

# SEED: Spectral Entropy-Guided Evaluation of Spatial-Temporal Dependencies for Multivariate Time Series Forecasting

Feng Xiong<sup>1</sup>, Zongxia Xie<sup>1\*</sup>, Yanru Sun<sup>1</sup>, Haoyu Wang<sup>2</sup>, Jianhong Lin<sup>1</sup>

<sup>1</sup>Tianjin University

<sup>2</sup>Fudan University

teddybear@tju.edu.cn, caddixie@hotmail.com, yanrusun@tju.edu.cn, wanghy24@m.fudan.edu.cn, ljh123@tju.edu.cn

## Abstract

Effective multivariate time series forecasting often benefits from accurately modeling complex inter-variable dependencies. However, existing attention- or graph-based methods face three key issues: (a) strong temporal self-dependencies are often disrupted by irrelevant variables; (b) softmax normalization ignores and reverses negative correlations; (c) variables struggle to perceive their temporal positions. To address these, we propose **SEED**, a Spectral Entropy-guided Evaluation framework for spatial-temporal Dependency modeling. SEED introduces a Dependency Evaluator, a key innovation that leverages spectral entropy to dynamically provide a preliminary evaluation of the spatial and temporal dependencies of each variable, enabling the model to adaptively balance Channel Independence (CI) and Channel Dependence (CD) strategies. To account for temporal regularities originating from the influence of other variables rather than intrinsic dynamics, we propose Spectral Entropy-based Fuser to further refine the evaluated dependency weights, effectively separating this part. Moreover, to preserve negative correlations, we introduce a Signed Graph Constructor that enables signed edge weights, overcoming the limitations of softmax. Finally, to help variables perceive their temporal positions and thereby construct more comprehensive spatial features, we introduce the Context Spatial Extractor, which leverages local contextual windows to extract spatial features. Extensive experiments on 12 real-world datasets from various application domains demonstrate that SEED achieves state-of-the-art performance, validating its effectiveness and generality.

**Code** — <https://github.com/saber1360/SEED>

## Introduction

Multivariate Time Series Forecasting (MTSF) is a fundamental yet highly challenging task, with broad applications in diverse domains such as financial market analysis (Baffour, Feng, and Taylor 2019), traffic flow prediction (Khan et al. 2023), energy consumption forecasting (Chengqing et al. 2023), and others (Sun et al. 2021, 2022; Wang et al. 2023). Recent advances in MTSF have primarily focused on attention-based (Liu et al. 2024; Zhang and Yan 2023;

\*Corresponding author  
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

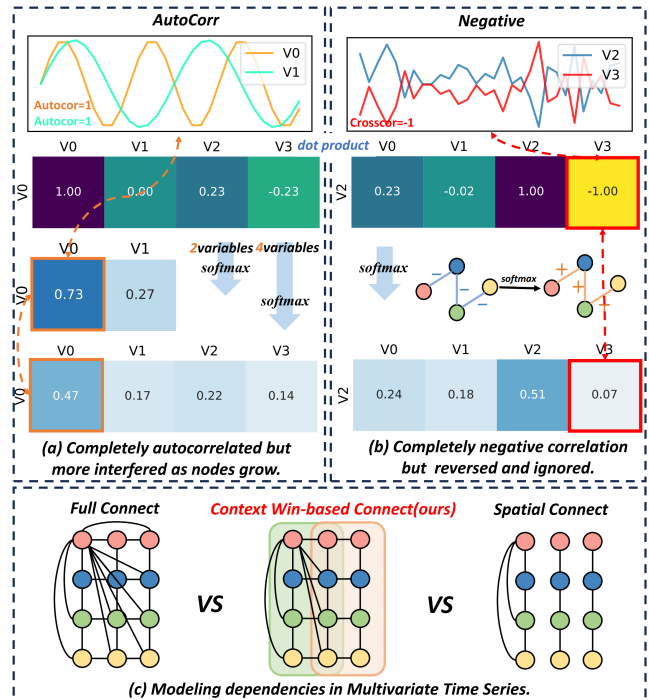


Figure 1: Challenges of temporal-spatial modeling. (a) show as the number of variables increases, they will be more influenced by other variables. (b) shows negative correlation in the time series, but is reversed by softmax function. (c) shows the method of modeling multivariate relationships. We employed local window modeling to enhance awareness of the temporal positions of spatial features.

Zhou et al. 2021a; Sun et al. 2025b) and graph-based (Cai et al. 2024; Shang et al. 2024; Wang et al. 2024a; Luo et al. 2025) frameworks to model spatial-temporal relationships. The former connects all node pairs based on the similarity of the features, while the latter restricts the connection to only consider neighboring nodes that are close in distance.

However, as shown in Figure 1, they still face the following three major challenges. **Firstly**, strong auto-correlation but as the number of nodes increases, the self-influence of these variables is weakened while interference from other

external variables is amplified. In many datasets, some variables exhibit regular patterns and are predominantly influenced by their own historical behaviors. Recent strategies, such as Channel Partiality (CP) (Qiu et al. 2025a), aim to mitigate such interference by limiting inter-variable connections and reducing noise from unrelated channels (Qiu et al. 2025b; Chen et al. 2024). Although this helps alleviate the problem to some extent, it does not fundamentally address the issue, as interference remains significant with the increase in node count. **Secondly**, strong negative correlations are often reversed and ignored. Existing methods typically apply softmax normalization to the relationship matrix, resulting in all graph or attention weights being non-negative. This causes negative correlations between variables to be reversed or overlooked. Moreover, from the perspective of graph learning, enforcing non-negativity via softmax acts as a smoothing operation that retains only low-frequency components among node pairs (Bo et al. 2021). In contrast, high-frequency components are also critical for learning node representations in heterogeneous graphs. Given the heterogeneity of variables in multivariate time series, existing graph construction methods fail to capture the diverse relationships among semantically different variables by using only non-negative scalar edge representations. **Finally**, the spatial structure cannot be modeled independently of the temporal context. Traditional spatial modeling methods typically consider only the current multivariate relationships (Xue et al. 2023) or use fully connected connections (Yi et al. 2023), the former ignored the temporal context, while the latter confused the temporal relationships. This naturally violate the real-world unified spatio-temporal dependencies, where the model lacks awareness of the temporal position of spatial features and cannot model continuous spatial-temporal interactions effectively.

To address the above challenges, we propose **SEED**, a Spectral Entropy-guided evaluation framework for spatial-temporal dependency modeling. Specifically, we decouple spatio-temporal dependencies into temporal and spatial components and introduce a Spectral Entropy to perform an initial estimation of their weights. Subsequently, we design a Spectral Entropy-based Fuser (SE-Fuser) to further fuse these two. Spectral entropy effectively characterizes the energy concentration of time series in the frequency domain, reflecting the complexity of the time series. Time series with lower spectral entropy have stronger regularity and periodicity, thus guiding each variable in balancing between channel-independent (CI) and channel-dependent (CD) strategies. For series with strong structures, more attention should be given to their own temporal dynamics, while more random series are more influenced by other variables and should focus on complex spatial dependencies. In addition, we design two Signed Graph Constructor to capture negative correlations between variables, one based on *tanh*, which provides a more dispersed weight distribution, and the other based on *softmax*, which yields more concentrated weights. Furthermore, inspired by the theories of multi-graph and signed graph modeling, we adopt a multi-head mechanism to capture richer and more diverse inter-variable relationships. Finally, to unify spatial and tempo-

ral perspectives and enhance complementarity, we introduce Context Spatial Extractor to construct more comprehensive spatial features that are aware of their temporal positions. Our main contributions are summarized as follows:

- We propose a spatial-temporal feature fusion method that reasonably balances CI and CD while preserving internal structures of individual variables.
- We introduce **SEED**, a novel spatial-temporal model that accounts for each variable’s specific attention to spatial-temporal features, and allows edges in graph to be represented by signed vectors to model complex and diverse inter-variable relationships.
- Extensive experiments demonstrate that SEED achieves state-of-the-art performance on multiple widely-used benchmark datasets.

## Related Work

### Multivariate Time Series Forecasting Models

Multivariate Time Series Forecasting (MTSF) is a key area in time series analysis (Ma et al. 2024; Qiu et al. 2024; Sun et al. 2025c). With the strong representation power of neural networks, deep learning has shown great potential in this field (Wang et al. 2025; Sun et al. 2025a). Existing MTSF models mainly focus on capturing inter-variable dependencies. CNN-based (Zhou et al. 2025, 2021b; Wu et al. 2021) and MLP-based (Ekambaram et al. 2023) methods model variable interactions adaptively, while Transformer variants employ attention for dynamic aggregation. GNN-based approaches (Yi et al. 2023; Zhao et al. 2023; Wang et al. 2024b) (e.g., GCNs, GATs) construct graphs to capture topological dependencies. Yet, some studies find that models without explicit inter-variable modeling can still perform well, likely because certain variables follow simpler dynamics, where complex inter-variable modeling is unnecessary for good predictions. Conversely, complex variables benefit from richer relational modeling. These findings inspire us to integrate both modeling philosophies for balanced performance.

### Spatial-Temporal Dependencies Modeling

In multivariate time series forecasting (MTSF), dependency modeling methods can be broadly categorized into three types: Channel-Independent (CI) strategies, Channel-Dependent (CD) strategies, and Channel Partiality (CP) strategies. CI strategies focus on temporal dependencies, typically resulting in simpler and more robust models (Sun et al. 2025b; Liu et al. 2025c). However, they often perform poorly when dealing with complex dynamics or high-dimensional data. CD strategies introduce explicit modeling of inter-variable dependencies, emphasizing spatial relationships (Wu et al. 2025). While this can provide useful information gain, it often increases model complexity and may introduce redundant information in tasks where self-dependence dominates. Recently, research has shifted toward Channel Partiality, which attempts to selectively model variable dependencies either statically or dynamically. However, such methods heavily rely on feature-level similarity

for clustering and often lack consideration of the inherent periodic patterns within the time series itself. To address this, we propose to leverage spectral entropy of the series as a dynamic criterion to balance the CI and CD strategies.

## Preliminary

### Problem Formulation

In the context of multivariate time series forecasting, let  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_C\} \in \mathbb{R}^{C \times L}$  be the input time series, where  $C$  denotes the number of channels and  $L$  denotes the length of the lookback window.  $\mathbf{x}_i \in \mathbb{R}^L$  represents one of the channels. The objective is to evaluate the future values  $\hat{X} \in \mathbb{R}^{C \times T}$ , where  $T$  denotes the length of the forecasting window. The forecast values are given by  $\hat{\mathbf{Y}} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_C\} \in \mathbb{R}^{C \times T}$ , where  $T$  denotes the forecasting horizon.

## Method

### Overall Architecture

The comprehensive architecture of SEED, as illustrated in Figure 2, comprised of five key components: **(a) Dependency Evaluator Module (DpdEva)** measures the relatively complex and random parts of variables by the spectral entropy after filtering; **(b) Temporal Attention Module (TAttn)** focuses on capturing the relational patterns within the variable itself; **(c) Context Spatial Extractor Module (CSE)** extracts the more complete spatial feature from the surrounding spatial-temporal context; **(c.1) Signed Graph Constructor Module (SGC)** constructs a non-symmetric graph with signed vector-valued edges, using a multi-head mechanism and adopting a relaxed learnable distance metric. **(d) Spectral Entropy-based Fuser Module (SE-Fuser)** considers the extent of the involvement of spatial feature in the the inherent regular patterns exhibited by the variables and fuses the temporal feature with spatial feature.

We first evaluate the inherent regular patterns of each variable in the input  $X \in \mathbb{R}^{C \times L}$  by using spectral entropy computed after filtering.

$$\text{SpEn} = \text{DpdEva}(X). \quad (1)$$

The resulting  $\text{SpEn} \in \mathbb{R}^C$  represents the spectral entropy, which is used to measure the inherent regular patterns of each variable.

We then divide the input multivariate time series  $X$  into non-overlapping patches  $\mathbf{X}'_p \in \mathbb{R}^{C \times N \times P}$ , where  $P$  denotes length of each patch and  $N = \lceil \frac{L}{P} \rceil$  denotes the number of patches. Then each patch is mapped to an embedded patch token  $\mathbf{X}_p \in \mathbb{R}^{C \times N \times D}$ .

$$\mathbf{X}'_p = \text{Patching}(X), \mathbf{X}_p = \text{Embedding}(\mathbf{X}'_p) + \text{PE}, \quad (2)$$

where PE represents the positional encoding, which is added to the embedded patches to retain the directional information and the relative positions among patches.

The patch token  $\mathbf{X}_p$  is then concurrently fed into two distinct pathways:

$$\mathcal{T} = \text{TAttn}(\mathcal{X}^{(l-1)}), \mathcal{E} = \text{CSE}(\mathcal{X}^{(l-1)}), \quad (3)$$

where  $\text{TAttn}(\cdot)$  and  $\text{CSE}(\cdot)$  are the Temporal Attention module and Context Spatial Extractor module, respectively. The output  $\mathcal{T} \in \mathbb{R}^{C \times N \times D}$  denotes temporal feature and  $\mathcal{E} \in \mathbb{R}^{C \times N \times D}$  denotes the spatial feature.

After that, the resulting  $\mathcal{T}$  and  $\mathcal{E}$  are fed into the SE-Fuser module and fused based on the SpEn.

$$\mathcal{F} = \text{SE-Fuser}(\mathcal{T}, \mathcal{E}, \text{SpEn}). \quad (4)$$

Subsequently, the output is passed through a feed-forward MLP block and skip connection:

$$\mathcal{H}^{(l)} = \text{LayerNorm}(\mathcal{X}^{(l-1)} + \mathcal{F}), \quad (5)$$

$$\mathcal{X}^{(l)} = \text{LayerNorm}(\mathcal{H}^{(l)} + \text{Linear}(\mathcal{H}^{(l)})). \quad (6)$$

Finally, we flatten the patch-level representations of each variable and project them to the output dimension:

$$\hat{Y} = \text{Projection}(\text{FlattenHead}(\mathcal{X}^{(o)})), \quad (7)$$

where  $\hat{Y} \in \mathbb{R}^{C \times T}$  denotes the forecasted series, and  $T$  is the forecasting horizon.

### Dependency Evaluator Module

The Dependency Evaluator Module aims to quantify how much of each variable's temporal dynamics can be confidently predicted using only its own historical information. To this end, we employ **spectral entropy** as a frequency-domain measure of structural complexity. Detailed spectral entropy analysis is provided in Appendix. However, given the sensitivity of spectral entropy to noise, we incorporate frequency-domain filtering to eliminate noise disturbances and improve robustness.

Given an input time series  $x_c \in \mathbb{R}^L$  for variable  $c$ , we first apply the Fourier Transform and a Plain Shaping Filter (Yi et al. 2024) to obtain a denoised frequency spectrum:

$$\mathcal{Z}_c = \mathcal{F}(x_c), \mathcal{S}_c = \mathcal{Z}_c \odot_L \mathcal{H}_\phi, \quad (8)$$

where  $\mathcal{F}$  denotes the Fourier Transform,  $\odot_L$  is the element-wise multiplication along the length dimension  $L$ ,  $\mathcal{H}_\phi \in \mathbb{C}^{1 \times L}$  is the universal plain shaping filter, and  $\mathcal{S}_c \in \mathbb{C}^{1 \times L}$  is the filtered spectrum.

We then compute the Power Spectral Density (PSD) of variable  $c$  as:

$$P_c(f_i) = |\mathcal{S}_c(f_i)|^2, \quad i = 1, 2, \dots, L, \quad (9)$$

where  $P_c(f_i)$  is the power at frequency bin  $f_i$  for variable  $c$ .

Finally, we calculate the spectral entropy of  $x_c$  as:

$$\text{SpEn}(x_c) = \frac{-\sum_{i=1}^L p_{c,i} \log(p_{c,i})}{\log(L)}, \quad (10)$$

where  $p_{c,i} = \frac{P_c(f_i)}{\sum_{j=1}^L P_c(f_j)}$  is the normalized power at frequency bin  $f_i$ , forming a valid probability distribution over frequencies.

The resulting spectral entropy  $\text{SpEn}(x_c) \in [0, 1]$  quantifies the level of uncertainty or disorder in the frequency domain. A lower SpEn value indicates that the spectral energy is more concentrated, implying stronger temporal regularity and higher self-prediction ability. In such cases, the

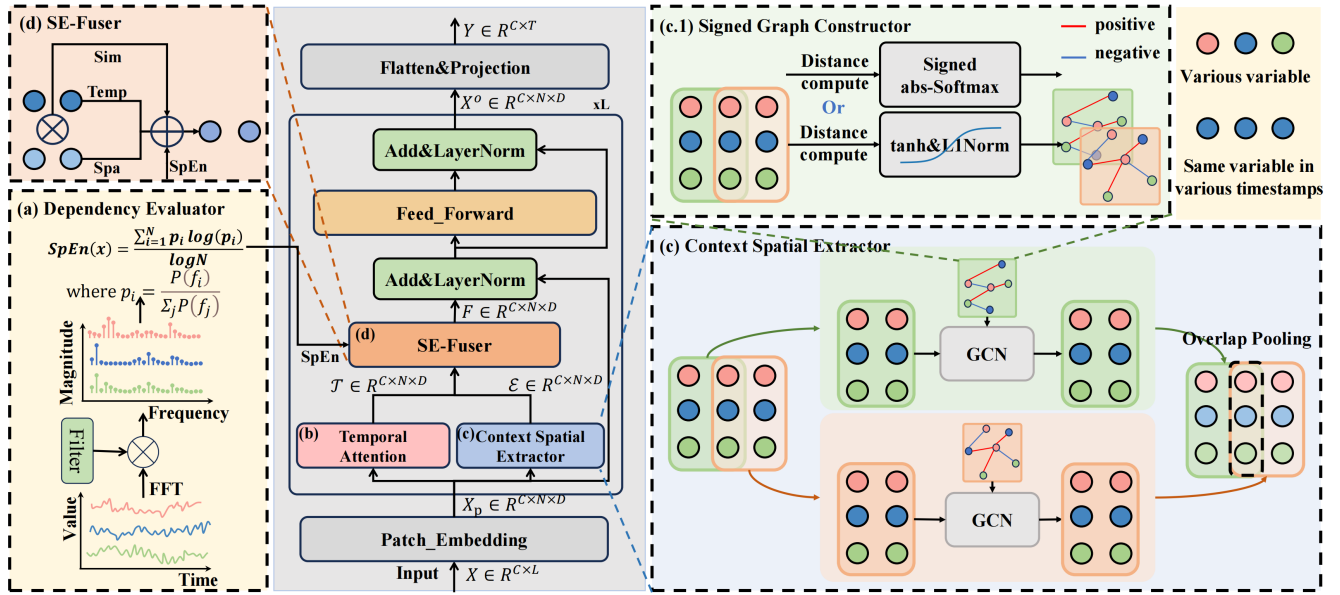


Figure 2: Overview of SEED, which comprises the following key modules: (a) **Dependency Evaluator** calculates the spectral entropy for evaluation of Spatial-Temporal Dependencies after filtering in the frequency domain; (b) **Temporal Attention** focuses on utilize its own information to capture own internal patterns; (c) **Context Spatial Extractor** extracts the more complete spatial feature from the surrounding spatial-temporal context; (c.1) **Signed Graph Constructor** allows the edge weights between nodes to be negative weights; (d) **SE-Fuser** dynamically fuses the variable’s temporal feature with spatial feature.

variable is considered to be more self-reliable, and thus self-modeling is more trustworthy. Conversely, a higher SpEn value suggests that the energy is more dispersed across frequencies, reflecting greater complexity and stochasticity, which requires the model to rely more on external contextual information gain for accurate forecasting.

We apply this computation to all variables  $c = 1, \dots, C$ , resulting in a spectral entropy vector  $\text{SpEn} \in \mathbb{R}^C$ .

### Temporal Attention Module

The Temporal Attention Module is designed to model in-travariabe dependencies across time, capturing how each variable evolves based on its own history. Similar to PatchTST (Nie et al. 2023), we apply temporal multi-head attention to each variable independently.

Given the embedded patch tokens  $\mathbf{X}_p \in \mathbb{R}^{C \times N \times D}$ , we compute temporal attention for each variable  $c$  as:

$$Q_c, K_c, V_c = \text{Linear}(\mathbf{X}_{p,c}), \quad (11)$$

$$\text{Attn}_c = \text{Softmax} \left( \frac{Q_c K_c^\top}{\sqrt{d_k}} \right) V_c, \quad (12)$$

$$\mathcal{T}_c = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O, \quad (13)$$

where  $d_k$  is the dimension of each head,  $h$  is the number of attention heads, and  $W^O$  is a learned projection matrix. The output  $\mathcal{T} \in \mathbb{R}^{C \times N \times D}$  preserves temporally structured representations for each variable.

### Context Spatial Extractor Module

To capture the complex and dynamic inter-variable dependencies, we design an Context Spatial Extractor module, which constructs local spatial-temporal interaction graphs from embedded patches and models the signed relationships across variables.

Given the embedded feature series  $X_p \in \mathbb{R}^{C \times N \times D}$ , where  $C$  is the number of variables,  $N$  is the number of patches per variable, and  $D$  is the feature dimension of each patch, we first partition the temporal dimension  $N$  into overlapping local windows of size 2 with stride 1. This results in a local patch tensor  $X_{win} \in \mathbb{R}^{C \times (N-1) \times 2 \times D}$ . Such a formulation allows the spatial interactions among variables to be aware of their temporal positions, achieving unified spatial-temporal perception.

**Signed Graph Constructor Module** For  $k$ -th local window  $X_{win}^k \in \mathbb{R}^{C \times 2 \times D}$ , we flatten the first two dimensions to obtain a set of spatial-temporal patches with  $n = C \times 2$  patches containing the enhanced local information  $\mathbf{X}_{local}^k \in \mathbb{R}^{n \times D}$ .

To model more diverse relationships among variables, we, inspired by multi-graph theory, signed graphs, attention mechanisms, divided the features into multiple heads, allowing the edges in the inter-variable relationship graph to be represented as a set of signed vectors. Specifically, we calculate the projection distances in a multi-head mechanism, allowing for modeling more complex relationships. We first divide  $\mathbf{X}_{local}^k$  into different heads  $\mathbf{X}_h^k \in \mathbb{R}^{H \times n \times \lfloor \frac{D}{H} \rfloor}$  and then calculate distances  $\text{Dist}(\cdot)$ . Here,  $H$  is the number of

heads and  $A^T$  is the transpose.

$$\text{Dist}(\mathbf{X}_h^k) = (\mathbf{X}_h^k)^T \mathbf{Q} \mathbf{X}_h^k \in \mathbb{R}^{H \times n \times n}, \quad (14)$$

where  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  denote a learnable matrix.

Next, based on above distances, we construct the signed graph. Considering that Softmax tends to highlight prominent associations, whereas methods for constructing inter-variable relationships based on CNN/MLP often result in more uniform relationships, we designed two distinct signed graph Constructor methods: one based on Softmax and another based on Tanh to explore these differences.

Specifically, we construct a signed interaction graph using two alternative strategies:

Let  $s_{ij}$  denotes the edge connecting variable  $i$  and variable  $j$  in distance  $\text{Dist}(\mathbf{X}_h^k)$ .

- **softmax-Based Signed Graph:** We then decouple the sign and magnitude: the magnitude  $|s_{ij}|$  is used to compute weights via softmax normalization, while the sign is preserved and multiplied with the attention, leading to the signed attention matrix:

$$\alpha_{ij} = \text{sign}(s_{ij}) \cdot \frac{\exp(|s_{ij}|)}{\sum_{j'=1}^n \exp(|s_{ij'}|)}. \quad (15)$$

- **tanh-Based Signed Graph:** Alternatively, we compute cosine similarity followed by a tanh transformation to retain bounded signed values. The resulting matrix is normalized using the L1 norm:

$$\alpha_{ij} = \frac{\tanh(s_{ij})}{\sum_j |\tanh(s_{ij})|}. \quad (16)$$

The absolute value of the above matrix is finally used as the criterion for construction of  $k$ -nearest neighbors (KNN), resulting in a set of graphs  $\mathbf{G} \in \mathbb{R}^{H \times n \times n}$ .

**GCN and Overlap Pooling** For the features of the  $k$ -th patch, we consider windows  $\mathbf{X}_{\text{win}}^{k-1}$  and  $\mathbf{X}_{\text{win}}^k$ .

$$E_{k-1} = \text{GCN}(\mathbf{X}_{\text{win}}^{k-1}, \mathbf{G}^{k-1}), E_k = \text{GCN}(\mathbf{X}_{\text{win}}^k, \mathbf{G}^k), \quad (17)$$

where  $\mathbf{E}_k \in \mathbb{R}^{C \times 2 \times D}$  These two are then aggregated (e.g., via mean or max pooling):

$$\mathcal{E}_k = \text{OverlapPooling}(E_{k-1}, E_k), \quad (18)$$

where OverlapPooling denotes pooling the overlapping parts from the preceding and succeeding windows, and  $\mathcal{E}_k \in \mathbb{R}^{C \times 1 \times D}$  denotes the features of the  $k$ -th patch. We apply this computation to all patch  $k = 1, \dots, N$ , resulting in a vector  $\mathcal{E} \in \mathbb{R}^{C \times N \times D}$ .

### Spectral Entropy-based Fuser Module

The Spectral Entropy-based Fuser (SE-Fuser) integrates the temporal features and spatial features based on the confidence signal from the DpdEva module. The fusion is guided by both patch-wise feature similarity and spectral entropy.

For each variable  $c$ , we compute a fusion coefficient  $\alpha_c \in [0, 1]$  based on its spectral entropy:

$$\alpha_c = 1 - \text{SpEn}(x_c), \quad (19)$$

where  $\text{SpEn}(x_c)$  denotes spectral entropy of variable  $c$ .

Let  $\text{Sim}_{c,n}$  denote the similarity between temporal features  $\mathcal{T}_{c,n}$  and spatial features  $\mathcal{E}_{c,n}$ , where  $c$  denote variable  $c$  and  $n$  denote  $n$ -th patch. The fusion weight becomes:

$$w_{c,n} = \alpha_c \cdot (1 - \text{Sim}_{c,n}). \quad (20)$$

When  $\text{Sim}_{c,n}$  is low, it suggests that the regularity exhibited by the variable has a weaker association with the spatial features, in which case we can appropriately strengthen the utilization of temporal features. Conversely, when  $\text{Sim}_{c,n}$  is high, it indicates that the variable's regularity relies more heavily on the spatial features, and we should reduce the reliance on temporal features.

Then the final fused representation is computed as a weighted combination:

$$\mathcal{F}_{c,n} = w_{c,n} \cdot \mathcal{T}_{c,n} + (1 - w_{c,n}) \cdot \mathcal{E}_{c,n}. \quad (21)$$

This allows the model to rely more on intrinsic temporal dynamics when the variable exhibits strong internal regularities, and to shift toward more complex spatial modeling when the variable exhibits higher complexity or uncertainty.

### Loss Function

The loss function of SEED consists of two components. For the predictors, the Mean Squared Error (MSE) loss is used to measure the variance between predicted values and ground truth.

$$\mathcal{L}_{\text{pred}} = \frac{1}{T} \sum_{i=0}^T \|\mathbf{y}_{:,i} - \hat{\mathbf{y}}_{:,i}\|_2^2. \quad (22)$$

Moreover, we introduce an Loss based on spectral entropy  $\mathcal{L}_{\text{SpEn}}$  to Guide the more reasonable distribution of weights.

$$\mathcal{L}_{\text{SpEn}} = \frac{1}{C} \sum_{c=1}^C \|\text{SpEn}(\mathbf{y}_{c,:}) - \text{SpEn}(\hat{\mathbf{y}}_{c,:})\|_2^2. \quad (23)$$

Therefore, the final loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda \mathcal{L}_{\text{SpEn}}, \quad (24)$$

where  $\lambda$  are the scaling factors.

## Experiments

### Experimental Details

**Datasets.** For long-term forecasting, we conduct extensive experiments on 8 widely-recognized multivariate time series forecasting datasets, including ETT (ETTh1, ETTh2, ETTm1, ETTm2), Traffic, Electricity, Weather, Solar-Energy (Miao et al. 2024a,b) datasets. For short-term forecasting, we selected 4 benchmarks from PEMS (PEMS03, PEMS04, PEMS07, PEMS08) (Liu et al. 2025b,a). The statistics of the dataset are shown in Appendix.

**Baselines.** We compare our method with SOTA representative methods, including GNN-based methods: MSGNet (Cai et al. 2024); Transformer-based methods: iTransformer (Liu et al. 2024), TQNet (Lin et al. 2025);, Cross-former (Zhang and Yan 2023) Linear-based methods: Leddam (Yu et al. 2024), SOFTS (Han et al. 2024), DUET (Qiu et al. 2025b);

Models	SEED (Ours)		TQNet CD (2025)		DUET CP (2025b)		iTransformer CD (2024)		MSGNet CD (2024)		SOFTS CD (2024)	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	<b>0.369</b>	<b>0.391</b>	<u>0.377</u>	<u>0.393</u>	0.390	<u>0.393</u>	0.407	0.410	0.398	0.411	0.393	0.403
ETTm2	<b>0.273</b>	<b>0.321</b>	<u>0.277</u>	<u>0.323</u>	0.280	0.324	0.288	0.332	0.288	0.330	0.287	0.330
ETTh1	<b>0.418</b>	<b>0.427</b>	<u>0.441</u>	<u>0.434</u>	0.443	0.436	0.454	0.447	0.452	0.452	0.449	0.442
ETTh2	<b>0.365</b>	<b>0.396</b>	<u>0.378</u>	0.402	<u>0.372</u>	<u>0.397</u>	0.383	0.407	0.396	0.417	0.385	0.408
Weather	<b>0.239</b>	<u>0.270</u>	<u>0.242</u>	<b>0.269</b>	0.251	0.273	0.258	0.279	0.249	0.278	0.255	0.278
ECL	<b>0.156</b>	<b>0.252</b>	<u>0.164</u>	0.259	0.172	<u>0.258</u>	0.178	0.270	0.194	0.300	0.174	0.264
Traffic	<b>0.404</b>	<b>0.265</b>	0.445	0.276	0.451	0.269	0.428	0.282	0.641	0.370	<u>0.409</u>	<u>0.267</u>
Solar	<u>0.218</u>	<u>0.252</u>	<b>0.198</b>	0.256	0.237	<b>0.233</b>	0.233	0.262	0.262	0.288	0.229	0.256
1st Cnt	<b>7</b>	<b>6</b>	1	1	0	1	0	0	0	0	0	0

Table 1: Results of the multivariate long-term time series forecasting task, evaluated using MSE and MAE (lower is better). The input series length  $L$  is set to 96 for all baselines. The best results are highlighted in **bold**, while the second-best results are underlined. See Appendix for full results.

Models	SEED (Ours)		iTransformer CD (2024)		Leddam CD (2024)		SOFTS CD (2024)		PatchTST CI (2023)		Crossformer CD (2023)	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	<b>0.084</b>	<b>0.188</b>	0.096	0.204	0.101	0.210	<u>0.087</u>	<u>0.192</u>	0.151	0.265	0.138	0.253
PEMS04	<b>0.080</b>	<b>0.182</b>	0.098	0.207	0.102	0.213	<u>0.091</u>	<u>0.196</u>	0.162	0.273	0.145	0.267
PEMS07	<b>0.068</b>	<b>0.165</b>	0.088	0.190	0.087	0.192	<u>0.075</u>	<u>0.173</u>	0.166	0.270	0.181	0.272
PEMS08	<b>0.080</b>	<b>0.182</b>	0.127	0.212	<u>0.102</u>	0.211	0.114	<u>0.208</u>	0.238	0.289	0.232	0.270
1st Cnt	<b>4</b>	<b>4</b>	0	0	0	0	0	0	0	0	0	0

Table 2: Short-term forecasting results. The input length  $L$  is 96. All results are averaged across three different forecasting horizons:  $T \in \{12, 24, 48\}$ . The best results are highlighted in **bold**, while the second-best results are underlined. See Appendix for full results.

**Setup.** All experiments are implemented in PyTorch and conducted on workstations equipped with 8 NVIDIA GeForce RTX 3090 24GB GPUs. We adopted a consistent experimental setup identical to that of iTransformer (Liu et al. 2024) to ensure a fair comparison. Specifically, the lookback length for all models was fixed at 96, and the SGC Module module employs a tanh-Based Signed Graph. Mean Absolute Error (MAE) and Mean Squared Error (MSE) was used as metrics.

## Main Results

**Long-term Forecasting.** Table 1 presents the predictive performance of SEED in multivariate long-term time series forecasting tasks in eight datasets. The input length  $L$  is 96 for our method and all baselines. The forecasting horizon  $T$  is  $\{96, 192, 336, 720\}$ . From the table, SEED demonstrates superior accuracy over current state-of-the-art models in most cases. Specifically, averaging MSE across all prediction lengths, SEED achieved the best performance on 6 out of the 8 datasets and secured the second-best performance on the remaining 2 datasets.

**Short-term Forecasting.** Table 2 presents the predictive performance of SEED in multivariate short-term time series forecasting tasks in eight datasets. The input length  $L$  is 96 for our method and all baselines. The forecasting horizon  $T$  is  $\{12, 24, 48\}$ . From the table, SEED consistently outperforms other methods across all 4 PEMS datasets. The insignificant improvement in SEED performance in PEMS03 and PEMS07 might be attributed to the high number of variables in these datasets, resulting in the temporal-dependent part having a relatively smaller effect.

## Ablation Study

To validate the effectiveness of SEED, we conduct a comprehensive ablation study on its architectural design.

**Spatial-Temporal Module Ablation.** **w/o-TAttn:** The Temporal Attention Module was removed. **w/o-CSE:** The Context Spatial Extractor Module was removed.

**Replace of SGC Module.** The Signed Graph Constructor Module in the aforementioned experiment uses the tanh-based signed graph. Its construction method ensures that the absolute values of the weights in the graph are relatively

Variants	ETT (avg)		Weather		ECL		Traffic		Solar-Energy	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o TAttn	0.364	0.387	0.244	0.272	0.159	0.254	0.407	0.268	0.230	0.257
w/o CSE	0.363	0.388	0.245	0.274	0.178	0.268	0.437	0.274	0.240	0.269
<b>re-S1</b>	0.360	0.386	0.243	0.272	0.177	0.272	0.411	0.257	0.226	0.253
<b>re-S2</b>	0.365	0.389	0.242	0.271	0.162	0.257	0.411	0.270	0.223	0.252
<b>re-F1</b>	0.362	0.387	0.242	0.272	0.161	0.257	0.407	0.269	0.231	0.260
<b>re-F2</b>	0.363	0.388	0.245	0.273	0.160	0.255	0.408	0.269	0.227	0.256
<b>re-F3</b>	0.374	0.392	0.374	0.392	0.170	0.256	0.235	0.300	0.297	0.340
<b>re-C1</b>	0.366	0.390	0.242	0.273	0.167	0.262	0.413	0.270	0.229	0.258
<b>re-C2</b>	0.363	0.388	0.240	0.270	0.164	0.260	0.413	0.269	0.231	0.255
<b>SEED</b>	<b>0.356</b>	<b>0.384</b>	<b>0.239</b>	<b>0.270</b>	<b>0.156</b>	<b>0.252</b>	<b>0.404</b>	<b>0.265</b>	<b>0.218</b>	<b>0.252</b>

Table 3: Ablation analysis showing averaged ETT results and other datasets. results. All results are averaged across three different forecasting horizons:  $T \in \{96, 192, 336, 720\}$ . The best results are highlighted in **bold**. See Appendix for full results.

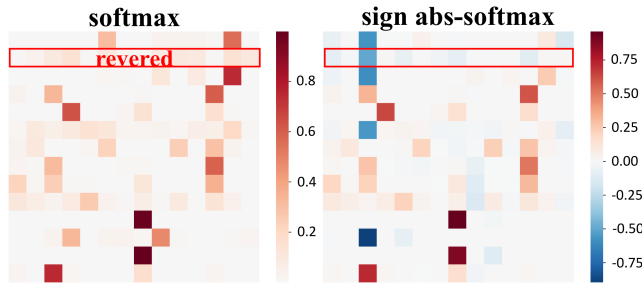


Figure 3: Above are the dependencies learned by SEED from the ETTm1. left: softmax function; right: our softmax-based signed graph.

smooth. Therefore, we further explored the differences between this construction method, which highlights key points using softmax, and the previous one. **re-S1**: The Signed Graph Constructor Module has been replaced by original softmax. That is, to ignore the negative correlation. **re-S2**: The Signed Graph Constructor Module use softmax-Based Signed Graph.

**Replace of SE-Fuser Module.** In order to verify the effectiveness of our spatiotemporal feature fusion method, we designed the following three variants: **re-F1**: Replace the spectral entropy with learnable weights as the weights. **re-F2**: The cross-use of the TAttn and CSE Module. **re-F3**: Concatenate the two features and pass them through a linear layer as the fusion weights.

**Replace of CSE Module.** We designed the following two spatial modeling methods to replace our CSE module. **re-C1**: Only consider the relationships between different variables at the same moment. **re-C2**: Use the full connection structure with all patches.

## Model Analysis

**Case study on Signed Graph modeling.** As shown in Figure 3, we visualize the dependencies of selected filters of one batch in the ETTm1. In fact, it shows that constructing

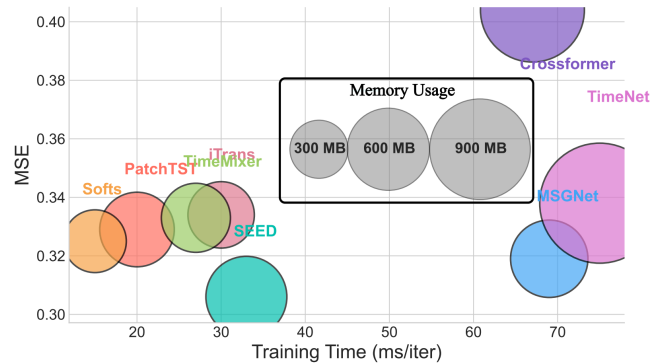


Figure 4: Performance Analysis of SEED: Assessing MSE, Training Time, and Memory Usage, evaluated on the ETTm1 Dataset with a 96-In/96-Out Setup.

the graph using softmax may lead to a group of negatively correlated connections being reversed into positive ones.

**Model Efficiency** We compare the SEED model against other models in terms of forecasting accuracy, memory usage, and training speed. The results, as shown in Figure 4, indicate SEED has achieved better prediction performance while maintaining a good training speed.

## Conclusion

Considering the varying complexity and internal regularity of different variables in multivariate time series, we propose SEED to adaptively evaluate and integrate spatial-temporal dependencies. Technically, the model uses spectral entropy to evaluate and measure the spatial-temporal dependencies. For complex prediction tasks, it prioritizes considering spatial dependencies, while for simpler patterns, it retains more temporal dependencies. Furthermore, our model utilizes multi-head mechanism and negative correlation to capture more intricate inter-variable relationships. Extensive experiments demonstrate that SEED consistently achieves state-of-the-art performance in both long-term and short-term forecasting tasks.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62376194; and in part by the China Scholarship Council under Grant 202406250137.

## References

- Baffour, A. A.; Feng, J.; and Taylor, E. K. 2019. A hybrid artificial neural network-GJR modeling approach to forecasting currency exchange rate volatility. *Neurocomputing*, 365: 285–301.
- Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 3950–3957.
- Cai, W.; Liang, Y.; Liu, X.; Feng, J.; and Wu, Y. 2024. MSGNet: Learning Multi-Scale Inter-series Correlations for Multivariate Time Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Chen, J.; Lenssen, J. E.; Feng, A.; Hu, W.; Fey, M.; Tassulas, L.; Leskovec, J.; and Ying, R. 2024. From similarity to superiority: Channel clustering for time series forecasting. *Advances in Neural Information Processing Systems*, 37: 130635–130663.
- Chengqing, Y.; Guangxi, Y.; Chengming, Y.; Yu, Z.; and Xiwei, M. 2023. A multi-factor driven spatiotemporal wind power prediction model based on ensemble deep graph attention reinforcement learning networks. *Energy*, 263: 126034.
- Ekambaram, V.; Jati, A.; Nguyen, N.; Sinthong, P.; and Kalagnanam, J. 2023. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 459–469.
- Han, L.; Chen, X.-Y.; Ye, H.-J.; and Zhan, D.-C. 2024. SOFTS: Efficient Multivariate Time Series Forecasting with Series-Core Fusion. In *Advances in Neural Information Processing Systems*.
- Khan, A.; Fouda, M. M.; Do, D.-T.; Almaleh, A.; and Rahman, A. U. 2023. Short-term traffic prediction using deep learning long short-term memory: Taxonomy, applications, challenges, and future trends. *IEEE Access*, 11: 94371–94391.
- Lin, S.; Chen, H.; Wu, H.; Qiu, C.; and Lin, W. 2025. Temporal Query Network for Efficient Multivariate Time Series Forecasting. *arXiv preprint arXiv:2505.12917*.
- Liu, C.; Miao, H.; Xu, Q.; Zhou, S.; Long, C.; Zhao, Y.; Li, Z.; and Zhao, R. 2025a. Efficient Multivariate Time Series Forecasting via Calibrated Language Models with Privileged Knowledge Distillation. In *ICDE*.
- Liu, C.; Xu, Q.; Miao, H.; Yang, S.; Zhang, L.; Long, C.; Li, Z.; and Zhao, R. 2025b. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *AAAI*, volume 39, 18780–18788.
- Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. *International Conference on Learning Representations*.
- Liu, Z.; Luo, Y.; Li, B.; Eldele, E.; Wu, M.; and Ma, Q. 2025c. Learning Soft Sparse Shapes for Efficient Time-Series Classification. In *Forty-second International Conference on Machine Learning*.
- Luo, Y.; Zhang, B.; Liu, Z.; and Ma, Q. 2025. Hi-Patch: Hierarchical Patch GNN for Irregular Multivariate Time Series. In *Forty-second International Conference on Machine Learning*.
- Ma, Q.; Liu, Z.; Zheng, Z.; Huang, Z.; Zhu, S.; Yu, Z.; and Kwok, J. T. 2024. A survey on time-series pre-trained models. *IEEE Transactions on Knowledge and Data Engineering*.
- Miao, H.; Liu, Z.; Zhao, Y.; Guo, C.; Yang, B.; Zheng, K.; and Jensen, C. S. 2024a. Less is more: Efficient time series dataset condensation via two-fold modal matching. *PVLDB*, 18(2): 226–238.
- Miao, H.; Zhao, Y.; Guo, C.; Yang, B.; Zheng, K.; Huang, F.; Xie, J.; and Jensen, C. S. 2024b. A unified replay-based continuous learning framework for spatio-temporal prediction on streaming data. In *ICDE*, 1050–1062.
- Nie, Y.; H. Nguyen, N.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations*.
- Qiu, X.; Cheng, H.; Wu, X.; Hu, J.; Guo, C.; and Yang, B. 2025a. A comprehensive survey of deep learning for multivariate time series forecasting: A channel strategy perspective. *arXiv preprint arXiv:2502.10721*.
- Qiu, X.; Hu, J.; Zhou, L.; Wu, X.; Du, J.; Zhang, B.; Guo, C.; Zhou, A.; Jensen, C. S.; Sheng, Z.; et al. 2024. TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods. *Proceedings of the VLDB Endowment*, 17(9): 2363–2377.
- Qiu, X.; Wu, X.; Lin, Y.; Guo, C.; Hu, J.; and Yang, B. 2025b. Duet: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 1185–1196.
- Shang, Z.; Chen, L.; Wu, B.; and Cui, D. 2024. AdaMSHyper: adaptive multi-scale hypergraph transformer for time series forecasting. *Advances in Neural Information Processing Systems*, 37: 33310–33337.
- Sun, Y.; Xie, Z.; Chen, D.; Eldele, E.; and Hu, Q. 2025a. Hierarchical classification auxiliary network for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 20743–20751.
- Sun, Y.; Xie, Z.; Chen, Y.; and Hu, Q. 2022. Accurate solar wind speed prediction with multimodality information. *Space: Science & Technology*.
- Sun, Y.; Xie, Z.; Chen, Y.; Huang, X.; and Hu, Q. 2021. Solar Wind Speed Prediction With Two-Dimensional Attention Mechanism. *Space Weather*, 19(7): e2020SW002707.
- Sun, Y.; Xie, Z.; Eldele, E.; Chen, D.; Hu, Q.; and Wu, M. 2025b. Learning Pattern-Specific Experts for Time Series

- Forecasting Under Patch-level Distribution Shift. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Sun, Y.; Xie, Z.; Xing, H.; Yu, H.; and Hu, Q. 2025c. PPGF: Probability Pattern-Guided Time Series Forecasting. *IEEE Transactions on Neural Networks and Learning Systems*.
- Wang, C.; Qi, Q.; Wang, J.; Sun, H.; Zhuang, Z.; Wu, J.; Zhang, L.; and Liao, J. 2025. Chattime: A unified multimodal time series foundation model bridging numerical and textual data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 12694–12702.
- Wang, C.; Zhuang, Z.; Qi, Q.; Wang, J.; Wang, X.; Sun, H.; and Liao, J. 2023. Drift doesn't matter: Dynamic decomposition with diffusion reconstruction for unstable multivariate time series anomaly detection. *Advances in neural information processing systems*, 36: 10758–10774.
- Wang, Y.; Sun, H.; Wang, C.; Zhu, M.; Wang, J.; Tang, W.; Qi, Q.; Zhuang, Z.; and Liao, J. 2024a. Interdependency matters: graph alignment for multivariate time series anomaly detection. In *2024 IEEE International Conference on Data Mining (ICDM)*, 869–874. IEEE.
- Wang, Y.; Xu, Y.; Yang, J.; Wu, M.; Li, X.; Xie, L.; and Chen, Z. 2024b. Fully-connected spatial-temporal graph for multivariate time-series data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 15715–15724.
- Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34: 22419–22430.
- Wu, X.; Qiu, X.; Li, Z.; Wang, Y.; Hu, J.; Guo, C.; Xiong, H.; and Yang, B. 2025. CATCH: Channel-Aware Multivariate Time Series Anomaly Detection via Frequency Patching. In *The Thirteenth International Conference on Learning Representations*.
- Xue, W.; Zhou, T.; Wen, Q.; Gao, J.; Ding, B.; and Jin, R. 2023. Card: Channel aligned robust blend transformer for time series forecasting. *arXiv preprint arXiv:2305.12095*.
- Yi, K.; Fei, J.; Zhang, Q.; He, H.; Hao, S.; Lian, D.; and Fan, W. 2024. Filtarnet: Harnessing frequency filters for time series forecasting. *Advances in Neural Information Processing Systems*, 37: 55115–55140.
- Yi, K.; Zhang, Q.; Fan, W.; He, H.; Hu, L.; Wang, P.; An, N.; Cao, L.; and Niu, Z. 2023. FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in neural information processing systems*, 36: 69638–69660.
- Yu, G.; Zou, J.; Hu, X.; Aviles-Rivero, A. I.; Qin, J.; and Wang, S. 2024. Revitalizing Multivariate Time Series Forecasting: Learnable Decomposition with Inter-Series Dependencies and Intra-Series Variations Modeling. In *Forty-first International Conference on Machine Learning*.
- Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*.
- Zhao, K.; Guo, C.; Cheng, Y.; Han, P.; Zhang, M.; and Yang, B. 2023. Multiple time series forecasting with dynamic graph modeling. *Proceedings of the VLDB Endowment*, 17(4): 753–765.
- Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021a. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11106–11115.
- Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021b. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11106–11115.
- Zhou, P.; Liu, Y.; Liang, J.; Song, Q.; and Li, X. 2025. CrossLinear: Plug-and-Play Cross-Correlation Embedding for Time Series Forecasting with Exogenous Variables. *arXiv preprint arXiv:2505.23116*.