# Temporal Branching-Graph Neural ODE without Prior Structure for Traffic Flow Forecasting

Shuang Wang, Hong Dai*, Lin Bai, Chengrui Liu, and Junhong Chen

*Abstract*—**Traffic flow prediction holds significant practical value in enabling efficient traffic management, enhancing transportation planning, and optimizing resource allocation. However, accurately forecasting traffic flow within a spatio-temporal context remains challenging due to complex dependencies within the traffic network. To model the spatio-temporal dependence of traffic sequence data, the existing traffic flow prediction models usually use the adjacency graph based on the prior spatial distribution as the input of shallow graph neural network (GNNs), but the over-expression of spatial adjacency relationship will affect the extraction effect of spatio-temporal dependence. Moreover, the spatio-temporal feature extraction ability of traditional GNNs is limited by the number of levels and the diffusion speed of node features. To address these issues, we propose a novel traffic flow prediction model with a unique historical time perspective: Temporal Branch Convolution Graph Neural ODE (TBC-GNODE), where Temporal Branch Convolution (TBC) is specifically designed to capture complex temporal dependencies in the data, and the Graph Neural ODE (GNODE) module uses adjacency graph with continuous correlation values to extract spatial dependencies, and applies optimized graph neural differential equations to represent the dynamic system of traffic flow variation. These modules are combined in a parallel structure and stacked in a residual manner to extract long-range spatio-temporal dependencies synchronously. Our experiments on four real datasets demonstrate that our proposed model improves prediction performance by over 4% on average compared to the baseline model. This showcases its practical application potential in the field.**

*Index Terms*—**spatio-temporal forecasting, dynamic system, graph neural ODE, dilated convolution**

## I. Introduction

In recent years, the spatio-temporal prediction task of traffic within the context of Intelligent Transportation Systems (ITS) has garnered significant attention. Advanced computing capabilities and the availability of vast amounts of spatio-temporal data have paved the way for applying deep learning methodologies in the prediction of traffic patterns. By using deep learning techniques, spatio-temporal data can be analyzed, which in turn generates situational awareness of the road network for intelligent transportation. This plays a crucial role in the downstream modules of ITS, such as traffic scheduling, intelligent signal control, and road planning [1]. Therefore, research in this field is of significant importance to the enhancement of human production and quality of life.

Thus far, the spatio-temporal prediction of traffic remains a challenging task. One of the primary obstacles is the susceptibility of traffic flow data to the spatial correlation of road sensors [2]. However, this correlation cannot be entirely determined by the distance between sensors. Fig. 1 depicts the diverse representations of regional connections in a transportation network. Fig. 1(a) illustrates the geographical topology of the traffic road network based on a prior distance. While Fig. 1(b) exhibits the resulting semantic correlation graph structure, obtained via calculation, which denotes the weight representing the strength of association. In the field of traffic prediction, a notable observation is the existence of similar traffic patterns between distant roads, despite their spatial separation. Conversely, for traffic nodes that are closely positioned yet not in each other's essential path, the adjacency relationship based solely on realistic distance may introduce noise that can adversely affect the predictive model [3]. Therefore, the problem of extracting the profound spatial associations of traffic nodes is a topic of considerable importance in traffic prediction research. Additionally, it is crucial to explore suitable approaches for integrating the extracted spatial features and temporal patterns to maximize the advantages of spatio-temporal features in traffic flow. This represents another significant challenge that warrants further study.

Within the realm of traffic prediction, graph neural networks (GNNs) have emerged as a popular choice among researchers due to their robust processing capabilities of graph topological data [4]. GNNs can efficiently aggregate the features of graph neighbor nodes in a hierarchical neural network structure and have demonstrated exceptional performance in tasks such as node classification [5, 6], link prediction [7], and scene graph generation [8]. In traffic prediction tasks, a significant amount of attention has been dedicated to the spatial-association representation of traffic nodes, such as the combination of multiple distance adjacency matrices with matrices representing their individual connections into spatio-temporal synchronization graphs [9, 10] or the complete replacement of the prior

Manuscript received May 11, 2023; revised September 5, 2023. The research work was supported by the Graduate student science and technology innovation project of University of Science and Technology Liaoning (No. LKDYC202221).

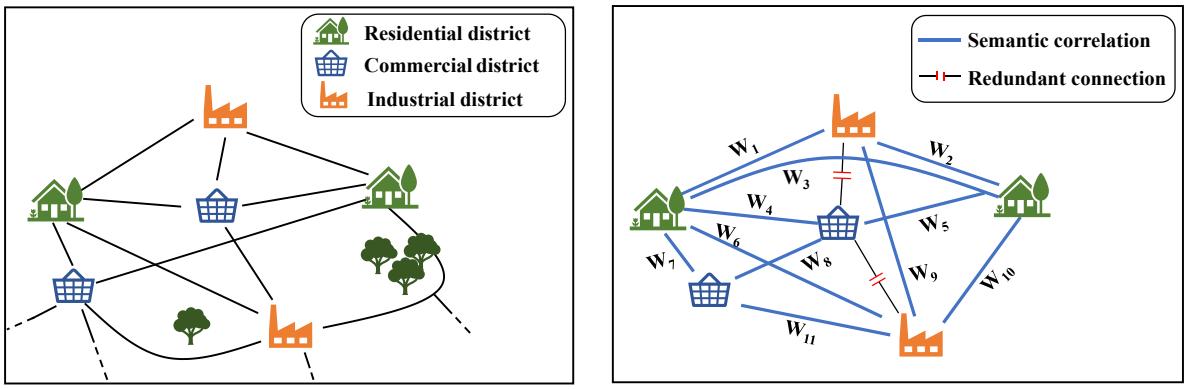
Shuang Wang is a postgraduate student of University of Science and Technology Liaoning, Anshan, Liaoning, CO 114051, China. (e-mail: 1657669526@qq.com).

Hong Dai* is a professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, China. (corresponding author to provide phone: +086-186-4226-8599; fax:0412-5929818; e-mail: dear_red9@163.com).

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

Junhong Chen is a postgraduate student of University of Science and Technology Liaoning, Anshan, Liaoning, CO 114051, China. (e-mail: shopkeeperakashi@hotmail.com).

Chengrui Liu is a postgraduate student of University of Science and Technology Liaoning, Anshan, Liaoning, CO 114051, China. (e-mail: cherryliumark@163.com).




**(a)** Correlation based on geographic distance **(b)** Correlation based on traffic semantics

Fig. 1. Geographical connection and semantic connection in a traffic road network

graphs adaptive adjacency matrices solely controlled by learnable variables [11]. Despite these advancements, the over-smoothing problem of GNNs poses difficulties for fully adaptive adjacency matrices to learn the complete correlation expression between nodes. Although the spliced space-time graph can capture the local and global spatial correlation of traffic patterns to a certain extent, the excessively large graph structure inevitably contains some superfluous connections, acting as a limiting factor for graph GNNs. In conclusion, for complex traffic flow data, the expression of adjacency relationship with necessary correlation information is the crucial problem to improve the accuracy of prediction results.

On the other hand, extracting time patterns from complex traffic data is also a subject of active research. The RNN-like structure of Long Short-Term Memory(LSTM) [12, 13] and Gated Recurrent Unit(GRU) [14] models, however, makes them susceptible to the issue of gradient vanishing or explosion when capturing ultra-remote sequence data [15]. Moreover, the Transformer model, though promising, presents a complex structure for spatio-temporal sequence data and entails a high time cost of reasoning [16]. Therefore, in the domain of traffic prediction, the development of a lightweight and efficient time feature extraction method constitutes another significant challenge in enhancing the predictive ability of the model.

In order to comprehensively capture the effective semantic correlation in space and improve the accuracy of the model in learning the spatio-temporal characteristics of traffic data, a spatial correlation graph without prior spatial structure was proposed in this study, inspired by the Dynamic Time Warping method [17]. An improved Graph Neural Ordinary Differential Equation (GNODE) module was constructed to capture the continuous dynamics of traffic flow and address the problem of spatial dependence extraction. In addition, this paper analyzes the time dimension features of traffic sequences from a new perspective and proposes two parallel multi-windows extended convolution modules to help the model capture multi-dimensional time dependence. In summary, the primary contributions of this study include:

1) We propose an innovative method for creating a spatial correlation graph solely based on sequence data. This approach allows us to represent the semantic correlation and specific correlation degree between traffic nodes without any prior spatial structure. In contrast to the existing integrated spatio-temporal graph model, our dynamic spatio-temporal correlation graph is more refined in size and contains more abundant and reliable spatial dependence information.
2) This paper presents a novel Graph Neural ODE (GNODE) module for effectively extracting association information. The GNODE module is derived and elaborated upon in detail, emphasizing its unique characteristics. Notably, the module incorporates a more reasonable parameter matrix and connection mode, capitalizing on the interpretability advantage of the Sylvester Differential Equation [18, 19]. Moreover, it circumvents the limitation of traditional Graph Neural Networks (GNN) regarding over-smoothing [20]. To further improve the robustness of the model against approximation errors, we designed a connection module for multi-step GNODE solutions, integrating the concept of residual connection.
3) To extract more comprehensive time-dependent features, this study reexamines the composition structure of historical data from a novel perspective, proposes a subsequence learning approach, and devises a Temporal Branch Convolution (TBC) module that is parallel to GNODE. The TBC module inherits the lightweight features of dilated convolution[21] and can focus on the temporal correlation of subsequence traffic sequences. This approach enhances the model's sensitivity to short-term oscillations, and the residual stacking module captures the temporal dependence on long-term time scales.

Furthermore, to improve GNODE's ability to capture the dynamics of the original data more accurately, we introduce the original data with only feature embedding as input to GNODE. We designed a refined residual GNODE spatio-temporal parallel learning framework to extract spatio-temporal features of the traffic sequence synchronously.

## II. Related Work

### *A. The Fusion of Spatio-Temporal Features*

In recent years, researchers in the field of traffic prediction have focused on extracting and fusing spatio-temporal features to utilize the characteristics of traffic flow fully. In 2018, Li Y. et al. proposed a traffic flow prediction model named DCRNN [22], based on Graph Neural Networks, which employs the bidirectional random walk

method to model the spatial information of the traffic graph, and GRU to extract temporal dynamic information. Additionally, Yu B. et al. proposed a spatio-temporal traffic flow prediction framework named Spatio-temporal Graph Convolutional Network (STGCN), which uses spatial graph convolution and one-dimensional temporal convolution for spatio-temporal feature extraction [23]. These methods aim to enhance prediction accuracy by effectively extracting and fusing spatio-temporal features from traffic flow data. Song et al. integrated the concept of time step into the spatio-temporal graph, extending the $N \times N$ dimensional adjacency matrix to a larger matrix($3N \times 3N$). This allowed for the simultaneous representation of spatial and local temporal correlations, with the spatio-temporal graph convolution module extracting spatial and temporal dependencies synchronously. The resulting model was named STSGCN [9]. In 2021, Li et al. proposed STFGNN[10], which utilizes a DTW similarity matrix as an extension of the original distance matrix to further enhance the size and expression capacity of the spatio-temporal graph. The resultant graph is then used as input to gated time convolution, facilitating the extraction of long-range spatio-temporal dependencies. Wu et al. designed an adjacency graph entirely defined by a learnable parameter matrix [11] and applied GNN and extended causal convolution to extract spatio-temporal features, respectively.

However, the majority of the aforementioned studies utilize the original distance matrix, which contains certain erroneous connections, rendering it challenging to enhance the precision of the models.

### *B. Neural Ordinary Differential Equation*

In 2018, Chen et al. proposed a novel method for modeling continuous-time dynamical systems using neural networks called Neural Ordinary Differential Equations [24] (Neural ODE). This was achieved by using a differential equation solver to compute the evolution of the hidden states of the neural network. This approach allows complex data to be modeled in continuous time. Based on the original Neural ODE method, researchers at home and abroad have begun to further study and apply this method in recent years. For example, Schildt et al. proposed Stiff Neural ODE[25] for classical problems with rigid conditions, which allows Neural ODE to be applied in practical issues such as chemical kinetics and environmental engineering with multiple time scales. An important extension of Neural ODE in graph structure modeling is Continuous Graph Neural Networks (CGNN), proposed by Xhonneux et al. [26]. This study first proposed the concept of continuous graph neural networks. Differential equations are used to model the discrete evolution process of the graph as continuous, and it is proven that the continuous graph neural network is robust to the over-smoothing of graph neural networks, meaning that the original graph neural network structure can be deeper to capture long-term dependencies between nodes. Moreover, the continuous nature of the neural network enables node features to quickly spread to the entire graph in continuous time, which improves stability and efficiency. Subsequently, Fang Z. et al. proposed the Spatio-Temporal Graph ODE Networks (STGODE) [27], which employ parameterized GNODEs instead of traditional GNNs for spatio-temporal prediction and have achieved outstanding predictive performance.

However, the CGNN's overall architecture is relatively lightweight and may not be well-suited for modeling complex traffic flows. Similarly, the GNODE module in STGODE only considers for temporal and feature transition matrices, with the extraction of spatial features still being locally constrained by prior spatial distance. Addressing these limitations may offer the potential for further enhancements in the model's predictive accuracy.

## III. PRELIMINARIES

### *A. Problem Formulation*

Traffic flow prediction falls under the category of sequence prediction, wherein the objective is to forecast future traffic flow data based on past observations. In the context of traffic flow prediction, the graph information $G$ represents the network topology, while the input sequence $X \in \mathbb{R}^{N \times H \times d}$ comprises the graph signal, which is the data recorded by $N$ sensors. The output sequence $Y \in \mathbb{R}^{N \times F}$ corresponds to the predicted traffic flow at a future time step $T'$. The objective is to learn a mapping function that can accurately predict the traffic flow $T'$ steps:

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

Where $0 < i \le T$ , $T < j \le T'$ , $X_n^i$ denotes the i-th step historical traffic data of the n-th node, and $Y_n^j$ denotes the j-th step future traffic data of the n-th node.

### *B. Neural ODE*

Neural ODE enables the integration of neural networks with dynamic systems. It involves a differential equation and employs a neural network to parameterize a vector field. The propagation process of a neural network is modeled as a solution process of a differential equation with the state of neurons regarded as the state variable in a dynamic system. Neural differential equations offer another perspective for comprehending and designing neural networks. An illustrative example of Neural ODE is as follows:

$$h(0) = h_0, \;\; h(t) = ODE\left(\theta, f, h(0), t\right) \tag{2}$$

Since Neural ODE is a continuous dynamic system, $t$ in the above equation denotes a continuous time range, $h(t)$ represents the state vector of the neuron at time $t$, $h(0)$ is the initial state, $\theta$ is the model's parameter, and $f : \mathbb{R} \times \mathbb{R}^{d_1 \times \ldots \times d_k} \to \mathbb{R}^{d_1 \times \ldots \times d_k}$ can be any standard neural network architecture. $ODE(\cdot)$ indicates the operation for solving differential equations, which can be expressed as a standard differential equation form:

$$h(0) = h_0, \;\; \frac{dh(t)}{dt} = f\left(\theta, h(t), t\right) \tag{3}$$

Where $f(\cdot)$ is considered as a function of the dynamic system in Neural ODE and $dh(t)/dt$ represents the derivative of the neuron state $h(t)$ with respect to time $t$. This equation signifies the progression of the neural network over time. The input of the Neural ODE consists of the initial state $h(0)$ and the dynamic system $f$ , and the independent variable is the time variable $t$. The hidden state $h(T)$ of the neural network at the target time point $T$ can be determined by solving the ODE.

## IV. Model

### A. Adjacency Matrix

The adjacency matrix is a matrix that describes the correlation information of the nodes in a graph and serves as the foundation of the graph convolution operation. In the previous work [9, 10, 23, 27], the adjacency matrix for traffic is usually defined using spatial distance information as follows:

$$A_{i,j} = \begin{cases} 1, & if\ d_{i,j} \le \partial \\ 0, & otherwise \end{cases} \tag{4}$$

Where $d_{i,j}$ represents the spatial distance between traffic detector node $i$ and node $j$, while $\partial$ is the threshold that regulates the sparsity of nodes.

To a certain extent, spatial distance can indicate the correlation between nodes. However, in a real complex traffic network, the adjacency matrix that uses the geographical distance between two nodes as a correlation index may not always accurately represent the correlation degree between nodes. For instance, consider two intersecting traffic routes in reality, such as overpasses. Although their starting and ending points are not similar, there may be detector nodes located near their intersection. In such cases, the spatial distance between the pair of detector nodes would be extremely small, but the similarity of their traffic flow may not be substantial.

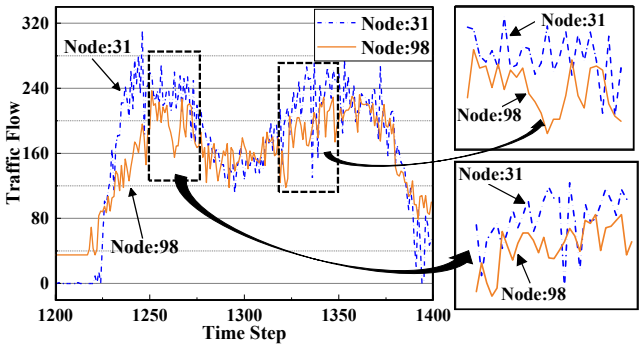



Fig. 2. The data comparison of two sample traffic nodes in the real data

Fig. 2 displays traffic flow data from two detector nodes during the same time period. It is evident that the traffic patterns between the two nodes differ significantly with respect to the onset time of fluctuations and the period of fluctuation. Consequently, an edge between these two nodes in the graph structure will not facilitate GNN learning and might even introduce noise. Unfortunately, despite the dissimilarity in traffic patterns, these nodes are situated in close proximity to each other in the actual traffic network. It results in an edge being present between the nodes in the graph structure constructed using traditional distance measurement methods, thereby contributing to noise in the GNN. In contrast to prior work, we utilize only the DTW distance to gauge the similarity of traffic nodes and create a graph adjacency matrix $A \in \mathbb{R}^{N \times N}$ that expresses the degree of node dependence. The refined size of $A$ represents the correlation of traffic patterns between graph nodes.

The DTW method[17] is a frequently used technique for measuring sequence similarity, boasting strong adaptability to fluctuations within sequences and an ability to handle local translation and scaling matching effectively. DTW can conduct similarity-matching computations on specific time steps within traffic flow sequences. Its specific calculation formula is presented below:

$$DTW(X,Y) = \sqrt{D(L_X, L_Y)} \tag{5}$$

In Equation (5), $L_X$ and $L_Y$ denote the data sequence length of nodes $X$ and $Y$, respectively, while $D$ represents the alignment cost matrix of the two nodes. When provided with a Euclidean distance matrix $M$, Equation (5) can be used to compute $D(i,j)$ via the recursive Equation (6):

$$\begin{aligned} D(i,j) &= M_{i,j} \\ &+ min\{D(i-1,j), D(i,j-1), D(i-1,j-1)\} \end{aligned} \tag{6}$$

Where $i$ and $j$ denote the recurrence term, while the initial condition for the recurrence formula is given as follows:

$$D(i_0,j_0) = \begin{cases} 0, & i_0 = 0\ and\ j_0 = 0 \\ \infty, & i_0 = 0\ and\ j_0 \ne 0 \\ \infty, & i_0 \ne 0\ and\ j_0 = 0 \end{cases} \tag{7}$$

Typically, the adjacency graph produced by Equation (4) can indicate whether nodes are similar enough to have connected edges. Nevertheless, when we use similarity as the basis for adjacency, this expression may not fully reflect the degree of similarity between their traffic patterns. Hence, we utilize similarity as the weight of edges to enhance the adjacency graph, enriching it with more comprehensive node dependence information. Ultimately, our adopted method for constructing the adjacency matrix is as follows:

$$A_{i,j} = \begin{cases} exp\left(-\left(\frac{DTW(i,j)}{\sigma}\right)^2\right), & if\ exp\left(-\left(\frac{DTW(i,j)}{\sigma}\right)^2\right) \le \delta \\ 0, & otherwise \end{cases} \tag{8}$$

To control the sparsity of the matrix, threshold parameters $\sigma$ and $\delta$ are used in Equation (8), to ensure the effectiveness of the adjacency information and mitigate the impact of poor associations on graph learning, any connections below a designated threshold $\delta$ are set to 0.

### B. Graph Neural ODE Module

The core idea of a graph neural network involves updating a node's feature information by aggregating its neighboring nodes' features within the graph. The commonly utilized graph convolution formula, grounded in spectral graph theory [28], can be expressed as follows:

$$H^{(l+1)} = \sigma(\tilde{A}H^{(l)}W) \tag{9}$$

Where $H^{(l+1)}$ represents the output state of layer $l$, $H^{(l)}$ represents the feature matrix of layer $l$, $\tilde{A} \in \mathbb{R}^{N \times N}$ represents the adjacency matrix after regularization, $W \in \mathbb{R}^{d_l \times d_{l+1}}$ represents the weight matrix between layer $l$ and layer $l+1$, which is one of the parameters that the model learns, and $\sigma$ represents the activation function. However, the main issue with this model is its tendency to over-smooth, which results in a limited depth of the convolutional layer. Drawing inspiration from [26] and [27], we view GNN based on spectral graph theory as a time-varying neural dynamic system that extends the capabilities of Neural ODE and Graph Neural Networks (GNNs) to handle time series data and adaptively adjusts model complexity. This allows us to effectively address the over-smoothing problem by

incorporating spatio-temporal information of graph structures into the continuous time domain. To enhance the spatial-temporal modeling and expression capability of GNN for graph data, we consider a discrete dynamic as follows:

$$H^{(l+1)} = H^{(l)} \times A \dot{\times} W_N \dot{\times} W_T \dot{\times} W_C + H^{(0)} \tag{10}$$

In Equation (10), the output feature of the $l$-th layer is denoted by $H^{(l)} \in \mathbb{R}^{N\times T\times C}$, while $H^{(0)}$ represents the residual term of the restart distribution which is added to mitigate the issue of over-smoothing. $W_C = Q_1 \Pi_1 Q_1^T$ and $W_T = Q_2 \Pi_2 Q_2^T$ are diagonalizable feature transformation matrix and time transformation matrix respectively, where $Q_i$ is the learnable orthogonal matrix, and $\Pi_i$ is the learnable vector with all elements greater than 0 and less than 1, both having all eigenvalues less than 1. Unlike previous works, our approach aims to make the node sequence learn synchronously in time and space dimensions by transforming or adjusting the feature matrices of different nodes, and further stimulating the strong diffusion potential of GNODE. For this purpose, we introduce an auxiliary space transformation matrix $W_N = Q_3 \Pi_3 Q_3^T$, which dynamically supplements the graph structure and participates in the time evolution of discrete dynamics. Finally, the product of the tensor and the corresponding dimension of the matrix is denoted by "$\dot{\times}$", this operation is often referred to as Einstein operation[29] and can be calculated as follows:

$$D_{\alpha,\beta,\gamma,\ldots} = \sum_{i,j,k\ldots} A_{i,\alpha} \dot{\times} B_{j,\beta} \dot{\times} C_{k,\gamma} \dot{\times} \ldots \tag{11}$$

In this equation, $A, B, C$ et al. is a tensor with appropriate indices, and the indices labeled $i, j, k$ et al. are summed, while the indices $\alpha, \beta, \gamma$ et al. that have not been summed are in the resulting tensor $D$. To ensure clarity in subsequent derivations of the dynamic system, it is essential to note that $\dot{\times}$ has specific mathematical properties that are similar to common matrix multiplication:

1) Associative property: The results of Einstein operation are controlled by the input tensor and the manipulated index, which ensures that the method is group-independent. For example:

$$A \dot{\times} B \dot{\times} C = A \dot{\times} \left(B \dot{\times} C\right) = \left(A \dot{\times} B\right) \dot{\times} C \tag{12}$$

2) Distributive property: Einstein operation is distributive over addition, meaning we can distribute the operation over a sum of tensors. For example:

$$\left(A + B\right) \dot{\times} C = A \dot{\times} C + B \dot{\times} C \tag{13}$$

In the context of the Einstein operation, many of the mathematical properties that apply to matrix multiplication also apply, including linearity, product rule, and chain rule of matrix derivatives. This enables us to compute derivatives of complex functions involving tensors using Einstein expressions. However, it should be noted that the Einstein operation is not the same as matrix multiplication and involves different indices that must be appropriately contracted to obtain the desired result. This extension allows us to compute derivatives of complex functions involve tensors.

Therefore, in the following derivations of ODEs, the Einstein operation can still be regarded as a matrix multiplication extended to the relevant dimension. The feature matrix tuple $\{W_N, W_T, W_C\}$ dynamically adjusts the spatial, temporal, and embedded features of the neural ODE in Equation (10), allowing for rapid diffusion of high-dimensional features across the entire graph structure. The explicit formula for this dynamic system can be obtained as follows:

$$H^{(l)} = \sum_{i=0}^{l} \left( A^i \times H^{(0)} \dot{\times} W_N^i \dot{\times} W_T^i \dot{\times} W_C^i \right) \tag{14}$$

In order to construct a continuous dynamic system, we replace the original discrete layer variable $l$ with a continuous variable $t \in \mathbb{R}_0^+$. We consider Equation (14) as the Riemann sum of the integrals from time $t = 0$ to time $t = l$, which enables the extension of the discrete propagation process to the continuous propagation case. We let $j = i - 1$, $\Delta t = (t + 1 - 0)/(n + 1)$, and $t = n$, yielding the following equation:

$$H^{(l)} = \sum_{j=0}^{l} (A^{j\times\Delta t} \times H^{(0)} \dot{\times} W_N^{j\times\Delta t} \dot{\times} W_T^{j\times\Delta t} \dot{\times} W_C^{j\times\Delta t}) \times \Delta t \tag{15}$$

When $n \to \infty$, the following equation holds true:

$$H^{(l)} = \int_0^{t+1} A^{\omega} \times H^{(0)} \dot{\times} W_N^{\omega} \dot{\times} W_T^{\omega} \dot{\times} W_C^{\omega} d_{\omega} \tag{16}$$

Furthermore, by solving the second derivative of the left and right terms in the equation as mentioned above and carrying out another integral operation, it is possible to obtain that the discrete dynamics in Equation (10) can be discretized from the following ODE:

$$\begin{aligned} \frac{dH^{(t)}}{dt} = {} & ln\tilde{A} \times H^{(t)} + H^{(t)} \dot{\times} lnW_N \\ & + H^{(t)} \dot{\times} lnW_T + H^{(t)} \dot{\times} lnW_C + H^{(0)} \end{aligned} \tag{17}$$

In model calculations, computing matrix logarithms as shown in Equation (17) can be a complicated process. To simplify this calculation, a first-order Taylor expansion[26] can be used as an approximation, where $lnM$ is approximated by $M - I$:

$$\begin{aligned} \frac{dH^{(t)}}{dt} = {} & (\tilde{A} - I) \times H^{(t)} + H^{(t)} \dot{\times} (W_N - I) \\ & + H^{(t)} \dot{\times} (W_T - I) \\ & + H^{(t)} \dot{\times} (W_C - I) + H^{(0)} \end{aligned} \tag{18}$$

Similar to the previous work [26], the differential equation presented above can also be reformulated into the general form of Sylvester Differential Equation [19], which is given by:

$$\begin{pmatrix} \dfrac{dX^{(t)}}{dt} = A \times X^{(t)} + B \times X^{(t)} + C \\ X^{(0)} = D \end{pmatrix} \tag{19}$$

By utilizing Equation (18), we can derive the following expression:

$$\begin{pmatrix} \dfrac{dH^{(t)}}{dt} = \left(A - I\right) \times H^{(t)} \\ \qquad\qquad + H^{(t)} \dot{\times} \left(W_N + W_T + W_C\right) + H_0 - 3H^{(t)} \\ H^{(0)} = H_0 \end{pmatrix} \tag{20}$$

In this context, let $W_N + W_T + W_C - 3I = W' \in \mathbb{R}^{N\times T\times C}$, we can obtain the following expression:

$$\begin{cases} \dfrac{dH^{(t)}}{dt} = (A-I)\times H^{(t)} + H^{(t)} \dot{\times} W' + H_0 \\ H^{(0)} = H_0 \end{cases} \tag{21}$$

In the context of predictive models, Sylvester Differential Equations possess two essential properties:

1) Given an initial value condition $H^{(0)}$, the Sylvester Differential Equation has a unique solution $H^{(t)}$, which is essential for ensuring the accuracy and reliability of the model predictions.
2) As the dynamic time approaches infinity, the solution of the equation approaches a stable value, ensuring that the model predictions are not affected by long time spans, which are typically represented by numerous layers in discretization graph neural networks.

Thus far, Equation (18) has provided an approximate representation of the differential equation used in the solver. Additionally, when the Sylvester operator $\boldsymbol{S} = (A-I)\times H^{(t)} + H^{(t)} \dot{\times} W'$ is set, Equation (18) has a unique solution in the following form:

$$\begin{aligned} H^{(t)} &= e^{t\boldsymbol{S}} H_0 + \int_0^t \left( e^{(t-\mu)\boldsymbol{S}} H_0 \right) d\mu \\ &= e^{t\boldsymbol{S}} H_0 + \left[ -\boldsymbol{S}^{-1} e^{(t-\mu)\boldsymbol{S}} H^{(0)} \right]_{\mu=0}^{\mu=t} \\ &= e^{t\boldsymbol{S}} H_0 - \boldsymbol{S}^{-1} H_0 + \boldsymbol{S}^{-1} e^{t\boldsymbol{S}} H_0 \end{aligned} \tag{22}$$

To validate the accuracy of the formula as mentioned above, we can substitute the initial value $H^{(0)} = H_0$ into the following expression:

$$H^{(0)} = H_0 - \boldsymbol{S}^{-1} H_0 + \boldsymbol{S}^{-1} H_0 = H_0 \tag{23}$$

In our approach, we interpret the given differential equation as a continuous dynamic system, and use the initial state $H_0$, obtained through feature embedding of the input data, to solve the system. We specify a time point tensor $t = [t_0, t_1, t_2]$ to solve the ODE at various time instants, with the input value $t_i \in [0, \infty)$ serving as the initialization point for the solver. To ensure that the subsequent residual connection is effective, we set the time step of the solution using the half-value link, which is $t_1 = t_2 / 2$. Numerical methods can then be employed to solve the problem at corresponding time points. By solving the ODE, we can obtain the numerical solutions at all time instants, representing the system's spatio-temporal dynamics.

In the previous simplification of the matrix logarithm, we utilized the first-order Taylor approximation, which yields a small error that accumulates as the ODE solving time step increases. To comprehensively incorporate the solution state of different time steps, we define the output solution state as $[X_0, X_1, X_2]$, and obtain the final output as follows:

$$X_{GODE} = (1-\xi) X_1 + \xi X_2 \tag{24}$$

In the equation above, $\xi$ is a linear weight parameter used to balance the contribution value. Notably, the GNODE applied to the ODE Solver considers the restart distribution term $H_0$, which means that the initial state $X_0$ is not considered in the residual connection of the equation above. This condition is crucial for ensuring the reliability of the dynamic system's results.

### *C. Temporal Branch Convolution Module*

In the context of spatio-temporal sequence prediction, while the GNODE above possesses robust spatial representation capabilities and stable performance, their accuracy is somewhat limited due to the complex and variable nature of traffic flow data. Consequently, the extraction of temporal features constitutes another crucial factor in this task, alongside spatial correlation feature extraction. To tackle this challenge, we developed a pioneering approach called Temporal Branch Convolution (TBC).

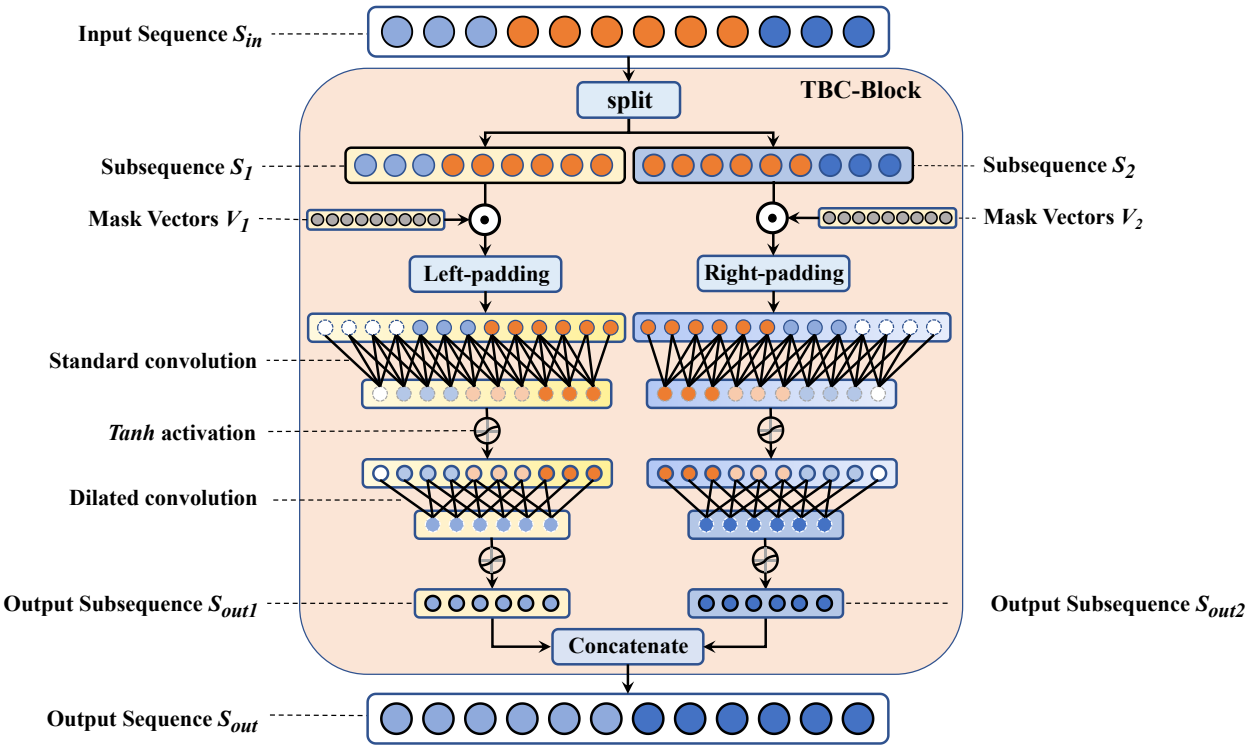



Fig. 3. Details of the TBC-Block

In the task of traffic prediction, the predicted future data is contingent upon the historical data, and therefore the extent to which the prediction results depend on the historical time data may vary depending on the frequency and amplitude of oscillations. In other words, different historical time steps may exert varying degrees of influence on the prediction results. In consideration of the time characteristics of traffic data, we partition the entire sequence into two subsequences, the front subsequence and the back subsequence. These subsequences represent the deep historical distribution containing the long-term potential distribution in the historical data and the future trend distribution with trend dynamic characteristics, respectively. In the task of data-driven sequence prediction, we aim to apply different temporal convolution operations to the subsequences with distinct potential features so as to acquire more comprehensive and diverse multi-dimensional features.

The details of the TBC-Block are shown in Fig. 3. where sequence elements with different stages and roles are represented by different colors, in the TBC-Block, $S_{in}$ refers to the input sequence, which is considered to have hierarchical temporal characteristics. This sequence is split into two sub-sequences: head subsequence $S_1$ and rear subsequence $S_2$ , representing the deep historical and shallow future trend sequences, respectively. Multi-dimensional hierarchical features are extracted from these sub-sequences and used as input for two synchronous dilated convolutions, which are then concatenated to generate the prediction output. Moreover, we propose two sets of Mask Vectors, $V_1$ and $V_2$, as learnable parameters in the model framework. This can be interpreted as an attention mechanism applied to historical time steps, allowing the model to adaptively adjust the contribution of different positions in the sub-sequence during training.

In the process of model training, it is common for the model to learn deep features that are not representative of the original sequence when two subsequences are completely separated. This phenomenon leads to the overfitting of branch granularity and negatively affects the overall model expression. To ensure the robustness and generalization of the deep features of the branch, we propose focusing on the middle sequence while learning multidimensional features. Specifically, we simultaneously shift the central part of the input sequence (the orange sequence elements in Fig. 3) to the right and left ends of the two subsequences. This approach ensures that the model always has comprehensive features with a robust overall representation during the learning process, thus avoiding overfitting issues in branch convolution learning.

In addition, we consider the iterative training mechanism and optimize the subsequence composition comprehensively. The sequence prediction model training can be regarded as a left-out-stack advancement process, where most data changes from the rear subsequence to the head subsequence as shown in Fig. 4.

To learn the time characteristics of the rear subsequence and consider its deep features when it becomes the head subsequence, a padding operation is performed on the right side of the rear subsequence. Similarly, the padding operation is performed on the left side of the head subsequence to retain the shallow features of the previous sequence as the rear subsequence when the rear subsequence becomes the head subsequence.

We perform these padding operations to ensure that the data in the head subsequence is retained during the learning process when it is used as a rear subsequence. Following this, two multilayer convolutions with the same structure extract features from the padded sequence, resulting in two time series $S_{out1}$ and $S_{out2}$ with length $T'/2$. Finally, these two series are concatenated in series according to the original sequence, where the notation $S_i^j$ refers to the $j$-th element in the output sequence $S_i$.

$$S_{out} = \left\{ S_{out1}^1, S_{out1}^2, \ldots, S_{out1}^{(T'/2)}, S_{out2}^1, S_{out2}^2, \ldots, S_{out2}^{(T'/2)} \right\} \quad (25)$$

*D. TBC-GNODE*

Given that the features extracted by the convolution layer typically correspond to nonlinear high-dimensional representations of the input data, the complex and nonlinear

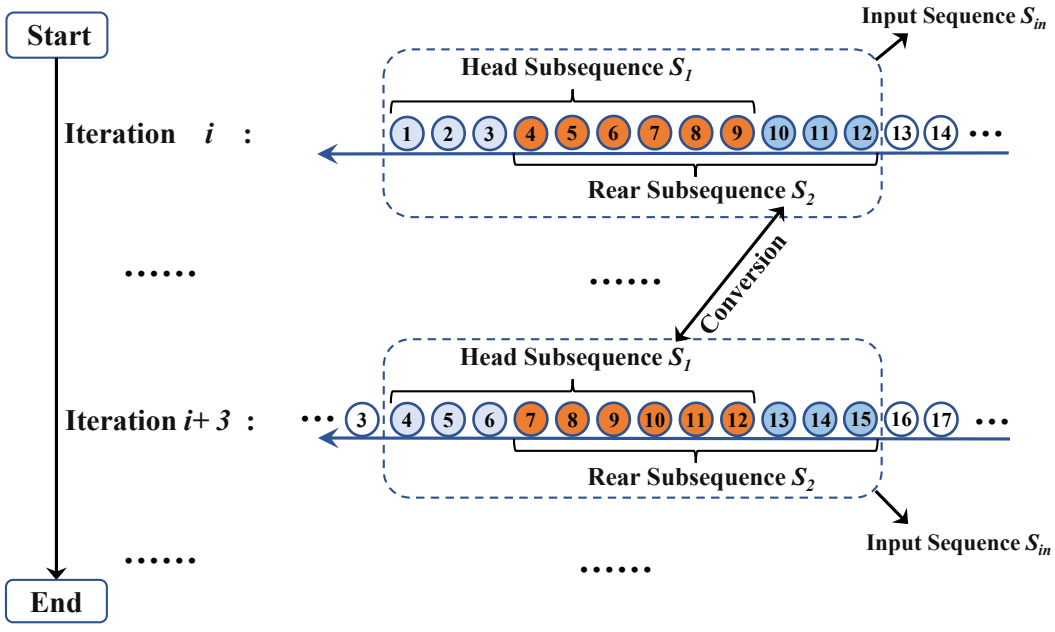



Fig. 4. Head and rear subsequence relationships in multiple iterations

nature of these features may impact the ability of the GNODE module to capture potential patterns and dynamics in the data effectively. Therefore, it is important to carefully consider the impact of the convolutional layer on the original data when inputting these features into the GNODE module.

To fully utilize the capabilities of the GNODE module and capture the continuous evolution of graph node characteristics over time or spatial position, we propose a parallel synchronous structure consisting of the GNODE module for capturing spatial dependence and the TBC module for acquiring deep traffic flow patterns. Compared to previous work with a "sandwich" structure, featuring three layers of modules arranged in sequence [9, 27], this synchronous architecture for feature extraction in the spatio-temporal domain can capture more complex nonlinear dynamics in the original data, thus enhancing the model's fitting ability.

Incorporating the design mentioned above considerations, the architecture and connectivity details of the final model are illustrated in Fig. 5. Where Fig. 5(a) displays the forecasting framework of TBC-GNODE, where the nodes in the graph are shaded with different connection colors to represent their degrees of association. The input sequence and output sequence lengths of the model are T and T', respectively. Fig. 5(b) presents the fusion mode of TBC-GNODE internal modules. After feature embedding, the original data and the prior-free spatio-temporal adjacency graph are input to the ODE solver to obtain richer spatio-temporal feature expressions. The TBC module extracts spatio-temporal correlation features of different historical depths synchronously, and the branch features obtained from it are fused element-wise. To capture long-range spatio-temporal dependencies without losing low-level features, residual connections are incorporated among multiple TBC-GNODE modules. Fig. 5(c) illustrates the stack architecture of TBC-GNODE connected by the residual structure.

In order to reduce the sensitivity to outliers, the model utilizes the Huber loss function [30], this loss function strikes a balance between the Mean Squared Error (MSE) and Mean Absolute Error (MAE) loss functions. Huber loss is expressed as:

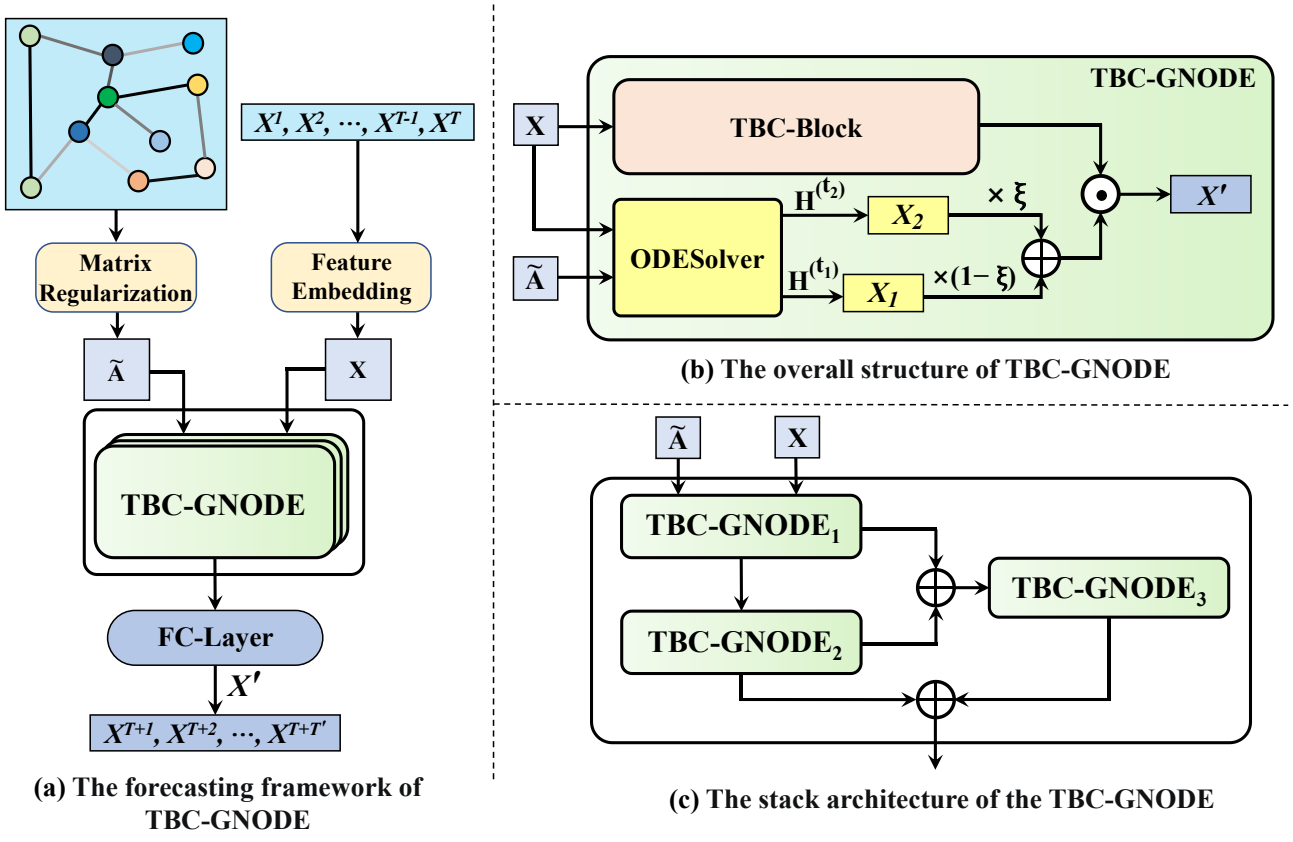
$$L_\rho\left(Y, f\left(X\right)\right)=\begin{cases}\frac{1}{2}\left(Y-f\left(X\right)\right)^2 & , \quad if\left|Y-f\left(X\right)\right|\le\rho\\ \rho\left|Y-f\left(X\right)\right|-\frac{1}{2}\rho^2, & if\left|Y-f\left(X\right)\right|>\rho\end{cases} \tag{26}$$

In the given equation, $Y$ represents the ground truth value, $f(X)$ represents the predicted value, and $\rho$ represents the hyperparameter in Huber loss, which is utilized to regulate the impact of outliers on the loss function.

## V. Experiments

### A. Datasets and Baselines

The performance of the TBC-GNODE was evaluated on four widely-used public datasets, namely PeMS03, PeMS04, PeMS07, and PeMS08, which comprise real traffic data collected and curated by California Transportation Performance Measurement System (PeMS) [31]. The specifics of these four datasets are presented in Table I, where each dataset provides traffic flow information at five-minute intervals.




(a) The forecasting framework of TBC-GNODE

(b) The overall structure of TBC-GNODE

(c) The stack architecture of the TBC-GNODE

Fig. 5. The architecture and connection method details of TBC-GNODE

TABLE I
DETAILS OF THE FOUR DATASETS

| 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 |

To evaluate the TBC-GNODE performance of our model, we conduct a comparative study against the following baseline models:

1) DCRNN [22]: Deep Convolutional Recurrent Neural Network, which combines graph convolutional networks and spatio-temporal GRU to capture spatial dependencies and temporal dynamics of traffic data.
2) STGCN [23]: Spatio-Temporal Graph Convolutional Network, which utilizes graph convolutional networks to capture the spatial and temporal patterns in time series data.
3) GraphWaveNet[11]: Graph WaveNet, which introduces adaptive adjacency matrices and utilizes dilated causal convolutions for spatio-temporal feature extraction..
4) STSGCN [9]: Spatio-Temporal Synchronous Graph Convolutional Networks, which is an extension of the STGCN model that utilizes a synchronous strategy to capture the temporal dependencies and spatial correlations in spatio-temporal data by exploiting the information from different times simultaneously.
5) STFGNN [10]: spatio-temporal Fusion Graph Neural Networks, which extends the original adjacency matrix using DTW distance and integrates spatio-temporal information through a gated mechanism.
6) STGODE [27]: Spatio-Temporal Graph ODE Networks, which is an extension of continuous GNNs and fuses TCN for spatio-temporal feature extraction, inherits the advantage of continuous GNNs that can avoid over-smoothing problem.

### *B. Experiment Settings*

To ensure a fair comparison with the baseline model, we partitioned the four baseline datasets into training, validation, and test sets in a ratio of 6:2:2, respectively. We used 12 consecutive historical time steps to predict the traffic flow data for the next 12 time steps. All experiments were conducted on a Linux server with an Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz and an NVIDIA RTX 3090 (24GB) GPU. We applied the Runge-Kutta method [32] in the ODEsolver. The hidden dimensions of the TBC module were set to 32 and 64, with each layer containing three TBC-GNODE blocks. We used three evaluation metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). Lower values for these metrics indicate better model performance in predicting spatio-temporal traffic flow.

### *C. Experiment Results and Analysis*

Table II presents a comparison between the performance of our proposed model and a baseline model. The cells with underscored data indicate where the baseline model achieved superior results and the bold part is the best result in all models.

From the comparison of the baseline model, STSGCN and STFGNN have achieved better results by taking into account spatio-temporal correlation. However, These models use large neighborhood graphs with redundantly connected primitive distances, potentially impacting the GNN's learning ability. Additionally, the discrete aggregation process in GNN restricts its capacity to model high-dimensional and spatio-temporal features, leading to over-smoothing. Therefore, the STGODE model, which decouples the original distance matrix from the semantic adjacency matrix and utilizes GNODE, exhibits superior performance. Nevertheless, the semantic adjacency matrix used in STGODE still represents adjacency relationships discretely, with matrix values limited to 0 or 1. While this approach simplifies the adjacency graph's complexity, it restricts the representation of spatio-temporal dependence. Furthermore, the STGODE model is structured as a "sandwich" [7], with an ODE module placed between two TCN modules. This could constrain ODE's powerful continuous representation ability for sequential data, which is crucial in traffic prediction tasks that rely heavily on the original data distribution.

TABLE II
PERFORMANCE COMPARISON OF TBC-GNODE AND BASELINE MODELS ON PEMS DATASETS

| Datasets | Metric | DCRNN | STGCN | GraphWaveNet | STSGCN | STFGNN | STGODE | TBC-GNODE |
|---|---|---|---|---|---|---|---|---|
| PeMS03 | MAE | 18.18 | 17.49 | 19.85 | 17.48 | 16.91 | 16.50 | **15.74** |
| | MAPE(%) | 18.91 | 17.15 | 19.31 | 16.78 | 16.42 | 16.69 | **14.89** |
| | RMSE | 30.31 | 30.12 | 32.94 | 29.21 | 28.37 | 27.84 | **26.45** |
| PeMS04 | MAE | 24.70 | 22.70 | 25.45 | 21.19 | 20.45 | 20.84 | **19.61** |
| | MAPE(%) | 17.12 | 14.59 | 17.29 | 13.90 | 16.74 | 13.77 | **13.19** |
| | RMSE | 38.12 | 35.55 | 39.70 | 33.65 | 32.49 | 32.82 | **31.52** |
| PeMS07 | MAE | 25.30 | 25.38 | 26.85 | 24.26 | 23.33 | 22.59 | **21.63** |
| | MAPE(%) | 11.66 | 11.08 | 12.12 | 10.21 | 9.15 | 10.14 | **8.78** |
| | RMSE | 38.58 | 38.78 | 42.78 | 39.03 | 36.50 | 37.54 | **35.32** |
| PeMS08 | MAE | 17.86 | 18.02 | 19.13 | 17.13 | 16.89 | 16.81 | **15.77** |
| | MAPE(%) | 11.45 | 11.40 | 12.68 | 10.96 | 10.53 | 10.01 | **9.64** |
| | RMSE | 27.83 | 27.83 | 31.05 | 26.82 | 26.20 | 25.97 | **24.92** |

Regarding MAE, our proposed TBC-GNODE model achieved a reduction of 4.6%, 4.1%, 4.2%, and 6.1% compared to the previously superior model on four datasets with varying regional and complexity characteristics. The other two error evaluation metrics also exhibited reductions of over 4% on average. Thus, TBC-GNODE demonstrated an absolute advantage over the four benchmark datasets.

*D. Ablation Experiments*

In order to validate the effectiveness of each module and the necessity of structural design, we conducted a series of ablation experiments on the PEMS04 and PEMS08 datasets, which exhibit high traffic information complexity. Specifically, we devised four variants of TBC-GNODE as follows:

1) TBC-GNN: We substituted the GNODE module with a general shallow GNN to verify the efficacy of the GNODE design. To ensure fairness, we incorporated the same form of hierarchical residual connection into GNN.
2) TCN-GNODE: We replaced the TBC module with the classic TCN module to test the effectiveness of TBC in extracting subsequence order features.
3) TBC-GNODE-with-Spatial: In addition to the DTW similarity matrix, we introduced the spatial distance matrix as the input graph of the TBC-GNODE module. We took the best result in the experiment to verify the effectiveness of using only DTW distance to represent distribution similarity.
4) TBC-GNODE-Sandwich: We placed the GNODE module in the middle of the model to accept the data characteristics fitted by TCN, following the settings of STSGCN and STGODE. This experiment was designed to verify the necessity of the GNODE module directly processing the raw data.

Table III presents the results of the ablation experiments conducted to evaluate the performance of our proposed model against various sub-models. Among all the sub-models, the performance degradation is most significant when the generic GNN module is used in place of the GNODE module, followed by the model using the 'Sandwich' structure. This suggests that using a suitable GNODE module effectively extracts spatio-temporal dependent representations of the original distribution for forecasting traffic series. The fusion of the original distance and the DTW distance as the adjacency matrix of the GNODE has relatively little impact on the PEMS08 dataset, where the overall oscillation amplitude is more stable. Still, the prediction performance decreases more significantly on the PEMS04 dataset. This indicates that when the oscillation frequency of the traffic is more complex, the redundant connections sometimes cause the node features with dissimilar traffic characteristics to be affected by the original distance correlations, and this effect will be amplified as the complexity of the flow pattern increases. Additionally, when replacing the TBC module, the RMSE error in the prediction results is significantly impacted compared to the MAE. This highlights the superior performance of the TBC module in capturing accurate sequence features, particularly in specific cases or outliers, surpassing the capabilities of the traditional TCN module.

*E. Hyperparameters Analysis*

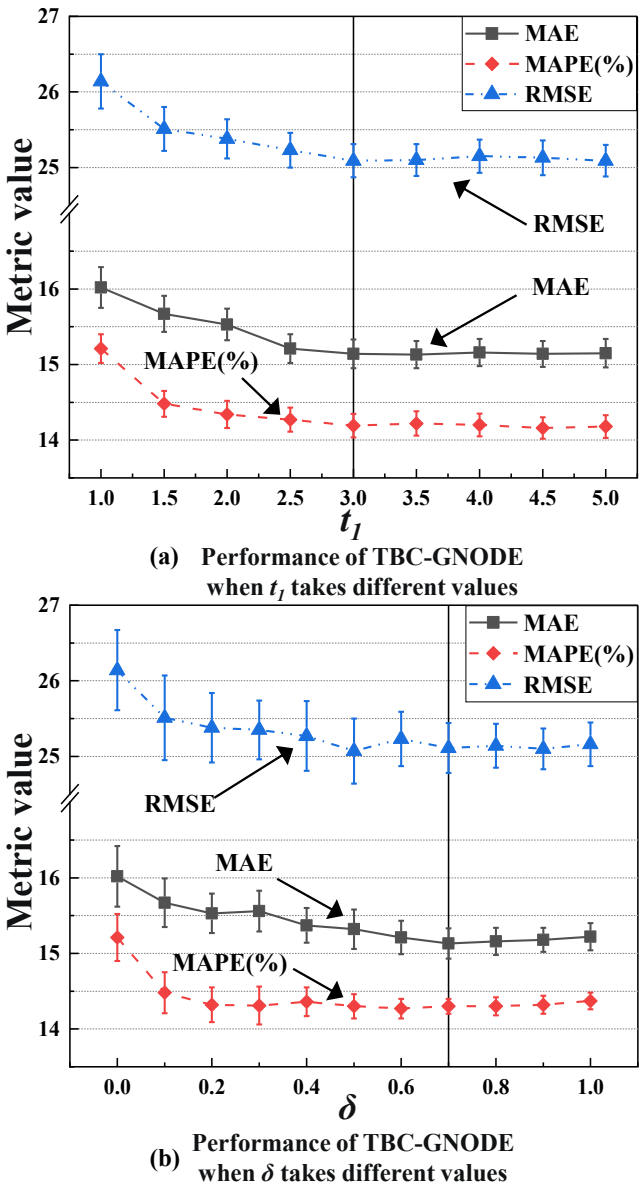
In TBC-GNODE, two critical hyperparameters $t_1$ and $\delta$ require careful consideration. For clarity, the variable $t_1$ represents the number of steps utilized in the neural ODE solver to approximate the solution of the ordinary differential equation. It is solely a numerical count and does not possess any specific physical dimension. Similarly, the variable $\delta$ represents the lower threshold of the adjacency matrix and is a numerical limit without physical units. Both $t_1$ and $\delta$ are dimensionless quantities, and no units will be added to them later in the hyperparameter tuning legend.

The threshold $\delta$ controls the model's tolerance for node similarity. A low $\delta$ may lead to the influence of semantically distant nodes, affecting prediction performance, while a high $\delta$ may hinder the model's ability to capture spatio-temporal dependencies effectively.

The time step $t_1$ used in the ODE solver directly affects the model's fitting ability, with a low $t_1$ limiting learning depth and a high $t_1$ increasing convergence time cost. It should be noted that the first output of the ODE solver is always the initial state of the dynamical system, but in practice, we typically focus more on the solution at subsequent time steps. Hyperparameter tuning can significantly affect the performance of the model on prediction tasks.

TABLE III
ABLATION EXPERIMENTS OF TBC-GNODE

| Datasets | The original model and Variants | MAE | MAPE(%) | RMSE |
|---|---|---|---|---|
| PEMS04 | TBC-GNN | 20.26 | 13.45 | 31.96 |
| | TCN-GNODE | 20.14 | 13.60 | 32.25 |
| | TBC-GNODE-with-Spatial | 19.86 | 13.41 | 32.06 |
| | TBC-GNODE-Sandwich | 20.03 | 13.53 | 32.17 |
| | TBC-GNODE | **19.61** | **13.19** | **31.52** |
| PEMS08 | TBC-GNN | 16.31 | 9.93 | 25.52 |
| | TCN-GNODE | 16.12 | 9.90 | 25.44 |
| | TBC-GNODE-with-Spatial | 15.83 | 9.81 | 25.31 |
| | TBC-GNODE-Sandwich | 15.95 | 9.86 | 25.36 |
| | TBC-GNODE | **15.77** | **9.64** | **24.92** |




(a) Performance of TBC-GNODE when $t_1$ takes different values

(b) Performance of TBC-GNODE when $\delta$ takes different values

Fig. 6. Performance of different hyperparameters

Fig. 6 presents the performance evaluation of our model with different choices of hyperparameters using the PEMS03 dataset. The solid vertical line in the figure represents the performance of our final selection of hyperparameters and the short vertical line segments at each value point are error bars for the results of 10 times of experiments. As depicted in Fig. 6(a), the experimental results confirm the stabilization of the prediction error when the time step $t_1$ of the ODE solver is set to a value greater than or equal to 3. This finding provides further evidence of the reliability and consistency of GNODE, showcasing its ability to yield stable predictions for the given task.

Fig. 6(b) reveals a significant trend: as the adjacency matrix threshold $\delta$ increases to 0.7, the error bar gradually converges. What excites us is the consistent demonstration of good stationarity in the experimental results, even when the threshold $\delta$ exceeds 0.7 We could attribute this achievement to our adjacency matrix construction method, which operates independently of the prior spatial structure, effectively mitigating the negative impact caused by certain realistic distance relations. Furthermore, the adaptive learning strategy for the matrix allows automatic adjustment of adjacent weights as semantic adjacent edges gradually increase. This crucial feature ensures the stability of prediction results as the model dynamically adapts to evolving semantic associations. Overall, these findings highlight the effectiveness and robustness of our proposed approach in handling spatio-temporal traffic flow forecasting tasks.

## VI. Conclusion

In spatio-temporal prediction tasks, current methods mainly rely on increasing the scale and dimension of the adjacency graph to capture spatial dependencies between nodes while applying one-dimensional TCN module to extract temporal dependencies. However, there is limited consideration of the structural refinement of the adjacency graph and the composition of historical temporal patterns. This paper proposes a new spatio-temporal traffic flow forecasting model, TBC-GNODE based on Graph Neural ODE. The model leverages a refined semantic adjacency graph to represent the complete semantic association of spatio-temporal data, mitigating the impact of redundant connection noise on the spatio-temporal model. Moreover, the model reanalyzes the historical data composition of the time series from a new perspective, utilizing a temporal branch convolution to fuse the deep historical distribution and future trend distribution in the established data. Additionally, a parallel spatio-temporal dependence extraction architecture without upstream modules is designed, enabling the acquisition of long-term spatio-temporal dependence by stacking the overall module. The comparison experiments demonstrate that TBC-GNODE outperforms the optimal baseline model by reducing the prediction error by over 4%. Finally, the detailed additional experiments further validate the effectiveness and stability of the proposed model.

In general, Our research is diligently focused on delivering a powerful solution idea and framework for spatio-temporal traffic flow prediction. The experiments demonstrate that the prior structure and distance of the traffic network are not necessary conditions for accurate predictions. By employing appropriate spatio-temporal feature extraction methods, we can even achieve superior models without relying on prior spatial information.

However, in the long run, the performance of traffic flow prediction may depend on various related factors, such as further optimization of Neural ODE and graph neural network technology, the expression of space-time dependence, and the extraction of global time trend features, among others. These factors can directly or indirectly enhance prediction accuracy. In future work, we will continue to devote ourselves to further research on traffic spatio-temporal prediction, contributing innovative thinking and new progress to the field of advanced intelligent transportation.

## References

[1] H. Li, X. J. Li, L. C. Su, D. Jin, J. B. Huang, and D. S. Huang, "Deep Spatio-temporal Adaptive 3D Convolutional Neural Networks for Traffic Flow Prediction," *ACM Transactions on Intelligent Systems and Technology*, vol.13, no.2, pp. 1-21, 2022.

[2] H. Q. Wang, R. Q. Zhang, X. Cheng, and L. Q. Yang, "Hierarchical Traffic Flow Prediction Based on Spatial-Temporal Graph Convolutional Network," *IEEE Transactions on Intelligent Transportation Systems*, vol.23, no.9, pp. 16137-16147, 2022.

[3] H. Zeng, Z. Y. Peng, X. H. Huang, Y. X. Yang, and R. Hu, "Deep spatio-temporal neural network based on interactive attention for traffic flow prediction," *Applied Intelligence*, vol.52, no.9, pp. 10285-10296, 2022.

[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] H. Hafidi, P. Ciblat, M. Ghogho, and A. Swami, "Graph-Assisted Bayesian Node Classifiers," *IEEE Access*, vol.11, pp. 23989-24002, 2023.

[6] C. Huang, and Y. Zhong, "A Network Representation Learning Method Fusing Multi-dimensional Classification Information of Nodes," *IAENG International Journal of Computer Science*, vol.50, no.1, pp. 94-105, 2023.

[7] J. Skarding, M. Hellmich, B. Gabrys, and K. Musial, "A Robust Comparative Analysis of Graph Neural Networks on Dynamic Link Prediction," *IEEE Access*, vol.10, pp. 64146-64160, 2022.

[8] A. A. Liu, H. S. Tian, N. Xu, W. Z. Nie, Y. D. Zhang, and M. Kankanhalli, "Toward Region-Aware Attention Learning for Scene Graph Generation," *IEEE Transactions on Neural Networks and Learning Systems*, vol.33, no.12, pp. 7655-7666, 2022.

[9] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," *Proceedings of the AAAI conference on artificial intelligence*, pp. 914-921, 2020

[10] M. Li, and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," *Proceedings of the AAAI conference on artificial intelligence*, pp. 4189-4196, 2021

[11] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for Deep Spatial-Temporal Graph Modeling," *The 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1907-1913, 2019

[12] H. Sen, "Time Series Prediction based on Improved Deep Learning," *IAENG International Journal of Computer Science*, vol.49, no.4, pp. 1133-1138, 2022.

[13] J. A. Adisa, S. Ojo, P. A. Owolawi, A. Pretorius, and S. O. Ojo, "Application of an Improved Optimization Using Learning Strategies and Long Short Term-Memory for Bankruptcy Prediction," *IAENG International Journal of Computer Science*, vol.50, no.2, pp. 512-524, 2023.

[14] D. Zhao, F. Liu, and X. Wang, "Bearing Remaining Useful Life Estimation Based on Encoder and Gated Recurrent Units," *IAENG International Journal of Computer Science*, vol.49, no.1, pp. 140-148, 2022.

[15] H. Li, H. Ge, H. Yang, J. Yan, and Y. Sang, "An Abnormal Traffic Detection Model Combined BiIndRNN With Global Attention," *IEEE Access*, vol.10, pp. 30899-30912, 2022.

[16] Z. Zeng, C. Liu, Z. Tang, K. Li, and K. Li, "AccTFM: An Effective Intra-Layer Model Parallelization Strategy for Training Large-Scale Transformer-Based Models," *IEEE Transactions on Parallel and Distributed Systems*, vol.33, no.12, pp. 4326-4338, 2022.

[17] 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

[18] M. Harker, and P. O'leary, "Sylvester Equations and the numerical solution of partial fractional differential equations," *Journal of Computational Physics*, vol.293, pp. 370-384, 2015.

[19] M. Behr, P. Benner, and J. Heiland, "Solution formulas for differential Sylvester and Lyapunov equations," *Calcolo*, vol.56, no.4, 2019.

[20] K. Oono, and T. Suzuki, "Graph Neural Networks Exponentially Lose Expressive Power for Node Classification," *8th International Conference on Learning Representations, (ICLR) 2020*, pp. 156-193, 2020

[21] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, pp. 472-480, 2017

[22] 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

[23] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3634-3640, 2018

[24] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *Advances in Neural Information Processing Systems*, vol.31, no.1, pp. 6571–6583, 2018.

[25] S. Kim, W. Q. Ji, S. L. Deng, Y. B. Ma, and C. Rackauckas, "Stiff neural ordinary differential equations," *Chaos*, vol.31, no.9, 2021.

[26] 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

[27] Z. Fang, Q. Long, G. Song, and K. Xie, "Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting," *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 364–373, 2021

[28] 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.

[29] W. Z. Wang, and X. W. Liu, "Intuitionistic Fuzzy Information Aggregation Using Einstein Operations," *IEEE Transactions on Fuzzy Systems*, vol.20, no.5, pp. 923-938, 2012.

[30] P. J. Huber. *Robust Estimation of a Location Parameter*. New York, NY: Springer New York, 1992.

[31] C. Chen. *Freeway Performance Measurement System (PeMS)*. Berkeley, CA, USA: University of California, Berkeley, 2002, ch. 1.

[32] W. W. Hager, "Runge-Kutta methods in optimal control and the transformed adjoint system," *Numerische Mathematik*, vol.87, no.2, pp. 247-282, 2000.