Adaptive Correlation Graph Neural Ordinary Differential Equation for Traffic Flow Forecasting

Lin Bai, Zheng Huang*, Shuang Wang and Hong Dai

Abstract—**Contemporary advancements in deep learning have spurred widespread adoption of spatio-temporal prediction across various scientific disciplines. Nonetheless, traffic flow prediction, as a quintessential spatio-temporal task, continues to present significant challenges, such as the accurate modeling of complex dependencies and dynamic changes over time and space. To address these issues, this paper introduces the Dual-Branch and Multi-Temporal Resolution Convolutional Network with an Adaptive Graph Neural Ordinary Differential Equation (DM-AGODE) model. This innovative approach integrates an optimized graph neural ordinary differential equation with an adaptive correlation adjacency graph, ensuring precise feature propagation across the network. The model incorporates a Dual-Branch Learning (DBL) mechanism to effectively differentiate between short-term dynamics and long-term trends, while the Multi-Temporal Resolution Convolution (MTRC) method enhances the processing of temporal data across multiple scales, critical for capturing the complex behaviors of traffic flow. Furthermore, to demonstrate the effectiveness of our model, we conducted a comprehensive evaluation of our model using six widely recognized real-world datasets, which highlighted its superior adaptability to complex traffic flow patterns. Compared to the leading baseline model, our approach achieves an improvement in prediction accuracy exceeding 8% and significantly enhances efficiency in processing.**

Keywords—**Traffic flow forecasting, Multi-step forecasting, Neural ODE, Spatio-temporal feature fusion, Training algorithm, Dilated Convolution**

I. Introduction

THE forecasting of spatio-temporal data, especially the task of spatio-temporal traffic forecasting in Intelligent Transport Systems(ITS) [1-3], has garnered significant attention in recent years. Accurate traffic predictions are vital for enhancing urban mobility and efficiency. By leveraging deep learning methods in conjunction with traffic network topology modeling, traffic management centers can accurately predict flow trends for individual road segments. This capability is fundamental for effective road planning, resource allocation, and scheduling within intelligent transportation systems [4].

Spatio-temporal data, which depends on both time and space, also finds applications in various other fields. For instance, similar methodologies are used in forecasting the capacity or load of wind power stations [5], monitoring air quality, and predicting weather conditions [6]. However, despite the advancements, spatio-temporal traffic flow forecasting still faces numerous challenges. The dynamic and complex nature of traffic patterns requires continuous improvement and refinement of predictive models to ensure accuracy and reliability [7]. However, the task of spatio-temporal traffic flow forecasting still encounters numerous challenges at present.

In academic research, it is crucial to consider the spatial correlation of traffic sensors when investigating spatio-temporal traffic forecasting. The correlation between traffic patterns at different nodes in the road network is not always directly related to their physical distance. For example, both the office area and the industrial area shown in Fig. 1 are work zones, and their traffic flow trends display similarities. This indicates a strong spatial association between these seemingly distant areas.

Nevertheless, it is important to recognize that significant differences in the functions of two traffic areas can reduce their similarity, even if they are geographically close. For example, as illustrated by the business district and the office area in Fig. 1, Fig. 1(a) shows the geographical topology of the traffic road network, while Fig. 1(b) presents the semantic association graph structure obtained after calculation. In such cases, relying solely on distance-based adjacency relationships may introduce noise, adversely affecting the model's predictive accuracy. Therefore, one of the key challenges addressed in this paper is developing an appropriate correlation graph design to extract the profound spatial correlations among traffic nodes.

Furthermore, in traffic flow prediction research, the predominant approach is data-driven [8, 9]. This method involves extracting time series features from historical data trends. However, it is crucial to acknowledge the intricate nature of oscillation periodicity in traffic flow, which becomes more pronounced in long-term predictions. Thus, accurately extracting node-specific time trend features is another vital factor contributing to the enhanced accuracy of prediction results in this study.

Addressing these key research issues may significantly improve traffic forecasting methodologies, resulting in more accurate and reliable predictions for intelligent transportation systems. This paper presents a novel spatio-temporal

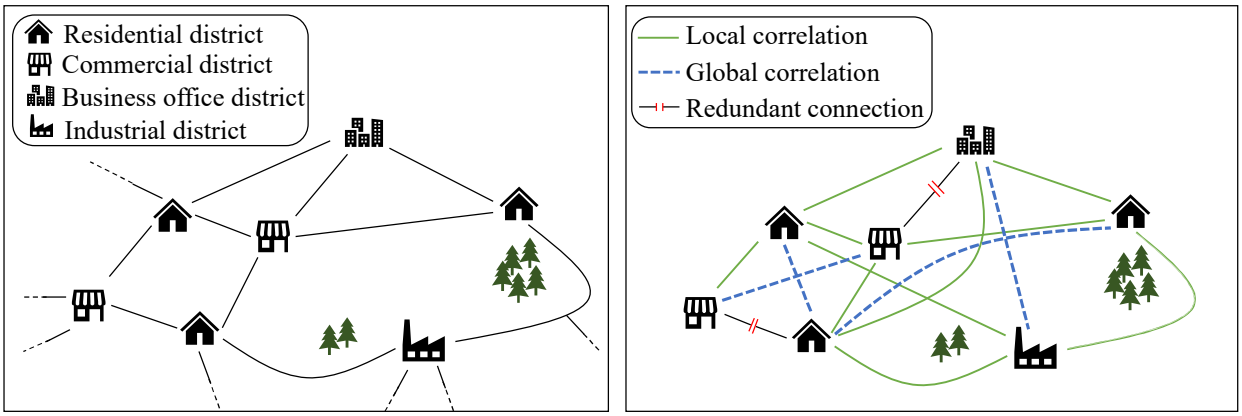
Manuscript received March 16, 2024, revised July 16, 2024. The research work was supported by the Graduate student science and technology innovation project of University of Science and Technology Liaoning (No. LKDYC202221).

Lin Bai is a postgraduate student of University of Science and Technology Liaoning, Anshan, Liaoning, CO 114051, China. (e-mail: 1294330160@qq.com).

Zheng Huang* is a professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, China. (corresponding author to provide phone: +086-151-4126-6160; fax: 0412-5929818; e-mail: huangzheng@ustl.edu.cn).

Shuang Wang is a postgraduate student of University of Science and Technology Liaoning, Anshan, Liaoning, CO 114051, China. (e-mail: 1657669526@qq.com).

Hong Dai is a professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, China. (email: dear_red9@163.com)



(a) Geographical topology structure of traffic road network

(b) Semantic association graph structure of traffic road network

Fig. 1. Real road network connection and semantic connection.

jointprediction framework, specifically designed to optimize the utilization of extracted spatio-temporal features. The main contributions of this paper are as follows:

1) We developed a novel spatio-temporal correlation graph to reflect traffic network node dependencies. This graph captures both global time pattern similarities and local spatial correlations among nodes. Additionally, we introduced an equal interval sampling technique to measure node similarity efficiently, thereby reducing the time required for pre-constructing the graph.
2) We constructed an optimized Adaptive Graph Neural Ordinary Differential Equation (AGODE) to extract association information, providing a detailed derivation. This method addresses the limitations of discrete layers, enabling efficient distribution of node features across the system.
3) We introduced the Dual-Branch Learning and Multi-Temporal Resolution scale joint learning method to capture time-dependent features. This approach uses two parallel modules for time feature extraction, leveraging dilated convolution in Temporal Convolutional Networks (TCN) to enhance efficiency. Each module captures distinct temporal correlations within traffic sequences, offering a comprehensive approach to feature extraction.

II. Related Work

A. Forecasting Methods based on Temporal Feature

In classical traffic flow prediction methods, time series prediction models are widely employed, utilizing the time series characteristics of traffic flow for further predictions. Initially, parametric methods such as the Autoregressive Integrated Moving Average (ARIMA) [10] model were commonly used to fit the changing trend of traffic flow. Over time, researchers have explored various ARIMA variants tailored for traffic flow forecasting [11, 12]. However, due to the nonlinear nature of traffic data, the focus has shifted from parametric to nonparametric time series prediction methods.

Lippi et al. introduced the Support Vector Regression (SVR) supervised learning algorithm [13], integrating it with the concept of the SARIMA model [14]. This hybrid approach leveraged the seasonality of traffic data to obtain more accurate forecasting results. Similarly, Lv et al. employed a stacked Autoencoder (SAE) model to learn general traffic flow features, which led to improved accuracy in predicting future traffic flow [15].

To better capture the nonlinear and stochastic characteristics of traffic flow, Fu et al. adopted two types of Recurrent Neural Network (RNN) models, specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Combining these two modules, they successfully predicted highway traffic flow with superior performance compared to classical parametric methods [16]. This shift towards nonparametric time series prediction techniques has opened new avenues for achieving more accurate and robust traffic flow forecasts.

B. Forecasting Methods based on Spatio-Temporal Features

Non-parametric time series prediction methods excel in capturing random and nonlinear features from time series data. However, in the context of traffic flow prediction, the detection nodes in the traffic network are not entirely independent. As the magnitude of available traffic data increases with the advancement of Internet of Things technology, researchers have recognized the importance of considering both spatial and temporal characteristics of the traffic network to achieve more accurate and convincing prediction results. Consequently, recent work in traffic flow prediction has increasingly focused on the extraction and fusion of spatio-temporal features.

In 2018, Yu et al. proposed the spatio-temporal feature extraction framework STGCN [17]. Their approach involved constructing a spatial graph based on the geographic distance of traffic detectors, employing graph convolution modules to extract spatial features, and using one-dimensional convolutional neural networks for capturing temporal dependencies. This model marked the first application of graph convolution in traffic prediction.
Li et al. represented pairwise spatial correlations between traffic sensors using a directed graph and introduced the Diffusion Convolutional Recurrent Neural Network (DCRNN) to model spatial dependence [18]. Zhou et al. extended the diffusion convolutional neural network to a directed graph and incorporated an attention mechanism for modeling the spatial correlation of sensors [19]. Wu et al. proposed Spatio-Temporal Synchronous Graph Convolutional Network (STSGCN) [20]. STSGCN considered the time step in the

spatial graph, integrating multiple adjacency matrices and self-connection matrices into a large matrix, and concurrently extracted spatial and temporal dependence through the graph convolution module. Building upon this progress, STFGNN introduced the notion of node similarity and further expanded the dimension of the spatio-temporal graph [21].

III. Methodology

A. Problem Definition

Traffic flow prediction is a sequence prediction task, wherein the objective is to forecast future data at time step T' based on past data at time step T. In the context of traffic flow prediction, we are provided with node graph information denoted as G. The sequence data of N input sensors, i.e., the graph signal, is represented by $X \in \mathbb{R}^{N\times H\times d}$, where H represents the time steps in the past, and d denotes the features of the input data.

To accomplish the traffic flow prediction task, we aim to learn a mapping function f with learnable parameters. This function takes the input sequence data X and produces the output sequence data $f(X)=Y\in\mathbb{R}^{N\times F}$ representing the predicted traffic flow at time step T'. The goal is to train the mapping function f to accurately forecast the traffic flow at steps T' into the future:

$$\left[X_n^i, G\right] \xrightarrow{f} Y_n^j \tag{1}$$

Where $0<i\le T$ represents the past time steps, and $T<j\le T'$ represents the future time steps to be predicted. The objective is to effectively model the dependencies between the graph signal and the graph information to achieve accurate and reliable traffic flow predictions at desired future time steps.

B. Adjacency Matrix Construction Method

In the context of traffic space modeling, the spatial distance between detector nodes in the traffic network is a straightforward indicator of spatial association. However, for complex traffic correlations, deriving the adjacency relationship demands a more encompassing similarity calculation method, considering macroscopic factors and broader spatial considerations.

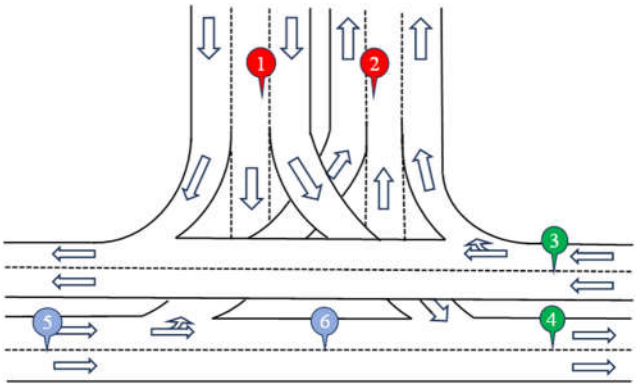


Fig. 2. Pairs of correlated nodes in a complex traffic road network

Consider two complex traffic routes that intersect or share similarities, such as the expressway network in Fig. 2. In this scenario, detector nodes 1 and 2 are on divergent roads with distinct destinations, while nodes 3 and 4 are on opposing routes. Despite differing endpoints, the spatial distances between these detector pairs are minimal. Relying solely on spatial distance for the correlation adjacency matrix could harm model learning due to strong node correlations. We term these links "pseudo-connections."

In practical traffic flow modeling, the presence of pseudo-connections in the adjacency matrix can significantly impact graph learning. Existing distance-based adjacency matrix construction methods are prone to generating numerous pseudo-connections. Nodes 5 and 6 in Fig. 2 are common in the road network, representing detector nodes before and after traffic diversion. Despite their considerable distance, they still display strong flow trend correlation. However, non-adjacent road sections' traffic flow exhibits a time-shift property as distance increases. Therefore, we propose a method fusing Earth Mover's Distance (EMD) [22] and Dynamic Time Warping (DTW) [23] to gauge node correlations, yielding a distinctive adjacency matrix. This method effectively blends global node correlation and trend similarity at comparable time points, enhancing the graph's refined and accurate spatio-temporal representation.

Different from other distance measures such as Euclidean distance and Manhattan distance, EMD evaluates the similarity between a pair of distributions based on the transportation cost required to transform one distribution into the other. For a pair of traffic flow sequences $X=\{X_1,X_2,\ldots,X_{L_1}\}$ and $Y=\{Y_1,Y_2,\ldots,Y_{L_2}\}$, EMD calculates the similarity of these distribution sequences using the following formula:

$$EMD(X,Y)=min_{\tau\in\Gamma(X,Y)}\sum_{i,j}\tau_{i,j}\times M_{i,j} \tag{2}$$

Where M is the distance matrix constructed using Euclidean distance, and $M_{i,j}$ is the distance between X_i and Y_j. $\tau\in\Gamma(X,Y)$ is a transition strategy from X to Y, represented as an $L_1\times L_2$ matrix, where $\tau_{i,j}$ denotes the transition cost of transferring the *i*-th element of X to the *j*-th element of Y. Γ is the set of all feasible transition strategies.

EMD is impervious to real-world node distances, considering distribution shape and topology comprehensively. This makes EMD well-suited for complex distribution series like traffic flow. Yet, it lacks time translation and sequence scaling capabilities within time series. To address these, we combine DTW distance for optimization. The DTW formula is as follows:

$$DTW(X,Y)=\sqrt{D(L_1,L_2)} \tag{3}$$

Given the same distance matrix M as defined in Equation (2), $D(i,j)$ in the above equation can be calculated using the following recursive formula:

$$\begin{aligned} D(i,j) &= M_{i,j} + \\ &\quad min\{D(i-1,j),D(i,j-1),D(i-1,j-1)\} \end{aligned} \tag{4}$$

The recursive formula for $D(i,j)$ is as follows:

$$D(i_0,j_0)=\begin{cases} 0, & i_0=0 \text{ and } j_0=0 \\ \infty, & i_0=0 \text{ and } j_0\neq 0 \\ \infty, & i_0\neq 0 \text{ and } j_0=0? \end{cases} \tag{5}$$

DTW method adapts well to sequence length changes and handles time translation and scaling matches effectively.

This complements EMD distance's correlation calculation on time-shifted sequences. For adjacency graph construction, we use normalized correlation values as weights. Hence, our final similarity measure is:

$$A_{i,j} = \begin{cases} exp\left(R_{DE}\right), & if\ exp\left(R_{DE}\right) \geq \partial \\ 0\ , & otherwise \end{cases} \tag{6}$$

Where R_{DE} is defined by the following equation:

$$R_{DE} = -\frac{\varepsilon(DTW_{i,j}^2) + (1-\varepsilon)\left(EMD_{i,j}^2\right)}{\sigma^2} \tag{7}$$

σ in Equation (6) and ∂ in Equation (7) are threshold parameters that control the sparsity of the matrix, and ε is the weight coefficient. These parameters play a crucial role in determining the strength and density of connections in the adjacency graph, allowing us to fine-tune the representation of the correlation between traffic nodes.

Since the above two methods are two sequence similarity evaluation methods with complex calculation, we have also proposed a dense sequence equal interval sampling method to efficiently preprocess the data and reduce the time cost. In this method, we set the interval step number as φ, and extract subsequences from the original data sequence $X^i = \{X_1^i, X_2^i, \ldots, X_L^i\}$ to create a new sequence $X^{i'} = \{X_{\varphi}^i, X_{2\times\varphi}^i, \ldots, X_{\lfloor L/\varphi \rfloor \times \varphi}^i\}$,where X^i represents the observed data sequences of node i , and $\lfloor \cdot \rfloor$ represents the integer operation. The resulting subsequence, $X^{i'}$, has a length of L' which satisfies the condition $L' \leq (1/\varphi) \times L$.

This sampling method retains most of the sequence characteristics, has minimal impact on the measurement of sequence similarity, and significantly reduces the time required for constructing the adjacency matrix. This is especially beneficial for datasets with dense intervals and large amounts of data.

C. Adaptive Graph Neural ODE Network

The graph neural network aggregates the features of neighbor nodes in an iterative manner [24], diffusing the feature information of each node on the graph to nearby nodes through discrete aggregation layers, thereby updating the feature information of each node. The common iterative formula of a graph neural network based on spectral graph theory [25] is as follows:

$$S^{(l+1)} = \sigma\left(\tilde{A} S^{(l)} W\right) \tag{8}$$

Where $S^{(l+1)}$ is the output state of the *l*-th layer, σ is the activation function, $S^{(l)}$ is the feature matrix obtained from the *l*-th layer, $W \in \mathbb{R}^{d_l \times d_{l+1}}$ is the weight matrix of the *l*-th layer, and $\tilde{A} \in \mathbb{R}^{N \times N}$ is the pre-constructed regularization adjacency matrix after normalization. This form of graph convolution is simple yet effective. However, there are still some problems with this model.

Firstly, traditional graph convolution methods based on spectral graph theory are discrete. They involve computing graph Laplacian eigenvectors and eigenvalues for spectral domain filtering, this can lead to over-smoothing when using multi-layer convolutions, limiting the depth to around 3 layers at most. Secondly, these models update node features discretely in layers, restricting graph propagation depth by layer count. This hampers hierarchical feature update structures, making it hard to determine the optimal number of layers. Lastly, real-world node correlation can change with time, yet traditional graph neural networks assume a static graph structure. This hinders effective learning of spatio-temporal sequences.

Considering these limitations and the intricate nature of real-world traffic flow, this paper aims to establish an efficient graph ODE to enhance traffic spatio-temporal learning. Building upon previous research [9, 26], we introduce a straightforward discrete dynamic as follows:

$$S^{(n+1)} = S^{(n)} \times \tilde{A} \times_1 W_N \times_2 W_T \times_3 W_C + S^{(0)} \tag{9}$$

Where $S^{(n)} \in \mathbb{R}^{N \times T \times C}$ represents the output state of the *n*-th layer, $\times_i$ denotes element-wise multiplication between tensors of different dimensions and the corresponding dimension in the matrix, ensuring smooth calculation during subsequent ODE derivations. $S^{(0)}$ corresponds to the residual term of the initial distribution, which is a crucial component to prevent over-smoothing issues [27]. $W_N = P_1 q_1 P_1^T$ is the spatial dimension weight, $W_T = P_2 q_2 P_2^T$ is the time dimension weight, and $W_C = P_3 q_3 P_3^T$ is the feature parameter matrix. These three weight matrices are constructed using learnable orthogonal matrices P and vectors q , guaranteeing their diagonalizability. The matrix tuple $\{W_N, W_T, W_C\}$ transforms and adjusts the feature matrix in the simulated dynamic system, allowing the ODE solver to better fit the data. This combination with the neural ODE forms the graph neural ODE, facilitating the continuous diffusion of high-dimensional features throughout the entire graph structure. By using inductive method, we can derive the explicit formula of this dynamic system in the following form:

$$S^{(n)} = \sum_{i=0}^{n} \left(S^{(0)} \times \tilde{A}^i \times_1 W_N^i \times_2 W_T^i \times_3 W_C^i \right) \tag{10}$$

Equation (10) with the discrete propagation layer n can be transformed into a continuous variable t to construct graph neural ODEs. By considering Equation (10) as a Riemann sum of integrals from time $t = 0$ to $t = n$, and setting $\Delta t = (t+1)/(n+1)$,we can derive the following equation when $t = n$:

$$\begin{aligned} S^{(n)} = \sum_{i=1}^{n} (& S^{(0)} \times \tilde{A}^{(i-1)\times\Delta t} \\ & \times_1 W_N^{(i-1)\times\Delta t} \times_2 W_T^{(i-1)\times\Delta t} \times_3 W_C^{(i-1)\times\Delta t}) \times \Delta t \end{aligned} \tag{11}$$

As n tends to infinity, we can further obtain the following equation:

$$S^{(n)} = \int_0^{t+1} S^{(0)} \times \tilde{A}^{\mu} \times_1 W_N^{\mu} \times_2 W_T^{\mu} \times_3 W_C^{\mu} d_{\mu} \tag{12}$$

According to the derivative rule, we can easily obtain the second derivative of the above formula as follows:

$$\begin{aligned} \frac{d^2 S^{(t)}}{dt^2} = & \frac{dS^{(t)}}{dt} \times \ln\tilde{A} + \frac{dS^{(t)}}{dt} \times_1 \ln W_N \\ & + \frac{dS^{(t)}}{dt} \times_2 \ln W_T + \frac{dS^{(t)}}{dt} \times_3 \ln W_C \end{aligned} \tag{13}$$

Integrating t from the Equation (13) reveals that the discrete dynamics in Equation (11) is a discretization of the following ODE:

$$\begin{aligned} \frac{dS^{(t)}}{dt} = & S^{(t)} \times ln\tilde{A} + S^{(t)} \times_1 ln W_N \\ & + S^{(t)} \times_2 ln W_T + S^{(t)} \times_3 ln W_C + S^{(0)} \end{aligned} \tag{14}$$

The logarithm of the matrix in Equation (14) is complex in the model calculation, and it is generally approximated by

the first-order Taylor expansion method in the neural ODE. The approximation is performed according to the following equation:

$$\ln(X+I) \approx X \overset{X+I=A}{\Leftrightarrow} \ln A \approx A - I \tag{15}$$

where I is the identity matrix of the same size as A. By using the above approximation, we can rewrite Equation (14) as follows:

$$\frac{dS^{(t)}}{dt} = (\tilde{A} - I) \times S^{(t)} + S^{(t)} \times_1 (W_N - I) + S^{(t)} \times_2 (W_T - I) + S^{(t)} \times_3 (W_C - I) + S^{(0)} \tag{16}$$

Equation (14) approximates the differential equation in the ODE solver. Traffic data dynamics are typically complex and varied, often displaying non-smooth behavior over time. This implies changing node associations over time. However, traditional spatio-temporal graph convolution models rely on time-invariant adjacency matrices. Their fixed graph structure can depict global connections for intricate traffic flow tasks but falls short in capturing local temporal correlation changes. To overcome this limitation, we introduced a dynamic optimization graph integrated with the graph ODE. Specifically, consider:

$$\tilde{A}_w = \frac{1}{2}\left(A_w + A_w^T\right) \odot \tilde{A} \tag{17}$$

Where A_w is the adaptive parameter matrix, which can be learned during the training process. This allows the graph adjacency matrix to be continuously fine-tuned with the gradient through A_w, leading to the gradual adjustment of the original system to a more accurate dynamic system. As a result, a more representative adjacency matrix is obtained. It is important to note that Equation (17) ensures that the optimized adjacency matrix $\tilde{A}_w$ of the traffic graph remains a real symmetric matrix, as the nodes in the traffic network are usually associated with each other.

Finally, the optimized form of Equation (16) is obtained as follows:

$$\frac{dS^{(t)}}{dt} = (\tilde{A}_w - I) \times S^{(t)} + S^{(t)} \times_1 (W_N - I) + S^{(t)} \times_2 (W_T - I) + S^{(t)} \times_3 (W_C - I) + S^{(0)} \tag{18}$$

Equation (18) represents the final form of our Adaptive Graph neural Ordinary Differential Equation (AGODE). We treat this ODE as a dynamic system to be solved, and the original data is only used through feature embedding as the initial state of the system $S_0^{'}$. We then adjust the time point tensor to be solved, $t = [t_0, t_1, \ldots, t_n]$, where $t_i \in [0, \infty)$ is used as input to initialize the solver. This represents any point in time at which the ODE is expected to be solved in continuous time. Finally, various numerical methods can be applied to solve the ODE at that point in time. In comparison, the explicit method is faster and has lower computational complexity. By solving the ODE, we can obtain the numerical solution of the dynamic system as follows:

$$S(0) = S_0^{'},\ S(t) = ODE\left(\frac{dS^{(t)}}{dt}, S(0), t\right) \tag{19}$$

Where $dS^{(t)}/dt$ is consistent with the expression in Equation (18), and $ODE(\cdot)$ represents the solution process of the differential equation. Finally, the numerical solution $S(t)$ at all time steps in t will be used for further analysis in the downstream module.

D. Dual-Branch and Multi-Temporal Resolution Convolutional Network

The Adaptive Graph Neural ODE offers a strong spatial representation capacity. However, traffic flow prediction inherently involves time series forecasting. Given the diverse factors affecting travel patterns, extracting temporal features becomes essential for spatio-temporal sequence prediction.

To address this, we propose the Dual-Branch and Multi-Temporal Resolution Convolutional Network (DB-MTRC), which combines two distinct techniques: Multi-Temporal Resolution Convolutional (MTRC) and Dual-Branch Sequence Learning (DBL). Specifically, distant historical data imparts long-range momentum to forecast outcomes, uncovering broader traffic flow patterns. Within this range, the future traffic flow exhibits variability due to the higher randomness stemming from the time interval between long-term-historical data and the data to be predicted, contributing to the model's improved accuracy in capturing overall historical patterns.

Conversely, recent-historical data, closer to the prediction time, provides real-time information, offering micro-fluctuation trends to predictive data. These points reveal more short-term fluctuations and trend characteristics, granting the model more control in its feature learning. Utilizing near-historical data brings in more current information, bolstering prediction stability. This stability fortifies the model's robustness in forecasting long-term traffic flows.

Considering the issue of overfitting and the presence of odd-numbered historical data, we set a shared intermediate time step section for the two sub-sequences. Specifically, if the input sequence has a time step of n, the shared time steps are denoted as c. To avoid interruptions between historical and predictive data, we prefer both n and c to be even numbers. However, if the historical time step is not even, we can pad the left end of the sequence with zeros. The complete historical time sequence is then divided into two parts, each having a length of $(n-c)/2$, as shown in the following equation:

$$\begin{aligned} S_1 &= \left[data_1, data_2, \ldots, data_{\frac{n}{2}}, \ldots, data_{\frac{n+c}{2}} \right] \\ S_2 &= \left[data_{\frac{n-c}{2}+1}, \ldots, data_{\frac{n}{2}+1}, \ldots, data_{n-1}, data_n \right] \end{aligned} \tag{20}$$

Moreover, for an improved ability to generalize the model to new data and enhance its directional understanding of the input sequence, we conducted one-sided padding operations on the two sub-sequences. The padding length, represented as p, relies on the length n of the original input sequence S and the sub-sequences' lengths:

$$p = \frac{n-c}{2} \tag{21}$$

MTRC focuses on capturing time series features at different temporal resolutions, enabling the model to consider various time-step dependencies and fluctuations, enhancing its understanding of temporal patterns. Conversely, DBL aims to comprehend the segmentation dependence between prediction results and historical data, providing a comprehensive viewof

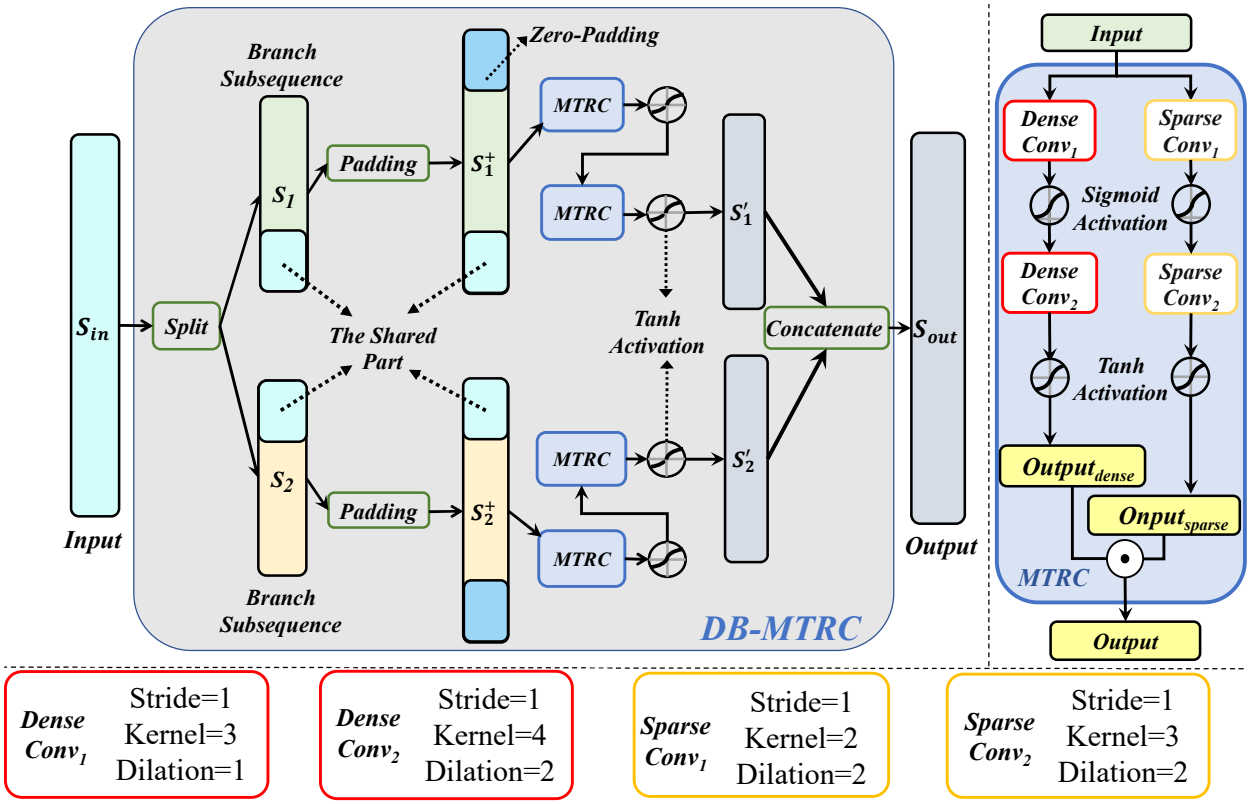


Fig. 3. Details of the DB-MTRC module

temporal dynamics in traffic flow. The integration of these approaches in DB-MTRC creates a potent framework that enhances the model's perception of time-domain fluctuations.

From a broad viewpoint, traffic flow data showcases abundant trends and periodic patterns. Yet, at a finer level, traffic flow changes aren't exclusively driven by periodic patterns. Hence, we factor in time features at various scales within the sequence and create sparse resolution and dense resolution blocks for MTRC, each tied to modules with distinct receptive field resolutions. The sparse module captures broader temporal context features, focusing on global patterns and overall trends, and the dense module captures finer-grained temporal context features, capturing local and specific patterns in the time series data.

To manage parameter costs, we employ dilated convolutions in the MTRC module. Finally, we combine predictions from the two branches at various time steps for the ultimate output. The overall architecture of the DB-MTRC module is depicted in Fig. 3.

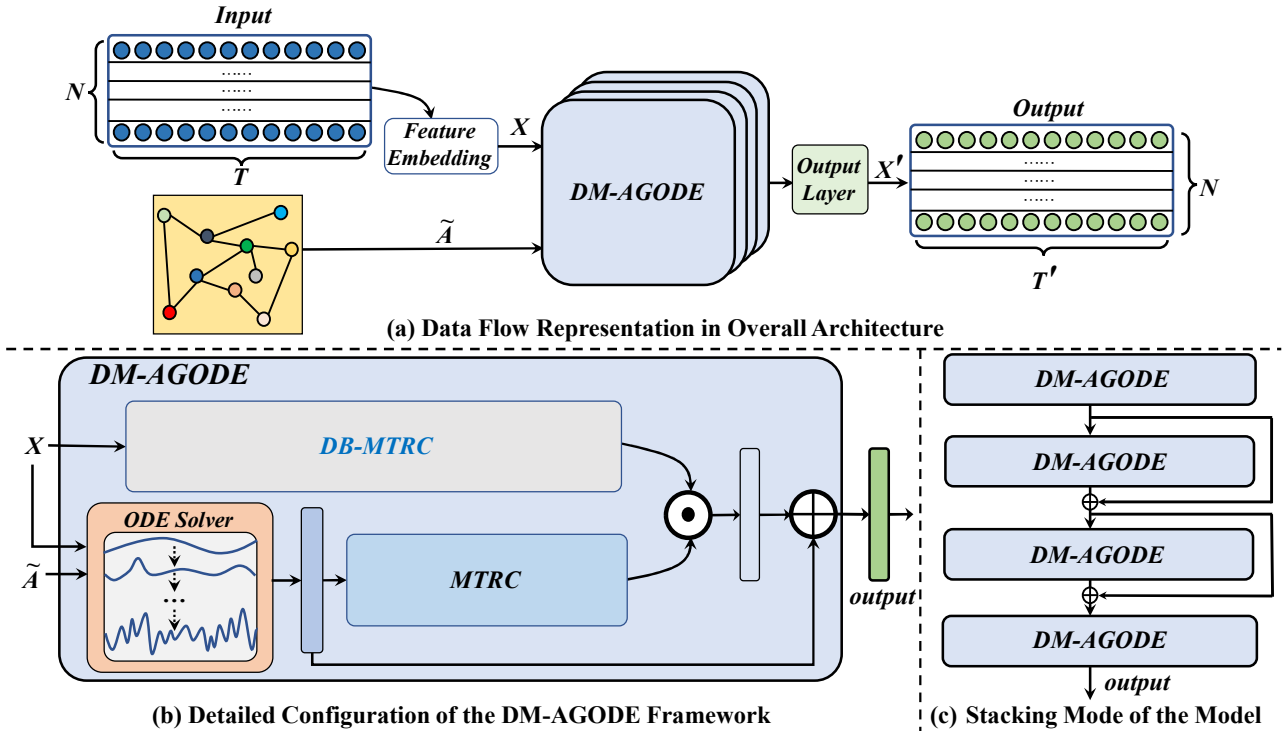


Fig. 4. Framework Details of DM-AGODE

E. DM-AGODE

Given the complexity of real-world transportation systems, it's vital to shield the original historical data from the intricacies of additional convolutional layers during modeling. To harness AGODE's capacity for capturing underlying patterns and DB-MTRC's perceptual aptitude for traffic flow patterns, we introduced a new parallel spatio-temporal feature synchronous extraction framework named DM-AGODE. This framework employs exclusively feature-embedded raw data as input for both DB-MTRC and ODE-solver. The final model is illustrated in Fig. 4, where Fig. 4(a) represents the input and output representation of the whole framework, graph structure $\tilde{A}$ is the adjacency matrix after normalization, Fig. 4(b) illustrates the fusion mode of internal modules of DM-AGODE, and Fig. 4(c) showcases the stacking mode of DM-AGODE, integrates residual connections in Fig. 4(b) and (c) to prevent loss of low-level features when capturing spatio-temporal dependencies. As shown in Fig. 4, the predicted sequence from ODE-solver is also input to the MTRC module, running in parallel with DB-MTRC to boost global temporal features.

IV. Experiments

A. Datasets

We validated the performance of the DM-AGODE model using six widely recognized public datasets, as detailed in Table I. These datasets include the popular PEMS series and the METR-LA dataset, both of which are commonly used in baseline models. The PeMS series datasets, collected and curated by the California Performance Measurement System (PeMS) [28]. They are divided into PeMS03, PeMS04, PeMS07, PeMS08, and PeMS-BAY, based on their collection time and region. All datasets offer traffic flow data with an initial time resolution of 5-minute intervals.

TABLE I
Datasets Details

Datasets	Data Source Region	Number of Sensors	Total Time Steps
PeMS03	Northern California	358	26208
PeMS04	San Francisco Bay Area	307	16992
PeMS07	Los Angeles	883	28224
PeMS08	Santa Clara County region	170	17856
PeMS-BAY	Los Angeles Bay Area	325	52116
METR-LA	Los Angeles	207	34272

Indeed, the PeMS series datasets include genuine distances between nodes, which are essential for creating accurate traffic adjacency matrices. In contrast, the METR-LA dataset employs an adjacency matrix formed through a thresholder Gaussian kernel, as provided by the authors of DCRNN. Although our model does not inherently depend on these matrices, we integrated them into our reproduction of baseline model testing experiments to ensure a consistent comparison.

B. Baseline Models for Comparison

We compare DM-AGODE with the following baseline models:

1) FC-LSTM [21]: Long Short-Term Memory Network, which is the classical contrast model constructed in the paper of STFGNN, It is a recurrent neural network with fully connected LSTM hidden units.
2) STGCN[17]: Spatial-Temporal Graph Convolutional Network, which first applied graph convolution to the traffic prediction problem, and the structure of each module consisted of two gated sequence convolution layers and a spatial graph convolution module in the middle. This temporal-spatial-temporal structure was later called "Sandwich structure".
3) DCRNN [18]: Deep Convolutional Recurrent Neural Network, which combines traffic flow with a diffusion process to model the spatial dependence, which is governed by a random walk strategy.
4) ASTGCN(r) [29]: Attention-based Spatial-Temporal Graph Convolutional Network with Residuals, which utilizes graph convolutional networks with attention mechanisms and residual connections to capture spatial and temporal dependencies.
5) GraphWaveNet [30]: Graph WaveNet, which designs a novel adaptive dependency matrix and learns the spatial dependencies between traffic nodes through node embedding, and adopts the methods of dilated convolution and causal convolution to capture the dependencies on the time axis.
6) STSGCN [20]: Spatial-Temporal Synchronous Graph Convolutional Networks, an extension of the STGCN model, introduces multiple modules with different time periods to capture local spatio-temporal graph heterogeneity. It also employs a spatio-temporal synchronization modeling mechanism to capture complex local spatio-temporal correlations.
7) STFGNN [21]: spatial-temporal Fusion Graph Neural Networks, which is an extension of the STSGCN model that proposes a temporal graph based on the similarity between time series, and integrates it with the original distance matrix to form a spatio-temporal fusion graph to capture the hidden spatio-temporal correlation.
8) STGODE [26]: Spatial-Temporal Graph Ordinary Differential Equation Networks, which still uses the semantic connection and the original distance connection to model the spatial dependence of traffic flow, and proposes a new optimized continuous representation of GCN to increase the depth of GCN to capture deeper spatio-temporal dependence.

C. Experiment Settings

For a fair comparison in our experiments, we divided all benchmark datasets into training, validation, and test sets in a 6:2:2 ratio. We used the Adam optimizer with an initial learning rate of 0.001 and applied a learning rate decay strategy, reducing the rate by 0.2 at 20 and 40 epochs. The batch size was set to 32, and training lasted for 50 epochs, utilizing the Euler method for ODE solving within the ODE-Solver. We employed three evaluation metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The loss function is Huber Loss [31]. All experiments were conducted on a Linux server with the following specifications: CPU - Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz, GPU - NVIDIA RTX 3090 (24GB).

TABLE II
PERFORMANCE COMPARISON OF DM-AGODE AND BASELINE MODELS ON PEMS DATASETS

Data	Metric \ Models	DCRNN	STGCN	ASTGCN(r)	GraphWaveNet	STSGCN	STFGNN	STGODE	DM-AGODE
PeMS03	MAE	18.18	17.49	17.69	19.85	17.48	16.91	16.50	**15.14**
	MAPE(%)	18.91	17.15	19.40	19.31	16.78	16.42	16.69	**14.19**
	RMSE	30.31	30.12	29.66	32.94	29.21	28.37	27.84	**25.09**
PeMS04	MAE	24.70	22.70	22.93	25.45	21.19	20.45	20.84	**19.13**
	MAPE(%)	17.12	14.59	16.56	17.29	13.90	16.74	13.77	**13.26**
	RMSE	38.12	35.55	35.22	39.70	33.65	32.49	32.82	**31.33**
PeMS07	MAE	25.30	25.38	28.05	26.85	24.26	23.33	22.59	**21.04**
	MAPE(%)	11.66	11.08	13.92	12.12	10.21	9.15	10.14	**8.91**
	RMSE	38.58	38.78	42.57	42.78	39.03	36.50	37.54	**34.14**
PeMS08	MAE	17.86	18.02	18.61	19.13	17.13	16.89	16.81	**15.34**
	MAPE(%)	11.45	11.40	13.08	12.68	10.96	10.53	10.01	**9.31**
	RMSE	27.83	27.83	28.16	31.05	26.80	26.20	25.97	**24.20**

In the MTRC block, the hidden layer dimensions were configured at 8 and 16, while the stacked residual layers included four DM-AGODE modules. The threshold values for the spatial adjacency matrix were set as follows: $\sigma = 10$ and $\partial = 0.1$.

D.Experiment Results and Analysis

Table II provides a comprehensive comparison of predictive performance between DM-AGODE and baseline models. Bold numbers highlight predictive results with the lowest errors, while underlined numbers indicate second-best results. Baseline models considering spatio-temporal correlations, like STGCN, STSGCN, and STFGNN, outperform models like FC-LSTM, which only account for temporal dependencies. The utilization of Neural ODE enables models to overcome the limitations introduced by discrete layers, making STGODE's performance relatively more favorable.

However, it's important to note that using DTW distance in STGODE mainly emphasizes local alignment. As a result, this approach might not fully capture the broader global pattern similarity in sequences. Moreover, within the model structure, STGODE employs a spatial distance matrix with pseudo connections and a fixed graph layout. This integration poses challenges in fully harnessing the potent fitting capabilities of ODEs due to the complex interaction between the spatial distance matrix and the static graph. DM-AGODE showcases a significant improvement in predictive Mean Absolute Error (MAE) across four varied datasets, achieving enhancements of 8.2%, 6.6%, 6.8%, and 8.7% compared to top-performing baseline models. This consistent superiority highlights DM-AGODE's notable advantage across all evaluation metrics.

Moreover, to confirm the model's stepwise predictive ability, we conducted comparative experiments on stepwise prediction using the PEMS-BAY and METR-LA datasets. The results in Table III reveal an interesting trend when compared to baseline models, models with better short-term predictive outcomes tend to experience a more pronounced decline in accuracy for long-term predictions. Conversely, models with weaker predictive performance exhibit relatively stable predictive errors. Their performance tends to produce smoother and simpler predictive outcomes, lacking the detail needed to accurately represent the fluctuations and variations in traffic flow. In Table III, while the performance enhancement of DM-AGODE in short-term prediction might not be conspicuous, it effectively circumvents the issue of rapid accuracy decline in mid-term and long-term predictions.

Consequently, it achieves superior predictive performance, thereby demonstrating that the branch network within our model effectively strikes a balance between long-term and short-term predictions. This advantage is particularly pronounced in the context of the METR-LA dataset, renowned for its high complexity.

E. Ablation Experiments

To validate the effectiveness of each module and the necessity of the structural design, we conducted a series of ablation experiments on the PEMS04 and PEMS08 datasets. Specifically, we designed seven variants of DM-AGODE, as follows:

1) DM-GNN: We replaced the AGODE module with a generic Graph Neural Network (GNN) to assess the efficacy of the AGODE design.
2) DM-GODE: We used the original static graph $\tilde{A}$ instead of the dynamically adjusted graph $\tilde{A}_w$, obtained through the parameter matrix A_w, to contrast the performance difference between using static and dynamic graphs.
3) M-AGODE: We solely employed the original sequence for temporal feature extraction to assess the effectiveness of extracting subsequence order features using DBL.
4) D-AGODE: Similarly, we replaced the MTRC module with the original TCN to validate the efficacy of MTRC in capturing multi-time-scale features.
5) DM-AGODE-no-EMD: We replaced the EMD distance with the original spatial distance to validate the effectiveness of EMD distance in representing overall distribution similarity.

TABLE III
STEPWISE PERFORMANCE COMPARISON OF DM-AGODE AND 6 BASELINE MODELS ON PEMS-BAY AND METR-LA DATASETS

Data	Metric / Models	15 min/3 time steps			30 min/6 time steps			60 min/12 time steps		
		MAE	**MAPE (%)**	**RMSE**	**MAE**	**MAPE (%)**	**RMSE**	**MAE**	**MAPE (%)**	**RMSE**
PeMS-BAY	FC-LSTM	2.05	4.80	4.19	2.20	5.20	4.55	2.37	5.70	4.96
	DCRNN	1.38	2.90	2.95	1.74	3.90	3.97	2.07	4.90	4.74
	STGCN	1.36	2.90	2.96	1.81	4.17	4.27	2.49	5.79	5.69
	ASTGCN(r)	1.43	3.25	3.05	1.79	4.40	4.06	2.10	5.30	4.77
	Graph WaveNet	1.30	2.73	2.74	1.63	3.67	3.70	1.95	4.63	4.52
	STGODE	**1.24**	**2.63**	**2.62**	1.55	3.67	**3.64**	1.86	4.34	4.35
	DM-AGODE	1.29	2.65	2.63	**1.53**	**3.52**	3.56	**1.73**	**4.05**	**4.11**
METR-LA	FC-LSTM	3.44	9.60	6.30	3.77	10.90	7.23	4.37	13.20	8.69
	DCRNN	2.77	7.30	5.38	3.15	8.80	6.45	3.60	10.50	7.60
	STGCN	2.88	7.62	5.74	3.47	9.57	7.24	4.59	12.70	9.40
	ASTGCN(r)	3.01	7.11	5.22	3.36	8.52	6.19	3.94	10.10	7.87
	Graph WaveNet	2.69	6.90	5.15	3.07	8.37	6.22	3.53	10.01	7.37
	STGODE	2.52	6.48	**4.85**	2.96	7.98	5.92	3.40	9.62	7.40
	DM-AGODE	**2.48**	**6.43**	4.87	**2.83**	**7.60**	**5.72**	**3.18**	**9.17**	**7.09**

6) DM-AGODE-Sandwich: We placed the GODE module after the model, allowing it to receive data features fitted by TCN, to determine the necessity of directly processing raw data with the GODE module.
7) DM-AGODE-Full-Seq: In this model, we utilized the entire sequence when computing the adjacency matrix, without considering time costs due to equidistant sampling. We solely compared evaluation metrics to assess the impact of equidistant sampling on the results.

The ablation experiment results are presented in Table IV. Models using static graphs or relying solely on DTW for adjacency matrix construction show more noticeable performance drops. This underscores the significance of dynamic graph fitting optimization and the global association extraction using EMD. In the sub-models without the temporal feature module, there are impacts on experimental results. Specifically, when the dual-branch learning mechanism is removed, RMSE errors in predictions are affected more than MAE errors. This suggests that the MTRC module alone captures general trends and patterns. However, for specific cases or notable fluctuations, the dual-branch learning mechanism is essential to capture precise sequence features.

When considering sequence similarity computation, utilizing the entire sequence data does contribute to relatively more accurate predictions. However, the overall improvement in accuracy is not significantly substantial. Interestingly, for the PEMS08 dataset, we noticed that constructing graphs with the entire sequence data resulted in slightly worse MAPE and RMSE prediction errors. This phenomenon arises due to the equidistant sampling method partially mitigating the impact of anomalous fluctuating values in cases where the data is relatively smooth. As a result, we choose to stick with the equidistant sampling approach for graph construction in the final model. This decision significantly reduces the time needed for graph pre-construction.

TABLE IV
ABLATION EXPERIMENTS OF DM-AGODE

Datasets	The original model and Variants	MAE	MAPE (%)	RMSE
PEMS04	DM-GNN	19.60	13.55	32.04
	DM-GODE	19.45	13.42	31.71
	M-AGODE	19.36	13.42	31.63
	D-AGODE	19.35	13.38	31.44
	DM-AGODE-no-EMD	19.23	13.34	31.51
	DM-AGODE-Sandwich	19.71	13.62	32.07
	DM-AGODE-Full-Seq	**19.10**	**13.26**	**31.28**
	DM-AGODE	19.13	**13.26**	31.33
PEMS08	DM-GNN	16.06	25.12	9.982
	DM-GODE	15.57	24.4	9.45
	M-AGODE	15.61	24.74	9.74
	D-AGODE	15.64	24.68	9.52
	DM-AGODE-no-EMD	15.59	24.87	9.81
	DM-AGODE-Sandwich	15.851	25.02	9.84
	DM-AGODE-Full-Seq	**15.32**	24.23	9.35
	DM-AGODE	15.34	**24.19**	**9.30**

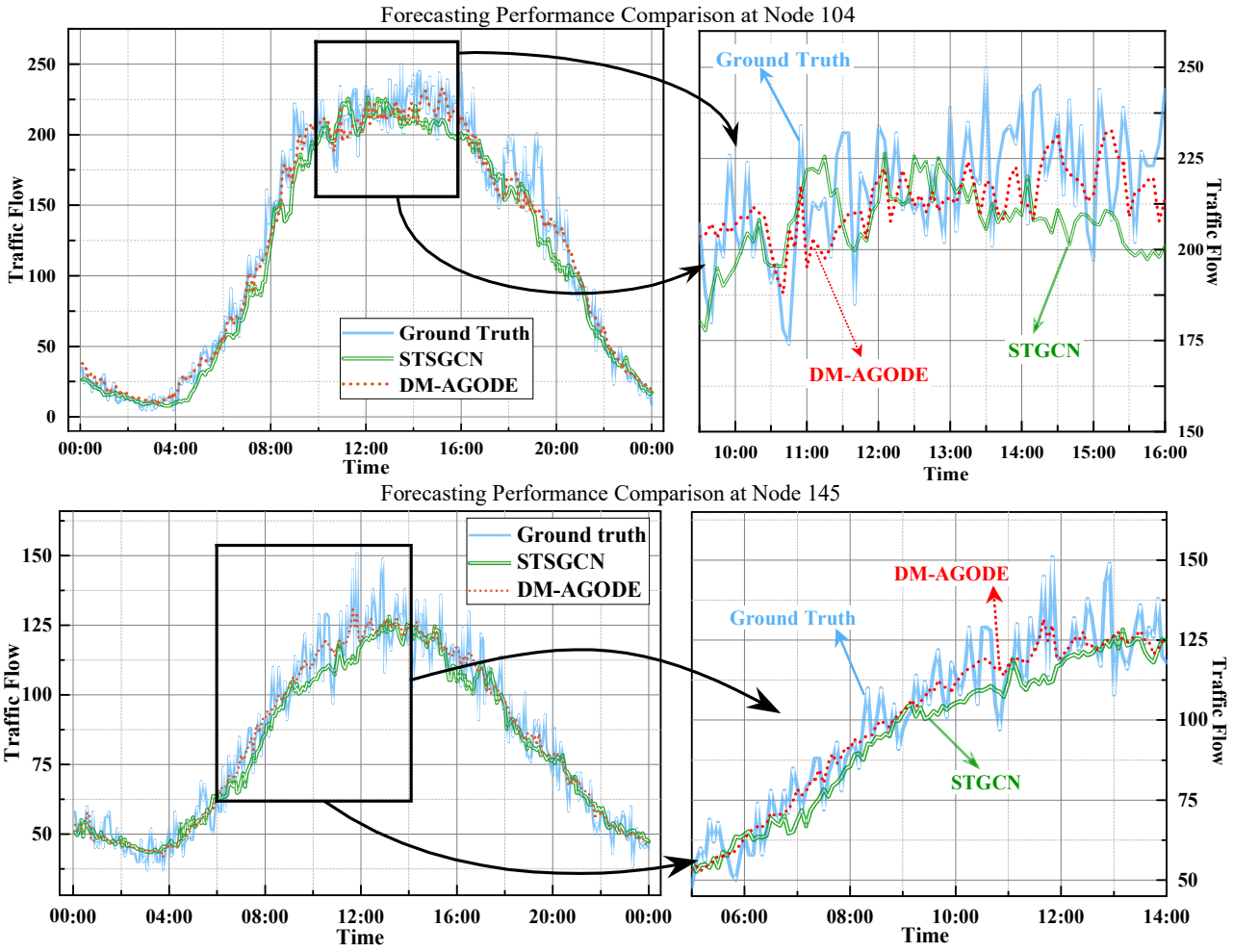


Fig. 5. Comparison of our model with STSGCN prediction case

F. Case Study

To provide a more intuitive demonstration of the model's effectiveness, we selected two traffic nodes from the testing dataset of PEMS04 with substantial differences in their daily average flow and volatility levels. A comparative analysis with the STSGCN model was performed to assess the model's performance. As illustrated in Fig. 5, in most cases, DM-AGODE exhibits higher predictive accuracy compared to the STSGCN model.

Excitingly, our model impressively captures the start and end of flow variations in traffic curves with noticeable, ongoing oscillations. This sharp awareness arises from our subsequence branch learning approach, which swiftly integrates deep historical data and emerging trends into its overall forecasts—especially beneficial in highly fluctuating environments. Additionally, the AGODE module blends global and local spatio-temporal relationships through an optimized dynamic adjacency matrix, enhancing our understanding of complex traffic patterns with the help of the ODE-solver's strategic framework.

In more stable situations, DM-AGODE continues to perform well, handling smoother data effectively while still detecting oscillatory patterns. This durability is thanks to the DB-MTRC module's robust ability to utilize multi-scale temporal features, ensuring the model remains adaptive and accurate across different traffic conditions.

G.Hyperparameters Analysis

In the model presented in this paper, the design of the dynamic graph enhances the model's resistance to correlation noise and allows it to adaptively adjust the correlation degree. The construction of the initial adjacency matrix is crucial, as this matrix serves as the starting point for the evolution of the dynamic system in the dynamic graph neural ODE, directly impacting the graph representation learning ability. As such, two hyperparameters warrant further analysis: the interval of sampling φ and the weight coefficient ε. These hyperparameters significantly influence the adjacency matrix's ability to accurately represent the associations among traffic nodes.

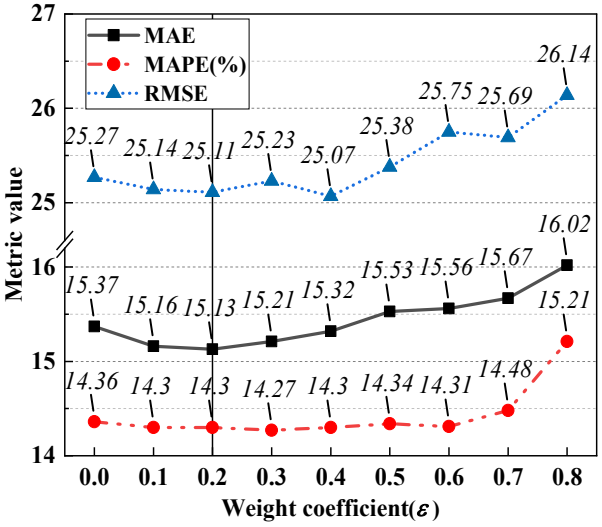


Fig. 6. Hyperparameter performance of Weight coefficient

The performance experiments for the hyperparameters interval of sampling and weight coefficient are shown in Fig.

6 and Fig. 7, respectively. We ultimately determined the optimal hyperparameter values to be ε=0.2 and φ=3, as indicated by the solid black vertical lines in the figures. Notably, while the model's accuracy is not the highest when φ=3, considering the significant increase in computation time for φ=2 in practical implementation, a comprehensive evaluation still suggests that an interval step size of 3 is optimal for the even interval sampling method.

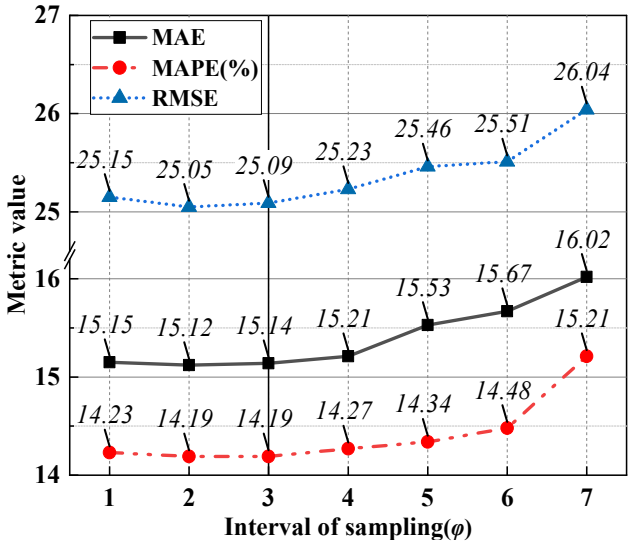


Fig. 7. Hyperparameter performance of interval of sampling

To ensure the fairness of our comparative experiments, we consistently applied the same hyperparameters across all step size prediction experiments. However, it is important to highlight that slightly reducing the ODE solver step size, denoted as t_{end}, can further decrease prediction errors for the 3-step and 6-step forecasts. This improvement occurs because the neural ODE in this study is a dynamic system parameterized by an adaptive graph neural network, which benefits from smaller integration steps when making short-term predictions.

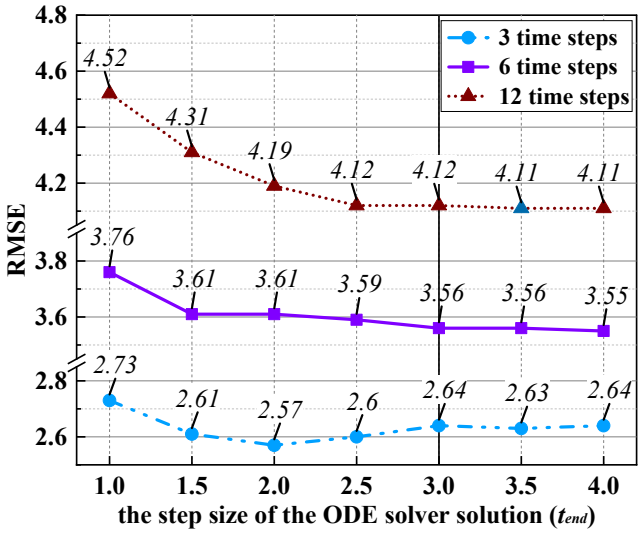


Fig. 8. Hyperparameter performance of step size of the ODE solver solution

Detailed experiments, shown in Fig. 8, analyzed the impact of the solver step size on short-term prediction errors using RMSE (Root Mean Square Error) as the metric. When the solver step size is smaller, the model focuses more on short-term accuracy. The finer granularity allows for more precise adjustments at each step, reducing short-term prediction errors. Conversely, with a larger step size, the model aims to minimize long-term errors. This approach can improve accuracy over extended horizons but may increase short-term errors due to less frequent updates.

The parameter performance in Fig. 8 illustrates the sensitivity of DM-AGODE to the ODE solution step size. In this paper, we place a greater emphasis on challenging long-term predictions, which led us to ultimately select a solution step size of 3.5 and uniformly apply it across all step size prediction experiments. The stability of the model error for t_{end}>3 further demonstrates the robustness of applying AGODE to address the over-smoothing problem.

H. Time Efficiency

In addition to focusing on accuracy, we also considered the efficiency of the model, which benefits from two aspects: Firstly, in terms of spatial feature extraction, our model uses only the adjacency matrix based on Dynamic Time Warping (DTW) and Earth Mover's Distance (EMD) as input to the AGODE module. This refined dimensionality reduction mitigates the time costs associated with handling large input graphs while still effectively capturing both local and global correlations of traffic nodes.

Secondly, the proposed temporal feature extraction module, DB-MTRC, also offers significant efficiency advantages. By leveraging dilated convolutions, the branches within the MTRC module extract features at varying scales, allowing for the use of lower time complexity element-wise products for complementary fusion. Furthermore, the parallel fusion structure of these two novel modules significantly reduces the time costs associated with the previous serial sequential structure.

In traffic flow and other time series prediction models, the convergence speed reflects the model's efficiency in capturing spatio-temporal features. We define DM-AGODE as an effective spatio-temporal traffic flow prediction model that achieves high prediction accuracy while significantly improving convergence rate and real-time inference speed. To validate this, we conducted comparative experiments with recently advanced spatio-temporal prediction models using the PEMS04 dataset, employing the same learning rates, batch sizes, and loss functions as specified in the original papers for each baseline model.

Fig. 9(a) and Fig. 9(b) compare the efficiency of our model with three recent advanced models that incorporate spatio-temporal features: STSGCN, STFGNN, and STGODE, which uses GODE for spatio-temporal feature extraction. It is evident that STFGNN, which employs a larger adjacency matrix input, initially exhibits higher training loss and validation set MAE. However, from the second round onward, both the STGODE and DM-AGODE models, utilizing the ODE-solver instead of GNN, demonstrate faster convergence. As training progresses, the training loss of all models gradually becomes similar. Yet, DM-AGODE consistently outperforms the others on the validation set.

Importantly, DM-AGODE's performance on the validation set remains more stable due to the ODE-solver and the DB-MTRC module directly extracting spatio-temporal features from raw data, enhancing the model's expressiveness in predictions.

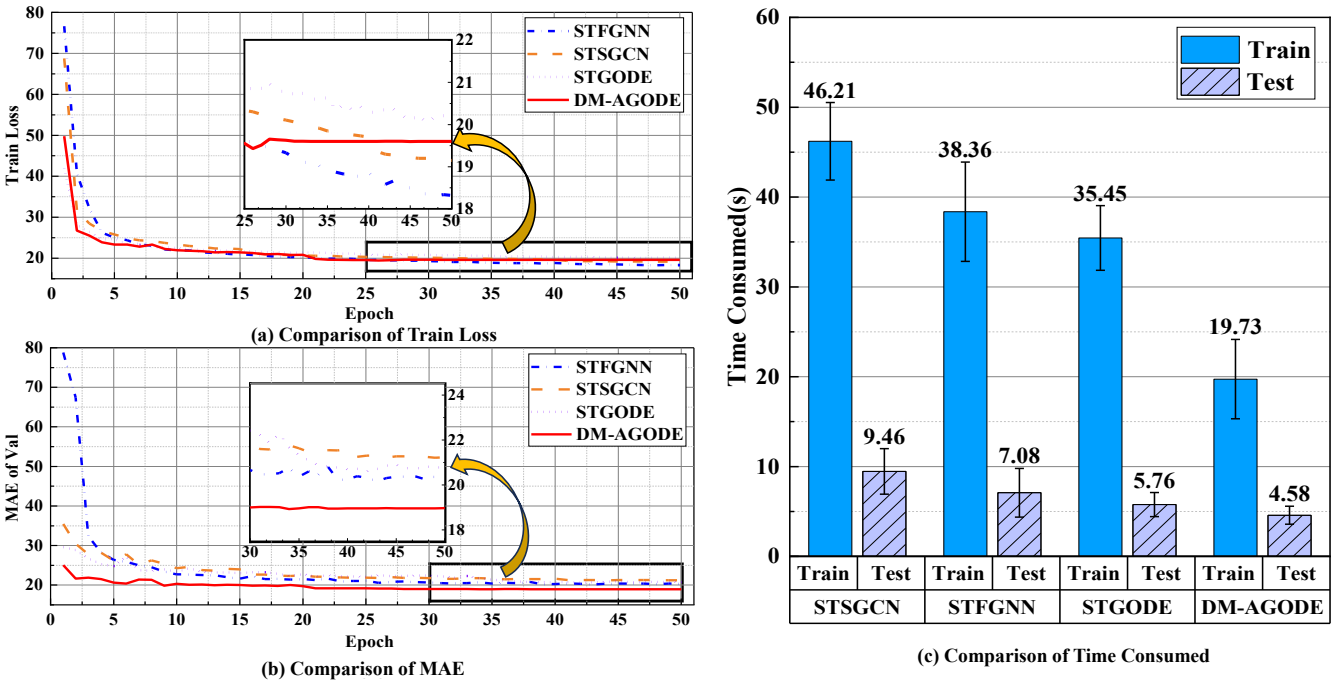


Fig. 9. Performance of different models

Regarding Fig. 9(c), as spatio-temporal prediction models have advanced, improvements have been made to both training and inference speeds. Compared to STGODE, DM-AGODE's training time is reduced by over 40%, resulting in a more than 25% reduction in inference time. In summary, DM-AGODE achieves substantial enhancements in both training efficiency and inference speed.

V. Conclusion

Traffic flow forecasting is crucial for advancing engineering and societal progress, enabling efficient transportation planning, congestion alleviation, resource allocation, and informed decision-making. This paper introduces DM-AGODE, an innovative spatio-temporal traffic flow prediction model. By employing a dual-branch learning approach, DM-AGODE leverages the strengths of both short-term and long-term historical data distributions for multi-step forecasting.

To enhance temporal feature extraction, we developed an efficient multi-temporal resolution convolution mechanism. Additionally, we established a robust, information-enriched adjacency graph that captures both global temporal correlations and local spatial relationships among nodes. This integration of dynamic graph propagation with an optimized adaptive graph neural ODE significantly enhances predictive fidelity and improves the model's understanding of underlying data dynamics.

The approaches explored in this study introduce novel viewpoints and effective solutions for accurate traffic flow prediction. Furthermore, we foresee the possible broader application of several of these methods. For instance, the data-driven adjacency graph construction and the adaptive graph neural ODE have promising potential for extending to diverse prediction tasks, even those without geographical spatial information. The parallel architecture with feature extraction branches warrants special attention, especially in scenarios prioritizing temporal efficiency. It's important to note that while such parallel architectures offer the potential for improved predictions, this isn't a universal guarantee. The success of this approach depends on the efficacy of information extraction branches, the design of residual structures, and the appropriateness of feature fusion strategies, among other factors. Rigorous experiments confirmed that our methodology not only enhances prediction accuracy but also achieves remarkable temporal efficiency in both training and inference, showcasing substantial practical advantages.

Amid technological advancements such as the Internet of Things and smart urban infrastructure, and the increasing availability of datasets, the methods discussed in this paper could play a crucial role in shaping more efficient intelligent transportation systems. Nonetheless, the field of traffic flow prediction still offers substantial potential for further exploration. Incorporating varied data sources beyond spatio-temporal aspects could enhance predictive accuracy, and integrating advanced anomaly detection techniques could improve the model's ability to handle unexpected disruptions in traffic patterns.

Our future research will explore broader traffic prediction tasks, including urban transit passenger flow evolution and more complex origin-destination flow predictions, while refining the model for greater robustness and accuracy, integrating diverse data sources, and addressing further challenges in traffic engineering.

References

[1] Z. Xiong, H. Sheng, W. G. Rong, and D. E. Cooper, "Intelligent transportation systems for smart cities: a progress review," *Science China-Information Sciences*, vol.55, no.12, pp. 2908-2914, 2012.

[2] H. Saleemi, Z. U. Rehman, A. Khan, and A. Aziz, "Effectiveness of Intelligent Transportation System: case study of Lahore safe city," *Transportation Letters-the International Journal of Transportation Research*, vol.14, no.8, pp. 898-908, 2022.

[3] Q. Wang, K. Zhang, C. Zhu, and X. Chen, "Traffic Accident Risk Prediction for Multi-factor Spatio-temporal Networks," *IAENG International Journal of Applied Mathematics*, vol.53, no.4, pp. 1315-1331, 2023.

[4] X. Zhang, G. Yu, J. Shang, and B. Zhang, "Short-term Traffic Flow Prediction With Residual Graph Attention Network," *Engineering Letters*, vol.30, no.4, pp. 1230-1236, 2022.

[5] L. Wang, C. Li, and C. Li, "A Short-term Wind Speed Prediction Method Based on the DGA-BP Neural Network," *IAENG International Journal of Computer Science*, vol.51, no.5, pp. 496-505, 2024.
[6] L. Huang, and L. Y. Xiang, "Method for Meteorological Early Warning of Precipitation-Induced Landslides Based on Deep Neural Network," *Neural Processing Letters*, vol.48, no.2, pp. 1243-1260, 2018.
[7] X. Yang, and L. Liu, "Research on Traffic Flow Prediction based on Chaotic Time Series," *IAENG International Journal of Applied Mathematics*, vol.53, no.3, pp. 1007-1011, 2023.
[8] D. Ma, X. Song, and P. Li, "Daily Traffic Flow Forecasting through a Contextual Convolutional Recurrent Neural Network Modeling Inter- And Intra-Day Traffic Patterns," *IEEE Transactions on Intelligent Transportation Systems*, vol.22, no.5, pp. 2627-2636, 2021.
[9] S. Wang, H. Dai, L. Bai, C. Liu, and J. Chen, "Temporal Branching-Graph Neural ODE without Prior Structure for Traffic Flow Forecasting," *Engineering Letters*, vol.31, no.4, pp. 1534-1545, 2023.
[10] L. Zhao, X. Wen, Y. Wang, and Y. Shao, "A novel hybrid model of ARIMA-MCC and CKDE-GARCH for urban short-term traffic flow prediction," *IET Intelligent Transport Systems*, vol.16, no.2, pp. 206-217, 2022.
[11] Y.-M. Chen, and D.-Y. Xiao, "Traffic network flow forecasting based on switching model," *Kongzhi yu Juece/Control and Decision*, vol.24, no.8, pp. 1177-1187, 2009.
[12] O. Giraka, and V. K. Selvaraj, "Short-term prediction of intersection turning volume using seasonal ARIMA model," *Transportation Letters*, vol.12, no.7, pp. 483-490, 2020.
[13] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Transactions on Intelligent Transportation Systems*, vol.14, no.2, pp. 871-882, 2013.
[14] Y. Wang, R. X. Jia, F. Dai, and Y. X. Ye, "Traffic Flow Prediction Method Based on Seasonal Characteristics and SARIMA-NAR Model," *Applied Sciences-Basel*, vol.12, no.4, pp. 2190, 2022.
[15] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic Flow Prediction with Big Data: A Deep Learning Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol.16, no.2, pp. 865-873, 2015.
[16] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," *31st Youth Academic Annual Conference of Chinese Association of Automation*, pp. 324-328, 2016
[17] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *27th International Joint Conference on Artificial Intelligence, IJCAI 2018*, pp. 3634-3640, 2018
[18] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *6th International Conference on Learning Representations, (ICLR) 2018*, pp. 361-376, 2018
[19] F. Zhou, Q. Yang, K. Zhang, G. Trajcevski, T. Zhong, and A. Khokhar, "Reinforced Spatiotemporal Attentive Graph Neural Networks for Traffic Forecasting," *IEEE Internet of Things Journal*, vol.7, no.7, pp. 6414-6428, 2020.
[20] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," *34th AAAI Conference on Artificial Intelligence, AAAI 2020*, pp. 914-921, 2020
[21] M. Li, and Z. Zhu, "Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting," *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, pp. 4189-4196, 2021
[22] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval," *International Journal of Computer Vision*, vol.40, no.2, pp. 99-121, 2000.
[23] D. J. Berndt, and J. Clifford, "Using dynamic time warping to find patterns in time series," *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pp. 359-370, 1994
[24] Q. Tan, X. Zhang, X. Huang, H. Chen, J. Li, and X. Hu, "Collaborative Graph Neural Networks for Attributed Network Embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol.36, no.3, pp. 972-986, 2024.
[25] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol.30, no.2, pp. 129-150, 2011.
[26] Z. Fang, Q. Long, G. Song, and K. Xie, "Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting," *27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2021, August 14, 2021 - August 18, 2021*, pp. 364-373, 2021
[27] L.-P. a. C. Xhonneux, M. Qu, and J. Tang, "Continuous graph neural networks," *37th International Conference on Machine Learning, (ICML) 2020*, pp. 10363-10372, 2020
[28] C. Chao. Freeway performance measurement system (PeMS). Berkeley, CA, USA: UC Berkeley: California Partners for Advanced Transportation Technology, 2003.
[29] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," *33rd AAAI Conference on Artificial Intelligence, AAAI 2019*, pp. 922-929, 2019
[30] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *28th International Joint Conference on Artificial Intelligence, IJCAI 2019*, pp. 1907-1913, 2019
[31] P. J. Huber. *Robust Estimation of a Location Parameter*. New York, NY: Springer New York, 1992.