2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Fine and Coarse-grained Graph Flow Neural Network for Traffic Forecasting

Yuhao Zhao[1], Zhanquan Wang[*]

[1, *]School of Information Science and Engineering
East China University of Science and Technology
Shanghai, China

zhqwang@ecust.edu.cn

Abstract. Recently, the Intelligent Transportation System has been developed to help relieve traffic congestion, which calls for the need to predict middle and long-term traffic flow accurately. However, existing models can't effectively obtain enough accuracy in middle and long-term prediction. To solve this, we propose a Fine and Coarse-grained Graph Flow Neural Network (FCGFNN), which makes better prediction by capturing both fluctuating and stable traffic patterns. Firstly, an asymmetric embedding layer is designed to integrate graph structure and temporal dependencies with two dimensions of data. Then, a Season-Trend Encoder is designed to extract essential spatial-temporal features as well as handling non-stationary flows. Finally, the pattern of traffic flow prediction is obtained. Experimental results on two real public traffic datasets shows average performance improvements of 5.9%, 7.5% and 7.7% across 30-minute, 45-minute and 60-minute prediction intervals.

Keywords: Traffic Forecasting, Decomposition Model, Attention Mechanism.

1 Introduction

In recent years, there are increasing pressures on transportation systems due to rapid urbanization. Intelligent Transportation System (ITS), as a pivotal solution, delivers feasible decisions for transportation management in urban cities [1]. As a foundation of ITS, traffic flow prediction utilizes historical data series to anticipate future traffic patterns, which has attracted much attention [2]. Through the strategic deployment of sensor networks across transportation infrastructure, ITS can capture real-time traffic data, alleviating congestion and optimizing traffic flow efficiency [3].

Traffic forecasting is a critical time series analysis challenge, distinguished by the complex inter-dependencies among spatial, temporal, and external factors [4] [5]. These dependencies substantially increase the complexity of the task. To recognize these uncertainties in raw traffic flow data, researchers have spent efforts on developing

static and dynamic representations. These representations represent the intrinsic characteristics of traffic flow from several dimensions, including periodicity, trends, and seasonal variations [6]. However, complex temporal dynamics, including mixed patterns and fluctuating series, present great challenges in accurately capturing and predicting traffic behavior [7].

Benefiting from deep learning techniques, many models have been proposed to capture the temporal patterns in traffic series. Notably, Graph Neural Networks (GNNs), integrated with Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), have received great popularity in recent years. By constructing data topology, GNNs improve the models' ability of capture spatial and temporal dependencies. The introduction of adjacency matrices also significantly improved the construction of spatial data topology. However, GNN-based approaches focus primarily on global traffic patterns, potentially neglecting inner-correlations embedded within the data. Capturing both intra-period temporal dynamics and cross-series dependencies is essential [8]. Firstly, traffic data may fluctuate suddenly due to severe weather conditions or signal interference, manifesting as high-frequency components in the raw data. Secondly, limited generalization capabilities may occur when applied to non-stationary traffic time series [5]. To solve this, decomposition techniques are employed in various models, such as Discrete Wavelet Transform (DWT) [5] and Discrete Fourier Transform (DFT) based methodologies [8]-[10]. Decomposition mechanism effectively captures traffic patterns across different dimensions while maintaining global temporal information. However, such kind of technique has limitation in completely isolating noise from the original data, thereby failing to fully extract crucial features from traffic series flows.

To address these challenges, we propose a novel Fine and Coarse-grained Graph Flow Neural Network (FCGFNN) for traffic condition prediction in road networks. Unlike previous studies that focus primarily on single-input or one-step decomposition approach, out model introduces a two-step decomposition framework. First, an asymmetric embedding layer is implemented to process fine and coarse-grained data flow, which have been disentangled through DWT in decomposition layer. Specifically, a general embedding is first applied to both data flows, and an additional embedding which contains spatial and temporal embedding is exclusively applied to coarse-grained layer. Second, this paper introduces a Season-Trend Encoder (ST-Encoder), which leverages DFT mechanism to extract high-frequency components from coarse-grained data flows. A temporal attention and sparse GAT attention is followed behind to further capture traffic flow patterns.

The principal contributions of this paper are summarized as follows:

- We propose a two-step decomposition model for traffic forecasting, which effectively captures intricate spatial-temporal dynamics through the decomposition layer and ST-Encoder.
- We incorporate an asymmetric embedding layer to model both spatial and temporal dependencies, enabling a comprehensive representation of the entire traffic graph and significantly enhancing the model's forecasting performance.

- To assess the performance of our proposed framework, an extensive baseline comparisons and ablation experiments are performed on two real-world public transportation datasets. The experimental results demonstrate the superior performance of our method over mainstream traffic forecasting baselines across 30-minute, 45-minute and 60-minute prediction intervals.

The rest of this paper is organized as follows. Section II provides a comprehensive literature review. Section III presents the problem formulation of traffic forecasting. Section IV, V, and VI detail the proposed method, introduce the experimental setup and results, and conclude the paper, respectively.

2 Related work

2.1 Traffic Forecasting

In this section, we will discuss main model architecture in traffic forecasting and special decomposition model. Research in traffic flow prediction has a long history. Conventional sequential models, including LSTM [11]-[13] and GRU [14] [15], are constrained by their inherent temporal processing architecture. Advancements in modeling spatial-temporal dependencies [4] in early years have demonstrated the remarkable performance of RNNs and CNNs in traffic flow prediction tasks. Notably, DCRNN [16] demonstrates superior capability in modeling complex spatial relationships and captures non-linear temporal patterns. The combination of GNNs [17] with RNN or CNN effectively leverages the graph topology to increase the spatial-temporal ability. For example, T-GCN [18], which integrates Graph Convolutional Networks (GCNs) with Gated Recurrent Units (GRUs), has demonstrated exceptional capability in learning intricate topological structures and capturing dynamic spatial dependencies. Graph WaveNet [19] innovatively combines GCN with an adaptive dependency matrix, which effectively models long-range temporal dependencies in traffic flows.

In addition to GNNs, the Transformer architecture has emerged as a prominent alternative framework [20] [21]. The core principle of attention mechanisms lies in their ability to assign weights to the most informative features based on input data characteristics [22]. Consequently, these mechanisms are mainly employed to model spatial dependencies in traffic networks. To capture dynamic spatial-temporal correlations in traffic systems, GMAN [23] introduces a hierarchical architecture integrating spatial and temporal attention blocks across multiple stages. Similarly, STGM [24] introduces an innovative attention mechanism specifically designed to capture the intricate relationships between temporal and spatial dependencies. Furthermore, recent advancements in traffic forecasting have verified the effectiveness of integrating attention mechanisms with graph-structured data [25]. For instance, RGDAN [26] incorporates both graph diffusion attention and temporal attention modules, enabling more accurate modeling of spatial dependencies in dynamic systems.

Despite these significant contributions, existing models predominantly emphasize the architectural enhancements for improved feature extraction, frequently failing to incorporate the intrinsic characteristics of traffic data itself.

2.2 Decomposition Model

Multi-scale decomposition frameworks have been widely adopted in time series analysis [27]. Unlike mainstream single-flow models, decomposition-based approaches divide traffic data into distinct dimensions, thereby capturing more intrinsic features. StemGNN [8] integrates Graph Fourier Transform (GFT) and DFT to effectively model intra-period correlations and temporal periodicity in complex datasets. STWave [5] leverages a disentangled-fusion framework to addresses distribution shift. By utilizing DWT, STWave decouples traffic data into stationary trend and non-stationary events, thereby improving model robustness. Furthermore, decomposing techniques have demonstrated effectiveness in long-period time series forecasting. TimeMixer [7] proposes a novel mixing architecture incorporating Past-Decomposable-Mixing, which effectively leverages disentangled time series components for enhanced predictive performance. Additionally, DRFormer [28] proposes a multi-scale Transformer model that incorporates a multi-scale sequence extraction mechanism, which is capable of capturing multi-resolution features. However, existing approaches typically neglect the inherent noise embedded in cross-scale traffic patterns, which can hinder their performance in middle and long-term predictions.

In this paper, we extend the exploration of feature extraction in traffic data flows. FCGFNN innovatively presents a two-step decomposable architecture based on previous designs. Building upon a fundamental decomposition layer, the model incorporates an asymmetric embedding layer that enriches decomposed traffic data with additional graph topology information. Furthermore, an ST-Encoder is implemented to capture spatial-temporal correlation at a more refined level.

3 Preliminaries

The traffic prediction problem is formally defined as a multivariate time-series forecasting task, where the road network topology is represented by a directed graph $\mathbb{G} = (V, E, A)$. Here, V represents the set of deployed traffic sensors and E encodes the adjacency relationships based on road connectivity, and $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix encoding the non-Euclidean distances between nodes i and j. The adjacency matrix is binary, consisting solely of zeros and ones, with its construction rule explicitly defined in Equation 1.

$$\mathrm{A}_{ij} = \begin{cases} 1 & \text{if } i, j \in \mathrm{V}, \mathrm{E}_{ij} \in \mathrm{E} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Given a sequence of observed traffic time series data $\{X_{c-S}, \cdots, X_{c-1}\}$ comprising S historical steps, the objective of multivariate time-series forecasting is to learn a mapping function that predicts the subsequent T future steps $\{Y_{c+1}, \cdots, Y_{c+T}\}$. This mapping can be formally expressed as the following mathematical function:

$$\{X_{c-S}, \cdots, X_{c-1}\} \overset{p}{\rightarrow} \{Y_{c+1}, \cdots, Y_{c+T}\} \tag{2}$$

4 Fine and Coarse-grained Graph Flow Neural Network

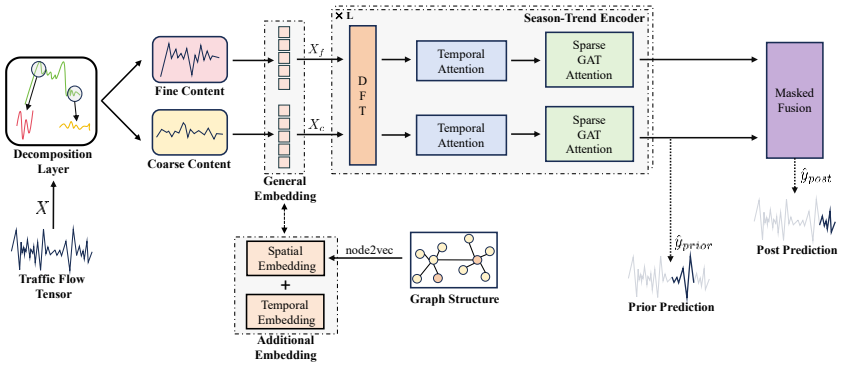

Fig. 1. The architecture of FCGFNN

Inspired by the decomposition layer in STWave[5], this paper introduces FCGFNN to address the challenge of feature extraction. As is illustrated in Figure 1, the proposed model is primarily composed of four key components: a decomposition layer, an asymmetric embedding layer, a season-trend encoder, and a masked fusion layer. The input traffic flow $X \in \mathbb{R}^{B\times T_i\times N\times D}$ is disentangled into fine and coarse content within the decomposition layer. Subsequently, these representations are embedded into $X_f \in \mathbb{R}^{B\times T_f\times N\times D}$ and $X_c \in \mathbb{R}^{B\times T_c\times N\times D}$ through the asymmetric embedding layer. Following the season-trend encoder, the fine and coarse-grained flows are fed into the masked fusion layer for further integration. Notably, FCGFNN employs two predictors: the coarse-grained data is output as $\hat{y}_{prior} \in \mathbb{R}^{B\times T_{out}\times N\times B}$ alone for prior prediction after the season-trend encoder, while the fusion data is output as $\hat{y}_{post} \in \mathbb{R}^{B\times T_{out}\times N\times B}$ after masked fusion for post prediction.

4.1 Asymmetric Embedding Layer

A fundamental bottleneck in urban traffic flow prediction lies in effectively capturing the intricate and dynamic spatial-temporal dependencies [29]. To address this challenge, we introduce an asymmetric embedding layer designed to comprehensively capture the structural features within the graph. As illustrated in Figure 1, the proposed embedding layer consists of two distinct components. The first component is a general embedding layer that applies to both fine-grained and coarse-grained flow data. The

second component is an additional embedding layer that exclusively applies spatial and temporal embedding to coarse-grained data.

General Embedding. Initially, we define an L-layer fully connected network, where each layer undergoes a linear transformation parameterized by W_i and the ReLU activation function follows behind. The final layer's output transforms the coarse-grained traffic flow X_c and fine-grained traffic flow X_f into their respective embedding, which is denoted as $\mathrm{E}_c \in \mathbb{R}^{\mathrm{N}c}$ and $\mathrm{E}_f \in \mathbb{R}^{\mathrm{N}f}$. Given traffic flow $\mathrm{X}_{c-S:c}$ which contains S historical steps, the multi-layer feed-forward propagation process is mathematically formulated as:

$$h_i = \begin{cases} \mathrm{W}_0\mathrm{X}_{c-S:c} + \mathrm{b}_0 & \text{if } i = 0 \\ \sigma(\mathrm{W}_i h_{i-1} + \mathrm{b}_i) & \text{if } 0 < i < L \\ \mathrm{W}_L\mathrm{h}_{L-1} + \mathrm{b}_L & \text{if } i = L \end{cases} \tag{3}$$

where W_i represents the weight matrix of the i^{th} layer, and $\sigma(\cdot)$ denotes the ReLU activation function. Following this process, the fine-grained embedding is directly output, whereas the coarse-grained embedding undergoes further integration with additional embedding.

Additional Embedding. Traffic dynamics, characterized by multi-dimensional spatial-temporal features, exhibit a strong correlation with the underlying road network structure. Consequently, we employ the node2vec algorithm [30] to improve low-dimensional node features representation. In a directed graph $\mathbb{G} = (\mathrm{V}, \mathrm{E})$, we define the random walk probability $\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx}$, where w_{vx} represents the edge weight between nodes v and x, and $\alpha_{pq}(t, x)$ is a probability function defined as follows:

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases} \tag{4}$$

where parameters p and q govern the direction of the random walk, while d_{tx} denotes the shortest path distance between nodes t and x. The random walk probability represents the likelihood of transitioning from node v to its neighboring node x. Based on the aforementioned transition probability, a graph neighborhood $N_S(v)$ is constructed using sampling strategy S. Subsequently, the Skip-gram architecture [31] [32] is adapted to learn node vector representations by maximizing the following objective function:

$$\max_f \sum_{v \in \mathrm{V}} \log Pr(N_S(v) | f(v)) \tag{5}$$

Here, f denotes a mapping function from nodes to their feature representations, and $Pr(N_S(v) \mid f(v))$ is the Softmax function which is defined as:

$$Pr(N_S(v)|f(v)) = \prod_{v_i \in N_S(v)} \frac{\exp(f(v_i)) \cdot f(v)}{\sum_{u \in V} \exp(f(u)) \cdot f(v))} \quad (6)$$

By processing the adjacency matrix using node2vec, we derive the spatial correlations among sensor nodes. Through the fully connected layer, the entire map is transformed into a spatial embedding $E_{se} \in \mathbb{R}^{N_{se}}$, where N_{se} represents the number of nodes. Additionally, the state of traffic flow is significantly influenced by temporal conditions. To capture inherent temporal features, we concatenate day-of-week and time-of-day embedding into $E_t \in \mathbb{R}^{N_w+N_d}$, where $N_w = 7$ and $N_d = 288$ represent the number of days in a week and timestamps per day, respectively. Finally, we concatenate all the embedding to obtain the fine-grained embedding $E_f \in \mathbb{R}^{N_f}$ and the coarse-grained embedding $E_c \in \mathbb{R}^{N_c+N_{se}+N_w+N_d}$.

4.2 Season-Trend Encoder Layer

The season-trend encoder layer serves as the second step to decompose the traffic data. The layer integrates DFT and self-attention mechanisms to effectively capture the intricate and dynamic spatial-temporal dependencies. The proposed encoder layer contains three distinct components. The first component is a DFT module, which refines the coarse-grained traffic flow by extracting noise, enabling the model to focus on stationary trends. The second component consists of temporal attention modules, designed to capture local and global temporal dependencies in traffic flow. Finally, the sparse GAT attention module further models the complex and dynamic spatial dependencies.

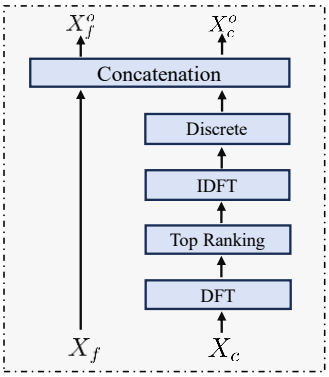

Fig. 2. DFT process

DFT module. Traffic flow can be conceptualized as an aggregation of effective information and noise. To address this, we employ a DFT module to extract the sudden pattern from coarse-grained time series while supplementing detailed information for the fine-grained scale. Fig. 2 details the framework of DFT mechanism, which consists of DFT, top-pool ranking, Inverse Discrete Fourier Transformation (IDFT), discrete module and concatenation mechanism. The DFT converts the time-domain signal x[n] into the frequency-domain signal X[k], formulated as follows:

$$\mathrm{X}[k] = \mathrm{DFT}(\mathrm{x}[n]) = \sum_{n=0}^{\mathrm{N}-1} \mathrm{x}[n] \cdot \mathrm{e}^{-j\frac{2\pi}{\mathrm{N}}kn}, k \in \mathbb{Z} \tag{7}$$

Compared to time-domain signal, frequency-domain signals offer distinct advantages in distinguishing between high and low-frequency components. To isolate noise, we apply a top-ranking technique, which involves defining a threshold. Frequency components below this threshold are set to zero, while the high-frequency portion of this signal, corresponding to values exceeding the threshold in X[k], is retained. Let $M = \min(\mathrm{top}_k(|X[k]|))$, where $\mathrm{top}_k(\cdot)$ returns the top k largest values. The top-ranking process can be formulated as:

$$\widetilde{\mathrm{X}}[k] = \begin{cases} X[k], & |X[k]| \geq M \\ 0, & |X[k]| < M \end{cases} \tag{8}$$

Subsequently, we use IDFT to reconstruct the time-domain signal from its frequency-domain representation, which generates the seasonal component $x_{season}[n]$. Mathematically, the IDFT is defined as:

$$x_{season}[n] = \mathrm{IDFT}(\tilde{X}[k]) = \frac{1}{N}\sum_{n=0}^{\mathrm{N}-1} \mathrm{x}[n] \cdot \mathrm{e}^{j\frac{2\pi}{\mathrm{N}}kn}, k \in \mathbb{Z} \tag{9}$$

The trend component $x_{trend}[n]$ is derived by subtracting the seasonal component from raw data in discrete module. This operation can be expressed as:

$$x_{trend}[n] = x[n] - x_{season}[n] \tag{10}$$

Finally, the trend component is combined with the fine-grained traffic flow to facilitate subsequent analysis.

Temporal attention. The characteristics of fine-grained fluctuation are sudden and intermittent, whereas coarse-grained flow exhibits stable and continuous behavior. To effectively model these distinct temporal patterns, we propose a temporal attention layer which leverages its global receptive field to capture overall traffic pattern while preserving high-frequency details. Specifically, fine-grained flow $x^f \in \mathbb{R}^{B\times T_f\times N\times D}$ and coarse-grained flow $x^c \in \mathbb{R}^{B\times T_c\times N\times D}$ are independently processed by the temporal attention respectively. The attention mechanism is mathematically formulated as:

$$\alpha^{c,f} = Softmax(\frac{Q^{c,f}K^{c,f}}{\sqrt{d}})V^{c,f} \tag{11}$$

Sparse GAT Attention. Graph Attention Network (GAT) implements a self-attentive mechanism to dynamically assign weights to adjacency relationships in traffic graphs, enabling edge-aware feature aggregation across spatially correlated road segments [33]. To effectively incorporate structural information into GAT, we extract Laplacian eigenvalues and eigenvectors from the temporal and spatial adjacency matrices and integrate them into the original input stream. The temporal adjacency matrix is defined as a symmetric matrix, as follows, while the spatial adjacency matrix represents the positional relationships of road sensors:

$$tem\_matrix = \begin{cases} 1 & \text{if } \mathrm{dtw}_{ij} < \mathrm{nth} \text{ or } \mathrm{dtw}_{ji} < \mathrm{nth} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

dtw denotes the Dynamic Time Warping (DTW) distance between nodes *i* and *j*, and nth represents the median value of the DTW distance distribution.

To reduce the computational complexity of GAT, we implement a query sampling strategy [5]. Specifically, a sampling key matrix K_{sample} is first generated based on the sensor adjacency matrix. By multiplying this with the query matrix Q, a sampling query-key matrix QK_{sample} is obtained, which encodes the correlations between the nodes represented in Q and K_{sample}. A top-k pooling operation is then applied to QK_{sample} to filter out the most informative nodes. Based on QK_{sample}, the final sampling query matrix Q_{sample} is derived, containing sensor nodes with significant information. Finally, a standard attention mechanism is applied, formulated as follows:

$$\mathrm{Attn} = \frac{exp((Q_{\text{sample}} \cdot K)^{\mathrm{T}}(Q_{\text{sample}} \cdot K))}{\sum_{k=1}^{n} exp(Q_{\text{sample}} \cdot K)} \tag{12}$$

4.3 Masked Fusion

Compared to single traffic flow model, fine-grained flow often exhibits fluctuating events that may deviate from predicted results. To retain useful patterns while filtering out noise, we incorporate a masked fusion module. Based on vanilla masked self-attention, this module computes attention scores while employing a mask to prevent the model from learning unnecessary timestamps, thereby ensuring accurate predictions. Given $\mathrm{Q} \in \{X_1^C, \ldots, X_N^C\}$, $\mathrm{K} \in \{X_1^F, \ldots, X_N^F\}$, $\mathrm{V} \in \{X_1^F, \ldots, X_N^F\}$, where X_i^C denotes coarse-grained flow while X_i^F denotes fine-grained flow, the vanilla attention is formulated as follows:

$$\hat{y} = Softmax(\frac{QK}{\sqrt{d}})V \tag{13}$$

4.4 Loss Function

To capture intrinsic data characteristics more effectively, we adopt a two-step prediction strategy. The coarse-grained data is firstly output by ST-Encoder to generate a prior prediction. The fusion data is then output by Masked Fusion to produce a post prediction. The L1 loss is employed to supervise the traffic forecasting task. The loss function is formulated as follows:

$$\mathrm{Loss} = \sum_{t=1}^{n} \left|\mathrm{Y}_t - \hat{y}_{prior}\right| + \sum_{t=1}^{n} \left|\mathrm{Y}_t - \hat{y}_{post}\right| \tag{14}$$

4.5 Complexity Analysis

The primary time-consuming processes in the proposed framework include the computation of eigenvalues and eigenvectors, the decomposition layer, and the ST-Encoder module. The computational complexity of calculating the eigenvalues and eigenvectors

of the Laplacian graph is $O(N^3)$, where N represents the number of nodes in the graph. Notably, this step can be preprocessed prior to the training phase, thereby not contributing to the overall model complexity during training. The decomposition layer exhibits a complexity of $O(NT)$, where T denotes the temporal dimension. Furthermore, the ST-Encoder module introduces a complexity of $O(N + L(TNK + NT^2 + TN\log N))$, with L representing the number of layers, and K indicating the dimension of the feature space. The pseudo-code for the proposed algorithm is provided as follows:

Algorithm 1 Training Procedure of FCGFNN

Input:

history data $X \in \mathbb{R}^{B\times T_{in}\times N\times D}$

time embedding $TE \in (R)^{B\times(N_w+N_d)\times D}$

adjacent matrix $ADJ \in \mathbb{R}^{N\times N}$

temporal eigenvector $\Phi_{\text{tem}} \in \mathbb{R}^{N\times d}$

spatial eigenvector $\Phi_{\text{spa}} \in \mathbb{R}^{N\times d}$

Output: Prediction Results $\hat{Y}_{prior} \in \mathbb{R}^{B\times T_{\text{out}}\times N\times D}, \hat{Y}_{post} \in \mathbb{R}^{B\times T_{\text{out}}\times N\times D}$

Decomposition Layer

$X^F, X^C \leftarrow Decompose(X)$

$X^F \leftarrow FC(X^F)$

$X^C \leftarrow FC(X^C)$

Asymmetric Embedding

$x^f \leftarrow \text{Emb}_L(X^F)$

$x^c \leftarrow \text{Emb}_H(X^C)$

$te \leftarrow \text{TeEmb}(TE)$

$ae \leftarrow \text{node2vec}(ADJ)$

Season-Trend Encoder

for $i = 1$ **to** L **do**

    $x^c_{\text{season}}, x^c_{\text{trend}} \leftarrow \text{DFT\_Decomp}(x^c_i)$

    $x^f_i \leftarrow x^f_i + x^c_{\text{trend}}$

    $\tilde{x}^c_i \leftarrow \text{TemporalAttention}(x^c_i + te + ae)$

    $\tilde{x}^f_i \leftarrow \text{TemporalAttention}(x^f_i)$

    $\tilde{x}^c_{i+1} \leftarrow \tilde{x}^c_i + \text{SparseGATAtt}(\tilde{x}^c_i, \Phi_{\text{spa}}, \Phi_{\text{tem}})$

    $\tilde{x}^f_{i+1} \leftarrow \tilde{x}^f_i + \text{SparseGATAtt}(\tilde{x}^f_i, \Phi_{\text{spa}}, \Phi_{\text{tem}})$

end for

$\hat{Y}_{prior} \leftarrow \tilde{x}^c_L$

Masked Fusion

$\hat{Y}_{post} \leftarrow \text{MaskedFusion}(\tilde{x}^c_L, \tilde{x}^f_L, te)$

return $\hat{Y}_{prior}, \hat{Y}_{post}$

5 Experiment

5.1 Datasets

We evaluate the performance of FCGFNN on two publicly available real datasets, namely PEMS04 and PEMS08, sourced from the California Transportation Agencies Performance Measurement System [34]. Each time interval is standardized to 5 minutes, resulting in 12 time slices per hour. To establish the prediction task, we utilize historical data from the past one hour to forecast traffic conditions for the subsequent one hour. Detailed statistical information regarding the datasets is presented in Table 1.

Table 1. Dataset Information

Datasets	Number of sensors	Interval	Time range
PEMS04	307	5min	1/1/2018-2/28/2018
PEMS08	170	5min	7/1/2016-8/31/2016

To ensure consistency and comparability across the dataset, we apply a standardization procedure using normalization. Following standardization, the dataset is partitioned into three subsets in chronological order: a training set (60%), a validation set (20%), and a test set (20%).

5.2 Parameter Setting

The training of FCGFNN is conducted using the Adam optimizer with a batch size of 64 and an initial learning rate of 0.001 over 100 epochs. To enhance convergence efficiency, a learning rate decay factor of 0.1 is applied when the loss does not decrease for 20 consecutive iterations during the training process. The default hyper-parameters of the model are configured as follows: the feature dimension d in FCGFNN is set to 128 for PEMS04 dataset and 64 for PEMS08 dataset. The decomposition level L of the DWT is set to 2, while the number of layers L in season-trend encoder is set as 2. Furthermore, the top k value of DFT is set to 5. Additionally, to optimize performance across different datasets, distinct discrete wavelets are employed: the Daubechies wavelet is utilized for PEMS04, while the Coiflets wavelet is applied to PEMS08.

5.3 Environment Setting

The proposed model is implemented in Python 3.8, leveraging the PyTorch 2.0.0 framework for efficient deep learning computations. All experimental evaluations are performed on a compute cluster featuring an AMD EPYC 9654 96-Core Processor and a single NVIDIA RTX 4090 GPU, running under Ubuntu 20.04 environment.

5.4 Baseline Models

We compare FCGFNN with following recently proposed deep learning spatial-temporal baseline models:

AGCRN [35]: A model utilizing GCN to capture fine-grained spatial and temporal dependencies in traffic series.

StemGNN [36]: A framework that integrates GFT and DFT mechanisms for spatio-temporal feature extraction.

STNorm [21]: A model incorporating temporal and spatial normalization techniques to enhance prediction accuracy.

DGCRN [3]: A RNN and GCN-based architecture designed to model the topology of dynamic graphs.

GMSDR [37]: An RNN-based approach that incorporates multiple historical time steps as input for each time unit.

MegaCRN [38]: A model that integrates a Meta-Graph Learner into the GCRN encoder-decoder framework.

STWave [5]: A disentangle-fusion framework that decouples complex traffic data into stable trends and fluctuating events for enhanced representation learning.

STG-NRDE [39]: A novel approach leveraging spatio-temporal graph neural rough differential equations for traffic flow modeling.

MHGNet [40]: A framework designed to model spatio-temporal multi-heterogeneous graphs for improved traffic prediction.

5.5 Metrics

To rigorously assess the performance of the proposed model, three widely adopted evaluation metrics are employed: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

5.6 Experiment Results

The performance of FCGFNN is evaluated across three prediction horizons: horizon 6, horizon 9, and horizon 12. Each horizon corresponds to a 5-minute interval in real-world scenarios. As demonstrated in Table 2, the best results are highlighted in bold and underlined, while the second-best results are marked with underlining. FCGFNN outperforms other models in most index.

AGCRN, DGCRN, and MegaCRN are all convolutional recurrent network-based models that leverage adaptive parameters, such as node embeddings, to enhance their performance. However, their approaches differ significantly. AGCRN infers node embeddings directly from the data and generates node-specific parameters from a shared pool of weights and biases. DGCRN employs a dynamic adjacency matrix generated by a hyper-network, which utilizes dynamic node features to improve graph generation effectiveness. MegaCRN integrates both adaptive and momentary embeddings into a meta-graph learning framework, enabling it to distinguish traffic patterns across different roads. Despite these advancements, these models exhibit limitations in capturing

deep spatial correlations between graph nodes. In contrast, FCGFNN's node embedding layer is designed to capture node features comprehensively, including static and dynamic embeddings, temporal and spatial features, as well as fine-grained and coarse-grained dimensions. This capability allows FCGFNN to construct a more intricate and comprehensive graph network, resulting in superior performance compared to these models.

Table 2. Experiment Results.

Dataset	Model	Horizon 6			Horizon 9			Horizon 12		
		MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
PEMS04	AGCRN	19.43	13.90	31.10	19.99	14.12	32.07	20.58	14.11	32.98
	StemGNN	23.01	16.67	35.37	25.44	18.69	38.76	28.13	20.44	42.50
	STNorm	19.17	12.87	31.95	19.70	13.15	32.66	20.37	13.86	33.50
	DGCRN	18.87	12.95	30.81	19.64	13.54	32.16	20.40	14.12	33.30
	GMSDR	19.30	13.20	30.82	20.07	13.75	31.98	20.87	14.39	33.09
	MegaCRN	18.75	12.76	30.57	19.54	13.24	31.98	20.27	13.69	33.13
	STWave	18.68	12.79	30.57	19.27	13.40	31.60	20.07	14.09	32.69
	STG-NRDE	19.24	12.78	31.01	19.87	13.25	32.04	20.69	13.87	33.16
	MHGNet	18.72	**12.44**	30.36	19.49	12.89	31.49	20.41	13.45	32.73
	FCGFNN	**18.46**	12.46	**30.22**	**18.96**	**12.84**	**31.02**	**19.60**	**13.37**	**31.94**
PEMS08	AGCRN	15.63	10.00	24.60	16.74	10.89	26.16	17.70	11.43	27.61
	StemGNN	16.49	11.26	26.02	17.83	12.19	28.07	19.11	12.95	30.02
	STNorm	15.44	9.75	25.01	16.17	10.21	26.43	16.93	10.69	27.58
	DGCRN	15.01	10.14	23.96	15.91	10.77	25.45	16.86	11.41	26.89
	GMSDR	16.35	10.74	25.33	17.25	11.45	26.63	18.13	11.99	27.95
	MegaCRN	14.75	9.44	23.81	15.64	10.02	25.24	16.46	10.54	26.53
	STWave	**13.80**	9.07	24.20	14.50	9.58	25.54	15.23	10.10	26.84
	STG-NRDE	15.69	10.09	24.85	16.49	10.55	26.10	17.46	11.45	27.51
	MHGNet	15.20	10.71	24.09	16.08	11.67	25.34	16.51	11.27	26.30
	FCGFNN	13.83	**9.05**	**23.49**	**13.34**	**9.36**	**24.44**	**14.95**	**10.02**	**25.41**

Like FCGFNN, STWave and MHGNet are both decomposition-based models. STWave disentangles complex traffic data into stationary trends and non-stationary events, while MHGNet decouples single-pattern traffic data into multi-pattern data. However, both

models fail to adequately address the residual noise present in the decoupled traffic data. FCGFNN's DFT module addresses this limitation by dividing coarse-grained flow into seasonal and trend components, effectively isolating noise from the stable trend. The trend component is then combined with fine-grained flow, enhancing the accuracy of long-term predictions by incorporating more stable trends.

FCGFNN also demonstrates superior performance compared to STG-NRDE, which models spatial and temporal information using neural rough differential equations (NRDE). This outcome underscores the effectiveness of FCGFNN's temporal and sparse GAT attention module in capturing spatio-temporal dependencies. STG-NRDE employs a two-step NRDE process to separately handle temporal and spatial dependencies, which may lead to the loss of essential correlations. In contrast, FCGFNN leverages its GAT attention mechanism to fully exploit the graph topology, enabling the model to extract more information from local correlations with neighboring nodes. Additionally, the asymmetric embedding layer in FCGFNN introduces additional node correlations that are not explicitly represented in the graph structure, further enhancing its predictive capabilities.

Table 3. Ablation Study on PEMS04 and PEMS08.

Dataset	Varients	Horizon 6			Horizon 9			Horizon 12		
		MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
PEMS04	w/o AE	21.13	14.18	34.21	23.17	15.70	37.30	25.30	17.31	40.46
	w/o DFT	18.48	12.94	30.25	19.03	13.40	31.14	19.82	14.02	32.23
	FCGFNN	**18.46**	**12.46**	**30.22**	**18.96**	**12.84**	**31.02**	**19.60**	**13.37**	**31.94**
PEMS08	w/o AE	14.70	9.56	25.68	15.99	10.33	27.96	17.23	11.22	29.97
	w/o DFT	14.01	9.42	23.93	14.60	9.66	25.10	15.23	10.26	26.11
	FCGFNN	**13.83**	**9.05**	**23.49**	**14.34**	**9.36**	**24.44**	**14.95**	**10.02**	**25.41**

5.7 Ablation Study

To comprehensively evaluate the contributions of individual components within FCGFNN, a series of ablation studies are conducted by comparing the full model with its variants. Specifically, the following configurations are examined:

1. FCGFNN: The complete model incorporates both the additional embedding and DFT modules.

2. FCGFNN (w/o AE): A variant of the model without the additional embedding, where fine-grained and coarse-grained embedding are identical.

3. FCGFNN (w/o DFT): A variant of the model with the DFT mechanism removed.

As depicted in Table 3, the removal of the additional embedding results in a noticeable decline in performance, while the exclusion of the DFT module leads to a performance degradation in MAPE. This observed behavior can be attributed to several crit-

ical factors. Firstly, the additional embedding plays a pivotal role by constructing dynamic graph representations that effectively capture spatial correlations among sensor nodes. Secondly, the additional embedding enriches the coarse-grained flow with supplementary information, thereby enhancing the effectiveness of the ST-Encoder. Thirdly, the DFT module serves a crucial function by segregating information-dense components from noise-dense elements within the original data. Collectively, these ablation studies underscore the indispensable contributions of both the additional embedding and DFT modules to the overall performance of FCGFNN, as evidenced by the experimental results.

6 Conclusion

To effectively capture the fluctuating and stable traffic patterns, the Fine and Coarse-grained Graph Flow Neural Network (FCGFNN) is proposed. Based on the foundational DWT decomposition mechanism, the asymmetric embedding layer is designed to integrate graph structure and temporal correlations with fine and coarse-grained traffic flow, which are sent into the ST-Encoder as input data. Furthermore, the DFT mechanism within the novel ST-Encoder is leveraged to effectively decompose the noise from coarse-grained data. Following DFT, the temporal attention mechanism and sparse GAT attention is utilized repeatedly in the ST-Encoder to learn spatial and temporal mappings. Finally, the pattern of traffic flow prediction is obtained. Extensive experiments are conducted on two real-world datasets, demonstrating superior performance compared to existing methods. As our future work, more external factors such as weather and temperature will be integrated to improve the performance.

REFERENCE

1. YIN C, XIONG Z, CHEN H, et al. A literature survey on smart cities[J/OL]. Science China Information Sciences, 2015: 1-18.
2. JIANG J, HAN C, ZHAO W, et al. PDFormer: Propagation Delay-aware Dynamic Long-range Transformer for Traffic Flow Prediction[J]. AAAI 2023, 2023.
3. LI F, FENG J, YAN H, et al. Dynamic Graph Convolutional Recurrent Network for Traffic Prediction: Benchmark and Solution[J/OL]. ACM Transactions on Knowledge Discovery from Data, 2023: 1-21.
4. LABLACK M, SHEN Y. Spatio-Temporal Graph Mixformer for Traffic Forecasting[J/OL]. Expert Systems With Applications, 2022.
5. Fang Y, Qin Y, Luo H, et al. When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks[C]//2023 IEEE 39th international conference on data engineering (ICDE). IEEE, 2023: 517-529.
6. WU H, HU T, LIU Y, et al. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis[J]. 2022.
7. Wang S, Wu H, Shi X, et al. Timemixer: Decomposable multiscale mixing for time series forecasting[J]. arXiv preprint arXiv:2405.14616, 2024.

8. Cao D, Wang Y, Duan J, et al. Spectral temporal graph neural network for multivariate time-series forecasting[J]. Advances in neural information processing systems, 2020, 33: 17766-17778.
9. ZHANG L, AGGARWAL C, QI G J. Stock Price Prediction via Discovering Multi-Frequency Trading Patterns[C/OL]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017.
10. REN H, XU B, WANG Y, et al. Time-Series Anomaly Detection Service at Microsoft[C/OL]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019.
11. HOCHREITER S, SCHMIDHUBER J. Long Short-Term Memory[J/OL]. Neural Computation, 1997: 1735-1780.
12. SUTSKEVER I, VINYALS O, LE QuocV. Sequence to Sequence Learning with Neural Networks[J]. 2014.
13. Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18). Association for Computing Machinery, New York, NY, USA, 95–104.
14. CHUNG J Y, GULCEHRE C, CHO K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv: Neural and Evolutionary Computing,arXiv: Neural and Evolutionary Computing, 2014.
15. LV Z, XU J, ZHENG K, et al. LC-RNN: A Deep Learning Model for Traffic Speed Prediction[C/OL]//Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden. 2018.
16. LI Y, YU R, SHAHABI C, et al. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting[J]. ICLR 2018, 2017.
17. Scarselli F, Gori M, Tsoi A C, et al. The graph neural network model[J]. IEEE transactions on neural networks, 2008, 20(1): 61-80.
18. ZHAO L, SONG Y, ZHANG C, et al. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction[J/OL]. IEEE Transactions on Intelligent Transportation Systems, 2020: 3848-3858.
19. WU Z, PAN S, LONG G, et al. Graph WaveNet for Deep Spatial-Temporal Graph Modeling[C/OL]//Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China. 2019.
20. VASWANI A, SHAZEER N, PARMAR N, et al. Attention is All you Need[J]. Neural Information Processing Systems,Neural Information Processing Systems, 2017.
21. ST-Norm: Spatial and Temporal Normalization for Multi-variate Time Series Forecasting[J].
22. CHENG W, SHEN Y, ZHU Y, et al. A Neural Attention Model for Urban Air Quality Inference: Learning the Weights of Monitoring Stations[J/OL]. Proceedings of the AAAI Conference on Artificial Intelligence, 2022.
23. ZHENG C, FAN X, WANG C, et al. GMAN: A Graph Multi-Attention Network for Traffic Prediction[J/OL]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020: 1234-1241.
24. LABLACK M, SHEN Y. Spatio-Temporal Graph Mixformer for Traffic Forecasting[J/OL]. Expert Systems With Applications, 2022.
25. LIU Z, ZHOU J. GRAPH ATTENTION NETWORKS[M/OL]//Synthesis Lectures on Artificial Intelligence and Machine Learning,Introduction to Graph Neural Networks. 2020: 39-41.

26. RGDAN: A random graph diffusion attention network for traffic prediction[J]. Neural Networks, 2024.
27. WU H, XU J, WANG J, et al. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting[J]. Cornell University - arXiv,Cornell University - arXiv, 2021.
28. Ding R, Chen Y, Lan Y T, et al. DRFormer: Multi-Scale Transformer Utilizing Diverse Receptive Fields for Long Time-Series Forecasting[C]//Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. 2024: 446-456.
29. YIN X, WU G, WEI J, et al. Deep Learning on Traffic Prediction: Methods, Analysis, and Future Directions[J/OL]. IEEE Transactions on Intelligent Transportation Systems, 2022: 4927-4943.
30. GROVER A, LESKOVEC J. node2vec: Scalable Feature Learning for Networks[J].
31. MIKOLOV T, CHEN K, CORRADO GregS, et al. Efficient Estimation of Word Representations in Vector Space[J]. International Conference on Learning Representations,International Conference on Learning Representations, 2013.
32. PEROZZI B, SKIENA S. DeepWalk: Online Learning of Social Representations[J].
33. LIU Z, ZHOU J. GRAPH ATTENTION NETWORKS[M/OL]//Synthesis Lectures on Artificial Intelligence and Machine Learning,Introduction to Graph Neural Networks. 2020: 39-41.
34. SONG C, LIN Y, GUO S, et al. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting[J/OL]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020: 914-921.
35. BAI L, YAO L, LI C, et al. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting[J]. NIPS 2020, 2020.
36. CAO D, WANG Y, DUAN J, et al. Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting[J]. Cornell University - arXiv,Cornell University - arXiv, 2021.
37. LIU D, WANG J, SHANG S, et al. MSDR: Multi-Step Dependency Relation Networks for Spatial Temporal Forecasting[J].
38. JIANG R, WANG Z, YONG J, et al. Spatio-Temporal Meta-Graph Learning for Traffic Forecasting[J]. AAAI 2023, 2022.
39. Choi J, Park N. Graph neural rough differential equations for traffic forecasting[J]. ACM Transactions on Intelligent Systems and Technology, 2023, 14(4): 1-27.
40. WU M, LIN Y, JIANG T, et al. MHGNet: Multi-Heterogeneous Graph Neural Network for Traffic Prediction[J].