AMGNet: Adaptive Multi-Scale Gated GNN and Hierarchical Fusion for Time Series Forecasting

1st Xin Gao
Software College
Northeastern University
Shenyang, Liaoning, China
2471422@stu.neu.edu.cn

2nd Tianyu Chang
Software College
Northeastern University
Shenyang, Liaoning, China
2410592@stu.neu.edu.cn

3rd Dongming Chen
Software College
Northeastern University
Shenyang, Liaoning, China
chendm@mail.neu.edu.cn

4nd Dongqi Wang*
Software College
Northeastern University
Shenyang, Liaoning, China
wangdq@swc.neu.edu.cn

***Abstract*—Real-world time series data contain multiple temporal patterns at different scales that are intricately intertwined, making forecasting extremely challenging. Recently, Graph Neural Networks (GNNs) have demonstrated significant potential in multivariate time series forecasting due to their ability to explicitly model inter-variable dependencies. However, existing methods typically rely on static and uniform aggregation rules, failing to adapt to the varying informational needs of different nodes across their multi-hop neighborhoods. This limitation hinders their effectiveness in modeling inter-series dependencies. Moreover, current models often suffer from error accumulation due to a lack of explicit cross-scale information interaction. To address these issues, we propose AMGNet. Our framework incorporates three key components: (1) A Node-Aware Gated Message Passing (NAGMP) module operating on scale-specific graph structures to adaptively capture node-specific correlations. This design effectively filters neighborhood noise and enhances the modeling of complex inter-series dependencies. (2) A coarse-to-fine hierarchical guidance structure that utilizes coarse-grained dominant trends to explicitly calibrate fine-grained feature learning, thereby facilitating cross-scale information interaction and mitigating error accumulation. (3) Finally, a Multi-Predictor Synthesis (MPS) module based on spectral energy is adopted to achieve a noise-resilient prediction ensemble. Extensive experiments on six datasets spanning diverse domains demonstrate the superiority of AMGNet.**

***Index Terms*—Graph Neural Networks, Multi-Scale, Multivariate Time Series Forecasting.**

I. Introduction

Time series forecasting plays a pivotal role in numerous real-world applications, such as traffic planning [1] and financial analysis [2]. Real-world data often contain interwoven multi-scale temporal patterns. For instance, in financial markets, short-term price fluctuations are constrained by long-term macroeconomic trends. Moreover, inter-variable correlations can shift across scales (e.g., positive in the long term, but negative during short-term turmoil). Therefore, accurate forecasting requires capturing complex inter-variable dependencies and exploiting multi-scale temporal interactions.

While various deep learning models [3]–[6] have advanced the field, Graph Neural Networks (GNNs) [7], [8] have emerged as a promising paradigm by treating variables as nodes to capture inter-series correlations. Despite their potential, integrating GNNs into complex forecasting tasks remains challenging, primarily due to two limitations.

The first is *how to effectively capture complex and dynamically evolving inter-series dependencies*. Early methods relying on pre-defined graphs [9] are suboptimal for complex variations. Although recent studies adaptively learn graph structures, they typically apply fixed aggregation rules uniformly across all nodes [8], [10]. This rigid approach overlooks the personalized informational needs of different nodes and indiscriminately incorporates irrelevant neighborhood information. Consequently, it introduces noise interference and limits the model's capability to effectively capture complex and dynamically evolving inter-series dependencies.

The second is *the lack of cross-scale information interaction*. Many multi-scale models [4], [8] process different scales in parallel, inherently ignoring the fact that macroscopic trends constrain microscopic fluctuations. While recent GNNs attempt cross-scale connections via hypergraphs [11], they rely on predefined rules and lack explicit hierarchical guidance. As a result, models are prone to error accumulation, especially when forecasting non-stationary series over long horizons. Furthermore, simple aggregation strategies in the prediction phase often fail to filter out high-frequency noise, leading to degraded performance.

To address these challenges, we propose AMGNet, a GNN-based method for multivariate time series forecasting. It effectively accommodates the personalized informational needs of different variables, fully utilizes the guiding influence of macroscopic trends on microscopic fluctuations, and integrates the predictive capabilities across different scales. Our main contributions are summarized as follows:

- We propose the Adaptive Gated Spatio-Temporal Modeling module to learn complex intra-series and inter-series dependencies within scales. The module begins by adaptively learning a graph structure for each scale, followed by the introduction of a Node-Aware Gated Message Passing module. This enables the model to adapt to the varying informational needs of different nodes, dynamically filtering noise and capturing complex variable dependencies. Finally, we employ a multi-head self-attention mechanism to capture temporal dependencies.

* Corresponding author.

- We design a Hierarchical Guided Fusion (HGF) module, which establishes a coarse-to-fine hierarchical feature fusion structure and promotes cross-scale information interaction. Additionally, we design a Multi-Predictor Synthesis (MPS) module that utilizes spectral energy weights to adaptively ensemble multi-scale predictions.
- Extensive experiments on 6 real-world datasets demonstrate that AMGNet consistently achieves leading performance among 12 mainstream methods, reducing MSE by an average of 1.24% compared to the best baseline. This confirms the significant superiority of AMGNet[1].

II. Related Works

A. Deep Models for Time Series Forecasting

Deep learning has revolutionized time series forecasting, evolving from early RNNs and CNNs [12], [13] to dominant Transformer-based [5], [14], [15] and MLP-based [6], [16] architectures. To capture complex temporal patterns, recent works have introduced specialized multi-scale designs, including multi-update frequencies [17], 2D variations [4], pyramidal attention [18], and frequency decomposition [19], [20]. Despite these advancements, many models adopt a Channel Independence strategy, which inherently sacrifices inter-variable synergies and creates information bottlenecks in highly coupled systems. Furthermore, existing multi-scale methods typically process different scales in parallel. This lacks explicit cross-scale information interaction, leading to the entanglement of different fluctuations with redundant information.

B. GNNs for Time Series Forecasting

Graph Neural Networks (GNNs) have become a powerful paradigm for modeling inter-series dependencies [7], [8], [11], [21]. Since predefined graphs [9] are often unavailable in real-world scenarios, Graph Structure Learning (GSL) methods [10], [21] were proposed to adaptively learn adjacency matrices from data. Recent efforts also attempt to combine GNNs with multi-scale modeling to capture spatial-temporal dynamics across different scales [7], [8], [11]. However, these methods typically rely on static or uniform graph aggregation mechanisms. They fail to accommodate the personalized information selection needs of different nodes and are prone to introducing noise. Regarding cross-scale interaction, most methods aggregate information via simple summation or concatenation. Without explicit hierarchical guidance, local errors tend to accumulate over long horizons, leading to performance degradation.

III. Method

A. Problem Definition

A multivariate time series can be formally represented as a matrix $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L\} \in \mathbb{R}^{L\times N}$, where L denotes the length of the historical look-back window and N represents the number of variables. Here, $\mathbf{x}_t \in \mathbb{R}^N$ captures the observations of all variables at time step t. Given the historical observations $\mathbf{X}$, the forecasting task aims to predict the future sequence $\mathbf{Y} = \{\mathbf{x}_{L+1}, \dots, \mathbf{x}_{L+T}\} \in \mathbb{R}^{T\times N}$ for the next T time steps. Mathematically, the objective is to learn a mapping function f_θ with learnable parameters θ to generate the prediction $\hat{\mathbf{Y}}$:

$$\hat{\mathbf{Y}} = f_\theta(\mathbf{X}), \quad \text{s.t. } \hat{\mathbf{Y}} \approx \mathbf{Y} \tag{1}$$

where $\hat{\mathbf{Y}} \in \mathbb{R}^{T\times N}$ denotes the predicted values corresponding to the ground truth $\mathbf{Y}$.

B. Overall Architecture

As illustrated in Fig. 1, the proposed **AMGNet** consists of four parts: (1) **Frequency-based Multi-scale Decomposition** (FMD) disentangles the input into multi-scale periodic components and extracts a global context. (2) **Adaptive Gated Spatio-Temporal Modeling** captures inter-series dependencies via adaptively learned graphs and a Node-Aware Gated Message Passing (NAGMP) module, followed by self-attention for intra-series temporal correlations. (3) **Hierarchical Guided Fusion** constructs a coarse-to-fine pathway, utilizing low-frequency trends to explicitly guide high-frequency detail learning. (4) **Multi-Predictor Synthesis** independently generates scale-specific forecasts and adaptively ensembles them for the final prediction.

C. Input Embedding and Multi-scale Decomposition

1) Input Embedding: To capture semantic correlations and temporal context, we map the normalized historical input $\hat{\mathbf{X}} \in \mathbb{R}^{L\times N}$ into a hidden dimension d:

$$\mathbf{X}_{emb} = \alpha\mathcal{C}(\hat{\mathbf{X}}) + \mathbf{PE} + \sum_{p=1}^{P} \mathbf{SE}_p. \tag{2}$$

Here, $\mathcal{C}(\cdot)$ is a 1-D convolution (kernel size 3), α balances magnitudes, and $\mathbf{PE}, \mathbf{SE}_p \in \mathbb{R}^{L\times d}$ denote positional and learnable global timestamp embeddings.

2) Frequency-based Multi-scale Decomposition: To disentangle intricate periodic patterns [4], we perform Fast Fourier Transform (FFT) on $\mathbf{X}_{emb}$ along the temporal dimension. By averaging amplitudes across channels, we obtain a global amplitude vector $\mathbf{A}_{mp} \in \mathbb{R}^L$. We identify the top-$K$ frequencies, corresponding to dominant periods $\{p_1, \dots, p_K\}$. Sorting these periods from low-frequency trends to high-frequency details, we reshape $\mathbf{X}_{emb}$ for the k-th scale:

$$\mathbf{X}^{(k)} = \text{Reshape}(\text{Pad}(\mathbf{X}_{emb}), p_k) \in \mathbb{R}^{\lceil \frac{L}{p_k} \rceil \times p_k \times N \times d}. \tag{3}$$

This decomposes 1D dependencies into intra-period and inter-period variations for specific scale modeling.

D. Intra-Scale: Adaptive Gated Spatio-Temporal Modeling

1) Adaptive Graph Structure Learning: To uncover dynamic inter-series correlations without predefined graphs, we learn an adaptive adjacency matrix $\mathbf{A}^{(k)} \in \mathbb{R}^{N\times N}$ for the k-th scale using learnable node embeddings $\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{R}^{N\times d_e}$:

$$\mathbf{A}^{(k)} = \text{Softmax}(\text{ReLU}(\mathbf{E}_1\mathbf{E}_2^\top)). \tag{4}$$

The ReLU activation ensures sparsity, while Softmax provides a probabilistic interpretation of edge weights.

[1]https://github.com/Gadiaola/AMGNet

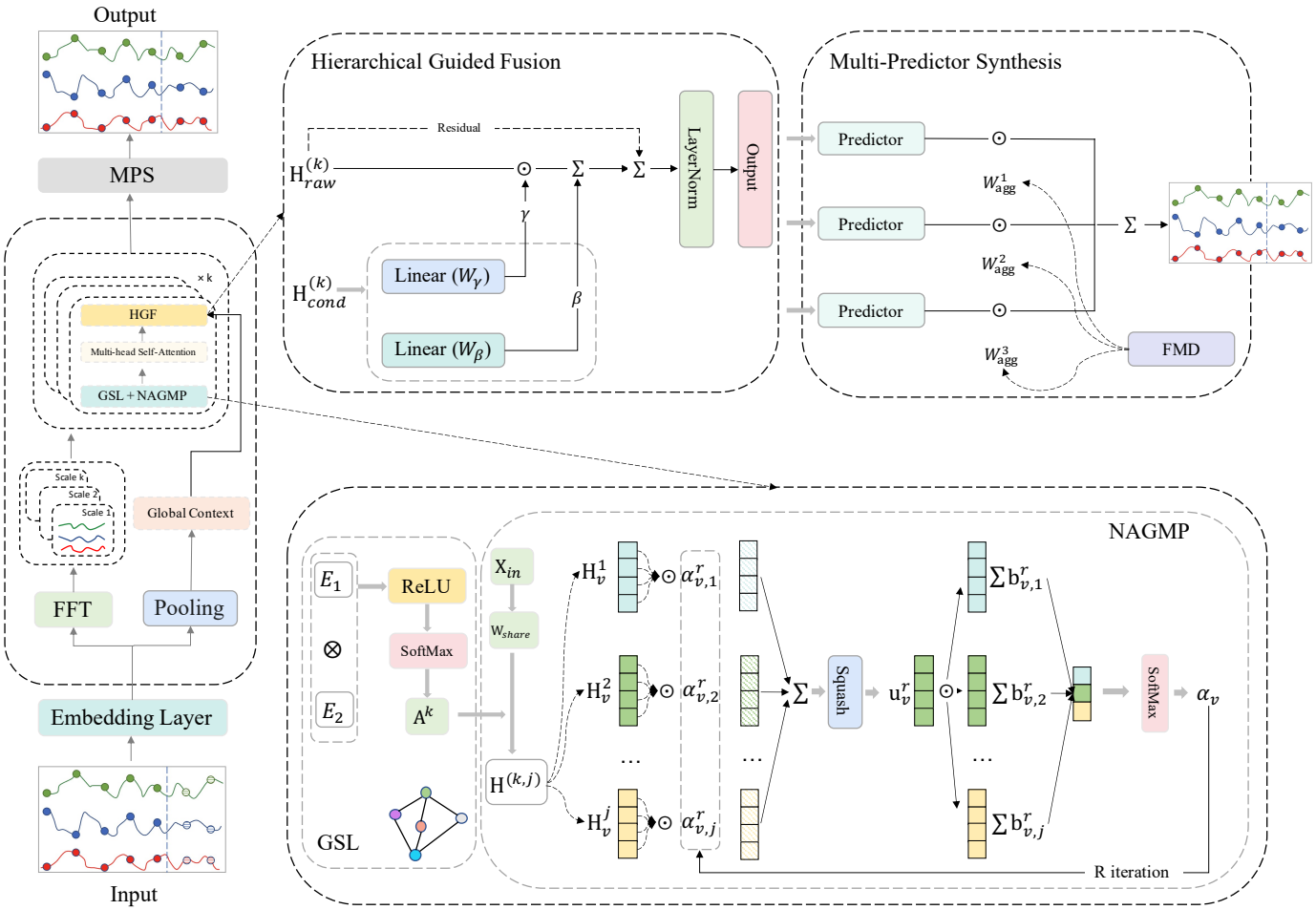

Fig. 1. The overall architecture of AMGNet, which begins by decomposing the input series into multi-scale representations via frequency-based analysis. Then, the Adaptive Gated Spatio-Temporal Modeling module captures inter- and intra-series correlations at different scales, followed by the HGF module which implements coarse-to-fine hierarchical feature fusion. Finally, the MPS module adaptively aggregates prediction results based on spectral energy weights.

2) Node-Aware Gated Message Passing: To overcome the limitations of fixed aggregation rules and noise from rigid multi-hop propagation, we introduce the NAGMP module inspired by dynamic routing [22]. We first flatten $\mathbf{X}^{(k)}$ into $\mathbf{X}_{in} \in \mathbb{R}^{L\times N\times d}$ and project it via a shared weight $\mathbf{W}_{share}$. The candidate feature at the j-th hop is:

$$\mathbf{H}^{(k,j)} = \mathrm{LN}\left(\left((\mathbf{A}^{(k)})^j(\mathbf{X}_{in}\mathbf{W}_{share})\right) \odot \mathbf{g}^{(j)}\right), \quad (5)$$

where $\mathbf{g}^{(j)} \in \mathbb{R}^d$ is a learnable scale vector for hop-wise calibration. Unlike static aggregation, NAGMP determines hop importance iteratively. For node v and hop j, the coupling coefficient $\alpha_{v,j}^{(r)}$ in iteration r is computed via softmax over routing logits $\mathbf{b}_{v,j}$:

$$\alpha_{v,j}^{(r)} = \frac{\exp(\mathbf{b}_{v,j}^{(r)})}{\sum_{j'=0}^{J}\exp(\mathbf{b}_{v,j'}^{(r)})}. \quad (6)$$

The current consensus vector $\mathbf{u}_v^{(r)}$ is obtained via the non-linear *Squash* function, compressing magnitude to represent confidence while preserving orientation:

$$\mathbf{u}_v^{(r)} = \mathrm{Squash}\left(\sum_{j=0}^{J}\alpha_{v,j}^{(r)}\mathbf{H}_v^{(k,j)}\right) = \frac{\|\mathbf{s}_v^{(r)}\|^2}{1+\|\mathbf{s}_v^{(r)}\|^2}\frac{\mathbf{s}_v^{(r)}}{\|\mathbf{s}_v^{(r)}\|}, \quad (7)$$

where $\mathbf{s}_v^{(r)}$ is the weighted sum. Crucially, routing logits are updated based on agreement (dot product):

$$\mathbf{b}_{v,j}^{(r+1)} = \mathbf{b}_{v,j}^{(r)} + \mathbf{H}_v^{(k,j)} \cdot \mathbf{u}_v^{(r)}. \quad (8)$$

After R iterations, the final spatial feature map $\mathbf{H}_{sp}^{(k)}$ is derived by adding a learnable bias $\boldsymbol{\beta}$ to the consensus matrix $\mathbf{U}^{(R)}$ and reshaping to restore temporal dimensions:

$$\mathbf{H}_{sp}^{(k)} = \mathrm{Reshape}\left(\mathbf{U}^{(R)} + \boldsymbol{\beta}\right) \in \mathbb{R}^{\lceil \frac{L}{p_k}\rceil \times p_k \times N\times d}. \quad (9)$$

3) Multi-head Self-Attention: To capture intra-series temporal correlations, we apply a standard Transformer encoder layer [23] (multi-head self-attention and feed-forward network) along the temporal dimension:

$$\mathbf{H}_{raw}^{(k)} = \mathrm{AttentionBlock}(\mathbf{H}_{sp}^{(k)}). \quad (10)$$

E. Inter-Scale: Hierarchical Guided Fusion

To prevent error accumulation and information isolation, HGF establishes a coarse-to-fine pathway where macroscopic trends explicitly guide microscopic fluctuations. We flatten the intra-scale output to $\mathbf{H}_{raw}^{(k)}$. The refined feature $\mathbf{H}_{refined}^{(k)}$ is generated sequentially from the coarsest trend ($k = 1$) to the finest detail ($k = K$). For the coarsest scale ($k = 1$), the conditioning feature $\mathbf{H}_{cond}^{(1)}$ is a global context vector $\mathbf{c}_{global}$ (derived via average pooling on $\mathbf{X}_{emb}$). For subsequent scales ($k > 1$), $\mathbf{H}_{cond}^{(k)} = \mathbf{H}_{refined}^{(k-1)}$. We inject guidance via feature-wise linear modulation [24]:

$$\mathbf{H}_{mod}^{(k)} = \mathbf{H}_{raw}^{(k)} \odot (1 + \tanh(\boldsymbol{\gamma})) + \boldsymbol{\beta}, \tag{11}$$

where $\boldsymbol{\gamma} = \mathbf{W}_{\gamma}\mathbf{H}_{cond}^{(k)}$ and $\boldsymbol{\beta} = \mathbf{W}_{\beta}\mathbf{H}_{cond}^{(k)}$ adjust the amplitude and baseline distribution dynamically. A residual connection and Layer Normalization yield the final refined representation:

$$\mathbf{H}_{refined}^{(k)} = \text{LN}(\mathbf{H}_{mod}^{(k)} + \text{Dropout}(\mathbf{H}_{raw}^{(k)})). \tag{12}$$

F. Multi-Predictor Synthesis

Since predictability varies across scales, MPS adopts a divide-and-conquer strategy. We assign K independent linear predictors to map each refined feature $\mathbf{H}_{refined}^{(k)}$ to a scale-specific forecast $\hat{\mathbf{Y}}^{(k)} \in \mathbb{R}^{T \times N}$. The final prediction $\hat{\mathbf{Y}}$ is adaptively ensembled using spectral energy weights:

$$\hat{\mathbf{Y}} = \sum_{k=1}^{K} \mathbf{W}_{agg}^{(k)} \hat{\mathbf{Y}}^{(k)}, \quad \mathbf{W}_{agg}^{(k)} = \frac{\exp(A_k)}{\sum_{j=1}^{K} \exp(A_j)}, \tag{13}$$

where A_k is the amplitude of the k-th frequency component from the FMD stage. This ensures dominant patterns contribute proportionally more while suppressing low-energy noise.

TABLE I
SUMMARY OF DATASETS

Dataset	Variates	Timesteps	Frequency	Information
ETTh1	7	17,420	Hourly	Temperature
ETTh2	7	17,420	Hourly	Temperature
ETTm1	7	69,680	15 mins	Temperature
ETTm2	7	69,680	15 mins	Temperature
Weather	21	52,696	10 mins	Weather
Electricity	321	26,304	Hourly	Electricity

IV. EXPERIMENTS

A. Datasets

We evaluate AMGNet on 6 widely used real-world benchmarks: four ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2), Weather, and Electricity. Detailed dataset statistics are summarized in Table I. We adopted the same data preprocessing and train-validation-test split scheme as previous studies [5], [11]. Each dataset is chronologically divided into training, validation, and test sets. The split ratio is set to 6:2:2 for the ETT datasets and 7:1:2 for the Weather and Electricity datasets.

B. Baselines

We selected 12 time series prediction methods for comparison, covering the current mainstream deep learning methods. (1) Transformer-based models: **iTransformer** [5], **PatchTST** [14], **Crossformer** [15], **Stationary** [25]. (2) CNN-based models: **TimesNet** [4], **SCINet** [13]. (3) MLP-based models: **FreTS** [26], **TiDE** [6], **DLinear** [16]. (4) GNN-based models: **MSHyper** [11], **MSGNet** [8], **MTGNN** [10].

C. Experimental Setups

To ensure a fair comparison, we standardize the input length to $L = 96$, while the output length is set to $T \in \{96, 192, 336, 720\}$. We employ the Adam optimizer with an initial learning rate of 10^{-4}. The batch size is set to 32, and the model is trained for a maximum of 15 epochs. Implement the early stop strategy. If no improvement is observed within 3 epochs, the training process will be terminated. The maximum hop count J and the number of routing iterations R are both selected from $\{2, 3\}$. We adopt Mean Squared Error (MSE) and Mean Absolute Error (MAE) as evaluation metrics, where lower values indicate superior performance. All the experiments were conducted on a Linux server equipped with an NVIDIA GeForce RTX 3090 24GB GPU.

D. Main Results and Analysis

Table II reports the prediction performance of all methods on 6 datasets. Bold and Underline denote the best and second best performance, respectively. As shown in the figure, AMGNet has achieved state-of-the-art performance across most datasets and prediction ranges. When evaluated across 30 comparison cases, our model achieved the best MSE performance in 16 cases and the best MAE performance in 20 cases. Significantly better than the competitive baseline. On average across all 6 datasets, our model reduces the MSE by 1.24% compared to the best-performing baseline for each metric. Notably, on the ETTh1 dataset and ETTh2 datasets, AMGNet demonstrates significant superiority, achieving MSE reductions of 3.98% and 3.13%, respectively. This superior performance can be attributed to AMGNet's ability to effectively integrate multi-scale dynamics while mitigating information redundancy and enhancing global awareness, thereby addressing key limitations in existing methods. PatchTST and DLinear performed poorly, which might be due to the fact that they usually handle variables independently. This overlooks complex inter-variable correlations and limits their performance. Although TimesNet and SCINet incorporate multi-scale information, their performance still lags behind AMGNet. This may be because they lack an explicit cross-scale interaction mechanism and ignore the mutual constraints among different time patterns, leading to error accumulation. MSHyper captures high-order correlations and scale interactions via hypergraphs, outperforming previous GNN methods, but still lags behind AMGNet. This indicates that relying on predefined rules often fails to capture implicit interactions and is prone to introducing noise, thereby damaging prediction accuracy.

TABLE II

LONG-TERM FORECASTING RESULTS WITH INPUT LENGTH $L = 96$ AND PREDICTION LENGTHS $T \in \{96, 192, 336, 720\}$. **BOLD** INDICATES THE BEST RESULT, AND UNDERLINE INDICATES THE SECOND BEST.

Models		AMGNet		iTransformer		MSHyper		FreTS		MSGNet		PatchTST		TiDE		Crossformer		TimesNet		DLinear		SCINet		Stationary		MTGNN	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.319**	**0.355**	0.334	0.368	0.348	0.369	0.335	0.372	**0.319**	0.366	0.329	0.367	0.364	0.387	0.404	0.426	0.338	0.375	0.345	0.372	0.418	0.438	0.386	0.398	0.381	0.415
	192	**0.362**	**0.380**	0.377	0.391	0.392	0.391	0.388	0.401	0.376	0.397	0.367	0.385	0.398	0.404	0.450	0.451	0.374	0.387	0.380	0.389	0.439	0.450	0.459	0.444	0.442	0.451
	336	0.402	**0.404**	0.426	0.420	0.426	0.410	0.421	0.426	0.417	0.422	**0.399**	0.410	0.428	0.425	0.532	0.515	0.410	0.411	0.413	0.413	0.490	0.485	0.495	0.464	0.475	0.475
	720	0.467	0.442	0.491	0.459	0.483	0.448	0.486	0.465	0.481	0.458	**0.454**	**0.439**	0.487	0.461	0.666	0.589	0.478	0.450	0.474	0.453	0.595	0.550	0.585	0.516	0.531	0.507
	Avg	0.388	**0.395**	0.407	0.410	0.412	0.405	0.408	0.416	0.398	0.411	**0.387**	0.400	0.419	0.419	0.513	0.495	0.400	0.406	0.403	0.407	0.485	0.481	0.481	0.456	0.457	0.462
ETTm2	96	**0.175**	**0.254**	0.180	0.264	0.183	0.267	0.189	0.277	0.177	0.262	**0.175**	0.259	0.207	0.305	0.287	0.366	0.187	0.267	0.193	0.292	0.286	0.377	0.192	0.274	0.240	0.343
	192	**0.238**	**0.294**	0.250	0.309	0.257	0.313	0.258	0.326	0.247	0.307	0.241	0.302	0.290	0.364	0.414	0.492	0.249	0.309	0.284	0.362	0.399	0.445	0.280	0.339	0.398	0.454
	336	**0.304**	**0.336**	0.311	0.348	0.335	0.361	0.343	0.390	0.312	0.346	0.305	0.343	0.377	0.422	0.597	0.542	0.321	0.351	0.369	0.427	0.637	0.591	0.334	0.361	0.568	0.555
	720	0.405	**0.396**	0.412	0.407	0.410	0.402	0.495	0.480	0.414	0.403	**0.402**	0.400	0.558	0.524	1.730	1.042	0.408	0.403	0.554	0.522	0.960	0.735	0.417	0.413	1.072	0.767
	Avg	**0.281**	**0.320**	0.288	0.332	0.296	0.336	0.321	0.368	0.288	0.330	**0.281**	0.326	0.358	0.404	0.757	0.611	0.291	0.333	0.350	0.401	0.571	0.537	0.306	0.347	0.570	0.530
ETTh1	96	0.385	**0.398**	0.386	0.405	0.392	0.407	0.395	0.407	0.390	0.411	0.414	0.419	0.479	0.464	0.423	0.448	**0.384**	0.402	0.386	0.400	0.654	0.599	0.513	0.491	0.440	0.450
	192	**0.432**	0.429	0.441	0.436	0.440	**0.426**	0.448	0.440	0.442	0.442	0.460	0.445	0.525	0.492	0.471	0.474	0.436	0.429	0.437	0.432	0.719	0.631	0.534	0.504	0.449	0.433
	336	**0.466**	**0.448**	0.487	0.458	0.480	0.453	0.499	0.472	0.480	0.468	0.501	0.466	0.565	0.515	0.570	0.546	0.491	0.469	0.481	0.459	0.778	0.659	0.588	0.535	0.598	0.554
	720	**0.451**	**0.461**	0.503	0.491	0.508	0.493	0.558	0.532	0.494	0.488	0.500	0.488	0.594	0.558	0.653	0.621	0.521	0.500	0.519	0.516	0.836	0.699	0.643	0.616	0.685	0.620
	Avg	**0.434**	**0.434**	0.454	0.448	0.455	0.445	0.475	0.463	0.452	0.452	0.469	0.455	0.541	0.507	0.529	0.522	0.458	0.450	0.456	0.452	0.747	0.647	0.570	0.537	0.543	0.514
ETTh2	96	0.307	0.350	**0.297**	0.349	0.300	0.351	0.309	0.364	0.328	0.371	0.302	**0.348**	0.400	0.440	0.745	0.584	0.340	0.374	0.333	0.387	0.707	0.621	0.476	0.458	0.496	0.509
	192	0.383	**0.398**	**0.380**	0.400	0.384	0.400	0.395	0.425	0.402	0.414	0.388	0.400	0.528	0.509	0.877	0.656	0.402	0.414	0.477	0.476	0.860	0.689	0.512	0.493	0.716	0.616
	336	**0.401**	**0.420**	0.428	0.432	0.443	0.438	0.462	0.467	0.435	0.443	0.426	0.433	0.643	0.571	1.043	0.731	0.452	0.452	0.594	0.541	1.000	0.744	0.552	0.551	0.718	0.614
	720	**0.393**	**0.419**	0.427	0.445	0.412	0.441	0.721	0.604	0.417	0.441	0.431	0.446	0.874	0.679	1.104	0.763	0.462	0.468	0.831	0.657	1.249	0.838	0.562	0.560	1.161	0.791
	Avg	**0.371**	**0.396**	0.383	0.407	0.385	0.408	0.472	0.465	0.396	0.417	0.387	0.407	0.611	0.550	0.942	0.684	0.414	0.427	0.559	0.515	0.954	0.723	0.526	0.516	0.773	0.633
ECL	96	0.161	0.266	**0.148**	**0.240**	0.176	0.261	0.176	0.258	0.165	0.274	0.181	0.270	0.237	0.329	0.219	0.314	0.168	0.272	0.197	0.282	0.247	0.345	0.169	0.273	0.211	0.305
	192	0.177	0.281	**0.162**	**0.253**	0.173	0.260	0.175	0.262	0.184	0.292	0.188	0.274	0.236	0.330	0.231	0.322	0.184	0.289	0.196	0.285	0.257	0.355	0.182	0.286	0.225	0.319
	336	0.189	0.293	**0.178**	**0.269**	0.195	0.297	0.185	0.278	0.195	0.302	0.204	0.293	0.249	0.344	0.246	0.337	0.198	0.300	0.209	0.301	0.269	0.369	0.200	0.304	0.247	0.340
	720	**0.212**	**0.310**	0.225	0.317	0.219	0.315	0.220	0.315	0.231	0.332	0.246	0.324	0.284	0.373	0.280	0.363	0.220	0.320	0.245	0.333	0.299	0.390	0.222	0.321	0.287	0.373
	Avg	0.185	0.288	**0.178**	**0.270**	0.191	0.283	0.189	0.278	0.194	0.300	0.205	0.290	0.251	0.344	0.244	0.334	0.193	0.295	0.212	0.300	0.268	0.365	0.193	0.296	0.243	0.334
Weather	96	0.160	**0.205**	0.174	0.214	0.170	0.223	0.174	0.208	0.163	0.212	0.177	0.218	0.202	0.261	**0.158**	0.230	0.172	0.220	0.196	0.255	0.221	0.306	0.173	0.223	0.171	0.231
	192	0.214	0.251	0.221	0.254	0.218	0.253	0.219	**0.250**	0.212	0.254	0.225	0.259	0.242	0.298	**0.206**	0.277	0.219	0.261	0.237	0.296	0.261	0.340	0.245	0.285	0.215	0.274
	336	**0.268**	**0.290**	0.278	0.296	0.269	0.300	0.273	**0.290**	0.272	0.299	0.278	0.297	0.287	0.335	0.272	0.335	0.280	0.306	0.283	0.335	0.309	0.378	0.321	0.338	0.266	0.313
	720	0.349	0.345	0.358	0.347	0.343	0.341	**0.334**	**0.332**	0.350	0.348	0.354	0.348	0.351	0.386	0.398	0.418	0.365	0.359	0.345	0.381	0.377	0.427	0.414	0.410	0.344	0.375
	Avg	**0.247**	0.272	0.258	0.278	0.250	0.279	0.250	**0.270**	0.249	0.278	0.259	0.281	0.271	0.320	0.259	0.315	0.259	0.287	0.265	0.317	0.292	0.363	0.288	0.314	0.249	0.298
1st Count		**16**	**20**	6	4	0	1	1	4	1	0	6	2	0	0	2	0	1	0	0	0	0	0	0	0	0	0

TABLE III

ABLATION STUDY OF AMGNET AND ITS VARIANTS.

Dataset	ETTh1		ETTh2		ETTm1		ETTm2	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
AMGNet	**0.434**	**0.434**	**0.371**	**0.396**	**0.388**	**0.395**	**0.281**	**0.320**
w/o-SpM	0.452	0.443	0.391	0.409	0.448	0.428	0.309	0.346
w/o-NAGMP	0.444	0.447	0.383	0.406	0.423	0.417	0.305	0.342
w/o-HGF	0.446	0.444	0.386	0.407	0.416	0.410	0.307	0.344
w/o-MPS	0.442	0.443	0.379	0.401	0.427	0.417	0.302	0.340

E. Ablation Study

To verify the effectiveness of each component in AMGNet, we considered 4 variants and conducted a comprehensive evaluation on 4 datasets. The specific variants are defined as follows:

- w/o-SpM: We removed the entire adaptive graph structure learning and NAGMP module.
- w/o-NAGMP: We replaced the NAGMP module with the MixHop convolution method [27].
- w/o-HGF: We removed the entire Hierarchical Guided Fusion module.
- w/o-MPS: We replaced the Multi-Predictor Synthesis with a simple linear layer for prediction (aggregate-then-predict).

Table III summarizes the results of the ablation study. As shown, removing any component leads to performance degradation. We observed that w/o-SpM caused the most significant performance drop across all datasets, demonstrating that explicitly modeling inter-series correlations is crucial for multivariate time series forecasting, and our adaptive graph structure learning and NAGMP module effectively captures these dependencies. The results of w/o-NAGMP indicate that substituting our NAGMP with MixHop leads to a decrease in accuracy. Although MixHop has previously been proven superior to traditional convolution [8], but it fundamentally relies on static uniform aggregation rules, failing to meet node-specific informational requirements, which demonstrates the effectiveness of the NAGMP. The results of w/o-HGF show that disrupting hierarchical structure harms performance. This suggests that the trend-guiding-detail strategy is essential. By allowing coarse-grained patterns to guide fine-grained feature learning, HGF successfully mitigates error accumulation in non-stationary series. The results of w/o-MPS indicate that MPS outperforms the aggregate-then-predict strategy, because it avoids the premature mixing of different frequency components, thereby preventing high-frequency noise from distorting the prediction of stable trends.

F. Parameter Study

To investigate the impact of the number of scales (k) on forecasting performance, we conducted a sensitivity analysis on the ETTh1 dataset. As illustrated in Fig. 2, the model achieves the best performance across all prediction horizons when $k = 3$. We observe that increasing k from 2 to 3 yields performance gains, benefiting from the incorporation of richer multi-resolution context. However, further increasing k beyond

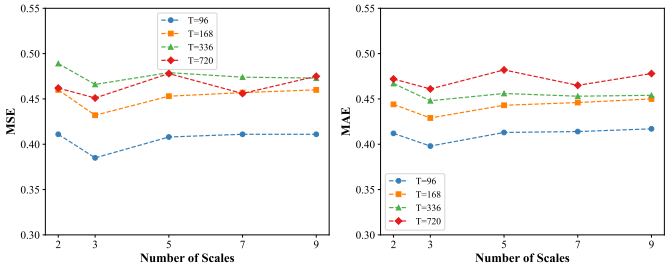

Fig. 2. Effect of different hyperparameters.

3 leads to performance degradation. This phenomenon can be attributed to the trade-off between information gain and noise accumulation: an excessively large k decomposes the series into too many high-frequency components, which may introduce irrelevant noise and redundancy, thereby interfering with the modeling of dominant temporal trends.

V. Conclusion

In this work, we proposed AMGNet for multivariate time series forecasting. Empowered by Adaptive Gated Spatio-Temporal Modeling, Hierarchical Guided Fusion, and Multi-Predictor Synthesis modules, AMGNet effectively captures node-specific dynamic correlations, establishes coarse-to-fine hierarchical guidance, and synthesizes complementary predictions across scales. This design tackles two key challenges: modeling complex inter-series dependencies within different scales and the insufficiency of cross-scale information interaction. Extensive experiments on real-world datasets demonstrate the powerful performance of AMGNet, validating its effectiveness in handling complex spatiotemporal dependencies under multi-scale temporal patterns. In future work, we plan to optimize the complexity and computational efficiency of the model framework to better address the broad challenges of non-stationary scenarios.

Acknowledgment

This work is funded by the Key Technologies Research and Development Program of Liaoning Province (Grant No.2024JH2/102400072), the Fundamental Research Funds for the Central Universities (Grant No. N25BSS062).

References

[1] R.-G. Cirstea, B. Yang, C. Guo, T. Kieu, and S. Pan, "Towards spatio-temporal aware traffic time series forecasting," in *IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 2900–2913.

[2] Z. Zeng, R. Kaur, S. Siddagangappa, S. Rahimi, T. Balch, and M. Veloso, "Financial time series forecasting using CNN and transformer," *arXiv preprint arXiv:2304.04912*, 2023.

[3] L. Chen, J. Cui, Z. Shang, and D. Cui, "TPRNN: A top-down pyramidal recurrent neural network for time series forecasting," *Information Sciences*, vol. 699, p. 121792, 2025.

[4] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[5] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "iTransformer: Inverted transformers are effective for time series forecasting," in *International Conference on Learning Representations*, 2024.

[6] A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, and R. Yu, "Long-term forecasting with TiDE: Time-series dense encoder," *arXiv preprint arXiv:2304.08424*, 2023.

[7] L. Chen, D. Chen, Z. Shang, B. Wu, C. Zheng, B. Wen, and W. Zhang, "Multi-scale adaptive graph neural network for multivariate time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 10 748–10 761, 2023.

[8] W. Cai, Y. Liang, X. Liu, J. Feng, and Y. Wu, "MSGNet: Learning multi-scale inter-series correlations for multivariate time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 10, 2024, pp. 11 141–11 149.

[9] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

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

[11] Z. Shang and L. Chen, "MSHyper: Multi-scale hypergraph transformer for long-range time series forecasting," *arXiv preprint arXiv:2401.09261*, 2024.

[12] S. Bai, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[13] M. Liu, A. Zeng, M. Chen, Z. Xu, Q. Lai, L. Ma, and Q. Xu, "SCINet: Time series modeling and forecasting with sample convolution and interaction," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5816–5828, 2022.

[14] Y. Nie, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[15] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The Eleventh International Conference on Learning Representations*, 2023.

[16] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.

[17] Z. Chen, Q. Ma, and Z. Lin, "Time-aware multi-scale RNNs for time series modeling," in *IJCAI*, 2021, pp. 2285–2291.

[18] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *International Conference on Learning Representations*, 2022.

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

[20] T. Zhou, Z. Ma, Q. Wen, L. Sun, T. Yao, W. Yin, R. Jin *et al.*, "FiLM: Frequency improved Legendre memory model for long-term time series forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12 677–12 690, 2022.

[21] K. Yi, Q. Zhang, W. Fan, H. He, L. Hu, P. Wang, N. An, L. Cao, and Z. Niu, "FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective," *Advances in Neural Information Processing Systems*, vol. 36, pp. 69 638–69 660, 2023.

[22] A. G. Ahsaei, R. Ewetz, and H. Zheng, "GAMMA: Gated multi-hop message passing for homophily-agnostic node representation in GNNs," in *Advances in Neural Information Processing Systems*, 2025.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[24] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[25] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9881–9893, 2022.

[26] K. Yi, Q. Zhang, W. Fan, S. Wang, P. Wang, H. He, N. An, D. Lian, L. Cao, and Z. Niu, "Frequency-domain MLPs are more effective learners in time series forecasting," *Advances in Neural Information Processing Systems*, vol. 36, pp. 76 656–76 679, 2023.

[27] S. Abu-El-Haifa, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. V. Steeg, and A. Galstyan, "MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *International Conference on Machine Learning*, 2019.