

Article

SpeQNet: Query-Enhanced Spectral Graph Filtering for Spatiotemporal Forecasting

Zongyao Feng and Konstantin Markov * 

Department of Computer and Information Systems, University of Aizu, Aizuwakamatsu 965-8580, Japan; d8252101@u-aizu.ac.jp

* Correspondence: markov@u-aizu.ac.jp

Abstract

Accurate spatiotemporal forecasting underpins high-stakes decision making in smart urban systems, from traffic control and energy scheduling to environment monitoring. Yet two persistent gaps limit current models: (i) spatial modules are often biased toward low-pass smoothing and struggle to reconcile slow global trends with sharp local dynamics; and (ii) the graph structure required for forecasting is frequently latent, while learned graphs can be unstable when built from temporally derived node features alone. We propose SpeQNet, a query-enhanced spectral graph filtering framework that jointly strengthens node representations and graph construction while enabling frequency-selective spatial reasoning. SpeQNet injects global spatial context into temporal embeddings via lightweight learnable spatiotemporal queries, learns a task-oriented adaptive adjacency matrix, and refines node features with an enhanced ChebNetII-based spectral filtering block equipped with channel-wise recalibration and nonlinear refinement. Across twelve real-world benchmarks spanning traffic, electricity, solar power, and weather, SpeQNet achieves state-of-the-art performance and delivers consistent gains on large-scale graphs. Beyond accuracy, SpeQNet is interpretable and robust: the learned spectral operators exhibit a consistent band-stop-like frequency shaping behavior, and performance remains stable across a wide range of Chebyshev polynomial orders. These results suggest that query-enhanced spatiotemporal representation learning and adaptive spectral filtering form a complementary and effective foundation for effective spatiotemporal forecasting.

Keywords: spatiotemporal modeling; time-series forecasting; spectral graph filtering; graph signal processing; multivariate forecasting



Academic Editors: Youness El Hamzaoui and José Alfredo Hernández Pérez

Received: 8 January 2026

Revised: 21 January 2026

Accepted: 22 January 2026

Published: 23 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In an era of smart urban systems [1], accurate spatiotemporal forecasting is essential for avoiding crises and optimizing resources, underpinning applications such as traffic management [2], energy scheduling [3], weather prediction [4], and environmental monitoring [5]. These systems generate multivariate time series (MTS) from spatially distributed sensors, who interact over an underlying spatial structure. This structure is often conveniently modeled as a graph, where each node corresponds to a time series and edges encode physical or functional relationships [6,7]. Graph neural networks (GNNs) are therefore a natural choice for modeling spatial dependencies and have been widely adopted in spatiotemporal forecasting architectures [8,9]. However, despite their theoretical appeal, graph-based models often fall short of strong Transformer-based architectures [10,11]. This shortfall arises from two central limitations. First, spatiotemporal data contain both slowly varying

global trends and rapidly changing local patterns [12], dynamics that are well suited for spectral analysis. Yet classic ChebNet [13], despite its strong theoretical expressiveness [14], frequently underperforms spatial GNNs like GCN on real-world datasets [15]. Second, real-world datasets rarely provide high-quality graph structures that faithfully reflect the spatial dependencies needed for forecasting, and existing graph learning methods often struggle to infer these dependencies reliably. Together, these limitations highlight the need for a spatiotemporal forecasting framework that models the spectral dependencies more effectively while learning graph structures in a robust and data-driven manner.

A large body of work on graph-based spatiotemporal forecasting is built on GCN [15], where each node aggregates information from its neighbors via a learned local kernel. While simple and effective, such neighbor aggregation behaves primarily as a low-pass filter on the graph [16–18]. It promotes smoothness across connected nodes and tends to suppress high-frequency components that encode sharp local variations or heterogeneous interactions. As a result, GCN often struggles to simultaneously capture slowly varying global trends, e.g., flow propagation across a corridor of roads, and rapidly changing local patterns, e.g., sudden congestion at intersection [18,19]. In contrast, spectral GNNs offer a more flexible alternative, and by parameterizing filters in the eigenbasis of the Laplacian graph, they operate directly in the frequency domain [20], enabling the learning of arbitrary spectral responses. This allows the model to selectively amplify or attenuate specific frequency bands, representing a much richer set of spatial dependencies that span both smooth global structures and localized, fast-changing behaviors. Although classic ChebNet often underperforms on real-world datasets [21] despite its theoretical expressiveness, recent advances such as ChebNetII [22] refine the Chebyshev approximation and demonstrate strong performance on node classification tasks, making spectral filtering a more promising backbone for spatiotemporal forecasting.

However, the expressive power of any GNN is fundamentally limited by the quality of the graph it operates on and the informativeness of the node representations propagated over that graph. Predefined physical graphs [8,23], such as distance- or connectivity-based structures, offer useful geometric priors but fail to capture functional dependencies such as hubs, bottlenecks, high-demand regions. To overcome these limitations, modern spatiotemporal forecasting models primarily adopt two families of graph learning strategies. Adaptive graph learning [24–26] generates a static, input-independent adjacency matrix by parameterizing node affinities through learnable embeddings, which is effective for capturing stable spatial patterns. Attention-based constructions [27,28], in contrast, derive spatial dependencies from the input itself, the query–key interactions implicitly define sample-specific adjacency patterns and can, in principle, capture non-recurring or abrupt spatial changes. Yet recent works employing these strategies often overlook the role of node representations, which are typically derived solely from temporal encoders and thus lack explicit spatial context. This way, they place the full burden of discovering meaningful spatial structure either on the graph parameters or on attention weights. This disconnect between node representation learning and graph construction leaves room for performance improvement, and calls for a framework where spatiotemporally enriched node features and adaptive graphs are learned in a mutually reinforcing way.

To address the above limitations, we propose SpeQNet, a spatiotemporal forecasting model that jointly learns a frequency-aware spatiotemporal representation and a task-relevant graph structure. Building on ChebNetII, our spectral graph filtering block integrates node-wise feature recalibration and nonlinear transformations to refine graph signal processing. To enhance the spatiotemporal learning, we introduce a learnable spatiotemporal query module that injects global spatial context into temporal node representations. These representations improve the forecasting accuracy while promoting adaptive graph

learning with a clearer topology. Extensive experiments on 12 real-world forecasting benchmarks spanning traffic, energy, and climate demonstrate that SpeQNet achieves state-of-the-art performance.

Our contributions are summarized as follows:

- Frequency-aware spatial modeling—We design an enhanced ChebNetII-based spectral graph filtering block that captures both global and local spatial dependencies, boosting forecasting accuracy.
- Query-enhanced node representations—We introduce a lightweight spatiotemporal query that injects global spatial context into temporal node representations. The enriched spatiotemporal representations contribute to consistent forecasting improvements.

The remainder of this paper is organized as follows. Section 2 reviews related work on multivariate time-series forecasting. Section 3 introduces the problem formulation and introduces the necessary graph spectral preliminaries. Section 4 presents the proposed SpeQNet framework in detail, describing the query-enhanced spatiotemporal representations, adaptive graph learning, and spectral graph filtering modules. Section 5 reports experimental settings and results on twelve real-world benchmarks, including Ablation Studies, Spectral Filter Response Analysis, Sensitivity to Chebyshev Polynomial Order Analysis, Scaling analysis, Computational Efficiency Analysis, Graph Interpretability Analysis, and Qualitative Forecasting Visualization. Finally, Section 6 concludes the paper and discusses limitations and directions for future work.

2. Related Works

Recent progress in multivariate time-series forecasting has been shaped by two contrasting modeling philosophies: treating each variable independently, or explicitly learning relationships across variables. Channel-independent models simplify multivariate forecasting by decomposing the problem into a collection of univariate tasks, an idea popularized by PatchTST [29] and the family of linear models such as DLinear and RLinear [30,31], and recent LLM-based methods like Chronos [32] and TimeLLM [33] that adapt large language models for univariate processing. These methods achieve strong long-term forecasting accuracy by focusing exclusively on temporal dynamics while avoiding the risk of spurious cross-variable correlations. Similar motivations underlie temporal CNN architectures such as TimesNet [34] and CycleNet [35], which extract periodic or multi-scale temporal patterns with minimal reliance on variable interactions. Although effective on long-sequence datasets where intra-variable temporal structures dominate, these approaches inherently discard spatial or functional relationships—an omission that limits their applicability to domains where interactions between variables are essential, such as traffic networks, environment systems, and energy grids.

Motivated by this limitation, a second line of works develop models that explicitly capture cross-variable dependencies. Transformer-based methods such as iTransformer [10] and Crossformer [36] treat each variable as a token and use attention mechanisms to discover multivariate correlations, while architectures like TimeXer [11] further combine patch-level and variate-level representations to reconcile temporal patterns with cross-series interactions. Recent extensions, such as VCformer [37] incorporating lagged correlations, MOIRAI [38] enabling universal pre-training, and UniTS [39] facilitating multi-task unification, build on these foundations to address diverse forecasting scenarios. These approaches provide flexible relational modeling but operate purely in the input space: attention weights are derived directly from observed time series, making them sensitive to noise, abrupt fluctuations, and nonstationarity [40].

Recent GNN-based architectures advance this paradigm by incorporating structural inductive biases, such as predefined or learned graphs over variables, to better encode

domain-specific topologies. Classical spatiotemporal GNNs pioneered the use of spatial graphs to capture interactions in applications like traffic and energy forecasting. Many rely on predefined graphs constructed from spatial distances, connectivity, or correlation matrices, as in STGCN [8], DCRNN [23], AST-GCN [41], and STSGCN [42]. To address scenarios where the true graph is latent, adaptive graph learning has emerged, inferring structures from learnable node embeddings or similarity functions, as demonstrated in MTGNN [26], Graph WaveNet [43], and AGCRN [25]. Building on these foundations, modern models like MSGNet [44] and CrossGNN [45] incorporate multi-scale or relation-aware mechanisms, leveraging adaptive graphs to handle complex, evolving dependencies. While effective, these approaches present opportunities to further integrate frequency-domain insights and enhance representation learning for improved robustness in noisy, nonstationary environments.

These observations highlight a gap in the current landscape: while channel-independent methods excel at temporal modeling and Transformer/GNN-based methods capture variable interactions, few provide a unified framework that combines stable, task-oriented graph structures with frequency-aware spatiotemporal representations.

3. Preliminaries

3.1. Problem Definition

We formulate multivariate time-series forecasting over a set of variables that form a graph. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ denote a graph with $N = |\mathcal{V}|$ nodes, where each node corresponds to a variate in the multivariate sequence. The edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ captures pairwise spatial dependencies, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the weighted adjacency matrix. In many real-world forecasting tasks, the graph structure is latent or partially known. In our SpeQNet, the adjacency matrix is adaptively learned and refined through the query-enhanced spatiotemporal representations (see Section 4). Given a historical sequence:

$$\mathbf{X}_{1:T} \in \mathbb{R}^{N \times T},$$

the goal is to predict the future values:

$$\hat{\mathbf{Y}}_{T+1:T+S} \in \mathbb{R}^{N \times S}.$$

The forecasting model learns a function f_θ that satisfies

$$\hat{\mathbf{Y}}_{T+1:T+S} = f_\theta(\mathbf{X}_{1:T}; \mathbf{A}_\theta), \quad (1)$$

where \mathbf{A}_θ denotes the adaptive adjacency matrix learned within the model. In SpeQNet, the adaptive adjacency matrix \mathbf{A}_θ is learned via a softmax normalization and therefore has non-negative entries with row-wise sums equal to one. The core challenge lies in modeling the multi-scale spatiotemporal dependencies among the N variables over T time steps, capturing both temporal patterns and spatial dependencies. SpeQNet addresses these challenges by jointly modeling the spatiotemporal representations via a frequency-aware spectral graph filtering block.

3.2. Graph Spectral Theory and Filtering

Given an adaptive adjacency matrix \mathbf{A}_θ and a degree matrix \mathbf{D} , we adopt the random-walk normalized Laplacian [46]:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1} \mathbf{A}_\theta, \quad (2)$$

which models stochastic diffusion dynamics.

The spectral graph theory provides a principled framework for analyzing graph-structured signals through the eigendecomposition of the Laplacian:

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top, \quad (3)$$

where \mathbf{U} contains orthonormal eigenvectors and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_N)$ contains eigenvalues that lie in $[0, 2]$ [22]. The eigenvectors form the graph Fourier basis, and the eigenvalues represent frequencies. Low frequencies correspond to smooth global modes, whereas high frequencies capture abrupt variations. The formulation of Equation (3) is introduced to convey the classical spectral interpretation of graph signals. In SpeQNet, the adjacency matrix \mathbf{A}_θ is adaptively learned and is not necessarily symmetric; accordingly, the model does not explicitly rely on an orthonormal eigendecomposition in practice. Instead, spectral filtering is implemented via polynomial approximations, as detailed in the next subsection, which remain well defined for general weighted graphs. Given a graph signal $x \in \mathbb{R}^N$, its graph Fourier transform and inverse-transform are

$$\hat{x} = \mathbf{U}^\top x, \quad x = \mathbf{U}\hat{x}. \quad (4)$$

A spectral filter $g(\mathbf{\Lambda})$ can modify graph signals by amplifying or suppressing specific frequencies:

$$y = \mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^\top x. \quad (5)$$

Learning such filters allows selective modeling of both smooth spatial correlations and localized variations. However, computing Equation (5) directly is computationally expensive for large graphs and is therefore typically approximated in practice.

3.3. Chebyshev-II Approximation

Classical ChebNet [13] approximate spectral filters using Chebyshev polynomials:

$$y \approx \sum_{k=0}^K w_k T_k(\hat{\mathbf{L}})x, \quad (6)$$

that satisfy the recurrence:

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_k(x) &= 2x T_{k-1}(x) - T_{k-2}(x). \end{aligned} \quad (7)$$

In Equation (6), the $K \in \mathbb{N}$ denotes the Chebyshev polynomial order, which determines the degree of the polynomial approximation and the effective K -hop spatial receptive field of the spectral filter. In principle, K can take any positive integer value, trading off spectral expressivity against computational cost. In this study, we fix $K = 5$ for all main experiments, representing a moderate polynomial order that balances spectral expressivity and computational efficiency. The w_k are the trainable Chebyshev coefficients and the scaled Laplacian $\hat{\mathbf{L}}$, whose eigenvalues lie in $[-1, 1]$, is defined as

$$\hat{\mathbf{L}} = \frac{2}{\lambda_{\max}} \mathbf{L} - \mathbf{I}. \quad (8)$$

In practice, λ_{\max} is usually set to 2 [15,22], which avoids the need for eigendecomposition.

Although efficient and expressive, ChebNet learns illegal coefficients when approximating analytic filter functions, causing degraded generalization due to over-fitting [22]. ChebNetII [22] addresses this limitation by replacing direct coefficient learning with Cheby-

shev interpolation. Instead of directly learning the Chebyshev coefficients w_k , ChebNetII learns the filter values at the Chebyshev nodes. These values are then projected into the Chebyshev basis, ensuring stable and theoretically valid spectral coefficients. Specifically, ChebNetII reparameterizes the coefficients as

$$w_k = \frac{2}{K+1} \sum_{j=0}^K \gamma_j T_k(x_j), \quad (9)$$

where $x_j = \cos\left(\frac{j+1/2}{K+1}\pi\right)$ are the Chebyshev nodes, and γ_j are learnable parameters representing the filter evaluated at those nodes. Substituting these coefficients into the spectral filtering operation yields the ChebNetII propagation rule:

$$y \approx \frac{2}{K+1} \sum_{k=0}^K \sum_{j=0}^K \gamma_j T_k(x_j) T_k(\hat{\mathbf{L}}) x. \quad (10)$$

The spectral filters are able to approximate a wide range of spectral shapes—including low-pass, band-pass, and band-rejection patterns—demonstrating strong expressive power and stable optimization. Therefore, ChebNetII is well suited for spatiotemporal forecasting scenarios where different frequency bands capture distinct spatial interaction patterns.

4. Methodology

4.1. Overview of SpeQNet

Given a multivariate time-series window $\mathbf{X} \in \mathbb{R}^{N \times T}$, where N denotes the number of variates and T is the lookback length, SpeQNet predicts the future horizon of length S for all nodes. Figure 1 shows the overview of SpeQNet, an encoder-only design with six main components: (1) Temporal Encoding: Raw time series are first embedded by a temporal module to capture temporal dependencies, producing node-wise temporal representations. (2) Query-Enhanced Spatial Embedding: A set of lightweight learnable spatiotemporal queries interacts with the input to inject global spatial context, a linear projection layer then aligns the spatial enhanced signal to the embedding space. The temporal and spatial embeddings are then fused together via addition, yielding the spatiotemporal representations. (3) Adaptive Graph Learning: Using two learnable low-rank embeddings, SpeQNet constructs a task-oriented adaptive adjacency matrix that captures latent and dataset-specific spatial dependencies. (4) Node Feature Extraction: Based on the query-enhanced spatiotemporal representations, a two-layer MLP extracts the node representation. (5) Spectral Graph Filtering: Node features are processed by an enhanced Chebyshev-based spectral filtering block, where ChebNetII enables flexible frequency-selective filtering, complemented by channel-wise recalibration and nonlinear refinement to improve expressiveness and stability. (6) Forecasting Head: The refined representations are finally mapped to future predictions via a lightweight prediction head. By integrating query-enhanced spatiotemporal representations with adaptive spectral graph filtering, SpeQNet enables expressive, interpretable, and robust spatiotemporal forecasting under latent graph structures.

Formally, starting from \mathbf{X} , SpeQNet computes temporal embeddings $\mathbf{E}_{\text{temp}} \in \mathbb{R}^{N \times E}$ and query-enhanced spatial embeddings $\mathbf{E}_{\text{spatial}} \in \mathbb{R}^{N \times E}$, where \mathbf{E} denotes the embedding dimension. These representations are then fused to form a spatiotemporal representation $\mathbf{E}_{\text{st}} \in \mathbb{R}^{N \times E}$. The adaptive graph learning constructs an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ from learnable low-rank embeddings and a two-layer MLP initializes node features $\mathbf{H}_0 \in \mathbb{R}^{N \times E}$ from \mathbf{E}_{st} . A stack of spectral graph filtering blocks then refines $\{\mathbf{H}_\ell\}_{\ell=1}^L$ using ChebNetII-based filters over the learned graph. Finally, the forecasting head applies a per-node linear projection to obtain $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times S}$.

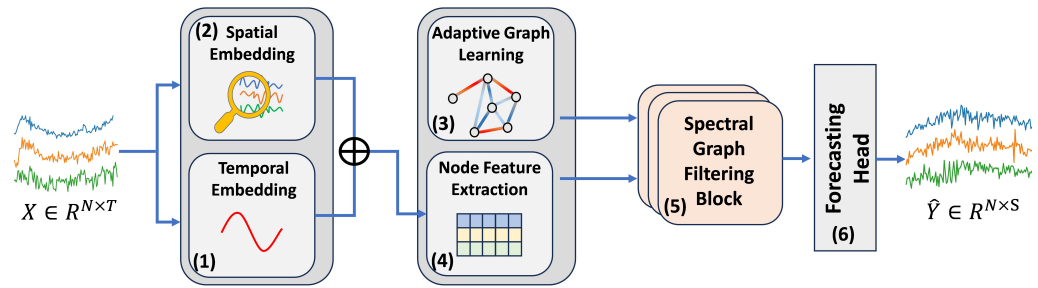


Figure 1. Overview of SpeQNet: query-enhanced spectral graph filtering for spatiotemporal forecasting. SpeQNet adopts an encoder-only architecture that integrates query-enhanced spatiotemporal representation learning, adaptive graph construction, and Chebyshev-based spectral graph filtering to model latent spatial dependencies and generate future forecasts.

4.2. Query-Enhanced Spatiotemporal Representation

Given X , SpeQNet first encodes the temporal dynamics of each variate and then injects global spatial context via learnable queries. Figure 2 illustrates the query-enhanced spatiotemporal representation module, where the Temporal Embedding Block (1), performs temporal embedding to encode historical dynamics of each variate, while the Query-Enhanced Spatial Embedding Block (2) applies spatial multi-head attention to inject global spatial context via learnable spatiotemporal queries. The two embeddings are fused to produce the spatiotemporal representation used for subsequent modeling.

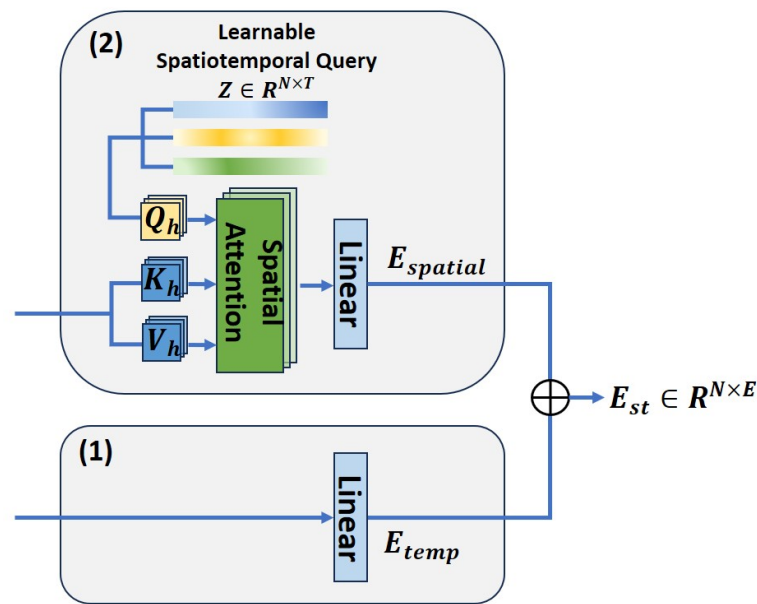


Figure 2. Query-enhanced spatiotemporal representation in SpeQNet. Block (1) encodes temporal embeddings, while Block (2) injects global spatial context via learnable queries; the two embeddings are fused to form the spatiotemporal representation.

4.2.1. Temporal Embedding

The temporal embedding layer extracts periodicities and trends from each variate’s length- T history into a compact representation of dimension E by applying a shared linear projection along the temporal dimension:

$$E_{temp} = XW_{temp}^T + b_{temp}, \quad E_{temp} \in \mathbb{R}^{N \times E}, \tag{11}$$

where $W_{temp} \in \mathbb{R}^{E \times T}$ and $b_{temp} \in \mathbb{R}^E$ are learnable parameters shared across all variates.

4.2.2. Query-Enhanced Spatial Embedding

While \mathbf{E}_{temp} summarizes the temporal evolution of each variate independently, it is agnostic to global spatial patterns shared across the dataset. To inject such information, SpeQNet initializes a learnable spatiotemporal query matrix $\mathbf{Z} \in \mathbb{R}^{N \times T}$, where the i -th row \mathbf{z}_i acts as a global query pattern associated with the i -th variate. We employ multi-head attention (MHA) to let these queries attend over the raw input series:

$$\mathbf{Q}_h = \mathbf{Z}\mathbf{W}_h^Q, \quad \mathbf{K}_h = \mathbf{X}\mathbf{W}_h^K, \quad \mathbf{V}_h = \mathbf{X}\mathbf{W}_h^V, \tag{12}$$

where $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{T \times d_h}$ are head-specific projection matrices and d_h is the head dimension. The spatially enriched signal for head h is then

$$\mathbf{X}_{\text{spatial},h} = \text{softmax}\left(\frac{\mathbf{Q}_h\mathbf{K}_h^\top}{\sqrt{d_h}}\right)\mathbf{V}_h \in \mathbb{R}^{N \times d_h}, \tag{13}$$

which aggregates information from all variates according to learned query–key similarities. After concatenating all heads and applying an output projection, we obtain

$$\mathbf{X}_{\text{spatial}} = \text{MHA}(\mathbf{Z}, \mathbf{X}, \mathbf{X}) \in \mathbb{R}^{N \times T}. \tag{14}$$

To align this spatial enriched signal with the temporal embedding space, $\mathbf{X}_{\text{spatial}}$ is passed through a linear projection:

$$\mathbf{E}_{\text{spatial}} = \mathbf{X}_{\text{spatial}}\mathbf{W}_{\text{spatial}}^\top + \mathbf{b}_{\text{spatial}}, \quad \mathbf{E}_{\text{spatial}} \in \mathbb{R}^{N \times E}, \tag{15}$$

where $\mathbf{W}_{\text{spatial}} \in \mathbb{R}^{E \times L}$ and $\mathbf{b}_{\text{spatial}} \in \mathbb{R}^E$ are learnable parameters.

4.2.3. Spatiotemporal Fusion

Finally, SpeQNet fuses temporal and spatial information via additive integration:

$$\mathbf{E}_{\text{st}} = \mathbf{E}_{\text{temp}} + \mathbf{E}_{\text{spatial}}, \tag{16}$$

yielding the query-enhanced spatiotemporal representation $\mathbf{E}_{\text{st}} \in \mathbb{R}^{N \times E}$ that serves as the input to the subsequent adaptive graph learning and spectral filtering blocks.

4.3. Adaptive Graph Learning and Node Feature Extraction

SpeQNet learns a task-specific adjacency matrix using learnable low-rank embeddings and initializes graph node features from the spatiotemporal representations. Figure 3 shows the adaptive graph learning and node feature extraction module, where Block (3) constructs a task-oriented adaptive adjacency matrix from the learnable low-rank embeddings and Block (4) extracts node features through a lightweight feature transformation of the query-enhanced spatiotemporal representations. Specifically, we introduce two learnable embedding matrices:

$$\mathbf{E}_1 \in \mathbb{R}^{N \times r}, \quad \mathbf{E}_2 \in \mathbb{R}^{N \times r},$$

where $r \ll N$ is a rank hyperparameter. We construct a dense affinity matrix and apply a row-wise softmax to obtain the adjacency matrix:

$$\mathbf{A}_\theta = \text{softmax}(\text{ReLU}(\mathbf{E}_1\mathbf{E}_2^\top)) \in \mathbb{R}^{N \times N}. \tag{17}$$

This low-rank factorization provides an efficient, input-independent graph that captures stable spatial patterns, while remaining fully learnable and task-driven. We then obtain the scaled Laplacian \hat{L} using Equations (2) and (8) with $\lambda_{max} = 2$.

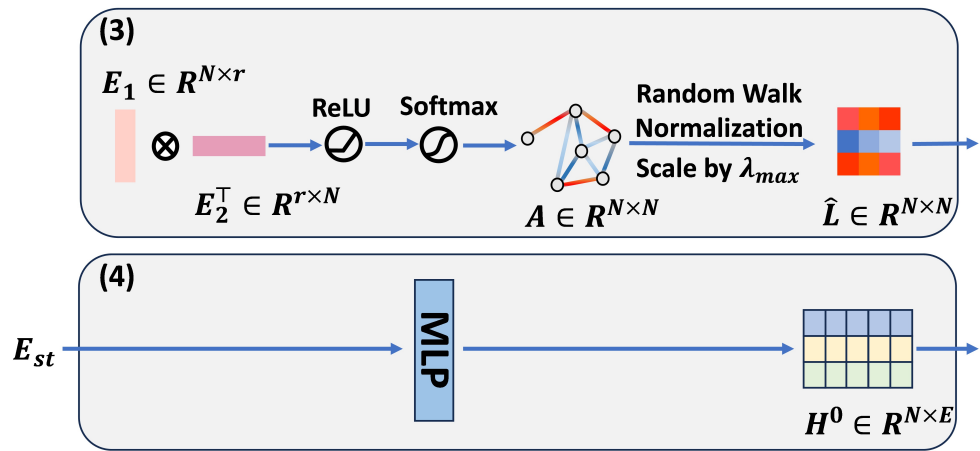


Figure 3. Adaptive graph learning and node feature extraction in SpeQNet. Block (3) learns a task-oriented adaptive adjacency matrix, and Block (4) initializes node features from the query-enhanced spatiotemporal representations for subsequent spectral graph filtering.

Simultaneously, given the spatiotemporal representation E_{st} , SpeQNet initializes the graph node features with a lightweight transformation. We apply a two-layer node-wise MLP with a ReLU activation to refine the representations and map them into the hidden feature space used by the spectral blocks. Denote this module by $MLP_{node}(\cdot)$, and the initial graph signal is

$$H^0 = MLP_{node}(E_{st}), \quad H^0 \in \mathbb{R}^{N \times E}. \tag{18}$$

This step improves feature expressiveness while keeping the architecture simple and computationally efficient.

4.4. Spectral Graph Filtering Block

To model spatial dependencies across the learned graph, SpeQNet applies \mathcal{L} spectral graph filtering blocks repeatedly over the node features, where

$$\ell \in \{1, \dots, \mathcal{L}\}$$

indexes the block. Each block combines (i) ChebNetII spectral filtering, (ii) node-wise feature recalibration, and (iii) a residual MLP, as visualized in Figure 4. The learnable filters enable flexible frequency-selective modulation of node representations, capturing critical spatial dependencies across various frequencies.

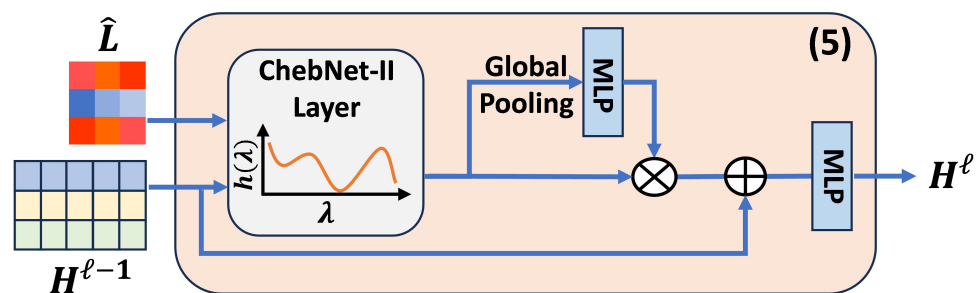


Figure 4. Spectral graph filtering block in SpeQNet. Given node features and the learned adaptive graph, the Block (5) applies ChebNetII-based spectral filtering to operate in the graph frequency domain.

Given node features $\mathbf{H}_{\ell-1} \in \mathbb{R}^{N \times E}$ and the scaled Laplacian $\hat{\mathbf{L}}$, the ChebNetII layer implements a spectral filter $g_\ell(\cdot)$ as in Equation (10):

$$\tilde{\mathbf{H}}^\ell = g_\ell(\hat{\mathbf{L}}, \mathbf{H}^{\ell-1}) \in \mathbb{R}^{N \times E}. \tag{19}$$

This yields flexible and stable spectral responses that can realize low-pass, band-pass, or band-stop behaviors over the learned graph.

To further emphasize informative channels and suppress noisy ones, we apply a squeeze-and-excitation [47] style recalibration on $\tilde{\mathbf{H}}^\ell$. First, we aggregate node-wise statistics:

$$\mathbf{s}_\ell = \frac{1}{N} \sum_{i=1}^N \tilde{\mathbf{H}}^\ell(i, :) \in \mathbb{R}^E. \tag{20}$$

Then, a two-layer bottleneck MLP followed by a sigmoid activation function produces channel-wise gates:

$$\alpha_\ell = \text{sigmoid}(\text{MLP}_{\text{se}}(\mathbf{s}_\ell)) \in \mathbb{R}^E. \tag{21}$$

The recalibrated features are

$$\hat{\mathbf{H}}^\ell(i, :) = \alpha_\ell \odot \tilde{\mathbf{H}}^\ell(i, :), \quad i = 1, \dots, N. \tag{22}$$

Finally, we refine $\hat{\mathbf{H}}^\ell$ with an MLP:

$$\mathbf{H}^\ell = \text{MLP}_{\text{spec}}(\hat{\mathbf{H}}^\ell + \mathbf{H}^{\ell-1}). \tag{23}$$

producing frequency-aware node representations that encode both global and local spatial dependencies on the learned graph.

4.5. Forecasting Head

The forecasting head maps the final node representations to the prediction horizon. Given the output of the last spectral graph filtering block, $\mathbf{H}^\mathcal{L} \in \mathbb{R}^{N \times E}$, we apply a shared linear projection along the feature dimension:

$$\hat{\mathbf{Y}} = \mathbf{H}^\mathcal{L} \mathbf{W}_{\text{proj}}^\top + \mathbf{b}_{\text{proj}}, \quad \hat{\mathbf{Y}} \in \mathbb{R}^{N \times S}, \tag{24}$$

where $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{S \times E}$ and $\mathbf{b}_{\text{proj}} \in \mathbb{R}^S$. The same projection is applied to every node, consistent with the view that each variate shares a common forecasting mechanism but interacts through the learned graph and spectral filters.

5. Experiments

5.1. Experimental Settings

5.1.1. Datasets

We evaluate SpeQNet on twelve widely used multivariate time-series benchmarks that cover both long-term and short-term forecasting. The long-term forecasting group includes four ETT subsets (ETTh1, ETTh2, ETTm1, and ETTm2), Weather, Solar-Energy, Electricity, and Traffic, which span energy, climate, electricity, and transportation domains. The short-term group contains four traffic datasets from the PEMS collection (PEMS03, PEMS04, PEMS07, and PEMS08). All datasets used in this work are publicly available and are accessed following the benchmark configurations and data links provided in the iTransformer repository [10]. For the four PEMS traffic datasets and the four ETT benchmarks, we follow a train/validation/test split of 6/2/2, consistent with prior work. The remaining datasets (Weather, Solar-Energy, Electricity, and Traffic) follow a standard

7/1/2 split. All reported results, including those of SpeQNet and the baselines, are based on these predefined splits. The main statistics are summarized in Table 1.

Table 1. Dataset statistics.

| Task | Dataset | Variates | Frequency | Domain |
|---------------------------|--------------|----------|-----------|----------------|
| Long-term Forecasting | ETTM1 | 7 | 15 min | Electricity |
| | ETTM2 | 7 | 15 min | Electricity |
| | ETTh1 | 7 | 1 h | Electricity |
| | ETTh2 | 7 | 1 h | Electricity |
| | Electricity | 321 | 1 h | Electricity |
| | Traffic | 862 | 1 h | Transportation |
| | Weather | 21 | 10 min | Weather |
| Short-term Forecasting | Solar-Energy | 137 | 10 min | Solar Power |
| | PEMS03 | 358 | 5 min | Transportation |
| | PEMS04 | 307 | 5 min | Transportation |
| | PEMS07 | 883 | 5 min | Transportation |
| | PEMS08 | 170 | 5 min | Transportation |

5.1.2. Baselines

We select baseline methods to provide a comprehensive and representative comparison across major modeling paradigms for multivariate time-series forecasting, including linear models, convolutional architectures, Transformer-based methods, and graph-based approaches. These methods are widely adopted in the recent spatiotemporal forecasting literature and are commonly used as reference baselines on the evaluated benchmarks. We compare SpeQNet with eleven representative baselines spanning a diverse range of multivariate time-series forecasting frameworks, including channel-independent models, transformer-based models, convolutional architectures, and graph-based methods. Specifically, the baselines include RLinear [31], DLinear [30], TiDE [48], PatchTST [29], TimesNet [34], CycleNet [35], SCINet [49], Crossformer [36], iTransformer [10], MSGNet [44], and TimeXer [11]. For baseline results, we primarily report results provided in the iTransformer paper, which evaluates a wide range of state-of-the-art methods across all considered datasets under a unified experimental protocol. For those baselines that are not reported in iTransformer such as TimeXer, CycleNet, and MSGNet, we report the results as provided in their original publications.

5.1.3. Experiment Setup

SpeQNet is implemented in PyTorch 2.3.1 using Python 3.11, and its overall training and inference procedure is summarized in Algorithm 1. All experiments are conducted on two NVIDIA RTX 3090 GPUs (NVIDIA Corporation, Santa Clara, CA, USA) with an AMD EPYC 7301 CPU (AMD, Santa Clara, CA, USA). We optimize the model using the Adam optimizer with Huber loss for a maximum of 30 epochs, and apply early stopping based on validation loss with a patience of 3 epochs. A learning rate scheduler with exponential decay is applied, where the learning rate is multiplied by a factor of 0.8 at each epoch. The model checkpoint achieving the lowest validation loss is saved and used for final evaluation. The hyperparameters are dataset-dependent and summarized in Table 2, where E denotes the model embedding dimension, Blocks indicates the number of stacked spectral graph filtering blocks, Batch is the batch size, and LR is the initial learning rate. For all datasets, we set the chebyshev polynomial order K to 5, and the number of attention-head to 4. We adopt a unified experimental protocol widely used in recent forecasting studies. The input window length is fixed to $T = 96$ time steps for all datasets. Prediction horizons are set to $S \in \{96, 192, 336, 720\}$ for long-term forecasting benchmarks

and $S \in \{12, 24, 48, 96\}$ for short-term forecasting benchmarks. This setting evaluates both short-range and long-range forecasting performance under a consistent lookback window, enabling fair and direct comparisons across methods. Forecasting performance is evaluated using Mean Squared Error (MSE) and Mean Absolute Error (MAE) on the test set for each prediction horizon, following standard practice in multivariate time-series forecasting.

Algorithm 1 SpeQNet forward pass and training procedure

Input: Historical window $\mathbf{X} \in \mathbb{R}^{N \times T}$, horizon S , Chebyshev order K , number of spectral blocks L

Output: Forecast $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times S}$

- 1: **Temporal embedding:** $\mathbf{E}_{\text{temp}} \leftarrow \mathbf{X}\mathbf{W}_{\text{temp}}^{\top} + \mathbf{b}_{\text{temp}}$ ▷ Equation (11)
- 2: **Query-enhanced spatial embedding:** initialize learnable queries $\mathbf{Z} \in \mathbb{R}^{N \times T}$
- 3: $\mathbf{X}_{\text{spatial}} \leftarrow \text{MHA}(\mathbf{Z}, \mathbf{X}, \mathbf{X})$ ▷ Equations (12)–(14)
- 4: $\mathbf{E}_{\text{spatial}} \leftarrow \mathbf{X}_{\text{spatial}}\mathbf{W}_{\text{spatial}}^{\top} + \mathbf{b}_{\text{spatial}}$ ▷ Equation (15)
- 5: **Fusion:** $\mathbf{E}_{st} \leftarrow \mathbf{E}_{\text{temp}} + \mathbf{E}_{\text{spatial}}$ ▷ Equation (16)
- 6: **Adaptive graph learning:** $\mathbf{A}_{\theta} \leftarrow \text{softmax}(\text{ReLU}(\mathbf{E}_1\mathbf{E}_2^{\top}))$ ▷ Equation (17)
- 7: Compute $\hat{\mathbf{L}}$ from \mathbf{A}_{θ} using Equations (2) and (8) with $\lambda_{\text{max}} = 2$
- 8: **Node feature init:** $\mathbf{H}^0 \leftarrow \text{MLP}_{\text{node}}(\mathbf{E}_{st})$ ▷ Equation (18)
- 9: **for** $\ell = 1$ to L **do**
- 10: $\tilde{\mathbf{H}}^{\ell} \leftarrow g_{\ell}(\hat{\mathbf{L}}, \mathbf{H}^{\ell-1}; K)$ ▷ Equation (19), ChebNetII (Equation (10))
- 11: $\mathbf{s}^{\ell} \leftarrow \frac{1}{N} \sum_{i=1}^N \tilde{\mathbf{H}}^{\ell}(i, :)$ ▷ Equation (20)
- 12: $\alpha^{\ell} \leftarrow \sigma(\text{MLP}_{\text{se}}(\mathbf{s}^{\ell}))$ ▷ Equation (21)
- 13: $\hat{\mathbf{H}}^{\ell}(i, :) \leftarrow \alpha^{\ell} \odot \tilde{\mathbf{H}}^{\ell}(i, :), \forall i$ ▷ Equation (22)
- 14: $\mathbf{H}^{\ell} \leftarrow \text{MLP}_{\text{spec}}(\hat{\mathbf{H}}^{\ell} + \mathbf{H}^{\ell-1})$ ▷ Equation (23)
- 15: **end for**
- 16: **Forecast head:** $\hat{\mathbf{Y}} \leftarrow \mathbf{H}^L\mathbf{W}_{\text{proj}}^{\top} + \mathbf{b}_{\text{proj}}$ ▷ Equation (24)
- 17: **Training:** minimize $\mathcal{L}(\hat{\mathbf{Y}}, \mathbf{Y})$ (Huber loss) with Adam (Section 5.1)
- 18: **return** $\hat{\mathbf{Y}}$

Table 2. Hyperparameter settings of SpeQNet across different datasets.

| Dataset | E | Blocks | Batch | LR |
|--------------|-----|--------|-------|--------|
| ETTh1 | 512 | 3 | 32 | 0.0001 |
| ETTh2 | 128 | 3 | 32 | 0.0001 |
| ETTm1 | 128 | 3 | 32 | 0.0001 |
| ETTm2 | 128 | 3 | 32 | 0.0001 |
| Electricity | 512 | 4 | 16 | 0.0005 |
| Solar-Energy | 512 | 3 | 32 | 0.0005 |
| Traffic | 512 | 5 | 16 | 0.001 |
| Weather | 512 | 4 | 64 | 0.0003 |
| PEMS03 | 512 | 5 | 32 | 0.001 |
| PEMS04 | 512 | 5 | 32 | 0.0005 |
| PEMS07 | 512 | 5 | 32 | 0.001 |
| PEMS08 | 512 | 5 | 32 | 0.0005 |

5.2. Main Results

We now present the main empirical results of SpeQNet across all twelve datasets. Detailed quantitative results are reported in Tables 3 and 4, which list MSE and MAE for each horizon as well as averages across horizons.

Table 3. Results for long-term multivariate time-series forecasting. All models use a fixed lookback length of $T = 96$. Baseline results are taken from original or authoritative subsequent publications, with the source of reported results cited in the table header. The row “# 1st” reports the number of times each method achieves the best average performance across all datasets, counted separately for MSE and MAE. Best results are highlighted in **red bold**, and second-best results are in **blue underlined**.

| Models | | SpeQNet | | TimeXer (2024) [11] | | CycleNet (2024) [35] | | iTransformer (2024) [10] | | MSGNet (2024) [44] | | TimesNet (2023) [10] | | PatchTST (2023) [10] | | Crossformer (2023) [10] | | DLinear (2023) [10] | | SCINet (2022) [10] | |
|-------------|-----|--------------|--------------|---------------------|--------------|----------------------|--------------|--------------------------|-------|--------------------|-------|----------------------|-------|----------------------|--------------|-------------------------|-------|---------------------|-------|--------------------|-------|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | 96 | 0.373 | <u>0.396</u> | 0.382 | 0.403 | <u>0.375</u> | 0.395 | 0.386 | 0.405 | 0.390 | 0.411 | 0.384 | 0.402 | 0.414 | 0.419 | 0.423 | 0.448 | 0.386 | 0.400 | 0.654 | 0.599 |
| | 192 | 0.428 | 0.423 | <u>0.429</u> | 0.435 | 0.436 | <u>0.428</u> | 0.441 | 0.436 | 0.443 | 0.442 | 0.436 | 0.429 | 0.460 | 0.445 | 0.471 | 0.474 | 0.437 | 0.432 | 0.719 | 0.631 |
| | 336 | <u>0.473</u> | 0.446 | 0.468 | <u>0.448</u> | 0.496 | 0.455 | 0.487 | 0.458 | 0.482 | 0.469 | 0.491 | 0.469 | 0.501 | 0.466 | 0.570 | 0.546 | 0.481 | 0.459 | 0.778 | 0.659 |
| | 720 | 0.509 | <u>0.483</u> | 0.469 | 0.461 | 0.520 | 0.484 | 0.503 | 0.491 | <u>0.496</u> | 0.488 | 0.521 | 0.500 | 0.500 | 0.488 | 0.653 | 0.621 | 0.519 | 0.516 | 0.836 | 0.699 |
| | Avg | <u>0.446</u> | 0.437 | 0.437 | 0.437 | 0.457 | 0.441 | 0.454 | 0.448 | 0.453 | 0.453 | 0.458 | 0.450 | 0.469 | 0.455 | 0.529 | 0.522 | 0.456 | 0.452 | 0.747 | 0.647 |
| ETTh2 | 96 | 0.263 | 0.329 | <u>0.286</u> | <u>0.338</u> | 0.298 | 0.344 | 0.297 | 0.349 | 0.329 | 0.371 | 0.340 | 0.374 | 0.302 | 0.348 | 0.745 | 0.584 | 0.333 | 0.387 | 0.707 | 0.621 |
| | 192 | 0.321 | 0.365 | <u>0.363</u> | <u>0.389</u> | 0.372 | 0.396 | 0.380 | 0.400 | 0.402 | 0.414 | 0.402 | 0.414 | 0.388 | 0.400 | 0.877 | 0.656 | 0.477 | 0.476 | 0.860 | 0.689 |
| | 336 | 0.365 | 0.396 | <u>0.414</u> | <u>0.423</u> | 0.431 | 0.439 | 0.428 | 0.432 | 0.440 | 0.445 | 0.452 | 0.452 | 0.426 | 0.433 | 1.043 | 0.731 | 0.594 | 0.541 | 1.000 | 0.744 |
| | 720 | <u>0.427</u> | <u>0.444</u> | 0.408 | 0.432 | 0.450 | 0.458 | <u>0.427</u> | 0.445 | 0.480 | 0.477 | 0.462 | 0.468 | 0.431 | 0.446 | 1.104 | 0.763 | 0.831 | 0.657 | 1.249 | 0.838 |
| | Avg | 0.344 | 0.384 | <u>0.368</u> | <u>0.396</u> | 0.388 | 0.409 | 0.383 | 0.407 | 0.413 | 0.427 | 0.414 | 0.427 | 0.387 | 0.407 | 0.942 | 0.684 | 0.559 | 0.515 | 0.954 | 0.723 |
| ETTm1 | 96 | 0.312 | 0.350 | <u>0.318</u> | <u>0.356</u> | 0.319 | 0.360 | 0.334 | 0.368 | 0.319 | 0.366 | 0.338 | 0.375 | 0.329 | 0.367 | 0.404 | 0.426 | 0.345 | 0.372 | 0.418 | 0.438 |
| | 192 | 0.358 | 0.373 | 0.362 | 0.383 | <u>0.360</u> | <u>0.381</u> | 0.377 | 0.391 | 0.377 | 0.397 | 0.374 | 0.387 | 0.367 | 0.385 | 0.450 | 0.451 | 0.380 | 0.389 | 0.439 | 0.450 |
| | 336 | 0.389 | 0.395 | <u>0.395</u> | 0.407 | 0.389 | <u>0.403</u> | 0.426 | 0.420 | 0.417 | 0.422 | 0.410 | 0.411 | 0.399 | 0.410 | 0.532 | 0.515 | 0.413 | 0.413 | 0.490 | 0.485 |
| | 720 | 0.457 | 0.434 | <u>0.452</u> | 0.441 | 0.447 | 0.441 | 0.491 | 0.459 | 0.487 | 0.463 | 0.478 | 0.450 | 0.454 | <u>0.439</u> | 0.666 | 0.589 | 0.474 | 0.453 | 0.595 | 0.550 |
| | Avg | 0.379 | 0.388 | <u>0.382</u> | 0.397 | 0.379 | <u>0.396</u> | 0.407 | 0.410 | 0.400 | 0.412 | 0.400 | 0.406 | 0.387 | 0.400 | 0.513 | 0.495 | 0.403 | 0.407 | 0.485 | 0.481 |
| ETTm2 | 96 | 0.175 | <u>0.256</u> | <u>0.171</u> | <u>0.256</u> | 0.163 | 0.246 | 0.180 | 0.264 | 0.182 | 0.266 | 0.187 | 0.267 | 0.175 | 0.259 | 0.287 | 0.366 | 0.193 | 0.292 | 0.286 | 0.377 |
| | 192 | 0.242 | 0.300 | <u>0.237</u> | <u>0.299</u> | 0.229 | 0.290 | 0.250 | 0.309 | 0.248 | 0.306 | 0.249 | 0.309 | 0.241 | 0.302 | 0.414 | 0.492 | 0.284 | 0.362 | 0.399 | 0.445 |
| | 336 | 0.298 | <u>0.336</u> | <u>0.296</u> | <u>0.338</u> | 0.284 | 0.327 | 0.311 | 0.348 | 0.312 | 0.346 | 0.321 | 0.351 | 0.305 | 0.343 | 0.597 | 0.542 | 0.369 | 0.427 | 0.637 | 0.591 |
| | 720 | 0.408 | 0.400 | <u>0.392</u> | <u>0.394</u> | 0.389 | 0.391 | 0.412 | 0.407 | 0.414 | 0.404 | 0.408 | 0.403 | 0.402 | 0.400 | 1.730 | 1.042 | 0.554 | 0.522 | 0.960 | 0.735 |
| | Avg | 0.281 | 0.323 | <u>0.274</u> | <u>0.322</u> | 0.266 | 0.314 | 0.288 | 0.332 | 0.289 | 0.330 | 0.291 | 0.333 | 0.281 | 0.326 | 0.757 | 0.611 | 0.350 | 0.401 | 0.571 | 0.537 |
| Electricity | 96 | 0.121 | 0.225 | 0.140 | 0.242 | <u>0.136</u> | <u>0.229</u> | 0.148 | 0.240 | 0.165 | 0.274 | 0.168 | 0.272 | 0.181 | 0.270 | 0.219 | 0.314 | 0.197 | 0.282 | 0.247 | 0.345 |
| | 192 | 0.141 | 0.242 | 0.157 | 0.256 | <u>0.152</u> | <u>0.244</u> | 0.162 | 0.253 | 0.185 | 0.292 | 0.184 | 0.289 | 0.188 | 0.274 | 0.231 | 0.322 | 0.196 | 0.285 | 0.257 | 0.255 |
| | 336 | 0.160 | 0.260 | 0.176 | 0.275 | <u>0.170</u> | <u>0.264</u> | 0.178 | 0.269 | 0.197 | 0.304 | 0.198 | 0.300 | 0.204 | 0.293 | 0.246 | 0.337 | 0.209 | 0.301 | 0.269 | 0.369 |
| | 720 | 0.196 | 0.290 | <u>0.211</u> | 0.306 | 0.212 | <u>0.299</u> | 0.225 | 0.317 | 0.231 | 0.332 | 0.220 | 0.320 | 0.246 | 0.324 | 1.730 | 1.042 | 0.245 | 0.333 | 0.299 | 0.390 |
| | Avg | 0.155 | 0.255 | 0.171 | 0.270 | <u>0.168</u> | <u>0.259</u> | 0.178 | 0.270 | 0.194 | 0.301 | 0.193 | 0.295 | 0.205 | 0.290 | 0.244 | 0.334 | 0.212 | 0.300 | 0.268 | 0.365 |

Table 3. Cont.

| Models | | SpeQNet | | TimeXer (2024) [11] | | CycleNet (2024) [35] | | iTransformer (2024) [10] | | MSGNet (2024) [44] | | TimesNet (2023) [10] | | PatchTST (2023) [10] | | Crossformer (2023) [10] | | DLinear (2023) [10] | | SCINet (2022) [10] | |
|--------------|----------|--------------|--------------|---------------------|--------------|----------------------|--------------|--------------------------|--------------|--------------------|-------|----------------------|-------|----------------------|--------------|-------------------------|-------|---------------------|-------|--------------------|-------|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| Solar-Energy | 96 | 0.170 | 0.230 | 0.215 | 0.295 | <u>0.190</u> | 0.247 | 0.203 | <u>0.237</u> | 0.210 | 0.246 | 0.250 | 0.292 | 0.234 | 0.286 | 0.310 | 0.331 | 0.290 | 0.378 | 0.237 | 0.344 |
| | 192 | 0.188 | 0.243 | 0.236 | 0.301 | <u>0.210</u> | 0.266 | 0.233 | <u>0.261</u> | 0.265 | 0.290 | 0.296 | 0.318 | 0.267 | 0.310 | 0.734 | 0.725 | 0.320 | 0.398 | 0.280 | 0.380 |
| | 336 | 0.202 | 0.251 | 0.252 | 0.307 | <u>0.217</u> | <u>0.266</u> | 0.248 | 0.273 | 0.294 | 0.318 | 0.319 | 0.330 | 0.290 | 0.315 | 0.750 | 0.735 | 0.353 | 0.415 | 0.304 | 0.389 |
| | 720 | 0.215 | 0.257 | 0.244 | 0.305 | <u>0.223</u> | <u>0.266</u> | 0.249 | 0.275 | 0.285 | 0.315 | 0.338 | 0.337 | 0.289 | 0.317 | 0.769 | 0.765 | 0.356 | 0.413 | 0.308 | 0.388 |
| | Avg | 0.194 | 0.245 | 0.237 | 0.302 | <u>0.210</u> | <u>0.261</u> | 0.233 | 0.262 | 0.263 | 0.292 | 0.301 | 0.319 | 0.270 | 0.307 | 0.641 | 0.639 | 0.330 | 0.401 | 0.282 | 0.375 |
| Traffic | 96 | <u>0.426</u> | 0.249 | 0.428 | 0.271 | 0.458 | 0.296 | 0.395 | <u>0.268</u> | 0.608 | 0.349 | 0.593 | 0.321 | 0.462 | 0.290 | 0.522 | 0.290 | 0.650 | 0.396 | 0.788 | 0.499 |
| | 192 | <u>0.446</u> | 0.262 | 0.448 | 0.282 | 0.457 | 0.294 | 0.417 | <u>0.276</u> | 0.634 | 0.371 | 0.617 | 0.336 | 0.466 | 0.290 | 0.530 | 0.293 | 0.598 | 0.370 | 0.789 | 0.505 |
| | 336 | <u>0.469</u> | 0.277 | 0.473 | 0.289 | 0.470 | 0.299 | 0.433 | <u>0.283</u> | 0.669 | 0.388 | 0.629 | 0.336 | 0.482 | 0.300 | 0.558 | 0.305 | 0.605 | 0.373 | 0.797 | 0.508 |
| | 720 | 0.564 | 0.335 | 0.516 | <u>0.307</u> | <u>0.502</u> | 0.314 | 0.467 | 0.302 | 0.729 | 0.420 | 0.640 | 0.350 | 0.514 | 0.320 | 0.589 | 0.328 | 0.645 | 0.394 | 0.841 | 0.523 |
| | Avg | 0.476 | 0.281 | <u>0.466</u> | 0.287 | 0.472 | 0.301 | 0.428 | <u>0.282</u> | 0.660 | 0.382 | 0.620 | 0.336 | 0.481 | 0.300 | 0.550 | 0.304 | 0.625 | 0.383 | 0.804 | 0.509 |
| Weather | 96 | <u>0.158</u> | 0.199 | 0.157 | <u>0.205</u> | <u>0.158</u> | <u>0.203</u> | 0.174 | 0.214 | 0.163 | 0.212 | 0.172 | 0.220 | 0.177 | 0.210 | <u>0.158</u> | 0.230 | 0.196 | 0.255 | 0.221 | 0.306 |
| | 192 | 0.207 | 0.251 | 0.204 | 0.247 | 0.207 | 0.247 | 0.221 | 0.254 | 0.211 | 0.254 | 0.219 | 0.261 | 0.225 | <u>0.250</u> | <u>0.206</u> | 0.277 | 0.237 | 0.296 | 0.261 | 0.340 |
| | 336 | <u>0.262</u> | <u>0.290</u> | 0.261 | <u>0.290</u> | <u>0.262</u> | 0.289 | 0.278 | 0.296 | 0.273 | 0.299 | 0.280 | 0.306 | 0.278 | <u>0.290</u> | 0.272 | 0.335 | 0.283 | 0.335 | 0.309 | 0.378 |
| | 720 | 0.347 | 0.343 | 0.340 | <u>0.341</u> | <u>0.344</u> | 0.344 | 0.358 | 0.349 | 0.351 | 0.348 | 0.365 | 0.359 | 0.354 | 0.340 | 0.398 | 0.418 | 0.345 | 0.381 | 0.377 | 0.427 |
| | Avg | <u>0.243</u> | 0.271 | 0.241 | 0.271 | <u>0.243</u> | 0.271 | 0.258 | 0.278 | 0.249 | 0.278 | 0.259 | 0.287 | 0.259 | <u>0.273</u> | 0.259 | 0.315 | 0.265 | 0.317 | 0.292 | 0.363 |
| # 1st | 4 | 7 | <u>2</u> | <u>2</u> | <u>2</u> | <u>2</u> | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 4. Results for short-term multivariate time-series forecasting. All models use a fixed lookback length of $T = 96$. Baseline results are taken from original or authoritative subsequent publications, with the source of reported results cited in the table header. The row “# 1st” reports the number of times each method achieves the best average performance across all datasets, counted separately for MSE and MAE. Best results are highlighted in **red bold**, and second-best results are in **blue underlined**.

| Models | | SpeQNet | | CycleNet (2024) [35] | | iTransformer (2024) [10] | | RLinear (2023) [10] | | TiDE (2023) [10] | | TimesNet (2023) [10] | | PatchTST (2023) [10] | | Crossformer (2023) [10] | | DLinear (2023) [10] | | SCINet (2022) [10] | |
|--------|-----|--------------|--------------|----------------------|--------------|--------------------------|--------------|---------------------|-------|------------------|-------|----------------------|-------|----------------------|-------|-------------------------|-------|---------------------|-------|--------------------|--------------|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| PEMS03 | 12 | 0.064 | 0.165 | <u>0.066</u> | <u>0.172</u> | 0.126 | 0.236 | 0.178 | 0.305 | 0.078 | 0.187 | 0.085 | 0.192 | 0.099 | 0.216 | 0.090 | 0.203 | 0.122 | 0.243 | <u>0.066</u> | <u>0.172</u> |
| | 24 | 0.078 | 0.182 | <u>0.089</u> | <u>0.201</u> | 0.093 | <u>0.201</u> | 0.246 | 0.334 | 0.257 | 0.371 | 0.118 | 0.223 | 0.142 | 0.259 | 0.121 | 0.240 | 0.201 | 0.317 | <u>0.085</u> | 0.198 |
| | 48 | 0.106 | 0.210 | 0.136 | 0.247 | <u>0.125</u> | <u>0.236</u> | 0.551 | 0.529 | 0.379 | 0.463 | 0.155 | 0.260 | 0.211 | 0.319 | 0.202 | 0.317 | 0.333 | 0.425 | 0.127 | 0.238 |
| | 96 | 0.131 | 0.235 | 0.182 | 0.282 | <u>0.164</u> | <u>0.275</u> | 1.057 | 0.787 | 0.490 | 0.539 | 0.269 | 0.370 | 0.228 | 0.317 | 0.262 | 0.367 | 0.457 | 0.515 | 0.178 | 0.287 |
| | Avg | 0.095 | 0.198 | 0.118 | 0.226 | <u>0.113</u> | <u>0.222</u> | 0.495 | 0.472 | 0.326 | 0.419 | 0.147 | 0.248 | 0.180 | 0.291 | 0.169 | 0.282 | 0.278 | 0.375 | 0.114 | 0.224 |
| PEMS04 | 12 | 0.069 | 0.173 | 0.078 | 0.186 | 0.078 | 0.183 | 0.138 | 0.252 | 0.219 | 0.340 | 0.087 | 0.195 | 0.105 | 0.224 | 0.098 | 0.218 | 0.148 | 0.272 | <u>0.073</u> | <u>0.177</u> |
| | 24 | 0.078 | 0.185 | 0.099 | 0.212 | 0.095 | 0.205 | 0.258 | 0.348 | 0.292 | 0.398 | 0.103 | 0.215 | 0.153 | 0.275 | 0.131 | 0.256 | 0.224 | 0.340 | <u>0.084</u> | <u>0.193</u> |
| | 48 | 0.097 | 0.205 | 0.133 | 0.248 | 0.120 | 0.233 | 0.572 | 0.544 | 0.409 | 0.478 | 0.136 | 0.250 | 0.229 | 0.339 | 0.205 | 0.326 | 0.355 | 0.437 | <u>0.099</u> | <u>0.211</u> |
| | 96 | 0.112 | 0.220 | 0.167 | 0.281 | 0.150 | 0.262 | 1.137 | 0.820 | 0.492 | 0.532 | 0.190 | 0.303 | 0.291 | 0.389 | 0.402 | 0.457 | 0.452 | 0.504 | <u>0.114</u> | <u>0.227</u> |
| | Avg | 0.089 | 0.196 | 0.119 | 0.232 | 0.111 | 0.221 | 0.526 | 0.491 | 0.353 | 0.437 | 0.129 | 0.241 | 0.195 | 0.307 | 0.209 | 0.314 | 0.295 | 0.388 | <u>0.093</u> | <u>0.202</u> |
| PEMS07 | 12 | 0.053 | 0.145 | <u>0.062</u> | <u>0.162</u> | 0.067 | 0.165 | 0.118 | 0.235 | 0.173 | 0.304 | 0.082 | 0.181 | 0.095 | 0.207 | 0.094 | 0.200 | 0.115 | 0.242 | 0.068 | 0.171 |
| | 24 | 0.062 | 0.154 | <u>0.086</u> | 0.192 | 0.088 | <u>0.190</u> | 0.242 | 0.341 | 0.271 | 0.383 | 0.101 | 0.204 | 0.150 | 0.262 | 0.139 | 0.247 | 0.210 | 0.329 | 0.119 | 0.225 |
| | 48 | 0.076 | 0.168 | 0.128 | 0.234 | <u>0.110</u> | <u>0.215</u> | 0.562 | 0.541 | 0.446 | 0.495 | 0.134 | 0.238 | 0.253 | 0.340 | 0.311 | 0.369 | 0.398 | 0.458 | 0.149 | 0.237 |
| | 96 | 0.100 | 0.191 | 0.176 | 0.268 | <u>0.139</u> | <u>0.245</u> | 1.096 | 0.795 | 0.628 | 0.577 | 0.181 | 0.279 | 0.346 | 0.404 | 0.396 | 0.442 | 0.594 | 0.553 | 0.141 | 0.234 |
| | Avg | 0.073 | 0.164 | 0.113 | 0.214 | <u>0.101</u> | <u>0.204</u> | 0.504 | 0.478 | 0.380 | 0.440 | 0.125 | 0.226 | 0.211 | 0.303 | 0.235 | 0.315 | 0.329 | 0.396 | 0.119 | 0.217 |
| PEMS08 | 12 | 0.070 | 0.168 | 0.082 | 0.185 | <u>0.079</u> | <u>0.182</u> | 0.133 | 0.247 | 0.227 | 0.343 | 0.112 | 0.212 | 0.168 | 0.232 | 0.165 | 0.214 | 0.154 | 0.276 | 0.087 | 0.184 |
| | 24 | 0.092 | 0.190 | 0.117 | 0.226 | <u>0.115</u> | <u>0.219</u> | 0.249 | 0.343 | 0.318 | 0.409 | 0.141 | 0.238 | 0.224 | 0.281 | 0.215 | 0.260 | 0.248 | 0.353 | 0.122 | 0.221 |
| | 48 | 0.141 | 0.230 | <u>0.169</u> | 0.268 | 0.186 | <u>0.235</u> | 0.569 | 0.544 | 0.497 | 0.510 | 0.198 | 0.283 | 0.321 | 0.354 | 0.315 | 0.355 | 0.440 | 0.470 | 0.189 | 0.270 |
| | 96 | 0.206 | 0.284 | 0.233 | 0.306 | <u>0.221</u> | <u>0.267</u> | 1.166 | 0.814 | 0.721 | 0.592 | 0.320 | 0.351 | 0.408 | 0.417 | 0.377 | 0.397 | 0.674 | 0.565 | 0.236 | 0.300 |
| | Avg | 0.127 | 0.218 | <u>0.150</u> | 0.246 | <u>0.150</u> | <u>0.226</u> | 0.529 | 0.487 | 0.441 | 0.464 | 0.193 | 0.271 | 0.280 | 0.321 | 0.268 | 0.307 | 0.379 | 0.416 | 0.159 | 0.244 |
| # 1st | | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

5.2.1. Long-Term Forecasting

Table 3 summarizes the long-term forecasting results on eight public benchmarks. Using dataset-level evaluation based on average performance, SpeQNet achieves the highest number of overall wins, ranking first on four datasets, compared with two for TimeXer and one for CycleNet. These results indicate that SpeQNet consistently delivers strong forecasting accuracy across diverse variate numbers and data domains, rather than excelling only at specific horizons or metrics. In particular, SpeQNet achieves consistent best performance across all metrics on Electricity (321 variates), and Solar-Energy (137 variates), where multivariate interactions play a critical role.

On datasets with simpler spatial structure, such as ETT subsets (7 variates) and Weather (21 variates), SpeQNet remains highly competitive and often achieves the best or second-best results across individual horizons. Notably, on ETTh1, ETTm1, and Weather, the performance gap between SpeQNet and other top models diminishes, reflecting the inherent noise and weaker spatial coupling of the variables. Even in this setting, SpeQNet remains competitive and consistently ranks among the top-performing models, indicating the robustness of SpeQNet.

The Traffic dataset presents a particularly challenging case. Although it contains the largest number of variates (862), SpeQNet does not exhibit the same level of consistent superiority observed on other high-dimensional benchmarks. We attribute this behavior primarily to the temporal resolution of the dataset. Unlike the PEMS benchmarks, which are sampled at 5 min intervals, the Traffic dataset is recorded at an hourly frequency. This coarser temporal granularity weakens spatiotemporal dependencies and reduces the amount of fine-grained information available for the model. As a result, the benefits of query-enhanced spatiotemporal representations are partially diminished. In contrast, on PEMS07, which has a comparable number of variates (883) but is sampled at a much finer 5-min interval, SpeQNet demonstrates consistent superiority across all prediction horizons and achieves the best average performance, see Table 4. This comparison highlights that SpeQNet is particularly effective in settings where dense spatial interactions are coupled with high-resolution temporal dynamics, a regime that is common in real-world spatiotemporal systems.

5.2.2. Short-Term Forecasting

Table 4 reports the results on four PEMS benchmarks, variates varying from 170 to 883, under the short-term forecasting setting. SpeQNet achieves dataset-level wins on all four datasets, and more importantly, exhibits consistent superiority across every horizon and both metrics. Overall, these results validate SpeQNet as a strong spatiotemporal forecaster in classical traffic regimes, it not only scales to hundreds of nodes but also delivers uniformly improved accuracy across short and longer prediction horizons, highlighting its effectiveness.

5.3. Ablation Studies

We conduct ablation studies to validate the core design choices of SpeQNet and to isolate the contribution of each major component. Specifically, we create four variants: (1) w/o ST-Query, where we remove the spatiotemporal query module and use only temporally encoded node representations to construct the graph, which evaluates the necessity of query-enhanced spatiotemporal representations. (2) Attn-Graph, where we replace the adaptive graph learning mechanism with an attention-based graph construction, where the adjacency matrix is computed via scaled dot-product similarity between node representations, i.e., $\mathbf{A} = \text{Softmax}(\mathbf{Q}\mathbf{K}^T / \sqrt{d})$, with \mathbf{Q} and \mathbf{K} obtained from linear projections of the node features. This variant assesses whether purely input-driven attention graphs can

substitute task-oriented adaptive graph learning. (3) Corr-Graph, where we replace the adaptive graph learning module with a fixed correlation-based graph constructed from the training data, where the adjacency matrix is obtained by applying a softmax normalization to the correlation matrix, which evaluates whether static statistical dependencies can substitute task-adaptive graph learning. (4) GCN, where we replace the spectral graph filtering block with an equal number of spatial GCN layers, in order to evaluate the impact of the proposed spectral graph filtering versus standard low-pass message passing. We evaluate all variants on four datasets spanning all major application domains and graph scales considered in this work: PEMS07 (Transportation, 883 variates), Solar-Energy (Solar Power, 137 variates), Electricity (Electricity, 321 variates), and Weather (Weather, 21 variates). This setup allows us to systematically examine how each design choice affects forecasting performance under varying numbers of variates and domains. Quantitative results are reported in Table 5.

The ablation results demonstrate that each design component of SpeQNet contributes meaningfully to forecasting performance, with their impact becoming increasingly pronounced as the number of variates and the complexity of spatial dependencies grow. Removing the spatiotemporal query (w/o ST-Query) consistently degrades performance across all datasets, confirming that query-enhanced spatiotemporal representations are crucial for spatiotemporal forecasting.

Table 5. Ablation study of SpeQNet. We evaluate the contribution of key architectural components by comparing the full model with four variants: w/o Query, without the spatiotemporal queries. Attn-Graph, replacing adaptive graph learning with attention weights. Corr-Graph, replacing adaptive graph learning with a fixed correlation-based graph. GCN, replacing spectral graph filtering block with GCN. Experiments are conducted on PEMS07, Solar-Energy, Electricity, and Weather datasets, spanning diverse domains and numbers of variates.

| Models | | SpeQNet | | w/o Query | | Attn-Graph | | Corr-Graph | | GCN | |
|-------------|-----|--------------|--------------|--------------|--------------|------------|--------------|------------|-------|-------|-------|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| Weather | 96 | 0.158 | 0.199 | 0.165 | 0.205 | 0.167 | 0.211 | 0.169 | 0.213 | 0.213 | 0.261 |
| | 192 | 0.207 | 0.251 | 0.210 | 0.248 | 0.211 | 0.250 | 0.212 | 0.253 | 0.258 | 0.294 |
| | 336 | 0.262 | 0.290 | 0.264 | 0.288 | 0.265 | 0.290 | 0.266 | 0.292 | 0.303 | 0.323 |
| | 720 | 0.347 | 0.343 | 0.346 | 0.341 | 0.347 | 0.342 | 0.348 | 0.343 | 0.379 | 0.366 |
| | Avg | 0.243 | 0.271 | 0.247 | 0.271 | 0.248 | 0.273 | 0.249 | 0.275 | 0.288 | 0.311 |
| Solar | 96 | 0.170 | 0.230 | 0.193 | 0.243 | 0.187 | 0.236 | 0.202 | 0.259 | 0.212 | 0.285 |
| | 192 | 0.188 | 0.243 | 0.209 | 0.256 | 0.208 | 0.250 | 0.224 | 0.271 | 0.233 | 0.298 |
| | 336 | 0.202 | 0.251 | 0.215 | 0.260 | 0.214 | 0.255 | 0.236 | 0.274 | 0.253 | 0.304 |
| | 720 | 0.215 | 0.257 | 0.222 | 0.261 | 0.222 | 0.256 | 0.244 | 0.275 | 0.256 | 0.298 |
| | Avg | 0.194 | 0.245 | 0.210 | 0.255 | 0.208 | 0.249 | 0.226 | 0.270 | 0.239 | 0.296 |
| Electricity | 96 | 0.121 | 0.225 | 0.129 | 0.229 | 0.126 | 0.229 | 0.127 | 0.232 | 0.185 | 0.287 |
| | 192 | 0.141 | 0.242 | 0.145 | 0.243 | 0.143 | 0.243 | 0.148 | 0.252 | 0.194 | 0.294 |
| | 336 | 0.160 | 0.260 | 0.162 | 0.259 | 0.162 | 0.261 | 0.176 | 0.278 | 0.212 | 0.309 |
| | 720 | 0.196 | 0.290 | 0.199 | 0.288 | 0.201 | 0.294 | 0.225 | 0.315 | 0.254 | 0.338 |
| | Avg | 0.155 | 0.255 | 0.159 | 0.255 | 0.158 | 0.257 | 0.169 | 0.269 | 0.211 | 0.307 |
| PEMS07 | 12 | 0.053 | 0.145 | 0.075 | 0.154 | 0.070 | 0.173 | 0.061 | 0.157 | 0.290 | 0.382 |
| | 24 | 0.062 | 0.154 | 0.084 | 0.165 | 0.081 | 0.185 | 0.070 | 0.168 | 0.297 | 0.387 |
| | 48 | 0.076 | 0.168 | 0.095 | 0.177 | 0.095 | 0.199 | 0.083 | 0.181 | 0.312 | 0.400 |
| | 96 | 0.100 | 0.191 | 0.107 | 0.188 | 0.114 | 0.214 | 0.097 | 0.193 | 0.322 | 0.404 |
| | Avg | 0.073 | 0.164 | 0.090 | 0.171 | 0.090 | 0.193 | 0.078 | 0.175 | 0.305 | 0.393 |

Note: Boldface is used to denote the best-performing results for each metric.

Replacing the adaptive graph learning with a fixed correlation-based graph (Corr-Graph) also leads to noticeable performance drops, particularly on datasets with larger

graph sizes. Although correlation graphs capture coarse pairwise dependencies, their static nature limits their ability to adapt to task objectives. Similarly, the Attn-Graph variant, which constructs the graph purely from input-driven attention, underperforms the proposed adaptive graph learner. This suggests that the adaptive graph learning is more effective than relying on similarity measures alone.

Among all variants, replacing the spectral graph filtering block with standard GCN layers (GCN) results in the most severe performance degradation across datasets. This finding underscores the limitation of purely low-pass message passing and highlights the importance of spectral graph filtering for modeling heterogeneous spatial interactions. Notably, performance differences between variants are relatively modest on the Weather dataset with limited spatial coupling, but widen substantially on larger graphs such as Solar-Energy and PEMS07. In particular, the pronounced degradation on PEMS07 confirms that SpeQNet's components are jointly necessary rather than interchangeable, and that their benefits scale with system size and spatial complexity.

5.4. Learned Spectral Filter Response Analysis

To better understand how SpeQNet exploits spectral graph information, we analyze the learned frequency responses of the spectral filtering module. Since SpeQNet employs Chebyshev polynomial approximation, the learned filter can be explicitly characterized by its spectral response

$$h(\lambda) = \sum_{k=0}^K c_k T_k(\lambda), \quad (25)$$

which describes how different graph frequencies, parameterized by Laplacian eigenvalues λ , are amplified or suppressed. Specifically, we examine the induced frequency responses, which provide an interpretable view of how information is propagated and reshaped in the graph spectral domain.

Figure 5 visualizes the learned spectral responses from both the first and the final spectral graph filtering layers on the PEMS07, Solar-Energy, and Weather datasets. Despite differences in shape, the learned spectral responses across all three datasets consistently exhibit a band-stop-like behavior, where mid-range graph frequencies are most strongly attenuated while both low- and high-frequency components are preserved. This shared pattern suggests that the proposed spectral filtering mechanism is particularly effective at suppressing intermediate-frequency variations that are less informative for forecasting, while retaining global trends and localized dynamics.

Beyond this common structure, the responses vary notably across datasets and across network depth. On PEMS07, the filters, especially in deeper layers, exhibit sharp transitions and pronounced curvature, indicating a highly expressive and aggressive frequency-selective behavior suited to complex and heterogeneous traffic interactions. In contrast, the Solar-Energy dataset yields a more moderated and piecewise response, where mid-range attenuation is achieved through a combination of steep and smooth variations, resulting in conservative yet structured spectral shaping. For the Weather dataset, the learned responses are considerably smoother across both early and late layers, characterized by gradual attenuation toward mid-range frequencies followed by a gentle and symmetric recovery at higher frequencies.

To complement the qualitative visualizations, Table 6 provides quantitative measurements of the learned spectral responses across predefined frequency bands. Following the Laplacian spectrum $\lambda \in [0, 2]$, we partition the spectrum into low ($[0, 0.67]$), mid ($[0.67, 1.33]$), and high ($[1.33, 2.0]$) frequency regions, and report the average gain, relative energy ratio, and within-band variance across bands for both first and last spectral filtering layers. Across all datasets, mid-frequency bands consistently exhibit the lowest average

gain and energy ratio, numerically confirming the band-stop behavior observed in Figure 5. In contrast, low- and high-frequency bands retain substantially higher energy, indicating preservation of both global trends and localized dynamics.

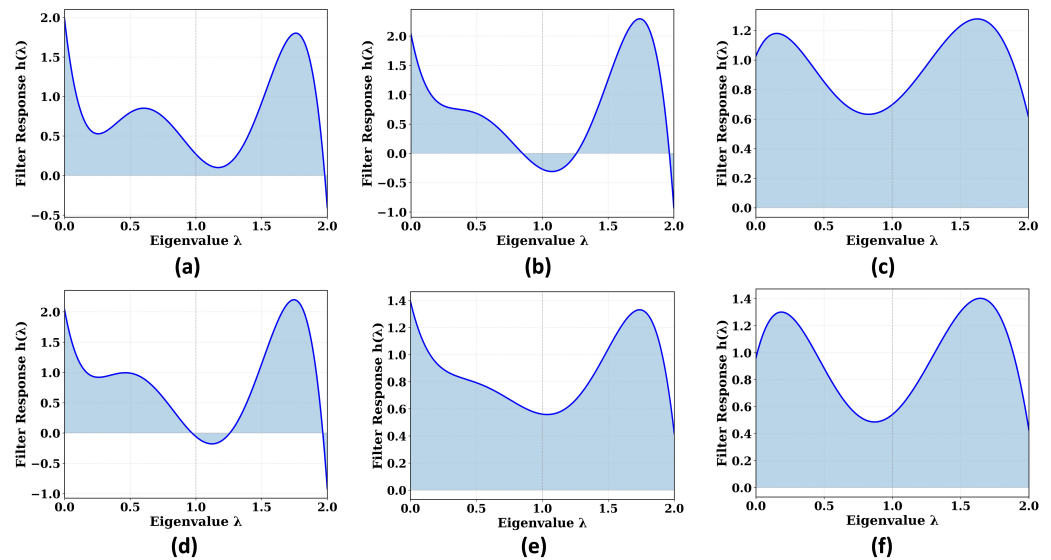


Figure 5. Learned spectral filter responses from SpeQNet. Subfigures (a,d), (b,e), and (c,f), respectively, correspond to the PEMS07, Solar-Energy, and Weather datasets, where (a–c) visualize the filters learned at the first spectral filtering layer and (d–f) show those from the final spectral filtering layer. The solid blue curve denotes the learned spectral filter response as a function of the Laplacian eigenvalue λ , while the shaded region highlights the filter magnitude across the spectrum.

Table 6. Quantitative spectral response statistics across frequency bands. For each dataset, we report band-wise metrics for the first and last spectral filtering layers. Metrics include average gain (GAIN), energy ratio (ENERGY, %), and gain variance (VAR) computed over low-, mid-, and high-frequency bands of the learned spectral response. Frequency bands are defined over Laplacian eigenvalues $\lambda \in [0, 2]$ as low ($[0, 0.67]$), mid ($[0.67, 1.33]$), and high ($[1.33, 2.0]$).

| Dataset | Layer | Low Freq. | | | Mid Freq. | | | High Freq. | | |
|---------|-------|-----------|--------|------|-----------|--------|------|------------|--------|------|
| | | GAIN | ENERGY | VAR | GAIN | ENERGY | VAR | GAIN | ENERGY | VAR |
| PEMS07 | First | 0.82 | 29.4 | 0.10 | 0.37 | 7.7 | 0.06 | 1.11 | 62.4 | 0.32 |
| | Last | 1.06 | 32.9 | 0.07 | 0.14 | 2.8 | 0.09 | 1.29 | 64.0 | 0.59 |
| Solar | First | 0.88 | 25.5 | 0.13 | −0.05 | 1.40 | 0.05 | 1.37 | 73.0 | 0.63 |
| | Last | 0.91 | 34.5 | 0.02 | 0.62 | 15.6 | 0.03 | 1.07 | 49.2 | 0.05 |
| Weather | First | 1.00 | 34.9 | 0.02 | 0.76 | 19.9 | 0.01 | 1.12 | 44.4 | 0.03 |
| | Last | 1.05 | 38.0 | 0.04 | 0.64 | 13.9 | 0.03 | 1.17 | 47.3 | 0.06 |

Overall, these quantitative statistics and response visualizations corroborate that SpeQNet learns depth- and domain-adaptive band-stop spectral operators, rather than relying on a fixed low-pass bias. The observed diversity in response curvature across datasets and network depth highlights the expressivity and flexibility of the proposed spectral filtering mechanism, which progressively refines frequency selectivity in alignment with the forecasting objective.

5.5. Sensitivity to Chebyshev Polynomial Order

We further investigate the sensitivity of SpeQNet to the Chebyshev polynomial order K , which controls the spectral expressivity of the graph filtering operation. To this end, we conduct a controlled study on the Solar-Energy dataset by varying K from 1 to 10 while

keeping all other settings fixed. Figure 6 reports the corresponding MSE and MAE results. SpeQNet exhibits stable performance across a wide range of K values, with both MSE and MAE fluctuating within a narrow margin. This observation indicates that the proposed model is not overly sensitive to the precise choice of spectral order and does not rely on carefully tuned high-degree polynomials to achieve good performance. Overall, the model demonstrates robustness to the choice of K , confirming that its effectiveness stems from adaptive spectral shaping rather than sensitivity to polynomial depth.

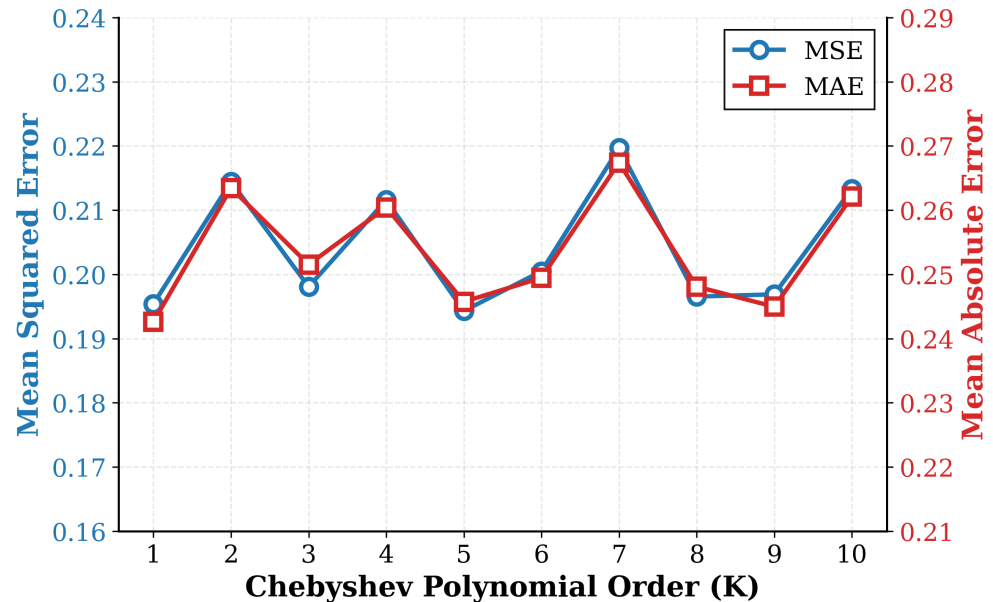


Figure 6. Sensitivity analysis of SpeQNet with respect to the Chebyshev polynomial order K on the Solar-Energy dataset.

5.6. Scaling Analysis

To examine how SpeQNet scales with increasing spatial complexity, we analyze the relative performance improvement over a strong and consistent baseline (iTransformer) as the number of variates increases across datasets, as shown in Figure 7. For short-term traffic forecasting, SpeQNet exhibits a clear scaling advantage: the relative improvement consistently increases with the number of sensors, reaching nearly 20% on PEMS07 with 883 variates. This pronounced trend indicates that query-guided spectral modeling becomes increasingly beneficial as spatial dependencies grow more complex. For long-term forecasting, SpeQNet maintains stable improvements across datasets with varying dimensionalities, achieving consistent gains on medium- to large-scale settings such as Solar-Energy and Electricity. Although the improvement on the largest Traffic dataset is more modest, this behavior is consistent with our earlier analysis attributing it to the coarse temporal sampling interval (one hour), which limits the amount of exploitable temporal and spatial variation. Overall, these results demonstrate that SpeQNet scales favorably with the number of variates, particularly in settings where complex and latent spatial interactions can be effectively leveraged.

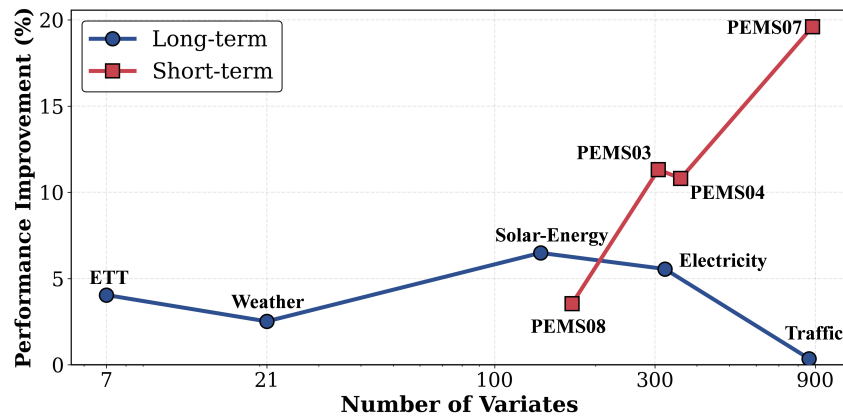


Figure 7. Scaling behavior of SpeQNet with respect to the number of variates.

5.7. Computational Efficiency Analysis

We further evaluate the computational efficiency and resource usage of SpeQNet and representative Transformer-based baselines under identical hardware and software settings. Table 7 reports the number of parameters, computational cost (GFLOPs), runtime, and memory consumption on two datasets with contrasting graph scales: Traffic (862 variates), representing large-scale systems, and ETTm1 (7 variates), representing small-scale networks.

Table 7. Computational efficiency comparison. We report model size, computational cost, runtime, and memory consumption for SpeQNet and representative Transformer-based baselines under identical hardware and software settings. Params (M) denotes the total number of trainable parameters in millions, GFLOPs denotes the number of giga floating-point operations required for a single forward pass, Train (ms) denotes the training time per iteration, Infer (ms) denotes the inference time per sample, and Memory (GB) denotes the peak GPU memory usage during training. Experiments are conducted on ETTm1 (7 variates) and Traffic (862 variates) with forecasting horizon $H = 720$.

| Dataset | | Traffic (862 Variates) | | | |
|--------------|------------|------------------------|------------|------------|-------------|
| Model | Params (M) | GFLOPs | Train (ms) | Infer (ms) | Memory (GB) |
| SpeQNet | 11.85 | 364.91 | 168.86 | 1.50 | 4.92 |
| iTransformer | 6.73 | 141.53 | 98.63 | 0.35 | 5.85 |
| PatchTST | 1.51 | 82.45 | 169.96 | 0.30 | 4.93 |
| Dataset | | ETTM1 (7 variates) | | | |
| Model | Params (M) | GFLOPs | Train (ms) | Infer (ms) | Memory (GB) |
| SpeQNet | 0.59 | 0.13 | 61.41 | 0.45 | 0.04 |
| iTransformer | 0.30 | 0.07 | 12.95 | 0.09 | 0.16 |
| PatchTST | 1.51 | 1.34 | 16.52 | 0.14 | 0.13 |

On Traffic, SpeQNet incurs higher computational cost than Transformer-based baselines due to adaptive graph learning and spectral filtering over a large number of nodes, while maintaining comparable or lower peak memory usage and practical inference latency. On ETTm1, all models exhibit negligible computational overhead, and the absolute efficiency differences become marginal. Overall, SpeQNet presents a moderate increase in computation in exchange for improved modeling capacity, while remaining practical in terms of memory footprint and inference efficiency.

5.8. Graph Interpretability Analysis

We analyze the structure of the learned adaptive graph across four representative domains. As shown in Figure 8, the learned graphs consistently exhibit pronounced hub structures, visible as vertical stripe patterns in the adjacency matrices, whereas the correlation-based

graphs are dominated by block-like connectivity induced by pairwise similarity. Confirming that the learned topology differs substantially from correlation structure.

The strength of the learned hubs varies systematically across domains. Traffic networks exhibit the strongest hub concentration (maximum degree 14.77), followed by Electricity (9.43), Solar-Energy (4.96), and Weather (2.02). This trend reflects dataset complexity, larger systems tend to develop stronger hub-dominated structures, where a small subset of nodes aggregates information from many others. Moreover, the difference maps (learned graph minus correlation graph) reveal systematic deviations between the two structures, with certain edges being emphasized or attenuated relative to correlation-based connectivity. These observations suggest that the adaptive graph encodes structural patterns that differ from simple pairwise similarity and may reflect task-specific relational characteristics present in the data.

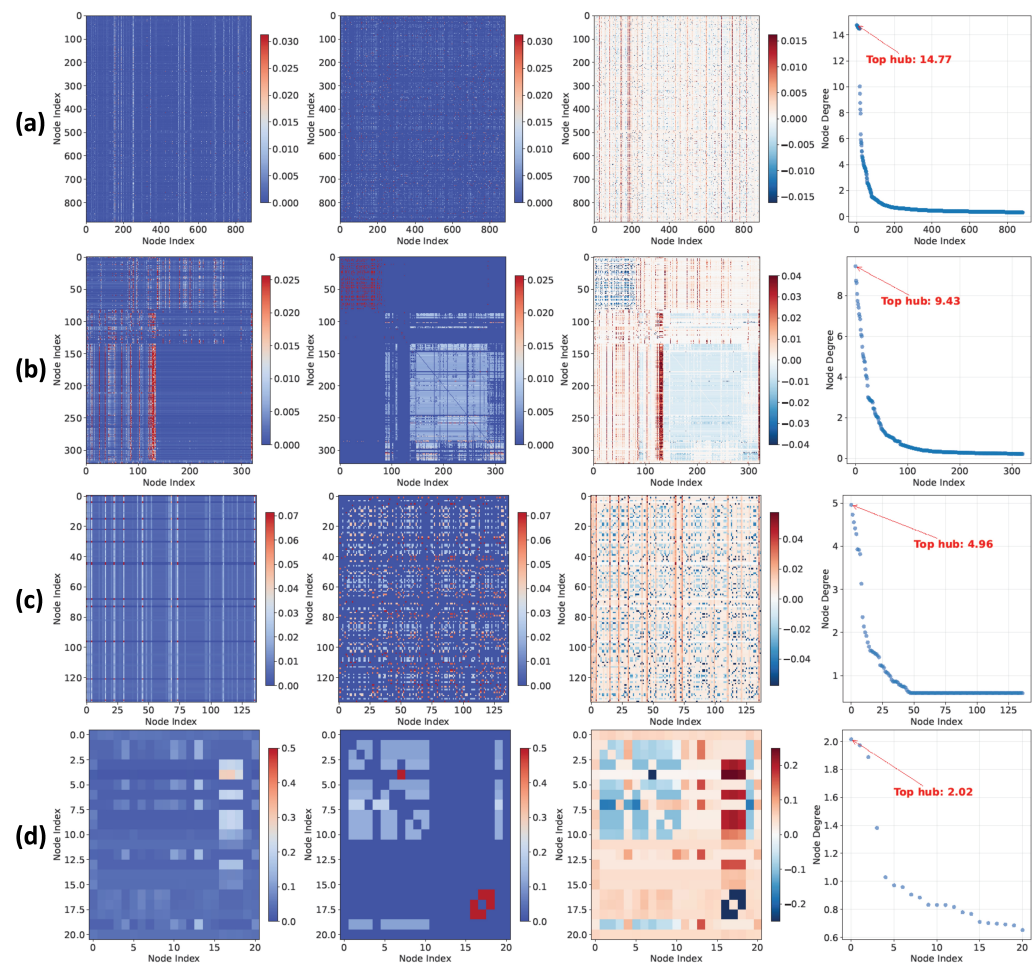


Figure 8. Learned vs. correlation graph comparison across domains. Each row corresponds to a dataset from a different domain: (a) PEMS07, (b) Electricity, (c) Solar-Energy, and (d) Weather. Columns show, from left to right: the learned adaptive graph, the correlation-based graph, their difference (learned minus correlation), and the node degree distribution of the learned graph.

5.9. Qualitative Forecasting Visualization

In addition to quantitative metrics, we provide qualitative visualizations of representative forecasting results to illustrate how SpeQNet captures temporal dynamics across different domains. Figure 9 shows example predictions for three randomly selected variables from each dataset (PEMS07, Electricity, Solar-Energy, and Weather), where the prediction horizons correspond to those evaluated in the main experiments (96, 336, 720, and 192, respectively). Across diverse temporal patterns, SpeQNet closely follows the ground-truth trajectories, accurately modeling both smooth trends and sharp periodic variations. These

visualizations complement the quantitative results by providing an intuitive illustration of the model's forecasting behavior.

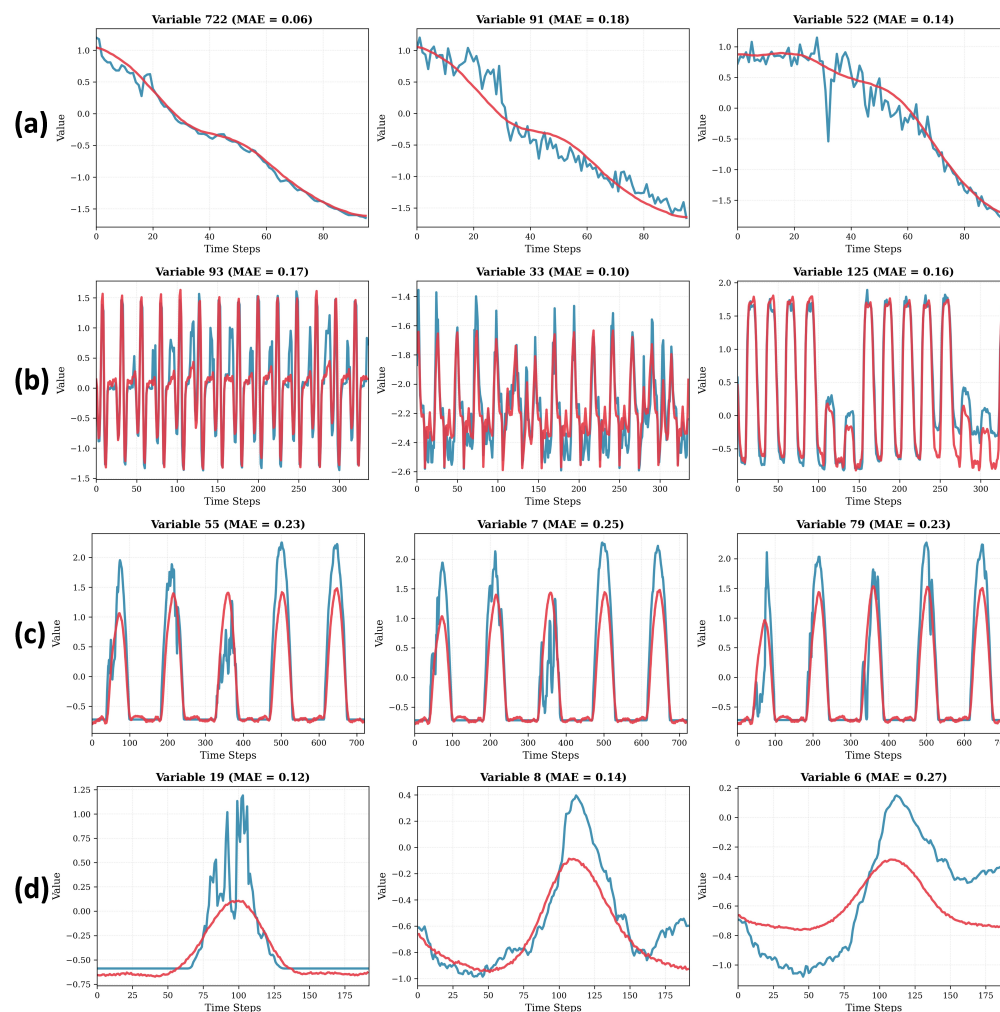


Figure 9. Qualitative forecasting results on representative datasets. For each dataset, three variables are randomly selected and visualized from the same sample. Blue curves denote ground-truth observations, while red curves indicate SpeQNet predictions. The mean MAE for each variable over the prediction horizon is reported in the corresponding subplot title. Rows (a–d) correspond to the PEMS07, Electricity, Solar-Energy, and Weather datasets, respectively.

6. Conclusions

Spatiotemporal forecasting demands models that can simultaneously capture global coordination across sensors and sharp, localized deviations—while operating under uncertain or latent graph structure. In this work, we introduced SpeQNet, a query-enhanced spectral graph filtering framework that couples (i) query-enhanced spatiotemporal representations, (ii) task-oriented adaptive graph construction, and (iii) a frequency-aware ChebNetII-based spectral filtering block for flexible spatial modeling. Extensive experiments on twelve benchmarks across traffic, energy, and climate domains demonstrate that SpeQNet consistently improves forecasting accuracy and scales effectively to large graphs with hundreds of variates.

Our analysis further sheds light on why SpeQNet works. Ablations show that the spatiotemporal query, adaptive graph learning, and spectral filtering are jointly necessary, with performance gaps widening as the number of variates increases. Moreover, SpeQNet learns domain-adaptive spectral operators with a consistent band-stop-like behavior, selectively attenuating mid-range graph frequencies while preserving both low- and high-

frequency components—an interpretable pattern aligned with suppressing less informative variations while retaining global trends and localized dynamics. Finally, SpeQNet is robust to the spectral depth hyperparameter: performance remains stable across a broad range of Chebyshev polynomial orders, indicating that its gains arise from adaptive spectral shaping rather than fragile tuning.

Limitations and Future Work

While SpeQNet demonstrates strong performance across a diverse set of spatiotemporal forecasting benchmarks, several limitations define the current scope of the proposed framework. First, SpeQNet assumes regularly sampled time series with a fixed set of nodes, and performs adaptive graph learning and spectral filtering on sample-wise node representations obtained after temporal compression. As a result, the current formulation does not explicitly model time-varying graph structures, irregular sampling patterns, or dynamically evolving node sets. Second, although the evaluated benchmarks span multiple domains, generalization to substantially different application domains or sensing regimes beyond these benchmarks has not been explicitly validated. Broader evaluation under more irregular, partially observed, or domain-shifted conditions therefore remains an important direction for future work. Promising extensions include incorporating explicit mechanisms for dynamic graph evolution, designing hierarchical or time-adaptive query representations, and coupling spectral filter generation more tightly with the query mechanism to enable context-aware frequency adaptation without increasing computation cost.

Author Contributions: Conceptualization, Z.F. and K.M.; methodology, Z.F.; software, Z.F.; validation, Z.F.; formal analysis, Z.F.; investigation, Z.F.; resources, Z.F.; data curation, Z.F.; writing—original draft preparation, Z.F.; writing—review and editing, Z.F. and K.M.; visualization, Z.F. and K.M.; supervision, K.M.; project administration, K.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study is publicly accessible and can be obtained from <https://github.com/thuml/iTransformer> (accessed on 10 June 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Satpathy, I.; Nayak, A.; Jain, V. The green city: Sustainable and smart urban living through artificial intelligence. In *Utilizing Technology to Manage Territories*; IGI Global: Hershey, PA, USA, 2025; pp. 273–304.
2. Fang, Y.; Liang, Y.; Hui, B.; Shao, Z.; Deng, L.; Liu, X.; Jiang, X.; Zheng, K. Efficient large-scale traffic forecasting with transformers: A spatial data management perspective. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, Toronto, ON, Canada, 3–7 August 2025; pp. 307–317.
3. Dong, Q.; Huang, R.; Cui, C.; Towey, D.; Zhou, L.; Tian, J.; Wang, J. Short-Term Electricity-Load Forecasting by deep learning: A comprehensive survey. *Eng. Appl. Artif. Intell.* **2025**, *154*, 110980. [[CrossRef](#)]
4. Zhang, H.; Liu, Y.; Zhang, C.; Li, N. Machine learning methods for weather forecasting: A survey. *Atmosphere* **2025**, *16*, 82. [[CrossRef](#)]
5. Kim, S.J.; Cho, Y. Enhancing environmental monitoring of harmful algal blooms with ConvLSTM image prediction. *Environ. Res. Commun.* **2025**, *7*, 025012. [[CrossRef](#)]
6. Jin, M.; Koh, H.Y.; Wen, Q.; Zambon, D.; Alippi, C.; Webb, G.I.; King, I.; Pan, S. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *arXiv* **2023**, arXiv: 2307.03759. [[CrossRef](#)] [[PubMed](#)]
7. Cini, A.; Marisca, I.; Zambon, D.; Alippi, C. Graph deep learning for time series forecasting. *ACM Comput. Surv.* **2025**, *57*, 321. [[CrossRef](#)]

8. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv: 1709.04875.
9. Pan, Z.; Liang, Y.; Wang, W.; Yu, Y.; Zheng, Y.; Zhang, J. Urban traffic prediction from spatio-temporal data using deep meta learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1720–1730.
10. Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; Long, M. itransformer: Inverted transformers are effective for time series forecasting. *arXiv* **2023**, arXiv: 2310.06625.
11. Wang, Y.; Wu, H.; Dong, J.; Qin, G.; Zhang, H.; Liu, Y.; Qiu, Y.; Wang, J.; Long, M. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 469–498.
12. Cini, A.; Marisca, I.; Zambon, D.; Alippi, C. Taming local effects in graph-based spatiotemporal forecasting. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 55375–55393.
13. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16), Barcelona, Spain, 5–10 December 2016; pp. 3844–3852.
14. Balcilar, M.; Renton, G.; Héroux, P.; Gaüzère, B.; Adam, S.; Honeine, P. Analyzing the expressive power of graph neural networks in a spectral perspective. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4 May 2021.
15. Kipf, T. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv: 1609.02907.
16. Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; Weinberger, K. Simplifying graph convolutional networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6861–6871.
17. Xu, B.; Shen, H.; Cao, Q.; Cen, K.; Cheng, X. Graph convolutional networks using heat kernel for semi-supervised learning. *arXiv* **2020**, arXiv: 2007.16002. [[CrossRef](#)]
18. Nt, H.; Maehara, T. Revisiting graph neural networks: All we have is low-pass filters. *arXiv* **2019**, arXiv: 1905.09550. [[CrossRef](#)]
19. Meng, G.; Jiang, Q.; Fu, K.; Lin, B.; Lu, C.T.; Chen, Z. Early forecasting of the impact of traffic accidents using a single shot observation. In Proceedings of the 2022 SIAM International Conference on Data Mining (SDM), Alexandria, VA, USA, 28–30 April 2022; pp. 100–108.
20. Wang, X.; Zhang, M. How powerful are spectral graph neural networks. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; pp. 23341–23362.
21. He, M.; Wei, Z.; Xu, H. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 14239–14251.
22. He, M.; Wei, Z.; Wen, J.R. Convolutional neural networks on graphs with chebyshev approximation, revisited. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 7264–7276.
23. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv: 1707.01926.
24. Shao, Z.; Zhang, Z.; Wei, W.; Wang, F.; Xu, Y.; Cao, X.; Jensen, C.S. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *arXiv* **2022**, arXiv: 2206.09112. [[CrossRef](#)]
25. Bai, L.; Yao, L.; Li, C.; Wang, X.; Wang, C. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. *arXiv* **2020**, arXiv: 2007.02842. [[CrossRef](#)]
26. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; Zhang, C. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. *arXiv* **2020**, arXiv: 2005.11650. [[CrossRef](#)]
27. Lee, H.; Ko, S. TESTAM: A time-enhanced spatio-temporal attention model with mixture of experts. *arXiv* **2024**, arXiv: 2403.02600.
28. Park, C.; Lee, C.; Bahng, H.; Tae, Y.; Jin, S.; Kim, K.; Ko, S.; Choo, J. ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, 19–23 October 2020; pp. 1215–1224.
29. Nie, Y. A Time Series is Worth 64Words: Long-term Forecasting with Transformers. *arXiv* **2022**, arXiv: 2211.14730.
30. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are transformers effective for time series forecasting? In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 11121–11128.
31. Li, Z.; Qi, S.; Li, Y.; Xu, Z. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv* **2023**, arXiv: 2305.10721. [[CrossRef](#)]
32. Ansari, A.F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S.S.; Arango, S.P.; Kapoor, S.; et al. Chronos: Learning the language of time series. *arXiv* **2024**, arXiv: 2403.07815. [[CrossRef](#)]
33. Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J.Y.; Shi, X.; Chen, P.Y.; Liang, Y.; Li, Y.F.; Pan, S.; et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv* **2023**, arXiv: 2310.01728.
34. Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv* **2022**, arXiv: 2210.02186.

35. Lin, S.; Lin, W.; Hu, X.; Wu, W.; Mo, R.; Zhong, H. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 106315–106345.
36. Zhang, Y.; Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In Proceedings of the Eleventh International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
37. Yang, Y.; Zhu, Q.; Chen, J. Vcformer: Variable correlation transformer with inherent lagged correlation for multivariate time series forecasting. *arXiv* **2024**, arXiv: 2405.11470. [[CrossRef](#)]
38. Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; Sahoo, D. Unified training of universal time series forecasting transformers. *arXiv* **2024**, arXiv: 2402.02592. [[CrossRef](#)]
39. Gao, S.; Koker, T.; Queen, O.; Hartvigsen, T.; Tsiglikaridis, T.; Zitnik, M. Units: A unified multi-task time series model. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 140589–140631.
40. Liu, Y.; Wu, H.; Wang, J.; Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 9881–9893.
41. Zhu, J.; Wang, Q.; Tao, C.; Deng, H.; Zhao, L.; Li, H. AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting. *IEEE Access* **2021**, *9*, 35973–35983. [[CrossRef](#)]
42. Song, C.; Lin, Y.; Guo, S.; Wan, H. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 914–921.
43. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. *arXiv* **2019**, arXiv: 1906.00121.
44. Cai, W.; Liang, Y.; Liu, X.; Feng, J.; Wu, Y. Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; Volume 38, pp. 11141–11149.
45. Huang, Q.; Shen, L.; Zhang, R.; Ding, S.; Wang, B.; Zhou, Z.; Wang, Y. Crossggn: Confronting noisy multivariate time series via cross interaction refinement. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 46885–46902.
46. Bauer, F. Normalized graph Laplacians for directed graphs. *Linear Algebra Appl.* **2012**, *436*, 4193–4222. [[CrossRef](#)]
47. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
48. Das, A.; Kong, W.; Leach, A.; Mathur, S.; Sen, R.; Yu, R. Long-term forecasting with tide: Time-series dense encoder. *arXiv* **2023**, arXiv: 2304.08424.
49. Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; Xu, Q. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 5816–5828.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.