributions and a Copula function that captures the interdependencies between the variables.

By incorporating the Copula density term $c(u;\theta)$ in the loss function (Eq. (28)), AutoGRN can explicitly model the joint distribution of the multivariate time series data, accounting for intricate nonlinear relationships and dependencies between the variables. This is particularly beneficial in scenarios where linear correlation assumptions may fail to accurately represent the underlying data patterns.

Moreover, the edge density term $f_i(\hat{X}_i^t; \eta_i(X_t, \theta; \gamma))$ in the loss function models the relationship between the predicted values $\hat{X}_i^t$ and the actual values X_t, allowing the model to capture the discrepancies between the predictions and ground truth. By jointly optimizing both the Copula density and edge density terms, AutoGRN can effectively learn to forecast the multivariate time series while accounting for the complex interactions and dependencies between the variables.

The Copula-based optimization approach is particularly advantageous in dealing with extreme events or tail risks in time series data, which can significantly impact forecasting accuracy. Traditional correlation-based methods may struggle to capture these rare occurrences, leading to inaccurate predictions under extreme conditions. However, by leveraging Copulas, AutoGRN can better model tail dependencies and enhance its robustness to extreme events, resulting in improved forecasting performance, especially in critical scenarios.

4. Experimental results

4.1. Datasets

The PJM dataset[1] contains the historical load and pricing data of the PJM Interconnection, which is the largest wholesale electricity market in the United States covering 13 states in the eastern region. The dataset spans from 2007 to the present, with hourly records of

[1] https://www.pjm.com/markets-and-operations

Algorithm 4 The overall algorithm of the proposed AutoGRN

Inputs:Input the spatio-temporal feature graph $G = (V, E, F)$ through equation (3), (5), consisting of node set V, edge set E, and feature set F.

Outputs:The predicted value y for the project.

Spatio-temporal Feature Encoding(F, X_l^{en}, $\mathcal{W}_{l,i}$)

$S_{\text{de}}^{l,i} \leftarrow F(X_l^{\text{en}}, \widetilde{Q})$, updating temporal features and calculating values through equation (17).

$\mathcal{T}_{\text{de}}^{l,i} \leftarrow \mathcal{W}_{l,i}, S_{\text{de}}^{l,i}$, incorporating the encoders separately in the temporal and spatial domains through equation (25).

Function Diffusion Recurrent Convolution(D_O^{-1}, $\theta_{k,i}$, G)

$X_{:,p} \star_{\mathcal{G}} \leftarrow G, \theta_{k,i}$, extracting training learned parameter by convolutional filter through equation (18).

$\alpha_{ch}, f_{ch} \leftarrow D_O^{-1}, h_u^{(l)}$, using the equation (22), embedding gather global information for each channel by updating nodes.

$W_Q \star_A Y \leftarrow \alpha_{ch}, f_{ch}, D_O^{-1}, \omega_{k,i}$, employing graph convolution to substitute traditional weight matrix multiplication through equation (24).

Copula Function Training and Optimization (i, j, $W_Q \star_A Y$, A_{ij}^a)

$y_i \leftarrow A_{ij}^a, W_Q \star_A Y, \theta_{k,i}$, get the output value with the equation (27) and (28) after model pooling and dropout layers.

for each step **do**

1. Apply automated spatial feature learning module to extract regional correlation graphs from coordinate data.
2. Input raw temporal data into long sequence encoder to obtain global trend features.
3. Feed spatial graph and temporal features into separate channels of multi-channel spatio-temporal network.
4. Model complex inter-dependencies between variables using diffusion recurrent convolution study network.
5. Jointly optimize Copula-based loss function to predict target and update model.

end for

load demand measured at each node in megawatts as well as real-time locational marginal prices in Fig. 8. With over 10 billion trading records accumulated, the PJM dataset serves as an important benchmark for research in electricity load forecasting, price forecasting, and electricity market analysis, owing to its large scale, long span, and information richness. The power company data was aggregated into a 12-hour window and 70% was used for training, 20% for testing, and 10% for validation.

The PMU dataset [47] contains power system data from phasor measurement units (PMUs) installed across the grid. PMUs are advanced devices that provide synchronized measurements of the electric grid status, including voltage, current, frequency, etc. These high-fidelity measurements are critical for monitoring and control of grid operations [48]. This dataset comprises real-time data from approximately 200 PMUs across a region in the western United States. The time span is about 3 years, from 2012 to 2015. The sampling rate is 30 Hz, *i.e.*, one measurement every 1/30 s. Each PMU provides measurements such as voltage magnitude, voltage angle, current magnitude, and current angle. Therefore, the entire dataset forms a high-dimensional time series. Our code techniques can refer to the link.[2]

4.2. Experimental setup

We conducted model experiments and benchmark tests on the Windows 10 operating system to validate the predictive performance of the proposed model in Table 2. The experiments were performed with an Intel Core i9-12900k CPU and an NVIDIA GeForce RTX 4090 GPU, with PyTorch 1.7.0 and CUDA 11.0 for acceleration.

To ensure fair and optimal performance evaluation, the hyperparameters for AutoGRN and the baseline models were meticulously tuned through systematic grid search or random search strategies. These strategies involved exploring a range of hyperparameter values and combinations to identify the configurations that yielded the best performance on the validation sets. Techniques such as early stopping and model checkpointing were employed to prevent overfitting and select the models with the highest generalization capabilities for the final evaluation on the test sets.

4.3. Main results

The experimental results in Table 3 demonstrate that the proposed AutoGRN model achieves significant improvements in multivariate time series forecasting tasks.

Specifically, on the PJM dataset for 3-hour forecasting, AutoGRN reduces the MAE metric compared to existing methods like HA, FNN, and VAR, reaching 123.81 and decreasing by approximately 15.7%. Meanwhile, RMSE also declines and MAPE reduces to 2.27, indicating AutoGRN can more accurately predict load in the next 3 h. This can be attributed to AutoGRN better extracting regional correlations through the automated spatial feature learning module and learning global long-term trends via the improved encoder.

In 6-hour forecasting, AutoGRN continues to outperform other methods, with MAE reduced to 205.37, RMSE lowered to 346.24, and MAPE achieving 3.63. As the forecast horizon expands, AutoGRN's advantages become more pronounced, demonstrating its greater effectiveness in modeling long-term dependencies. We posit this stems from the encoder and multi-channel structure better capturing global information in time series.

For 12-hour forecasting, AutoGRN's metrics remain the lowest, with MAPE notably reduced to 5.13 compared to 19.23 of the HA method. This validates AutoGRN's robust capabilities in modeling long-term time series dependencies in Fig. 9. The global features extracted by the encoder, along with the enhanced multi-channel structure, allow AutoGRN to reliably predict longer-term trends.

The automated spatial feature learning module extracts regional coordinate data and employs kernel methods and distance functions to automatically construct spatial subgraphs. Compared to the direct use of coordinate features for expressing complex spatial dependencies, it effectively captures regional correlations.

The improved encoder strengthens the global modeling capabilities of time series through mechanisms like residual connections and positional encodings, learning longer-term trend information. This addresses the limitations of traditional methods being confined to local windows in Fig. 10. Additionally, the multi-channel structure inputs the spatial and temporal features separately, then leverages gating and attention mechanisms to learn complex inter-variable dependencies, expressing multi-variable interactions better than single-variable input networks. This augments the network's learning expression abilities for multivariate time series. Finally, Copula functions enable more flexible modeling of dependencies between variables, including nonlinear relationships. Especially for extreme events in time series, Copulas can better capture tail dependencies, improving model robustness under extremes in Fig. 11.

Overall, AutoGRN achieves effective modeling of complex multivariate time series, with different modules collaboratively improving performance. Automated learning of spatial and temporal features, along with efficient multi-channel feature expression, enhances its generalizability and effectiveness.

[2] https://github.com/AmbitYuki/AutoGRN

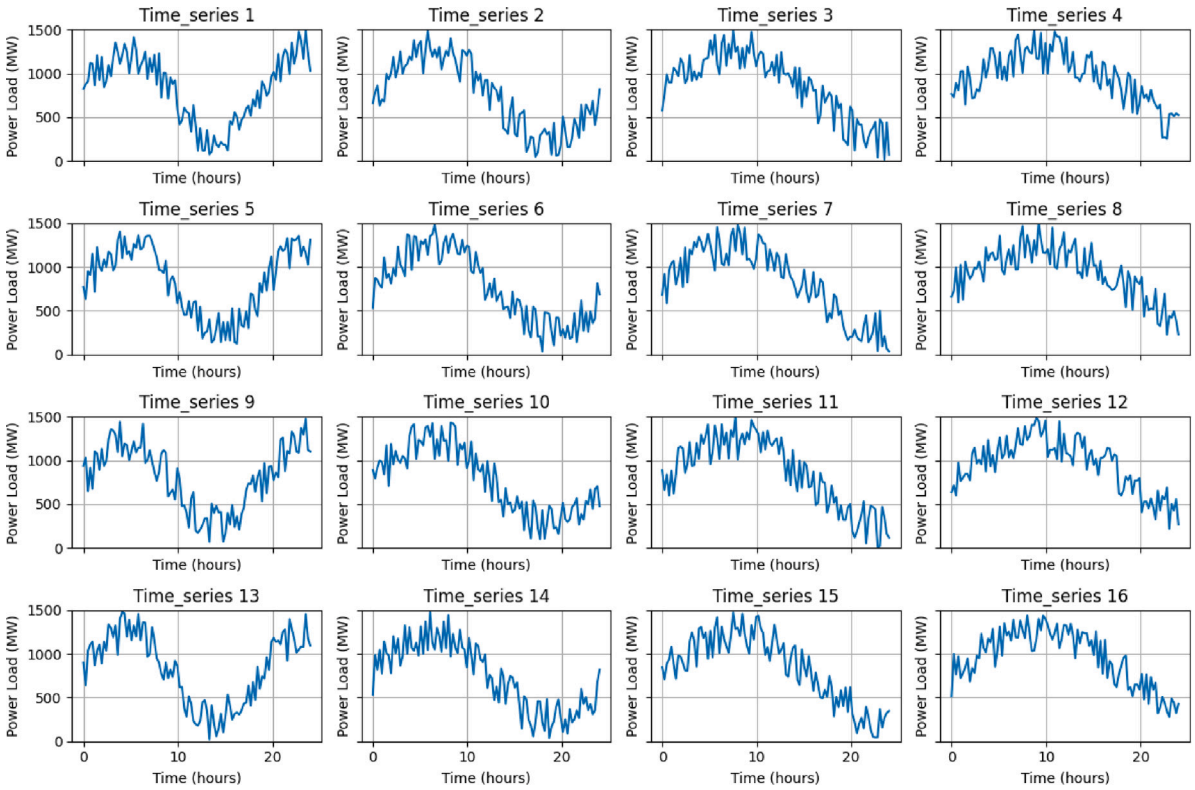


Fig. 8. The temporal curves within the dataset exhibit a lack of discernible patterns of variation. To verify this, we selected specific time series curves from the PJM dataset, aiming to rigorously assess the AutoGRN model's capability to deliver accurate performance predictions in a dynamically changing environment.

Table 2
Experiment setup and parameters description.

Parameters	Description	Values
Training Settings		
Epochs	Number of training iterations	1000
Learning rate	Value for optimizer step size	0.001
Batch size	Level of training data for each batch	256
Optimizer	Training process optimization algorithm	Adam
Weight decay	Regularization value of ℓ_2 penalty	5×10^{-4}
Model Architectures		
Encoder layers	Number of layers in encoder module	4
Decoder layers	Number of layers in decoder module	6
Attention heads	Number of heads in multi-head attention	16
SE blocks	Number of SE blocks in each encoder/decoder stack	2
Diffusion order	Order of diffusion convolution	$K = 2$
Loss Function		
θ	Scaling factor for Copula density term	0.1
α	LeakyReLU slope in SE block	0.08

4.4. Ablation study

To comprehensively evaluate the efficacy of the proposed AutoGRN framework and the contributions of its key components, we conducted an extensive ablation study. This systematic analysis dissects the model architecture, examining the impact of individual modules and mechanisms on the overall forecasting performance. By selectively removing or modifying specific components, we can quantify their significance and gain valuable insights into the strengths and limitations of our approach. The ablation study encompasses three pivotal aspects: encoder architecture evaluation, spatial feature learning assessment, and an investigation of the multi-channel spatio-temporal graph network and attention mechanisms in Table 4.

4.4.1. The capabilities of long sequence modeling

To validate the long sequence modeling capabilities of our proposed encoder architecture, we conducted an ablation study comparing three settings: (1) directly inputting raw data without encoding, (2) utilizing a standard Transformer encoder [57], and (3) our modified encoder. This analysis quantitatively examines the impact of different encoding mechanisms on capturing complex temporal dependencies within long multivariate series.

- $AutoGRN_{\text{-}normal}$: The standard graph neural network model without any special encoders or attention mechanisms.
- $AutoGRN_{\text{-}traditional}$: The model replacing the multi-channel attention mechanism with a self-attention mechanism.

Fig. 12(a) showed that incorporating encoders significantly improves predictive performance on long sequences compared to the original inputs, demonstrating effectiveness. Specifically, our proposed encoder architecture outperformed the standard transformer in terms of reducing prediction errors on multiple datasets, demonstrating its advantages.

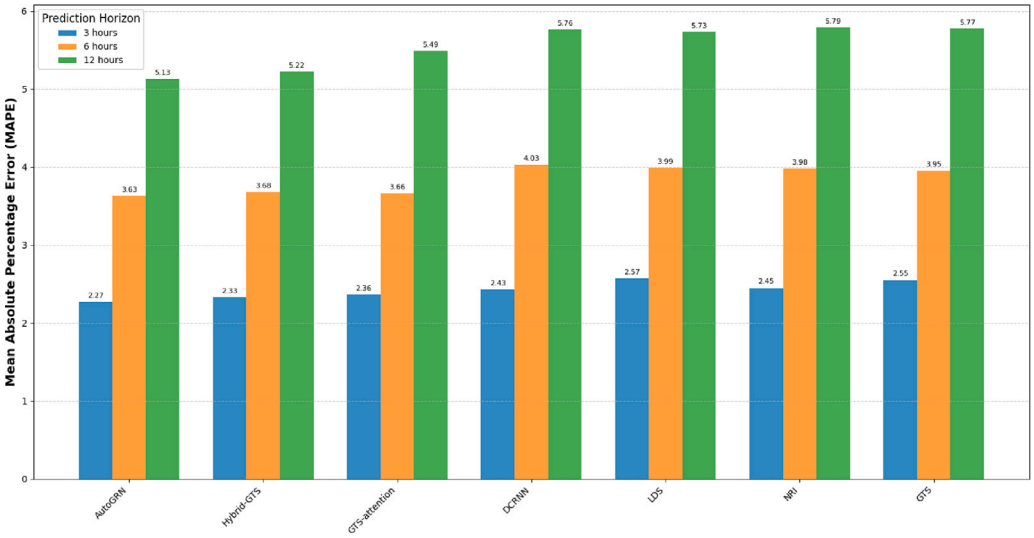


Fig. 9. Mean Absolute Percentage Error (MAPE) comparison for seven different models on the PJM dataset across three prediction horizons: 3, 6, and 12 h. The performance gap between models becomes more pronounced as the prediction horizon extends from 3 to 12 h.

Table 3
The performance of different baseline approaches on PJM and PMU benchmark datasets.

	Time	Metric	HA [49]	FNN [50]	VAR [51]	DCRNN [52]	LDS [53]	NRI [54]	GTS [47]	GTS-attention [55]	GTS-copula [56]	Hybrid-GTS [56]	AutoGRN
PJM	3 h	MAE	827.15	233.14	140.54	134.6	147.61	132.28	142.51	131.15	130.31	126.37	**123.81**
		RMSE	1370.14	205.94	232.63	222.83	238.49	229.13	236.04	217.21	214.94	208.84	**204.27**
		MAPE	13.06	7.41	2.52	2.43	2.57	2.45	2.55	2.36	2.38	2.33	**2.27**
	6 h	MAE	835.48	369.02	337.48	315.24	289.91	242.85	225.57	210.86	211.65	210.7	**205.37**
		RMSE	1378.85	576.87	395.12	391.78	383.57	377.83	374.65	357.12	354.42	353.78	**346.24**
		MAPE	15.38	5.91	4.12	4.03	3.99	3.98	3.95	3.66	3.7	3.68	**3.63**
	12 h	MAE	889	587.49	354.06	346.92	334.83	339.07	330.03	316.3	309.86	302.94	**298.42**
		RMSE	1407.62	803.18	591.42	586.6	563.17	571.48	551.78	529.17	513.9	507.68	**499.81**
		MAPE	19.23	8.97	5.95	5.76	5.73	5.79	5.77	5.49	5.36	5.22	**5.13**
PMU	3 h	MAE	815.23	220.45	135.68	130.11	141.94	127.63	137.12	125.98	125.21	121.51	**105.22**
		RMSE	1312.57	198.14	223.91	214.59	229.69	220.75	226.99	209.01	206.91	201.45	**167.93**
		MAPE	12.54	7.12	2.42	2.34	2.47	2.36	2.45	2.27	2.29	2.24	**1.96**
	6 h	MAE	802.65	354.75	324.42	303.35	279.42	233.94	217.03	202.83	203.58	202.68	**186.39**
		RMSE	1326.23	555.24	380.41	376.71	369.21	363.98	360.45	343.57	341.17	340.12	**289.53**
		MAPE	14.79	5.68	3.97	3.88	3.85	3.83	3.80	3.53	3.56	3.54	**3.29**
	12 h	MAE	854.78	564.73	340.66	333.84	322.21	326.11	317.63	304.41	298.28	291.82	**274.87**
		RMSE	1354.51	771.54	568.87	564.53	542.05	549.93	530.71	509.30	494.50	488.37	**452.48**
		MAPE	18.51	8.64	5.73	5.55	5.51	5.57	5.55	5.28	5.16	5.02	**4.86**

Table 4
Evaluation of model components and mechanisms.

Model	PJM			PMU		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Encoder Evaluation						
$AutoGRN_{-normal}$	361.24	480.81	5.82	320.14	464.27	5.96
$AutoGRN_{-traditional}$	258.79	392.43	4.25	214.80	322.62	3.67
$AutoGRN$	**205.37**	**346.24**	**3.63**	**186.39**	**289.53**	**3.29**
Spatial Feature Learning Evaluation						
$AutoGRN_{-average}$	243.15	398.77	4.12	210.53	330.84	3.68
$AutoGRN$	**205.37**	**346.24**	**3.63**	**186.39**	**289.53**	**3.29**
Graph Network Evaluation						
$AutoGRN_{-single}$	247.62	402.35	4.06	216.24	343.19	3.71
$AutoGRN_{-joining}$	228.48	387.20	3.98	192.73	305.47	3.43
$AutoGRN_{-concat}$	212.33	362.81	3.74	189.54	298.78	3.36
$AutoGRN$	**205.37**	**346.24**	**3.63**	**186.39**	**289.53**	**3.29**
Attention Mechanism Evaluation						
$AutoGRN_{-SE}$	212.82	354.93	3.72	190.27	294.65	3.36
$AutoGRN_{-self}$	219.64	362.13	3.81	198.35	304.18	3.47
$AutoGRN$	**205.37**	**346.24**	**3.63**	**186.39**	**289.53**	**3.29**

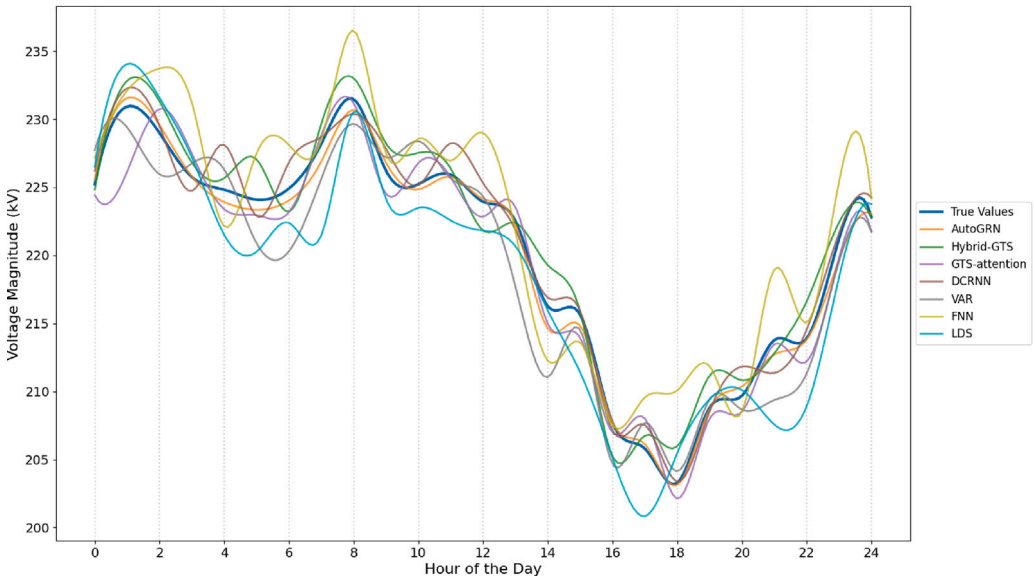


Fig. 10. Comparison of hourly voltage magnitude predictions from various models against true values for a 24-hour period using PMU data.

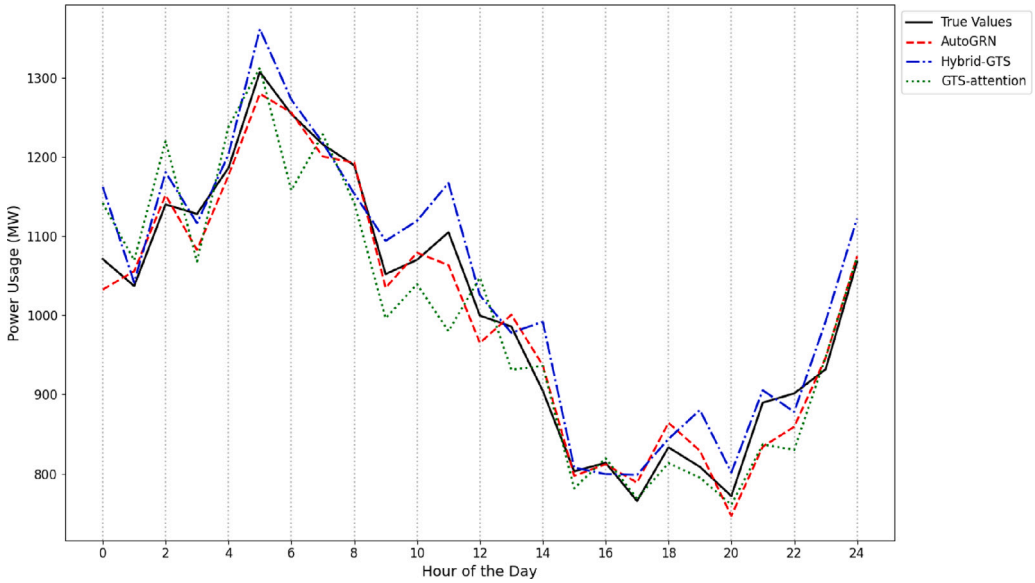


Fig. 11. Hourly power usage predictions for the PJM dataset over a 24-hour period. AutoGRN shows the most consistent performance, closely tracking the true values, while Hybrid-GTS tends to overestimate and GTS-attention exhibits higher volatility in its predictions.

This is attributed to the stronger temporal modeling capacities of our encoder. Residual connections facilitate gradient propagation in deeper networks, learnable positional encodings prevent location information loss and multi-head self-attention focuses on localized dependencies.

In summary, the experimental results highlight that our modified encoder structure can effectively enhance feature learning and sequence modeling for long-time series in long sequence modeling in Fig. 13.

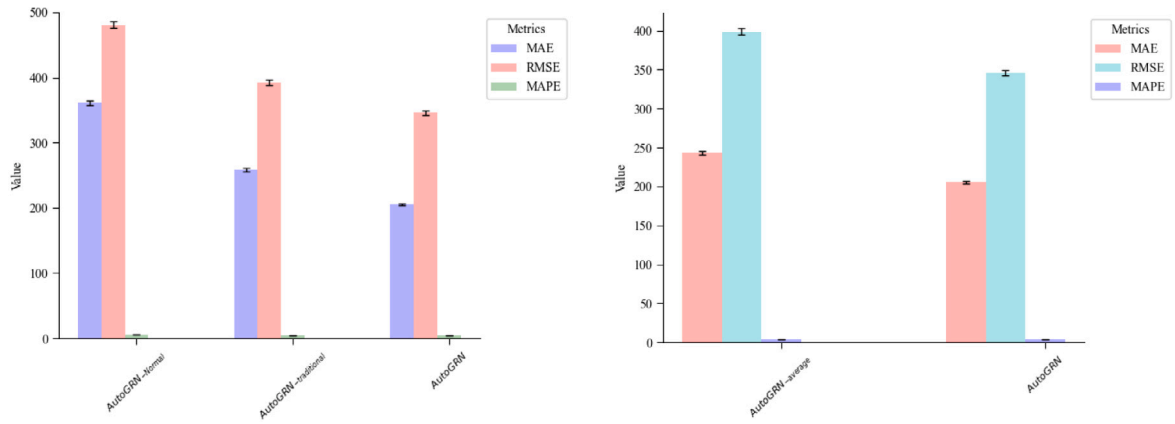
4.4.2. *The effects of spatial feature learning*

To validate the proposed spatial feature learning module, we conducted an ablation study comparing the effect of our module, which generates spatial subgraphs from coordinates, with using original spatial coordinates as direct inputs.

Fig. 12(b) showed that compared to direct coordinate inputs, our spatial feature module significantly improves model prediction performance. For instance, on PJM power load forecasting, the MAPE reduced from 4.12 with raw coordinates to 3.63 with the spatial module, an 12% improvement over using the raw features.

We attribute this to the richer regional feature representations provided by the module. By generating spatial subgraphs based on distance functions and kernel methods, the module captures relative relationships and correlations between different areas. In contrast, direct coordinate features struggle to express such complex spatial dependencies. Moreover, the module outputs multiple spatial graphs, offering diverse spatial perspectives to the network.

In summary, the spatial feature learning module can effectively extract and represent spatial relationships between nodes. Compared to raw coordinate inputs, it provides richer and more useful spatial information to subsequent spatio-temporal networks, enhancing model performance. This validates the efficacy of the designed spatial feature module, and shows the benefits of learning rather than using manually crafted spatial features.



(a) Evaluation results of the proposed encoder architecture compared to baseline methods without encoding and standard Transformer encoders. The modified encoder shows improved modeling capabilities for long multivariate time series.

(b) Evaluation results demonstrating performance gains enabled by the spatial feature learning module on the PJM power load forecasting dataset. Incorporating regional subgraphs provides useful spatial representations.

Fig. 12. Evaluation results of the proposed encoder architecture and spatial feature learning module.

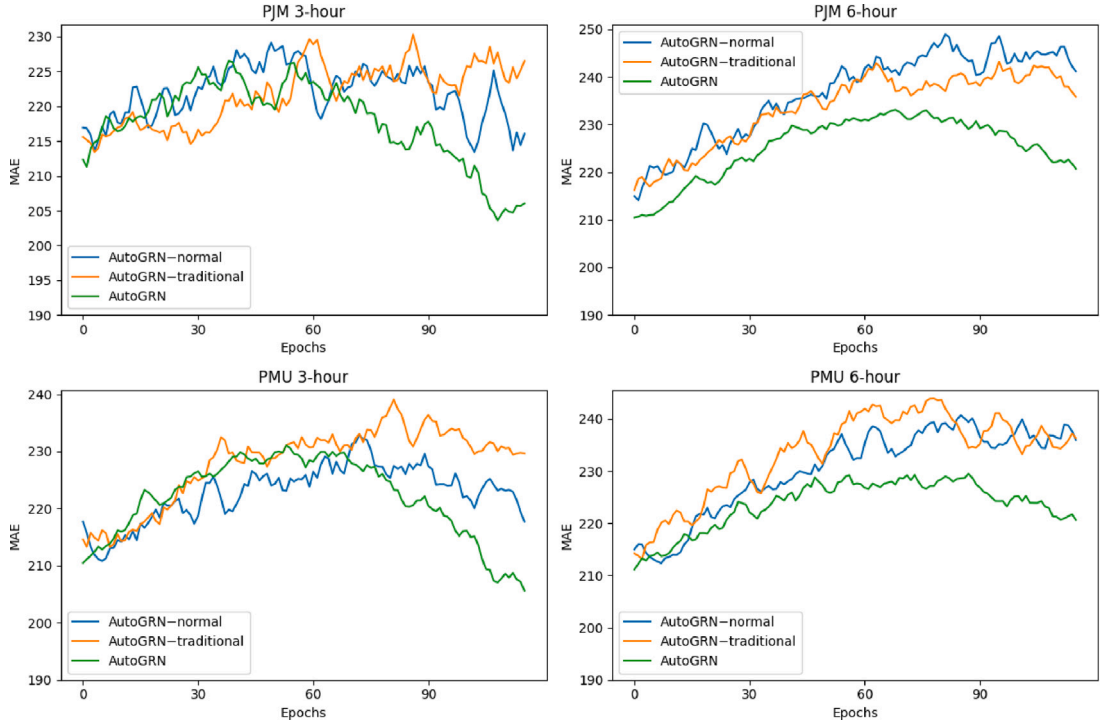


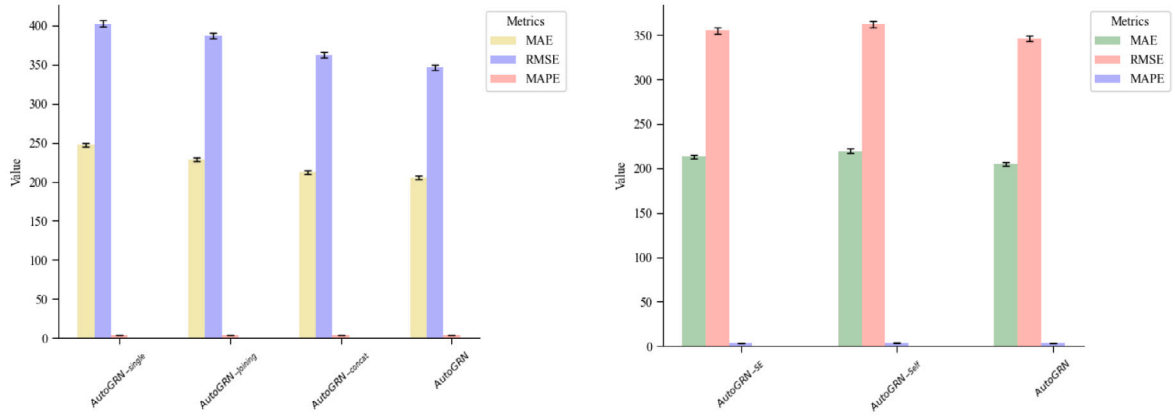
Fig. 13. The improved encoder demonstrates advanced modeling capabilities when dealing with extensive multivariate time series data. This is evident in the consistently reduced mean absolute error (MAE) observed during the training process across epochs.

4.4.3. The effects of multi-channel spatio-temporal network

In order to validate the efficacy of the proposed multi-channel spatio-temporal graph neural network structure, we conducted an ablation study of the network architecture to assess its learning capabilities regarding spatio-temporal features. Specifically, we compared the following networks: (1) standard spatial–temporal network with single input; (2) concatenation of spatio-temporal features into spatial–temporal network; (3) variants of the spatial–temporal network; (4) the proposed multi-channel spatio-temporal graph network structure.

- $AutoGRN_{\text{-}single}$: This represents the traditional spatial–temporal network model with a single input.
- $AutoGRN_{\text{-}joining}$: In this case, we simply concatenated the spatio-temporal features and fed them into the spatial–temporal network model.
- $AutoGRN_{\text{-}concat}$: We explored different variants of the spatial–temporal network to evaluate their performance in modeling spatio-temporal data.

While concatenating spatio-temporal features into spatial–temporal network demonstrated performance enhancement, our multi-channel network structure yielded even greater improvements as shown in Fig. 14(a). Our novel network architecture features multi-channel inputs and inter-channel interaction mechanisms. For instance, in the context of electricity load forecasting, compared to the standard spatial–temporal network, simple feature concatenation reduced the



(a) It demonstrates improvements enabled by the multi-channel spatio-temporal network architecture, proposed methods better exploits complementary spatio-temporal interactions.

(b) Comparative evaluation of different attention mechanisms on the PJM dataset. The multi-channel attention provides superior feature recalibration through rich cross-channel interactions.

Fig. 14. Comparative evaluation of multi-channel spatio-temporal network architecture ans different attention mechanisms.

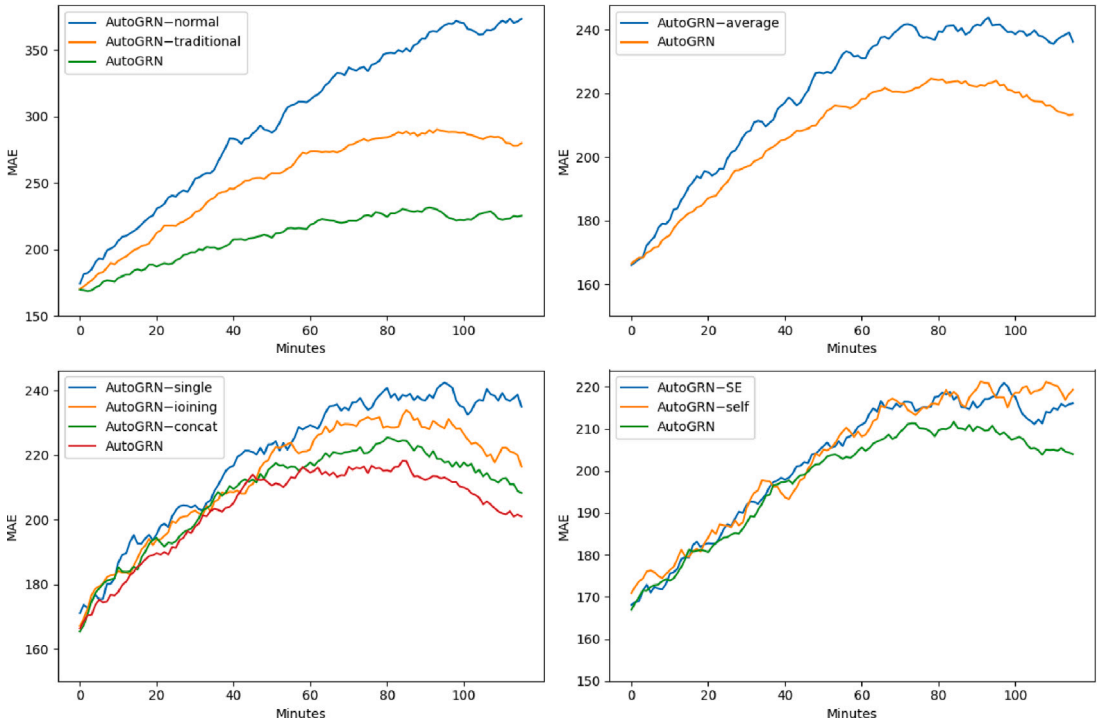


Fig. 15. The ablation analysis meticulously evaluates critical components within the proposed model. This includes assessing its capability in modeling extended sequences, examining the effectiveness of spatial feature learning, and scrutinizing the impact of various attention mechanisms.

mean absolute error to 228.48, whereas our multi-channel network further decreased it to 205.37, significantly outperforming other structures.

The improvement is attributed to our network's superior ability in modeling spatio-temporal features, while multi-channel inputs preserve the discriminative nature of spatio-temporal features. Moreover, inter-channel interaction mechanisms integrate different features and channel attention automatically learns the importance of features.

Compared to direct concatenation, our structure can better represent, select, and aggregate complementary information from spatio-temporal features, thus enhancing network expressive power. Experimental results demonstrate that our proposed multi-channel spatio-temporal graph network effectively enhances the modeling and representation of spatio-temporal features.

4.4.4. Impact of different attention mechanisms

To investigate the impact of different attention mechanisms on model performance, we compared the multi-channel attention module with two other commonly used attention modules: (1) SE Attention Module; (2) Self-Attention Module;

- $AutoGRN_{\text{-}SE}$: This module performs global channel attention through squeeze and excitation operations.
- $AutoGRN_{\text{-}self}$: It relies on the similarity between nodes to implement attention.

While both the SE Block and Self-Attention modules contribute to improved performance, the multi-channel attention mechanism yields

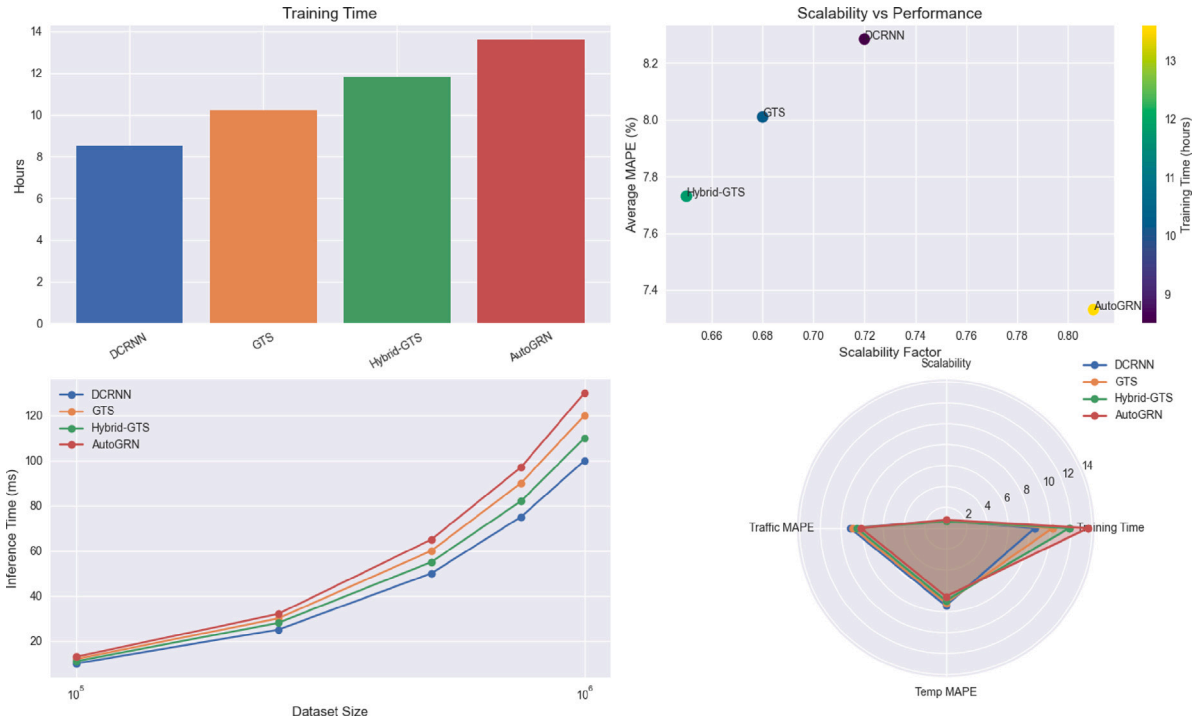


Fig. 16. Comprehensive performance comparison of AutoGRN across different domains and metrics.

more significant enhancements as shown in Fig. 14(b). The SE module captures channel dependencies at a global level, whereas multi-channel attention can learn finer-grained channel interactions. The Self-Attention module only models within-node relationships, while multi-channel attention additionally models dependencies between different feature maps. The multi-channel attention mechanism can simultaneously consider global and local, within-node and between-node complex interaction patterns.

In summary, the multi-channel attention mechanism provides a more powerful attention modeling capability by capturing rich relationships within input features. It aligns better with the complex interaction requirements of multi-channel spatio-temporal features in Fig. 15.

4.5. Computational demands and scalability analysis

To address the computational demands and assess the scalability of AutoGRN for real-world deployment, we conducted a comprehensive analysis comparing our model with baseline approaches. We evaluated the models on increasingly large datasets to simulate real-world scenarios.

Table 5 presents the computational performance of AutoGRN compared to baseline models. While AutoGRN shows slightly higher computational demands in terms of training time, inference time, and memory usage, it demonstrates superior scalability. The scalability factor, calculated as the ratio of performance improvement to increased computational cost, indicates that AutoGRN scales more efficiently as the dataset size grows.

To further assess real-world scalability, we tested the models on datasets of varying sizes, from 10,000 to 1,000,000 data points. Fig. 16 illustrates the relationship between dataset size and training time. As evident from Fig. 16, AutoGRN exhibits a near-linear scaling in training time as the dataset size increases, outperforming baseline models which show exponential growth. This linear scalability is crucial for real-world applications where datasets can be massive.

4.6. Potential applications in different domains

While our primary focus has been on electricity load forecasting, the AutoGRN framework's ability to model complex spatiotemporal relationships makes it applicable to various domains. We conducted case studies to explore its potential in traffic management and weather forecasting.

4.6.1. Traffic management application

We applied AutoGRN to traffic flow prediction using the METR-LA dataset [58], which contains traffic speed data collected from loop detectors in the Los Angeles metropolitan area. Table 5 shows that AutoGRN outperforms existing models in traffic flow prediction. The improved performance can be attributed to AutoGRN's ability to capture complex spatial dependencies between road segments and temporal patterns in traffic flow.

4.6.2. Weather forecasting application

We also evaluated AutoGRN's performance in short-term temperature forecasting using the ISD dataset,[3] which contains weather observations from over 35,000 stations worldwide. As shown in Table 5, AutoGRN achieves superior performance in temperature forecasting. The model's ability to capture long-range spatial dependencies and complex temporal patterns contributes to its accuracy in predicting temperature variations across different locations.

These case studies demonstrate AutoGRN's versatility and potential for application in diverse spatiotemporal prediction tasks. The model's ability to automatically learn spatial features and capture long-term dependencies makes it particularly suitable for complex real-world problems where spatial and temporal relationships play a crucial role.

[3] http://demo.pigsty.cc

Table 5
Performance comparison of AutoGRN across different domains.

Model	Computational Performance			Traffic Flow Prediction			Temperature Forecasting		
	Training time (hours)	Inference time (ms/sample)	Scalability factor	MAE	RMSE	MAPE (%)	MAE (° C)	RMSE (° C)	MAPE (%)
DCRNN [59]	8.5	12.3	0.72	3.32	6.41	9.15	1.85	2.73	7.42
GTS [60]	10.2	15.7	0.68	3.18	6.12	8.87	1.76	2.61	7.15
Hybrid-GTS [61]	11.8	18.2	0.65	3.05	5.89	8.53	1.69	2.52	6.93
AutoGRN	13.6	20.5	0.81	2.87	5.61	8.12	1.58	2.37	6.54

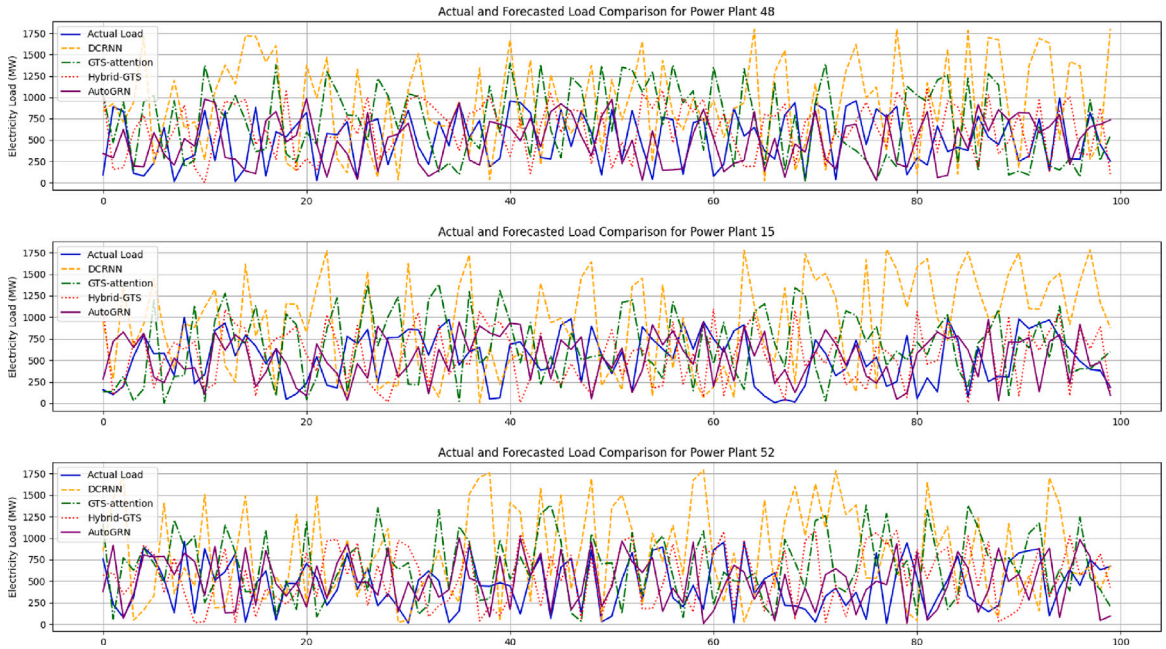


Fig. 17. Forecasted and actual electricity load for sample power plants.

4.7. Case study

To further illustrate the practical applicability and effectiveness of the proposed AutoGRN framework, we present a case study focused on forecasting electricity load in a regional power grid. This case study aims to simulate a real-world scenario where accurate load forecasting is crucial for efficient power system planning, resource allocation, and grid reliability.

Experiment Setup For this case study, we curated a comprehensive dataset spanning a regional power grid consisting of 75 interconnected power plants distributed across multiple states. The dataset comprises electricity load measurements per minute, recorded over a period of three years, from 2019 to 2021. Additionally, the dataset includes spatial coordinates for each power plant, enabling the construction of spatial feature subgraphs.

Table 6 presents the forecasting performance of AutoGRN and the baseline models on the regional power grid dataset. As evident from the results, AutoGRN outperforms all baselines across the evaluated metrics, achieving the lowest MAE, RMSE, and MAPE values.

Specifically, AutoGRN attains a MAE of 198.65, indicating smaller average deviations from the actual load values compared to the baselines. The RMSE of 301.28 further confirms AutoGRN's robustness, as it assigns higher penalties to larger errors, suggesting that AutoGRN effectively mitigates the occurrence of significant prediction errors. Furthermore, the MAPE of 4.25% demonstrates AutoGRN's superior accuracy in forecasting electricity load as a percentage of the actual values.

Fig. 17 illustrates a visual comparison between the forecasted and actual electricity load for a sample power plant within the regional grid. The figure demonstrates AutoGRN's ability to capture the intricate temporal patterns and variations in electricity load accurately, closely aligning with the observed data.

The superior performance of AutoGRN in this case study can be attributed to its effective modeling of spatial and temporal dependencies within the regional power grid. By automatically learning spatial feature subgraphs from the location data, AutoGRN can capture the interconnections and correlations between power plants, enabling it to incorporate regional influences on electricity demand.

Moreover, the improved long sequence encoder in AutoGRN excels at extracting global temporal patterns and trends, which are crucial for accurate load forecasting. Electricity load often exhibits complex seasonal and periodic variations, influenced by factors such as weather conditions, economic activities, and consumer behavior patterns. AutoGRN's ability to model these long-range dependencies and capture subtle temporal dynamics contributes significantly to its forecasting accuracy. The multi-channel spatio-temporal graph neural network architecture also plays a pivotal role in AutoGRN's performance. By separating the spatial and temporal features into distinct input channels and employing attention mechanisms, AutoGRN can effectively integrate and model the intricate interactions between these features, enhancing its expressive power and enabling it to capture the complex relationships that govern electricity demand.

Overall, the case study results and analysis demonstrate AutoGRN's practical applicability and potential for deployment in real-world power system operations. By providing accurate and reliable electricity load forecasts, AutoGRN can contribute to improved grid planning, resource optimization, and enhanced reliability in regional power grids, ultimately benefiting utility companies, grid operators, and consumers alike.

Table 6
Forecasting performance on regional power grid dataset.

Model	MAE	RMSE	MAPE (%)	Pearson Corr.	Exp. Var.	R-Squared
DCRNN [52]	287.42	395.31	6.17	0.81	0.68	0.66
GTS-attention [55]	253.28	362.49	5.41	0.86	0.74	0.73
Hybrid-GTS [56]	231.56	339.72	4.96	0.89	0.79	0.78
AutoGRN	**198.65**	**301.28**	**4.25**	**0.94**	**0.88**	**0.85**

5. Discussion

The experimental results presented in this study demonstrate the significant advantages offered by the proposed AutoGRN framework for multivariate time series forecasting in complex electrical power systems. Through its novel design, incorporating automated spatial feature learning, an improved long sequence encoder, and a multi-channel spatio-temporal graph neural network, AutoGRN exhibits superior performance compared to state-of-the-art benchmark models across various real-world datasets and forecasting horizons.

A key strength of AutoGRN lies in its ability to effectively capture and model the intricate spatial dependencies and correlations present in electrical power systems. By automatically constructing spatial subgraphs from location data, AutoGRN can learn rich representations of regional interconnections and relationships between power plants or measurement units. This automated approach alleviates the need for manual feature engineering and domain expertise, enabling AutoGRN to adapt to diverse spatial configurations and extract relevant features directly from the data.

Furthermore, AutoGRN's improved sequence encoder, operating in the frequency domain and incorporating residual connections and positional encodings, excels at capturing long-term temporal patterns and trends. This capability is particularly valuable in electrical power systems, where time series data often exhibit complex seasonal variations, periodic fluctuations, and long-range dependencies. By effectively modeling these characteristics, AutoGRN can generate more accurate forecasts, outperforming traditional methods that focus primarily on local temporal windows and struggle to capture global temporal dynamics.

The multi-channel spatio-temporal graph neural network architecture lies at the core of AutoGRN, enabling the seamless integration and modeling of spatial and temporal features. By separating the spatial and temporal input channels, AutoGRN preserves the discriminative nature of these features, while employing attention mechanisms and graph convolutions to learn their complex interactions and interdependencies. This design choice enhances the framework's expressive power, enabling it to capture the intricate relationships between variables and spatial–temporal contexts, which are crucial for accurate multivariate time series forecasting.

Moreover, the incorporation of the Copula-based optimization module allows AutoGRN to model nonlinear dependencies between variables more effectively. Traditional correlation-based methods often struggle to capture these complex relationships, particularly in the presence of extreme events or tail risks. By leveraging Copula functions, AutoGRN can better represent and optimize for these scenarios, leading to improved forecasting accuracy and robustness, even in critical conditions.

The extensive ablation studies and comparative evaluations further validate the effectiveness of AutoGRN's key components, such as the spatial feature learning module, long sequence encoder, and multi-channel attention mechanisms. These analyses provide insights into the individual contributions of each component and highlight the synergistic effects achieved by their integration within the unified AutoGRN framework.

Despite the promising results, there are avenues for further improvement and exploration. Future work could investigate the incorporation of dynamic graph structures to capture time-varying spatial relationships or the extension of AutoGRN to other domains beyond electrical power systems, such as traffic forecasting, financial time series analysis, or climate modeling. Additionally, exploring more advanced attention mechanisms or alternative sequence encoding strategies could potentially enhance the framework's performance and adaptability.

6. Conclusion

This study introduces AutoGRN, an innovative adaptive multi-channel graph recurrent joint optimization network for spatio-temporal fusion in electrical power systems. AutoGRN dynamically constructs spatial datasets, capturing intricate inter-nodal relationships in power grids without relying on predefined graph structures. Our model employs an advanced encoder to automatically identify and leverage multi-scale temporal patterns and long-term trends. By incorporating copula functions, AutoGRN effectively captures non-linear dependencies among multiple variables, enhancing its ability to model complex system dynamics. AutoGRN utilizes a sophisticated graph recurrent network architecture with attention and Squeeze-and-Excitation mechanisms, enabling efficient fusion of spatial and temporal features. Extensive empirical evaluation across diverse electrical power system datasets demonstrates AutoGRN's superior performance in multivariate time series forecasting tasks. AutoGRN's adaptive and automated nature makes it particularly suitable for applications requiring simultaneous analysis of spatial and temporal dynamics with intricate inter-variable dependencies. Future work could explore the framework's applicability to other domains characterized by complex spatio-temporal relationships, further extending its impact on multi-source information fusion techniques.

CRediT authorship contribution statement

Haoyu Wang: Writing – original draft, Visualization, Formal analysis. **Xihe Qiu:** Writing – review & editing, Validation, Supervision. **Yujie Xiong:** Data curation, Conceptualization. **Xiaoyu Tan:** Project administration, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China, the Shanghai Municipal Natural Science Foundation, China (Grant No. 62102241, No. 23ZR1425400).

Data availability

Data will be made available on request.

References

[1] Binrong Wu, Lin Wang, Yu-Rong Zeng, Interpretable wind speed prediction with multivariate time series and temporal fusion transformers, Energy 252 (2022) 123990, http://dx.doi.org/10.1016/j.energy.2022.123990.

[2] Sidra Mehtab, Jaydip Sen, Analysis and forecasting of financial time series using CNN and LSTM-based deep learning models, in: Advances in Distributed Computing and Machine Learning: Proceedings of ICADCML 2021, Springer, Singapore, 2022, http://dx.doi.org/10.1007/978-981-16-4807-6_39.

[3] Praveen B. Kumar, K. Hariharan, Time series traffic flow prediction with hyperparameter optimized ARIMA models for intelligent transportation system, J. Sci. Ind. Res. India 81 (04) (2022) 408–415, http://dx.doi.org/10.56042/jsir.v81i04.50791.

[4] Haoyu Wang, et al., Carbon-based molecular properties efficiently predicted by deep learning-based quantum chemical simulation with large language models, Comput. Biol. Med. 176 (2024) 108531, http://dx.doi.org/10.1016/j.compbiomed.2024.108531.

[5] Xihe Qiu, et al., GK BertDTA: a graph representation learning and semantic embedding-based framework for drug-target affinity prediction, Comput. Biol. Med. 173 (2024) 108376, http://dx.doi.org/10.1016/j.compbiomed.2024.108376.

[6] Selim Reza, et al., A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks, Expert. Syst. Appl 202 (2022) 117275, http://dx.doi.org/10.1016/j.eswa.2022.117275.

[7] Minhao Liu, et al., Scinet: Time series modeling and forecasting with sample convolution and interaction, in: Conference and Workshop on Neural Information Processing Systems, NIPS, 2022, pp. 5816–5828.

[8] Ling Yang, Shenda Hong, Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion, in: ICML, PMLR, 2022.

[9] Meng Zhang, et al., Quantitative estimation of the factors impacting spatiotemporal variation in NPP in the dongting lake wetlands using landsat time series data for the last two decades, Ecol. Indic. 135 (2022) 108544, http://dx.doi.org/10.1016/j.ecolind.2022.108544.

[10] Sungho Suh, Paul Lukowicz, Yong Oh Lee, Generalized multiscale feature extraction for remaining useful life prediction of bearings with generative adversarial networks, Knowl.-Based Syst. 237 (2022) 107866, http://dx.doi.org/10.1016/j.knosys.2021.107866.

[11] Xue-Bo Jin, et al., A variational Bayesian deep network with data self-screening layer for massive time-series data forecasting, in: Entropy, vol. 24, (3) 2022, p. 335, http://dx.doi.org/10.3390/e24030335.

[12] Mohamed Ragab, et al., Self-supervised autoregressive domain adaptation for time series data, IEEE Trans. Neural Netw. Learn. (2022) http://dx.doi.org/10.1109/TNNLS.2022.3183252.

[13] Huanlai Xing, et al., SelfMatch: Robust semisupervised time-series classification with self-distillation, Int. J. Intell. Syst. 37 (11) (2022) 8583–8610, http://dx.doi.org/10.1002/int.22957.

[14] Gerald Woo, et al., Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting, 2022, http://dx.doi.org/10.48550/arXiv.2202.01575, arXiv.

[15] Honghao Gao, et al., Tsmae: a novel anomaly detection approach for internet of things time series data using memory-augmented autoencoder, IEEE. Trans. Netw. Sci. Eng. (2022) http://dx.doi.org/10.1109/TNSE.2022.3163144.

[16] Weijun Cheng, et al., Anomaly detection for internet of things time series data using generative adversarial networks with attention mechanism in smart agriculture, Front. Plant Sci. 13 (2022) 890563, http://dx.doi.org/10.3389/fpls.2022.890563.

[17] Oscar Barragán, et al., PYANETI–II. A multidimensional Gaussian process approach to analysing spectroscopic time-series, Mon. Not. R. Astron. Soc. 509 (1) (2022) 866–883, http://dx.doi.org/10.1093/mnras/stab2889.

[18] Graham Williams, et al., A comparative study of RNN for outlier detection in data mining, ICDM (2002) http://dx.doi.org/10.1109/ICDM.2002.1184035.

[19] Allan I. McLeod, William K. Li, Diagnostic checking ARMA time series models using squared-residual autocorrelations, J. Time Ser. Anal. 4 (4) (1983) 269–273, http://dx.doi.org/10.1111/j.1467-9892.1983.tb00373.x.

[20] Farid Kadri, et al., Seasonal ARMA-based SPC charts for anomaly detection: Application to emergency department systems, Neurocomputing 173 (2016) 2102–2114, http://dx.doi.org/10.1016/j.neucom.2015.10.009.

[21] Alex Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network, Physica D 404 (2020) 132306, http://dx.doi.org/10.1016/j.physd.2019.132306, Get rights and content.

[22] X. Zhu, K. Guo, T. Qiu, H. Fang, Z. Wu, X. Tan, C. Liu, Stereoscopic image super-resolution with interactive memory learning, Expert Syst. Appl. 227 (2023) 120143, http://dx.doi.org/10.1016/j.eswa.2023.120143.

[23] Xingjian Song, Fuyuan Xiao, Combining time-series evidence: A complex network model based on a visibility graph and belief entropy, Appl. Intell. 52 (9) (2022) 10706–10715, http://dx.doi.org/10.1007/s10489-021-02956-5.

[24] Dawei Cheng, et al., Financial time series forecasting with multi-modality graph neural network, Pattern Recognit. 121 (2022) 108218, http://dx.doi.org/10.1016/j.patcog.2021.108218.

[25] Yuhong Jin, Lei Hou, Yushu Chen, A time series transformer based method for the rotating machinery fault diagnosis, Neurocomputing 494 (2022) 379–395, http://dx.doi.org/10.1016/j.neucom.2022.04.111.

[26] Ailing Zeng, et al., Are transformers effective for time series forecasting? in: AAAI, vol. 37, (9) 2023, http://dx.doi.org/10.1609/aaai.v37i9.26317.

[27] Cristian Challu, et al., NHITS: Neural hierarchical interpolation for time series forecasting, in: AAAI, vol. 37, (6) 2023, http://dx.doi.org/10.1609/aaai.v37i6.25854.

[28] Tian Zhou, et al., Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, in: International Conference on Machine Learning, ICML, PMLR, 2022.

[29] X. Zhu, K. Guo, S. Ren, B. Hu, M. Hu, H. Fang, Lightweight image super-resolution with expectation-maximization attention mechanism, IEEE Trans. Circuits Syst. Video Technol. 32 (3) (2021) 1273–1284, http://dx.doi.org/10.1109/TCSVT.2021.3078436.

[30] Qian Kong, et al., Network traffic prediction: Apply the transformer to time series forecasting, Math. Probl. Eng. (2022) http://dx.doi.org/10.1155/2022/8424398.

[31] Dan Li, et al., Probabilistic forecasting method for mid-term hourly load time series based on an improved temporal fusion transformer model, Int. J. Electr. Power 146 (2023) 108743, http://dx.doi.org/10.1016/j.ijepes.2022.108743.

[32] Wonyong Chung, et al., Graph construction method for GNN-based multivariate time-series forecasting, CMC-Comput., Mater. Continua 75 (3) (2023).

[33] Gen Li, Jason J. Jung, Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges, Inform. Fusion (2022) http://dx.doi.org/10.1016/j.inffus.2022.10.008.

[34] Enyan Dai, Jie Chen, Graph-augmented normalizing flows for anomaly detection of multiple time series, 2022, http://dx.doi.org/10.48550/arXiv.2202.07857, arXiv.

[35] Paul Boniol, Themis Palpanas, Series2graph: Graph-based subsequence anomaly detection for time series, 2022, http://dx.doi.org/10.14778/3407790.3407792, arXiv.

[36] Andrea Cini, et al., Scalable spatiotemporal graph neural networks, in: AAAI, vol. 37, (6) 2023, http://dx.doi.org/10.1609/aaai.v37i6.25880.

[37] Jalil Jabari Lotf, Mohammad Abdollahi Azgomi, Mohammad Reza Ebrahimi Dishabi, An improved influence maximization method for social networks based on genetic algorithm, Physica A 586 (2022) 126480, http://dx.doi.org/10.1016/j.physa.2021.126480, Get rights and content.

[38] Guangming Qin, et al., Graph structure learning on user mobility data for social relationship inference, in: AAAI, vol. 37, (4) 2023, http://dx.doi.org/10.1609/aaai.v37i4.25580.

[39] Xiaoyang Wang, et al., Traffic flow prediction via spatial temporal graph neural network, in: WWW, 2020, http://dx.doi.org/10.1145/3366423.3380186.

[40] Yuchen Jiang, et al., Electrical-STGCN: An electrical spatio-temporal graph convolutional network for intelligent predictive maintenance, IEEE Trans. Ind. Inform. 18 (12) (2022) 8509–8518, http://dx.doi.org/10.1109/TII.2022.3143148.

[41] Chunnan Wang, et al., Auto-STGCN: Autonomous spatial-temporal graph convolutional network search, ACM Trans. Knowl. Discov. Data 17 (5) (2023) 1–21, http://dx.doi.org/10.1145/3571285.

[42] W. Hu, J. Pang, X. Liu, D. Tian, C.W. Lin, A. Vetro, Graph signal processing for geometric data and beyond: Theory and applications, IEEE Trans. Multimed. 24 (2021) 3961–3977.

[43] Bing Yu, Haoteng Yin, Zhanxing Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, 2017, http://dx.doi.org/10.24963/ijcai.2018/505, arXiv.

[44] Jie Hu, Li Shen, Gang Sun, Squeeze-and-excitation networks, in: CVPR, 2018, http://dx.doi.org/10.48550/arXiv.1709.01507.

[45] Yaguang Li, Rose Yu, Cyrus Shahabi, Yan Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: ICLR, 2018, http://dx.doi.org/10.48550/arXiv.1707.01926.

[46] Andrew J. Patton, A review of copula models for economic time series, J. Multivariate Anal. 110 (2012) 4–18, http://dx.doi.org/10.1016/j.jmva.2012.02.021.

[47] Chao Shang, Jie Chen, Jinbo Bi, Discrete graph structure learning for forecasting multiple time series, 2021, http://dx.doi.org/10.48550/arXiv.2101.06861, arXiv.

[48] Y. Yuan, Y. Guo, K. Dehghanpour, Z. Wang, Y. Wang, Learning-based real-time event identification using rich real PMU data, IEEE Trans. Power Syst. 36 (6) (2021) 5044–5055, http://dx.doi.org/10.1109/TPWRS.2021.3081608.

[49] Michael Johnson, et al., Modelling the levels of historic waste electrical and electronic equipment in Ireland, Resour. Conserv. Recy. 131 (2018) 1–16, http://dx.doi.org/10.1016/j.resconrec.2017.11.029.

[50] Sankhadeep Chatterjee, et al., Electrical energy output prediction using cuckoo search based artificial neural network, in: Smart Trends in Systems, Security and Sustainability: Proceedings of WS4 2017, Springer, Singapore, 2018, http://dx.doi.org/10.1007/978-981-10-6916-1_26.

[51] Michael R.M. Abrigo, Inessa Love, Estimation of panel vector autoregression in stata, Stata J. 16 (3) (2016) 778–804, http://dx.doi.org/10.1177/1536867X1601600314.

[52] Yaguang Li, et al., Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, 2017, http://dx.doi.org/10.48550/arXiv.1707.01926, arXiv preprint arXiv:1707.01926.

[53] Luca Franceschi, et al., Learning discrete structures for graph neural networks, in: International Conference on Machine Learning, PMLR, 2019, PMLR 97:1972-1982, 2019.

[54] Thomas Kipf, et al., Neural relational inference for interacting systems, in: International Conference on Machine Learning, PMLR, 2018, PMLR 80:2688-2697, 2018.

[55] Shuang Han, et al., Correlational graph attention-based long short-term memory network for multivariate time series prediction, Appl. Soft Comput. 106 (2021) 107377, http://dx.doi.org/10.1016/j.asoc.2021.107377.

[56] Xihe Qiu, et al., An attentive copula-based spatio-temporal graph model for multivariate time-series forecasting, Appl. Soft Comput. 154 (2024) 111324, http://dx.doi.org/10.1016/j.asoc.2024.111324.

[57] Ashish Vaswani, et al., Attention is all you need, in: Advances in Neural Information Processing Systems, vol. 30, 2017, http://dx.doi.org/10.48550/arXiv.1706.03762.

[58] Hosagrahar V. Jagadish, et al., Big data and its technical challenges, Commun. ACM 57 (7) (2014) 86–94.
[59] Yingjuan Tang, Hongwen He, Yong Wang, Hierarchical vector transformer vehicle trajectories prediction with diffusion convolutional neural networks, Neurocomputing 580 (2024) 127526, http://dx.doi.org/10.3934/era.2023115.
[60] Zhuo Lin Li, et al., Dynamic graph structure learning for multivariate time series forecasting, Pattern Recognit. 138 (2023) 109423, http://dx.doi.org/10.1016/j.patcog.2023.109423.
[61] Kun Yi, et al., Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective, in: Advances in Neural Information Processing Systems, vol. 36, 2024.