



Research article

Reconstructed graph spatio-temporal stochastic controlled differential equation for traffic flow forecasting

Ruxin Xue¹, Jinggui Huang^{2,*}, Zaitang Huang¹ and Bingyan Li¹

¹ School of Mathematics and Statistics, Nanning Normal University, Nanning 530100, China

² Guangxi University of Finance and Economics, China-Asean Institute of Statistics, Nanning 530003, China

* **Correspondence:** Email: HJG88245675@gxufe.edu.cn.

Abstract: Spatio-temporal graph data have been widely applied in traffic flow prediction tasks. Traditional methods often combine graph convolutional networks with recurrent neural networks based on the original graph structure. However, these methods typically struggle with issues such as incomplete sensor distributions and unresolved potential dependencies between different parallel intersections. Moreover, the model structure often fails to capture the noise inherent in traffic predictions, which becomes a significant barrier to accurate forecasting. To address these challenges, we proposed a novel approach that diverges from traditional traffic flow models, which rely on predefined graph structures. Instead, our method uncovers hidden edge connectivity and integrates the data into a unified framework, enabling the model to better capture the underlying spatial relationships. Specifically, we introduced a new framework that combines neural control differential equations with stochastic differential equations to enhance spatio-temporal graph modeling. This integration improves the model's capacity to capture complex spatio-temporal patterns, thereby enhancing the accuracy of prediction tasks. Extensive experiments conducted on benchmark datasets validate the effectiveness of our method.

Keywords: spatio-temporal graph modeling; latent edges; spatio-temporal prediction; graph structure reconstruction; differential equations

1. Introduction

In the era of accelerated digitalization and urbanization, transportation systems are facing unprecedented challenges and opportunities. Efficient and accurate traffic flow prediction plays a crucial role in intelligent transportation systems. It not only helps alleviate urban congestion and enhance traffic safety but also optimizes the allocation of transportation resources, providing strong support for the sustainable development of cities [1–3]. Spatio-temporal graph data, with their powerful modeling capabilities,

have significantly promoted the development of prediction technologies in various fields such as traffic flow prediction, climate modeling, social network analysis, and epidemic spread modeling. In the traffic flow prediction scenario, spatio-temporal graphs can effectively model the spatial relationships between road segments and the temporal dynamics of traffic patterns, thus accurately predicting congestion and traffic flow changes, as confirmed in relevant studies [4–6].

Past research on traffic flow prediction has been fruitful. Traditional models such as the autoregressive integrated moving average model and the Kalman filter model were widely used for a period due to their well-established theories and easy implementation. These models are based on certain linear assumptions and can achieve satisfactory prediction results when the traffic flow is relatively stable and its change pattern is relatively obvious. With the rise of machine learning techniques, models such as support vector machines (SVMs) and random forests have also been introduced into traffic flow prediction. They can handle more complex nonlinear relationships and improve the prediction accuracy to some extent [7,8].

In recent years, deep learning techniques have shown great potential in the field of traffic flow prediction. Deep learning models represented by convolutional neural networks, recurrent neural networks, and their variants (such as long short-term memory (LSTM) networks and gated recurrent units (GRUs)) can automatically learn complex spatio-temporal features from large amounts of traffic data and have achieved better prediction performance than traditional models on multiple public datasets. Moreover, the inherent structure of graph data gives them certain natural advantages and such data are widely used. Several methods for processing spatio-temporal graph data have been developed. Spatio-temporal graph convolutional networks are commonly used to jointly model spatial and temporal dependencies [9]. Graph convolutional networks are used to capture the initial relationships, while long short-term memory networks handle time series [10]. Techniques for dynamic or time-varying graph data, such as dynamic graph convolutional networks or dynamically constructed graph neural networks, are also used to address the evolutionary nature of spatio-temporal relationships [11].

Currently, most studies focus more on the exploration of embedding vectors in the graph neural network prediction process. MSTDFGRN (multi-view spatio-temporal dynamic fusion graph recurrent network) introduces a multi-view spatial convolution module to dynamically fuse static graphs and adaptive graphs in multiple subspaces to learn the intrinsic and potential spatial dependencies of nodes [12]. SSGCRTN (space-specific graph convolutional recurrent transformer network) introduces a spatio-temporal interaction module and a transformer-based global time fusion module to capture global spatio-temporal correlations [13]. SDSINet (spatio-temporal dual-scale interaction network) enhances temporal feature modeling via implicit temporal identity embedding, reducing the computation costs while capturing global dependencies [14]. Regarding the definition of graph structure, PSTCGCN (principal spatio-temporal causal graph convolutional network) proposes a principal spatio-temporal causal graph convolution and introduces a data-driven method for generating graph embeddings to reconstruct the drifted data through adaptive transformation [15]. DHHNN (dynamic hypergraph hyperbolic neural network) dynamically updates the hyper-graph structure through node clustering and distance calculation, removes redundant edges, and applies a dynamic reconstruction mechanism to enhance the model's ability to model complex network structures [16]. DDMGCRN (decomposition dynamic multi-graph convolutional recurrent network) introduces sensor-specific spatial identity embeddings and timestamp embeddings to construct dynamic graphs and further integrates static graphs for multi-graph fusion [17]. STSGCN (spatio-temporal synchronous graph convolutional networks) considers the time

steps in the spatial graph, integrates multiple adjacency matrices and self-connection matrices into a large matrix, and extracts spatio-temporal correlations simultaneously through the graph convolution module [18].

In the existing frameworks, spatial and temporal dependencies are typically modeled independently using distinct mechanisms. For example, graph attention networks are adept at capturing spatial dependencies [19]. However, they also face the following challenges: for some irregular time-series data in static graphs, fixed-length uniform sampling of time steps is required, and their ability to simulate complex spatio-temporal interactions is limited. There is a possibility of gradient vanishing in long-term time-series data, and there are issues in handling dynamic graph topologies as well as problems with the stability and gradients of long-term dependency modeling. In 2022, a breakthrough concept for processing sequence data was proposed, which designed two neural controlled differential equations, one for time processing and the other for spatial processing. Its improvement lies in the ability to directly model continuous time series without interpolation. Moreover, the Lipschitz continuity condition ensures the stability of the solution, and the support of differential equation theory further alleviates the black-box nature. Neural differential equations and neural controlled differential equations have been introduced to model continuous hidden states, offering a robust solution for handling irregular spatio-temporal data in real-world scenarios [20], as shown in Figure 1.

Therefore, it is necessary to consider the spatio-temporal fusion of differential equations in traffic flow prediction. Currently, graph neural networks with differential equation modeling either focus on the construction and improvement of the neural network structure or on the combination of equations and graph structures, and there has not been much exploration of spatio-temporal fusion from the perspective of equations. To address this shortcoming, this paper improves the fusion method of spatio-temporal differential equations. Instead of using traditional ordinary differential equations in the spatial dimension, a neural stochastic differential equation model is proposed to capture more uncertainties. Since the transportation system itself has a high degree of uncertainty, including the randomness of vehicle arrival times and the fluctuations in road capacity, traditional models may ignore these uncertainty factors, resulting in a large deviation between the predicted results and the actual situation. The neural stochastic differential equation model can effectively capture these uncertainties and provide more reliable results for traffic flow prediction.

In addition to improving the basic settings of the graph neural traffic flow prediction differential equation, this paper further directly addresses the pain points in traffic prediction: the mining and addition of hidden edges in the basic graph structure. The graph neural networks usually deal with sparse adjacency matrices, which are only based on the number of sensors set in the actual road network. If only the relationships between directly connected nodes are considered, the potential connections between two roads that are not directly connected in space may be ignored. For example, roads that are not directly connected may show similar traffic flows during the same time period. Currently, most of the research on the reconstruction of adjacency matrices focuses on discussing the topological structure of the graph using equations such as the clustering coefficient based on the original connection relationships. To solve this problem, while ensuring the existence of actual edges, we no longer make judgments based on the traditional local clustering coefficient. Instead, a quantitative method is introduced to identify the hidden edges between nodes. The heterogeneity and dependence of nodes are explored through local full connections to determine the number of these added edges, and the positions of the added edges are further determined on the basis of the Pearson correlation coefficient.

The innovations of this study are mainly reflected in the following aspects. A new graph neural controlled traffic flow prediction differential equation is proposed. A random perturbation term is added in the spatial dimension to capture cognitive uncertainties, and the excellent characteristics of spatio-temporal fusion are retained. The improved spatio-temporal equation is used for subsequent convolutional feature extraction and prediction. On the other hand, considering the uneven distribution, insufficiency, and irrationality of sensors in traffic flow prediction, as well as the potential influence of parallel roads, we start from the original graph structure and mine the potential connections between nodes. Hidden edges are added while ensuring the actual connections, and the edge information is fully exploited. Through this study, we expect to solve the problems of existing traffic flow prediction models in handling complex dynamics, their lack of interpretability, and in ignoring graph structure connections.

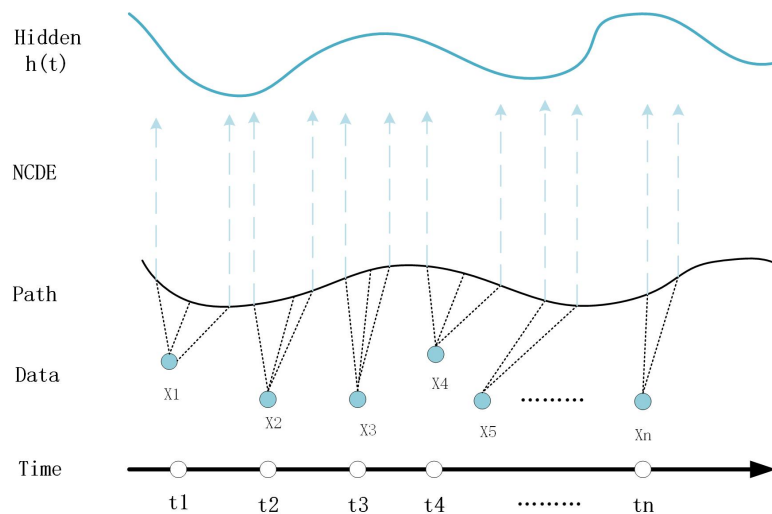


Figure 1. Neural controlled differential equations for processing time-series.

2. Analysis

2.1. Methods and analysis

This paper investigates spatio-temporal graph data with a fixed topological structure, a common assumption in traffic flow prediction, transportation networks, social networks, and financial networks. For example, in transportation, road network topologies typically remain static. Similarly, the relationships in social networks (such as friendships) or financial networks (transaction relationships between institutions) are often assumed to be stable over the short term.

However, directly applying a fixed road network topology for modeling and analysis does not fully leverage the spatial information. Graph neural networks inherently assume homogeneity, where only nodes with similar features or identical labels are connected in the graph [21, 22]. This assumption is particularly prominent in traffic flow prediction, where a fixed graph structure is typically used to make forecasts. The problem arises when this fixed graph structure is not accurate. Traffic flow graphs are usually based on data collection systems and sensor deployments, which are limited by resources. As a result, comprehensive sensor deployment cannot be guaranteed, and some connections may be missing between nodes that are related in terms of connectivity. This leads to sparse adjacency matrices

in datasets like PEMS4 and PEMS8, where some nodes are not directly connected.

In reconstructing adjacency matrices, methods such as incorporating the local clustering coefficient equation (2.1) are based on the original graph structure [23], where e_i represents the number of edges between the neighbors of node i and k_i is the degree of the node. The denominator $k_i(k_i - 1)/2$ represents the maximum possible number of edges between neighbors. But they do not necessarily reflect real-world relationships between nodes.

$$C_i = \frac{2e_i}{k_i(k_i - 1)}. \quad (2.1)$$

Currently, no method has been developed to directly extract fixed implicit nodes for traffic flow graph data structures. Our approach involves reconstructing the road network graph structure with an assumption of a fully connected graph. This is followed by clustering and restructuring the graph through spatial heterogeneity analysis. By incorporating implicit node connections into the nodes' adjacency relationships, we improve the model's accuracy. The selection of additional nodes is guided by Pearson's correlation coefficient, helping uncover hidden connectivity relationships between roads.

According to the spatial heterogeneity, let n_i be the total number of nodes in a subregion and let m_i be the aggregation number. Assuming that the size of each class remains within a certain range, we can reasonably infer that within the subregion, at least $\lfloor \frac{n_i}{m_i} \rfloor$ nodes are closely connected. Therefore, we can assume that these $\lfloor \frac{n_i}{m_i} \rfloor$ nodes within each subregion are fully connected. To avoid model overfitting or excessive computational resource usage, we introduce a threshold based on the actual connection density, restricting the applicability of the full connection assumption.

Moreover, the original actual connections should remain unchanged, and sparsity constraints must be preserved. Specifically, for a node n_k within its subregion, if the actual connection point is s_k , it must hold that $s_k < \lfloor \frac{n_i}{m_i} + 1 \rfloor$. The number of mined hidden edges is given by $\lfloor \frac{n_i}{m_i} \rfloor - s_k$.

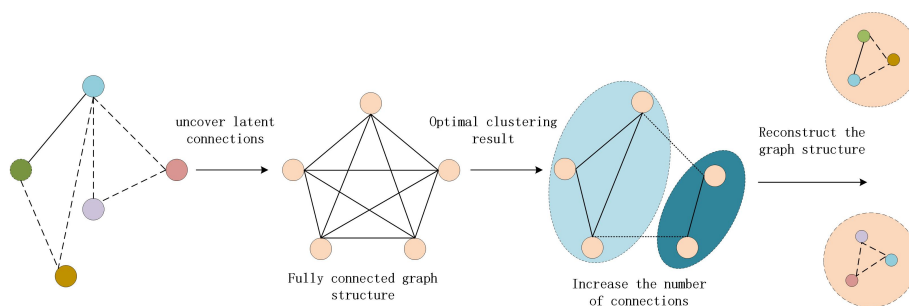


Figure 2. Graph reconstruction and clustering process.

This process is depicted in the Figure 2, showcasing the steps involved in improving graph data processing by revealing the latent connections, enhancing connections, performing clustering optimization, and reconstructing the structure. The objective is to optimize the connections and structure of the graph to enhance its representational power.

In the initial state, the connections between nodes are incomplete. For instance, the green, blue, purple, and red nodes are connected by dashed lines, indicating latent connections that are yet to be defined in the actual graph structure. These connections are typically introduced to strengthen the graph's structure and improve information flow. Clustering analysis is then applied to group the

nodes, with the goal of finding the optimal clustering partition, ensuring that the nodes within each cluster exhibit stronger connectivity. Ultimately, a new graph structure is generated by rearranging the connections. This new structure incorporates not only the original node connections but also the enhanced connections and clustering information, making the graph better suited for subsequent analysis or processing. Once the graph structure is updated (with the addition of hidden edges), the input features of the nodes are dynamically updated over time, accurately reflecting the evolving nature of the real system.

2.2. Assumptions and definitions

In traffic flow prediction tasks, traffic networks can be effectively modeled using graph structures to capture spatial dependencies and temporal dynamics. This paper proposes a novel modeling method based on temporal graph structures, which integrates graph convolutional networks with temporal modeling techniques to achieve accurate predictions of future traffic flow.

- $\{G_{t_i} = (\mathcal{V}, \mathcal{E}, \mathbf{F}_{t_i})\}_{i=0}^N$: The set of nodes and the set of edges in the graph structure are fixed to represent the topology of the traffic network, which is mathematically represented by the adjacency matrix or the graph Laplacian matrix. The node features change over time to reflect the dynamic characteristics of the traffic's state, such as fluctuations in the volume of traffic flow and speed.
- \mathcal{V} : This represents the set of nodes in the traffic network. Each node corresponds to a specific geographic location, such as a traffic intersection or a sensor location.
- \mathcal{E} : This represents the set of edges in the traffic network. Each edge defines a physical connection, such as a road, or other dependencies, such as neighborhood relationships between nodes.
- $\mathbf{F}_{t_i} \in \mathbb{R}^{|\mathcal{V}| \times D}$: This represents the node feature matrix at time t_i , containing D -dimensional feature vectors for each node.

Such an approach enhances the ability to model long-term dependencies effectively. For instance, in transportation networks, the road layout remains static, and in electricity networks, the connections between power stations (nodes) and transmission lines (edges) are predefined. Similarly, this assumption applies to social networks, where friendship relationships between users remain stable in the short term, or to financial networks, where transaction relationships between banks or companies can be considered fixed. This assumption also extends to meteorological networks, where weather stations' locations and their interconnections are geographically stable.

Traditional continuous dynamic equation methods embed discrete time-series inputs as continuous dynamic processes, as illustrated in Figure 1. While these methods excel in processing time-series data, they do not inherently model spatial information in graphs. Additional modules such as graph attention networks are often employed to capture spatial correlations. However, this modular approach can weaken the integration of spatial information with spatio-temporal relationships, thereby reducing the overall coherence and effectiveness of the model.

To address these challenges, this paper proposes an innovative framework that further explores hidden edges by analyzing a large volume of node feature data, enriching spatial representations (as shown in Figure 2). Meanwhile, a spatio-temporal modeling framework is constructed, as illustrated in Figure 3. Notably, the constructed spatial function retains the spatial adjacency matrix in its formulation, ensuring that the spatial relationships are consistently integrated. Temporal dynamics are captured by obtaining the temporal hidden states of the nodes through neural controlled differential equations, which

changes in the graph's spatial topology and the presence of randomness or uncertainty in the spatial relationships. The inclusion of random diffusion terms in the stochastic differential equation (SDE) smooths spatial processing and enhances the model's robustness to noise. This innovation tightly couples temporal and spatial modeling, improving the overall stability and performance of the model.

Finally, the integration of neural controlled differential equations (NCDE) and SDE provides a unified spatio-temporal modeling framework, which not only captures the spatio-temporal information more effectively but also handles uncertainty in the spatial relationships, thereby further improving the model's performance and robustness in complex data environments.

3. Process

3.1. Adaptive adjacency matrix construction

Within a reasonably defined region, we assume that the initial adjacency matrix is fully connected, meaning that all nodes have the potential to be connected. This assumption helps to capture both global and local patterns, allowing us to extract key regional features and hidden correlations between roads. However, directly using a fully connected adjacency matrix increases the computational complexity and introduces redundant information. Therefore, we apply a data-driven approach to optimize the adjacency matrix, ensuring that it retains meaningful road connections while reducing unnecessary links.

Clustering methods are applied under the assumption of a fully connected graph structure to ensure the identification of roads with similar traffic patterns. To determine the optimal number of clusters n^* , we use the silhouette score, which is calculated as:

$$S(n) = \frac{1}{N} \sum_{i=1}^N \frac{b_i - a_i}{\max(a_i, b_i)}, \quad (3.1)$$

where a_i represents the average intra-cluster distance of a node, calculated as

$$a_i = \frac{1}{|C_k| - 1} \sum_{j \in C_k, j \neq i} d(X_i, X_j), \quad (3.2)$$

and b_i represents the minimum average inter-cluster distance to another cluster:

$$b_i = \min_{C_m, m \neq k} \frac{1}{|C_m|} \sum_{j \in C_m} d(X_i, X_j). \quad (3.3)$$

To ensure a balanced cluster size, we introduce a minimum node constraint: and require that each cluster satisfies

$$|C_k| \geq N_{\min}, \quad \forall k = 1, 2, \dots, n. \quad (3.4)$$

The optimal cluster number n^* is then selected by maximizing the silhouette score while ensuring that the minimum cluster size constraint is met

$$n^* = \arg \max_n S(n), \quad \text{subject to } |C_k| \geq N_{\min}, \forall k. \quad (3.5)$$

As the first step in optimizing the adjacency matrix, we introduce Pearson's correlation analysis to measure the similarity between road traffic patterns. The Pearson correlation coefficient between two nodes is given by

$$r_{ij} = \frac{\sum_t (X_i(t) - \bar{X}_i)(X_j(t) - \bar{X}_j)}{\sqrt{\sum_t (X_i(t) - \bar{X}_i)^2} \sqrt{\sum_t (X_j(t) - \bar{X}_j)^2}}. \quad (3.6)$$

After obtaining the optimal cluster number n^* , we compute the number of additional edges each node needs to establish. The number of additional edges for a node is given by

$$\Delta d_i = \left\lfloor \frac{N}{n^*} \right\rfloor - d_i, \quad (3.7)$$

where d_i represents the current number of connections of node i , and $\left\lfloor \frac{N}{n^*} \right\rfloor$ represents the expected average number of connections each node should have under the optimal clustering number n^* . The additional edges are determined on the basis of the Pearson correlation coefficients r_{ij} , which are sorted in descending order.

Define and select the top edges: First, sort all possible edges by the Pearson correlation coefficient r_{ij} in descending order. The sorted edge set is denoted as E_{sorted} as follows:

$$E_{\text{sorted}} = \text{sort}(E, \text{by } r_{ij}, \text{in descending order}), \quad (3.8)$$

where E is the set of all possible edges, and r_{ij} is the Pearson correlation coefficient between nodes i and j .

Select the top Δd_i edges from E_{sorted} : From the sorted edge set E_{sorted} , choose the top Δd_i edges, i.e., the Δd_i edges with the highest Pearson correlation values, to fulfill the additional edge requirement for node i .

$$E_{\text{selected}} = \{e_1, e_2, \dots, e_{\Delta d_i}\} \quad \text{where} \quad e_k \in E_{\text{sorted}}, \forall k \in [1, \Delta d_i]. \quad (3.9)$$

After clustering and Pearson correlation-based optimization, we construct the following fixed adjacency matrix:

$$A_{\text{fixed},ij}(n^*) = \begin{cases} r_{ij}, & \text{if } (i, j) \in E_{\text{selected}} \\ 0, & \text{otherwise.} \end{cases} \quad (3.10)$$

In addition to the fixed adjacency matrix, we introduce node embedding learning to compute a data-driven adaptive adjacency matrix, which is formulated as

$$A_{\text{adaptive}} = \text{Softmax}(\sigma(EE^T)). \quad (3.11)$$

where E represents the node embedding matrix, and $\sigma(\cdot)$ is an activation function such as relu (rectified linear unit) or sigmoid. The Softmax function is applied to normalize the adjacency matrix and ensure stability.

The final adjacency matrix is computed as a weighted combination of the fixed and adaptive adjacency matrices as follows:

$$GS O_{\text{final}} = \beta A_{\text{fixed}}(n^*) + (1 - \beta) A_{\text{adaptive}}, \quad (3.12)$$

where β is a trainable parameter that controls the tradeoff between physical topology optimization (based on Pearson's correlation) and data-driven learning (based on node embeddings). The fusion coefficient β is constrained within $\beta \in [0, 1]$.

3.2. Neural controlled differential equations

NCDEs parameterize the control function $f(\cdot)$ through neural networks, enabling dynamic adjustments to hidden states based on input paths. This capability allows NCDEs to effectively accommodate irregular sampling and partial observations, which are common in real-world data. By leveraging neural networks to parameterize the control function, the model's expressive power is significantly enhanced.

The mathematical formulation of NCDEs is as follows

$$\mathbf{h}(T) = \mathbf{h}(0) + \int_0^T f(\mathbf{h}(t); \boldsymbol{\theta}) d\mathbf{X}(t). \quad (3.13)$$

Equation (3.13) clarifies its dynamic evolution mechanism, and we can expand Eq (3.13) for representation.

$$\mathbf{h}(T) = \mathbf{h}(0) + \int_0^T f(\mathbf{h}(t); \boldsymbol{\theta}_f) \frac{d\mathbf{X}(t)}{dt} dt. \quad (3.14)$$

Equation (3.14) expresses the rate of change $\frac{d\mathbf{X}(t)}{dt}$ of the control path $\mathbf{X}(t)$ explicitly. It reveals that the update of the hidden state $\mathbf{h}(t)$ is not only affected by the current state $\mathbf{h}(0)$ and the control function f , but is also directly driven by the rate of change of the control path $\mathbf{X}(t)$.

3.3. Stochastic differential equations

In spatio-temporal prediction tasks, future data distributions cannot be predicted with complete accuracy solely on the basis of historical data. The discrepancy between predicted values and actual outcomes primarily arises from the randomness inherent in real-world systems. To address this, randomness is incorporated into the modeling process to reduce prediction error. At the same time step, some predicted values may exceed the true values, while others may fall below them. To mitigate this, noisy data with spatial correlation are randomly generated, allowing the predicted values to align more closely with the true values.

SDEs combine deterministic and stochastic terms to describe the dynamic behavior of a system. To integrate the temporal and spatial dynamics, a global temporal dynamics matrix $\mathbf{H}(t)$ is employed. This matrix encodes hidden state information in the temporal dimension, serving as input for subsequent modeling of spatial dynamics. This setup effectively separates and interacts between temporal and spatial dimensions while capturing the randomness of real-world systems through the stochastic term

$$\mathbf{Z}(T) = \mathbf{Z}(0) + \int_0^T g(\mathbf{Z}(t); \boldsymbol{\theta}_g) d\mathbf{H}(t) + \int_0^T \mathbf{W}(\mathbf{Z}(t); \boldsymbol{\theta}_g) d\mathbf{W}(t). \quad (3.15)$$

In this context, g and \mathbf{W} represent the drift and diffusion terms, which can be any arbitrary continuous functions. The term $d\mathbf{W}(t)$ denotes a Wiener process, where

$$d\mathbf{W}(t) = \boldsymbol{\varepsilon} \cdot \sqrt{dt}, \quad \boldsymbol{\varepsilon} \sim N(0, 1). \quad (3.16)$$

To enable the stochastic differential equations to generate stochastic data with spatial characteristics, the ability to capture spatial information must be incorporated into the SDEs. Therefore, we introduce graph neural networks as the drift term $g(\mathbf{Z}(t); \boldsymbol{\theta}_g)$.

In spatio-temporal prediction tasks, randomness often impacts multiple nodes simultaneously. For example, an increase in data at one node may result in a decrease at another due to the underlying

dependencies. Such spatially correlated randomness is modeled using spatial correlations extracted from graph neural networks. Our neural SDE efficiently captures these spatio-temporal correlations, making it highly suitable for real-world prediction tasks.

3.4. Graph neural controlled space-time fusion equation

3.4.1. Time processing

The first NCDE used for time processing can be written as follows:

$$\mathbf{H}(T) = \mathbf{H}(0) + \int_0^T f(\mathbf{H}(t); \theta_f) \frac{d\mathbf{X}(t)}{dt} dt. \quad (3.17)$$

Define a matrix $\mathbf{H}(t) \in \mathbb{R}^{|\mathcal{V}| \times \dim(\mathbf{h}(\mathbf{v}))}$, where $\mathbf{h}(\mathbf{v})(t)$ is the hidden trajectory of the temporal information of node \mathbf{v} (at time $t \in [0, T]$). Thus, the trajectory formed by $\mathbf{H}(t)$ at time t contains the hidden information of temporal processing results.

3.4.2. Spatial process

After that, we start the SDE with the following spatial process:

$$\mathbf{Z}(T) = \mathbf{Z}(0) + \int_0^T g(\mathbf{Z}(t); \theta_g; \text{GSO}_{final}) d\mathbf{H}(t) + \int_0^T \mathbf{W}(\mathbf{Z}(t); \theta_g) d\mathbf{W}(t). \quad (3.18)$$

The hidden trajectory $\mathbf{Z}(t)$ is controlled by $\mathbf{H}(t)$, which is generated by the temporal processing module and subsequently used in the spatial processing module. $\mathbf{H}(t)$ injects temporal information into the spatial processing module, enabling spatial aggregation to account for the temporal context simultaneously. This integration allows the model to capture spatio-temporal interactions with greater accuracy.

The stochastic differential equation effectively captures the random noise and uncertainty within the system by introducing a random noise term. This addition enhances the model's robustness to unknown data and improves its generalization ability, making it well-suited for complex dynamic systems.

By combining Eqs (3.17) and (3.18), the following single equation is derived, incorporating both temporal and spatial processing:

$$\mathbf{Z}(T) = \mathbf{Z}(0) + \int_0^T g(\mathbf{Z}(t); \theta_g; \text{GSO}_{final}) f(\mathbf{H}(t); \theta_f) \frac{d\mathbf{X}(t)}{dt} dt + \int_0^T \mathbf{W}(\mathbf{Z}(t); \theta_g) d\mathbf{W}(t). \quad (3.19)$$

$\mathbf{Z}(t) \in \mathbb{R}^{|\mathcal{V}| \times \dim(\mathbf{Z}(\mathcal{V}))}$ denotes the matrix formed by superimposing the hidden trajectories $\mathbf{Z}(\mathcal{V})$ of all nodes \mathcal{V} . This matrix is generated by combining the hidden trajectories of all nodes \mathcal{V} with those of their neighboring nodes. In this equation, the hidden trajectories $\mathbf{Z}(\mathcal{V})$ are determined by integrating the trajectories of the nodes themselves and their neighbors.

Unlike traditional methods that rely on normalized adaptive adjacency matrices, the settings of graph signal operator parameters integrate both the latent graph structures and adaptive components, providing the graph with “prior” knowledge. This enables the model to not only adapt to changes in the graph but also to leverage hidden relationships (edges that are not explicitly present). This is a significant innovation, as traditional adaptive adjacency matrix methods typically compute the graph structure on the basis of node similarities, overlooking hidden relationships that may be critical to the model's

performance. The introduction of GSO_{final} greatly enhances the model's capacity to capture complex and latent relationships within the graph, which is crucial for modeling real-world dynamic systems, such as traffic flow. In the context of traffic flow prediction, the incorporation of GSO_{final} enables the model to better capture the latent dependencies between different road segments, even when these dependencies are not explicitly modeled by traditional graph structures. Traffic flow often exhibits complex, nonlinear interactions between adjacent roads and regions, which traditional models may miss. By integrating hidden relationships through GSO_{final} , the model can improve predictions of traffic congestion and flow patterns, especially during peak hours or unexpected disruptions, where traditional models often struggle to adjust dynamically.

Furthermore, designing the parameterizations f and g is a critical component of spatial processing. By parameterizing these functions as a graph neural network, the model is able to more effectively learn and represent the intricate spatial relationships between nodes and their neighbors. This leads to more accurate node trajectories $Z(t)$, particularly in data with dynamic and evolving graph structures, such as traffic networks that change depending on the time of day, accidents, or road closures. The ability to adaptively learn these spatial relationships, rather than relying on predefined assumptions, marks a key innovation in the model's ability to handle complex traffic dynamics. Finally, by combining the stochastic properties of stochastic differential equations with the controlled dynamics of controlled differential equations, the proposed equations offer a robust and flexible solution to spatio-temporal prediction tasks. In traffic flow prediction, this combination is particularly advantageous because traffic systems often exhibit both deterministic and random behavior, such as predictable rush hours and unpredictable accidents or weather disruptions. SDEs provide the ability to model the randomness in traffic flow, while controlled differential equations (CDEs) offer precise control over the underlying deterministic flow patterns. This integration significantly improves the model's ability to forecast traffic conditions under various scenarios, enhancing its adaptability and robustness to dynamic, real-world traffic systems.

4. Experimental study

In this section, we conduct experiments on three real-world datasets to demonstrate the effectiveness of the proposed model. In all cases, the model outperforms the others in terms of accuracy. Additionally, a comprehensive evaluation is performed to analyze the impact of key hyperparameters and the sensitivity of the model. Specifically, the model exhibits varying levels of sensitivity under different time horizons and missing rate scenarios.

The model is compared against the following three categories of baseline methods. Models based on graph convolutional networks, including AGCRN (adaptive graph convolutional recurrent network), DDGCRN (decomposition dynamic graph convolutional recurrent network), ASTGCN (attention based spatio-temporal graph convolutional networks), and STFGNN (spatio-temporal fusion graph neural networks), focus on handling spatio-temporal data with inherent spatial dependencies. Similarly, the proposed model is compared with recent spatio-temporal models such as MGCN (mamba-integrated spatio-temporal graph convolutional network), TSHDNet (temporal-spatial heterogeneity decoupling network), FasterSTS, Minusformer, and MambaTS. Finally, models based on neural differential equations, such as STGODE (spatio-temporal graph ode network) and STG-NCDE (graph structure neural differential equations on spatio-temporal prediction), leverage ordinary differential equations to model the dynamic

evolution of spatio-temporal data, which is particularly effective for long-term time series forecasting.

Experiments are conducted on three widely used datasets, including PEMS04, a traffic flow dataset from the California Department of Transportation (CalTrans) Performance Measurement System. Starting from January 1st, 2018, the dataset records traffic information at five-minute intervals over a period of 59 days, encompassing 16,992 time slices. The dataset contains traffic flow data and a 307×307 adjacency matrix representing the connectivity and distances between 307 intersections. Table 1 summarizes the statistical details of these datasets. Further details of the other datasets are omitted here for brevity.

Table 1. Details of the datasets.

No.	Name	Type	v	Length
1	PEMSD3	Traffic flow	358	26,208
2	PEMSD4	Traffic flow	307	16,992
3	PEMSD8	Traffic flow	170	17,856

4.1. Settings

The dataset was divided into training, validation, and test sets in a 6:2:2 ratio. Predictions were made for 12 consecutive time steps, with a 5-minute interval between each time point. The performance of different models was evaluated using three metrics: mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE).

The following hyperparameter configurations were used for testing: The Adam optimizer was employed to train the model for 100 epochs, with a batch size of 64 across all datasets. The learning rate was selected from the set $\{1 \times 10^{-2}, 5 \times 10^{-3}, 1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}\}$, and the weight decay coefficient was chosen from $\{1 \times 10^{-4}, 1 \times 10^{-3}, 1 \times 10^{-2}\}$. Following the methodology of Kipf [34], the number of Chebyshev polynomial terms, $K \in \{1, 2, 3\}$, was tested. The predictive performance varied slightly across datasets as K increased. To balance computational efficiency and model performance, the optimal value of K was selected for each dataset, with $K = 2$ for PEMS08 and $K = 3$ for the others, depending on the best prediction results.

4.2. Baseline models

The proposed new model is compared with 11 baseline models. All baseline models are implemented using publicly available official implementations to ensure consistency. The baseline models are as follows.

- 1) AGCRN [24]: A deep learning model that combines graph convolutional networks, recurrent neural networks, and attention mechanisms. It is particularly suited for handling dynamic data with temporal dependencies and graph structures, such as traffic flow prediction and social network analysis.
- 2) DDGCRN [25]: A model that captures the structural relationships between nodes through graph convolutional layers while leveraging recurrent neural network to capture temporal information. This allows effective time-series forecasting with dynamically changing graph structures.
- 3) ASTGCN [26]: A model designed to simultaneously capture the spatial and temporal (sequential) dependencies using graph convolutional networks. A key feature of ASTGCN is the incorporation of

attention mechanisms, enabling the model to automatically learn and focus on the most important spatio-temporal nodes, thus improving forecasting accuracy.

- 4) STGODE [27]: A model that uses ordinary differential equations to describe dynamic changes in spatio-temporal graphs. The model also captures spatial dependencies between nodes through graph convolutional networks.
- 5) STFGNN [28]: This model focuses on handling spatial dependencies using graph neural networks (GNNs) while incorporating time-series modeling to capture dynamic changes in spatio-temporal data.
- 6) MambaTS [29]: An enhanced time series forecasting model based on the Mamba architecture, utilizing selective state space models for linear complexity and incorporating temporary Mamba blocks for improved long-term forecasting.
- 7) Minusformer [30]: A transformer variant that employs subtractive attention to reduce computational complexity, enhancing efficiency while maintaining performance, especially for long-sequence modeling.
- 8) MGCN [31]: A multi-scale graph convolutional network designed for spatio-temporal data, capturing spatial dependencies at multiple scales and modeling dynamic changes through temporal convolutions, excelling in tasks like traffic forecasting and meteorological analysis.
- 9) TSHDNet [32]: A time-series hierarchical decomposition network that decomposes signals into trend, seasonality, and residual components for independent modeling, enhancing accuracy in complex time series with clear periodicity and trends.
- 10) FasterSTS [33]: An efficient spatio-temporal sequence modeling framework using lightweight spatio-temporal separable convolutions, reducing the computational overhead while preserving modeling capabilities, which is suitable for real-time prediction scenarios.
- 11) STG-NCDE [20]: A model that is capable of efficiently capturing the spatial dependencies in data while using neural continuous dynamic equations to model the temporal evolution of the data, making it highly-suited for handling long-term time-series data.

4.3. Results

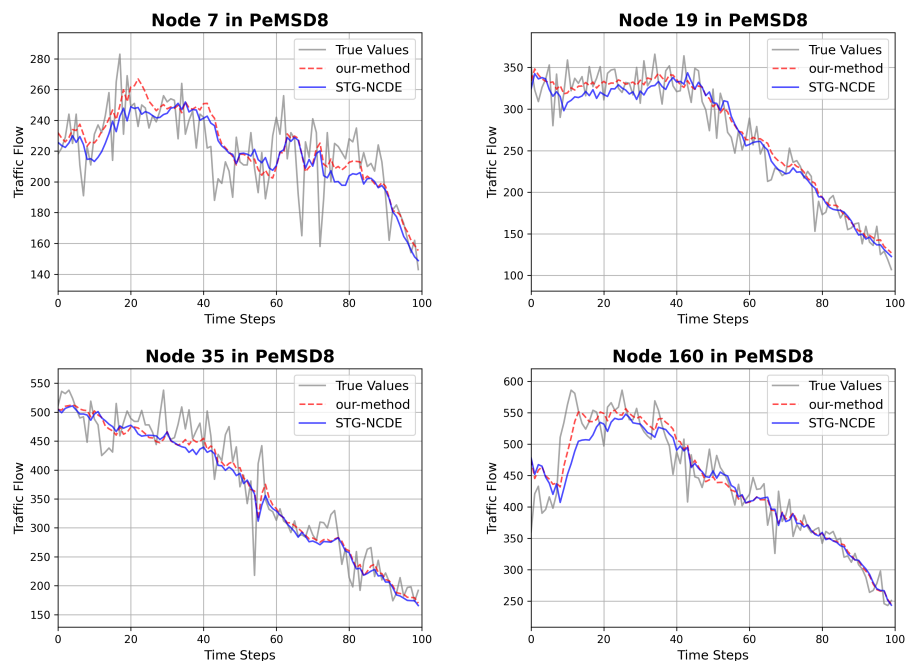
Table 2 presents the average prediction performance of the baseline models across three real-world traffic datasets. The best result in each column is highlighted in bold, while the second-best result is underlined. To further demonstrate the scalability of our approach, the penultimate row in the table shows the performance of our model with a single convolutional layer (reducing convolutional black-box complexity).

In the final row, we design a two-layer convolutional network with an activation function placed between the layers for the output passed to the convolutional network.

The performance of our model outperforms almost all of the baselines, achieving superior traffic flow prediction results. For the PEMS3 and PEMS4 datasets, our method achieves the best performance across all metrics except for the MAPE for PEMS3, where it ranks second. For the PEMS8 dataset, while our model does not achieve the lowest MAE, RMSE, and MAPE compared with FasterSTS and MGCN, it achieves the second-best RMSE and MAPE overall. FasterSTS has a lower MAPE and its RMSE is not the best, whereas MGCN's performance in terms of both MAE and MAPE is worse than ours.

Table 2. Performance comparison on PEMS datasets.

Model	PEMSD3			PEMSD4			PEMSD8		
	MAE	RMSE	MAPE%	MAE	RMSE	MAPE%	MAE	RMSE	MAPE%
AGCRN	19.41	32.34	25.92	19.83	32.26	12.97	16.69	26.43	10.22
DDGCRN	17.96	28.49	16.09	19.28	32.61	12.82	15.82	26.21	10.89
ASTGCN	20.17	33.33	19.48	22.93	35.22	16.56	23.71	35.56	14.21
STGODE	16.77	28.34	16.30	20.84	32.82	13.77	16.81	25.97	10.62
STFGNN	16.50	27.84	16.69	20.48	32.51	16.77	15.76	25.11	10.61
MambaTS	16.61	26.22	16.23	22.52	35.94	14.91	17.1	27.43	11.58
Minusformer	16.43	25.95	16.14	22.21	35.31	14.32	16.82	27.02	10.65
MGCN	16.93	29.52	16.45	19.75	31.83	12.86	15.42	24.83	10.15
TSHDNet	15.30	<u>25.70</u>	10.19	19.72	32.13	13.20	<u>14.97</u>	25.56	10.15
FasterSTS	16.30	27.03	16.52	19.64	31.67	12.98	14.93	25.42	9.90
STG-NCDE	15.65	28.93	15.11	19.33	31.18	12.75	15.93	25.42	11.56
Our method	<u>15.24</u>	27.15	14.21	<u>19.27</u>	<u>31.14</u>	<u>12.69</u>	15.82	25.07	10.59
Our method (2conv)	15.15	25.03	<u>13.85</u>	19.13	31.05	12.32	15.77	<u>24.97</u>	<u>10.12</u>

**Figure 4.** Traffic flow prediction at various nodes in PEMS8 (our method).

Therefore, while the performance of our model on the PEMS8 dataset is slightly weaker compared with the other two datasets, its overall performance is still strong, indicating that the model is generally accurate for most predictions in real-world scenarios. However, the model tends to show slightly higher

errors on datasets with fewer sensors, such as PEMS08, which also illustrates that data sparsity behavior can still pose a challenge to the performance of our approach. Our model performs better in environments with dense sensor coverage and more complex traffic patterns, but further improvements may be needed for datasets with sparse traffic flows. The performance of our method is also outstanding on three datasets, with MAPE values of 14.21% (PEMSD3), 12.69% (PEMSD4), and 10.59% (PEMSD8), demonstrating both stability and accuracy. Particularly, it outperforms on the PEMS03 and PEMS04 datasets, capturing the spatial dependencies and complex traffic patterns effectively.

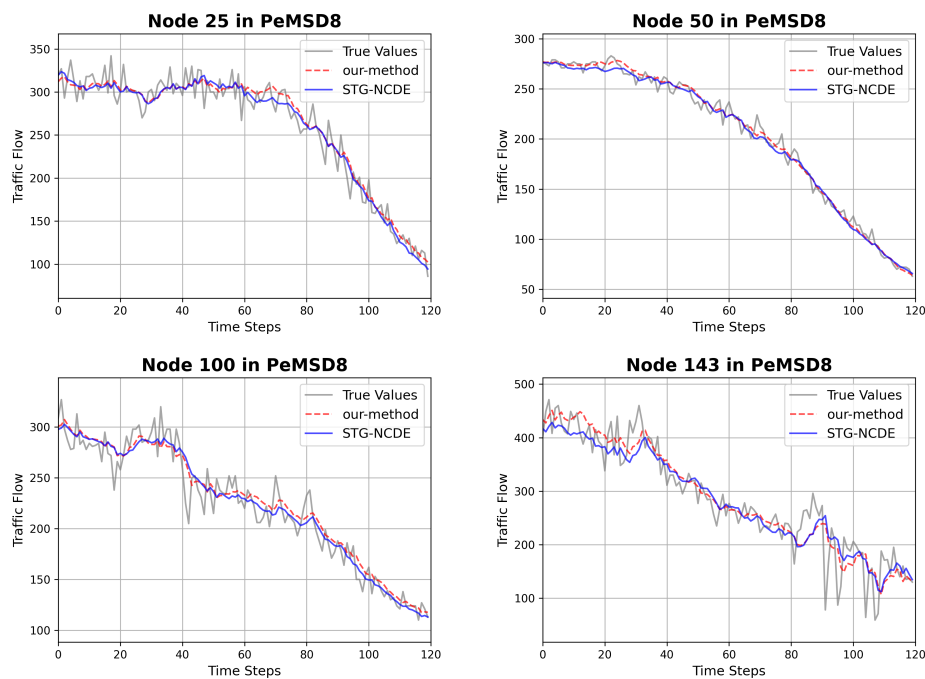


Figure 5. Traffic flow prediction at various nodes in PEMS08 (our method (2conv)).

Figures 4 and 5 present the traffic flow prediction results for different nodes of the model on the PEMS08 dataset. Figure 4 corresponds to Table 2, where our model does not include additional convolutional layers. While Figure 5 shows the model for the last row of Table 2 “Our method (2-conv)”. Each figure represent predictions for time periods, with comparisons with the real data and the baseline model’s (STG-NCDE) predictions. The model demonstrates reasonable performance, with the STG-NCDE prediction curve exhibiting similarities to our method at many time points. Both methods effectively capture the overall traffic flow variations.

However, our method achieves a better fit to the true values, resulting in lower errors and more precise predictions. As shown in Figures 4 and 5, our method provides more accurate predictions in challenging cases. Specifically, for the highlighted time points at nodes in the PEMS08 dataset, our method significantly outperforms STG-NCDE. Although STG-NCDE captures the overall trend, it exhibits larger errors at these specific nodes, indicating suboptimal performance in capturing fine-grained details at certain time points.

In Figure 6, the left panel of the figure illustrates that as the prediction horizon increases (with the horizontal axis representing the prediction length), the error progressively rises. The error analysis of the PEMS04 and PEMS03 datasets indicates that with the extension of the prediction time window, all error metrics exhibit a gradually increasing trend. This aligns with a common phenomenon in time-series forecasting, where errors tend to accumulate as the prediction horizon extends. Compared with the PEMS03 dataset, the model demonstrates relatively stable performance in short-term forecasting (within 30 minutes) on the PEMS04 dataset, with minimal variations in errors. However, as the prediction length increases, the rise in MAE and RMSE becomes more pronounced, suggesting that in long-term forecasting, the model's predictive capacity is constrained, making it challenging to capture long-term traffic flow patterns. The upward shift in error is considered inevitable. Compared with simple graph convolution methods, differential equation-based models, after training on long-term dependencies, are often better at extrapolating future states.

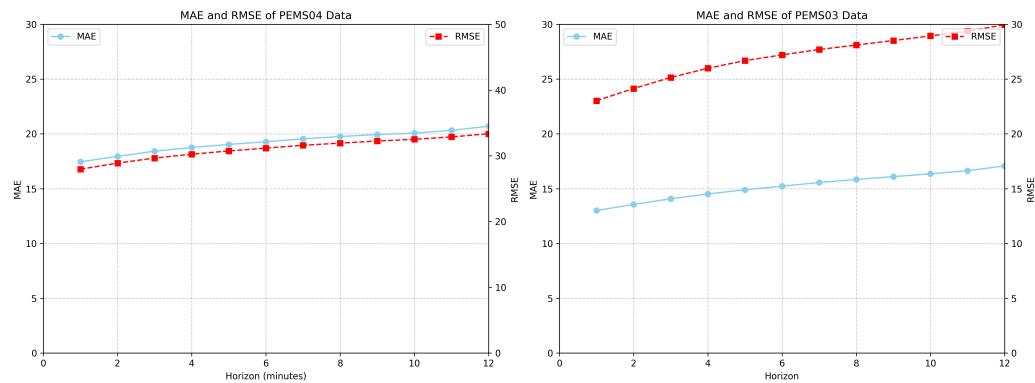


Figure 6. MAE and RMSE comparison on PEMS04 and PEMS03.

The baseline model (our method) performs best across all metrics, with an MAE of 15.77, an RMSE of 24.97, and an MAPE of 10.1223% in Table 3. Hidden connections play an important role in the model's performance, as their removal increases prediction errors. Removing the spatial SDE leads to a more significant decline, with an MAE of 17.83, an RMSE of 27.50, and an MAPE of 11.7740%, showing that the spatial SDE component is crucial for improving the model's accuracy. The variant with no hidden edges and a fixed spatial SDE performs the worst, suggesting that both hidden edges and spatial SDE contribute significantly to reducing prediction errors. Similarly, Table 4 presents the results of the ablation experiments conducted on the PEMS04 dataset. It clarifies how each factor impacts the model's performance. The results suggest that incorporating both hidden connections and spatial SDE is crucial for accurate traffic flow prediction on the PEMS04 and PEMS08 datasets.

Table 3. Ablation experiments on the PEMS08 dataset.

Datasets	PEMS08		
The original model and variants	MAE	RMSE	MAPE %
Our method (baseline)	15.77	24.97	10.1223
Our method (no hidden connections)	16.78	26.45	11.0843
Our method (no spatial SDE)	17.83	27.50	11.7740
Our method (no hidden edges, fixed spatial SDE)	18.33	27.90	12.4072

Table 4. Ablation experiments on the PEMS4 dataset.

Datasets	PEMS4		
The original model and variants	MAE	RMSE	MAPE %
Our method (baseline)	19.13	31.05	12.3224
Our method (no hidden connections)	19.39	31.43	12.7729
Our method (no spatial SDE)	19.55	31.50	12.8323
Our method (no hidden edges, fixed spatial SDE)	20.09	32.13	13.4218

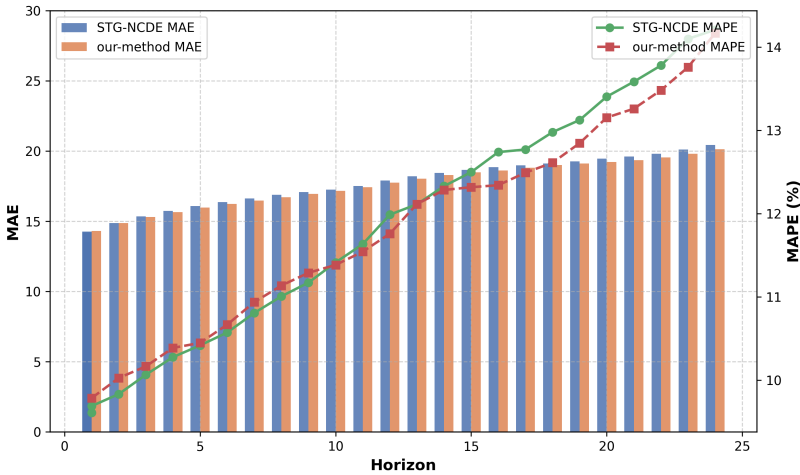


Figure 7. Comparison of MAE and MAPE for different models at a 10% missing rate.

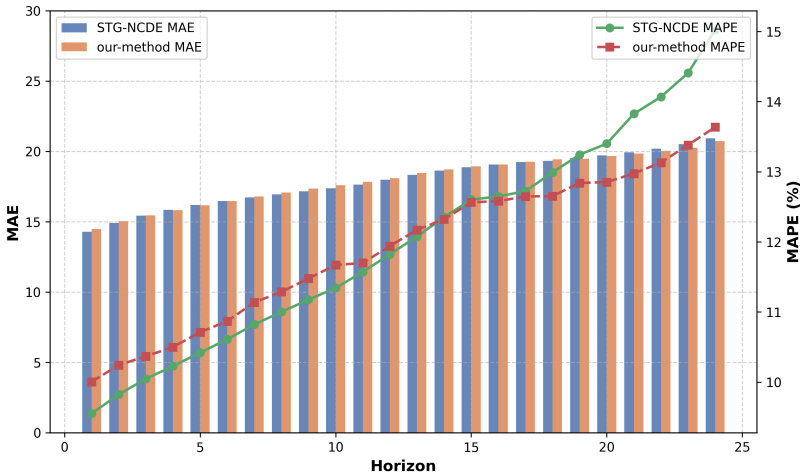


Figure 8. Comparison of MAE and MAPE for different models at a 30% missing rate.

To further demonstrate the advantages of our model, we conducted long-term forecasting experiments

under missing data conditions by extending the prediction horizon to 24 steps and comparing it with the baseline model, STG-NCDE. As shown in Figures 7 and 8, in long-term sequence forecasting, our model's MAE and MAPE increase at a significantly slower rate than those of STG-NCDE as the prediction horizon grows. Specifically, with a missing data ratio of 0.1, our model's average growth rates are 1.23% for MAE, 1.07% for RMSE, and 1.42% for MAPE. In contrast, STG-NCDE shows average growth rates of 1.68% for MAE, 1.67% for RMSE, and 1.99% for MAPE. Furthermore, when the missing ratio increases to 0.3, the MAE growth rate of STG-NCDE surpasses that of our model by 0.08%.

As the prediction horizon extends, the error in STG-NCDE accumulates and increases rapidly, while the error growth in our model slows down. For instance, in the early steps of prediction, both models exhibit similar errors, but when the forecast horizon is extended to longer periods (such as 12 steps or beyond), the MAE and MAPE growth rates of STG-NCDE increase sharply, while our model shows a much more gradual increase.

Performance comparisons under different missing rates showed that in the presence of missing data, our model shows stronger stability in predictive performance compared with STG-NCDE. As the missing ratio increases from 10% to 30%, both models experience an increase in prediction errors, but the performance degradation of STG-NCDE is more significant. When the missing rate is 10%, our model still maintains relatively low MAE and MAPE, with only a slight increase compared with the case with no missing data. On the other hand, STG-NCDE exhibits relatively higher errors, indicating that our model already shows an advantage under minimal missing data. When the missing ratio increases to 30%, this gap widens further, with the MAE and MAPE of STG-NCDE increasing much more sharply than ours.

4.3.1. Sensor failure scenarios

Table 5 presents the improvement or deterioration of three different methods—STGODE, STG-NCDE, and the proposed new method—compared with the baseline under various missing rates. The evaluation metrics include MAE, RMSE, and MAPE.

Table 5. Performance comparison under different missing rates.

Missing rate	STGODE			STG-NCDE			New method		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
10%	7.99↑	5.72↑	5.43%↑	0.23↑	0.15↑	0.13%↑	0.01↑	0.32↑	0.16%↓
30%	8.47↑	6.42↑	4.62%↑	0.76↑	0.83↑	0.51%↑	0.17↑	0.38↑	0.2408%↓
50%	—	—	—	1.23↑	1.36↑	0.75%↑	0.64↑	1.03↑	0.2554%↑

Across different missing rates, the new method consistently demonstrates certain advantages. The only exception occurs at the 10% missing rate, where the RMSE increases by 0.32, exceeding the lowest STG-NCDE increase of 0.15. Apart from this case, our model remains the optimal approach in all scenarios.

Furthermore, it is important to emphasize that even in the scenario with no missing data, the prediction performance of the new method already achieves the ideal MAE, RMSE, and MAPE values, further validating its robustness and effectiveness.

In real-world traffic forecasting, traffic sensors may malfunction, and repairing these sensors requires a certain amount of time. Consequently, data collection in some regions becomes impossible within

specific timeframes. To reflect this situation, we randomly set missing rates of 10%, 30%, and 50% for each node independently.

Since the newly proposed model, TGODe, and STG-NCDE are all neural differential equation models from a macroscopic perspective, we compare our method with these two representative baselines. The proposed model demonstrates superior performance due to its ability to mine the static graph structure of hidden edges and integrate SDEs, which enhances both flexibility and adaptability.

From the experimental results, it is evident that under a 50% missing rate, the STGODE model fails to correctly compute gradients during training, leading to not a number loss values. Empirical analysis indicates that when the missing data exceed 40%, normal prediction becomes infeasible. Therefore, under large-scale sensor failures, STGODE can no longer provide reliable predictions.

In contrast, the performance degradation of our model remains minimal. Notably, under 10% and 30% missing rates, the MAPE even decreases, as shown in Figure 9. This phenomenon is likely due to the incorporation of a hidden-edge adjacency matrix during construction of the spatial function, which effectively compensates for missing nodes through hidden connections. These findings suggest that when sensor failures occur at a missing rate below 30%, normal forecasting remains unaffected. Even in cases of large-scale failures (50% missing rate), our model's MAPE increases by only 0.2554%, which is the smallest increase among all models, as demonstrated in the table.

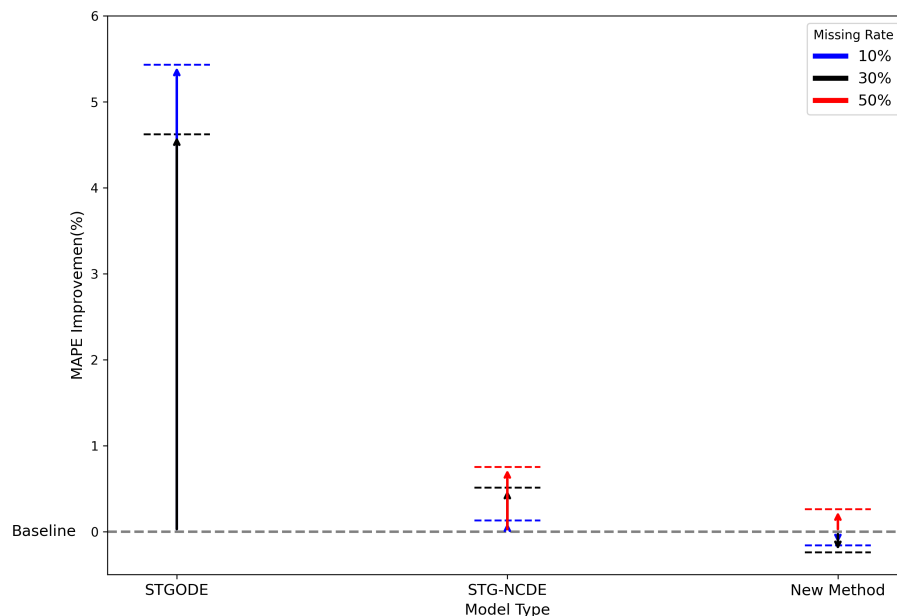


Figure 9. Relative MAPE growth for different models at 10%, 30%, and 50% missing rates.

5. Conclusions

This paper proposes a novel spatio-temporal neural-controlled differential equation model for traffic prediction. The model is extensible to scenarios consistent with our assumptions, including regional forecasting and other real-world applications. By incorporating hidden edges to reshape the graph structure and embedding their features into the spatial predictive equation, our approach uniquely differs

from previous methods that rely solely on adaptive adjacency matrices. This innovation enhances the model's ability to capture complex spatio-temporal dynamics, which traditional methods often fail to represent. A critical innovation in our approach is the construction of the spatial process using stochastic differential equations, which better accounts for the spatial uncertainty and dynamic relationships inherent in real-world systems, such as traffic flow. Incorporating SDEs into the spatial model allows us to handle not only deterministic traffic flow patterns but also the random fluctuations that are common in urban environments, such as sudden congestion or accidents. This spatial SDE framework improves the model's flexibility and robustness, ensuring that it can handle unpredictable changes in the traffic system more effectively.

Notably, the structural innovations in the use of controlled differential equations and SDEs significantly enhance the model's robustness and adaptability, making it capable of maintaining high accuracy even in the presence of sensor failures and missing data. This is particularly important for regional predictions, where data gaps are common. The incorporation of CDEs and SDEs ensures that the model not only captures the deterministic aspects of the system but also accounts for stochastic influences, providing a more comprehensive understanding of the traffic dynamics. The proposed model shows great potential in applications such as traffic flow forecasting, urban planning, congestion management, and other intelligent transportation systems. Its flexibility in modeling irregular regional data further supports its applicability in real-world scenarios. The spatial SDE framework enables the model to capture the complex spatial relationships between different regions of a transportation network, enhancing its ability to forecast traffic flow even under dynamic and uncertain conditions.

Future research directions include exploring adaptive node aggregation mechanisms, extending the model to cross-domain applications, improving interpretability, and further advancing the practical implementation and theoretical understanding of spatio-temporal data processing.

Experiments conducted on three datasets and eleven baseline models consistently demonstrated the superior overall accuracy of our approach. Furthermore, the model supports irregular regional forecasting, maintaining high accuracy even in regional block prediction scenarios—a challenge frequently encountered by existing methods. We believe that the proposed model is well-suited for a wide range of real-world data scenarios, enabling in-depth analysis and offering a promising direction for future research in spatio-temporal data processing.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the Research Projects of Guangxi Philosophy and Social Sciences Planning (Grant No. 20FTJ002). The authors express their thanks to all referees for their constructive suggestions, which have significantly enhanced the quality and comprehensiveness of this paper.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. T. Zhang, W. Ding, T. Chen, Z. Wang, J. Chen, A graph convolutional method for traffic flow prediction in highway network, *Wireless Commun. Mobile Comput.*, **2021** (2021), 1997212. <https://doi.org/10.1155/2021/1997212>
2. B. N. Oreshkin, A. Amini, L. Coyle, M. Coates, FC-GAGA: Fully connected gated graph architecture for spatio-temporal traffic forecasting, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **35** (2021), 9233–9241. <https://doi.org/10.1609/aaai.v35i10.17114>
3. X. Zhang, C. Huang, Y. Xu, L. Xia, P. Dai, L. Bo, et al., Traffic flow forecasting with spatial-temporal graph diffusion network, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **35** (2021), 15008–15015. <https://doi.org/10.1609/aaai.v35i17.17761>
4. M. A. Zaytar, C. E. Amrani, Sequence to sequence weather forecasting with long short-term memory recurrent neural networks, *Int. J. Comput. Appl.*, **143** (2016), 7–11. <https://doi.org/10.5120/ijca2016910497>
5. S. F. Tekin, O. Karaahmetoglu, F. Ilhan, I. Balaban, S. S. Kozat, Spatio-temporal weather forecasting and attention mechanism on convolutional lstms, preprint, arXiv:2102.00696.
6. M. Hossain, B. Rekabdar, S. J. Louis, S. Dascalu, Forecasting the weather of Nevada: a deep learning approach, in *2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, (2015), 1–6. <https://doi.org/10.1109/IJCNN.2015.7280812>
7. T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf, S. Shah, Forecasting traffic congestion using ARIMA modeling, in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, Tangier, Morocco, (2019), 1227–1232. <https://doi.org/10.1109/IWCMC.2019.8766698>
8. Y. Yılmaz, Machine learning-enhanced traffic light optimization system prioritizing emergency vehicle passage using SVM and random forest models, *Gazi Univ. J. Sci. Part A: Eng. Innov.*, **12** (2025), 175–196. <https://doi.org/10.54287/gujisa.1581105>
9. B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting, preprint, arXiv:1709.04875. <https://doi.org/10.24963/ijcai.2018/505>
10. G. Chen, L. Hu, Q. Zhang, Z. Ren, X. Gao, J. Cheng, ST-LSTM: spatio-temporal graph based long short-term memory network for vehicle trajectory prediction, in *2020 IEEE International Conference on Image Processing (ICIP)*, (2020), 608–612. <https://doi.org/10.1109/ICIP40778.2020.9191332>
11. Z. Qiu, K. Qiu, J. Fu, D. Fu, Dgcnn: dynamic graph convolutional network for efficient multi-person pose estimation, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2020), 11924–11931. <https://doi.org/10.1609/aaai.v34i07.6867>
12. S. Yang, Q. Wu, Y. Wang, Z. Zhou, MSTDFGRN: a multi-view spatio-temporal dynamic fusion graph recurrent network for traffic flow prediction, *Comput. Electr. Eng.*, **123** (2025), 110046. <https://doi.org/10.1016/j.compeleceng.2024.110046>
13. S. Yang, Q. Wu, Y. Wang, T. Lin, SSGCRTN: a space-specific graph convolutional recurrent transformer network for traffic prediction, *Appl. Intell.*, **54** (2024), 11978–11994. <https://doi.org/10.1007/s10489-024-05815-1>

14. S. Yang, Q. Wu, SDSINet: A spatiotemporal dual-scale interaction network for traffic prediction, *Appl. Soft Comput.*, **173** (2025), 112892. <https://doi.org/10.1016/j.asoc.2025.112892>
15. S. Yang, Q. Wu, Z. Li, K. Wang, PSTCGCN: principal spatio-temporal causal graph convolutional network for traffic flow prediction, *Neural Comput. Appl.*, **2024** (2024). <https://doi.org/10.1007/s00521-024-10591-7>
16. Z. Mei, X. Bi, D. Li, W. Xia, F. Yang, H. Wu, DHHNN: a dynamic hypergraph hyperbolic neural network based on variational autoencoder for multimodal data integration and node classification, *Inf. Fusion*, **119** (2025), 103016. <https://doi.org/10.1016/j.inffus.2025.103016>
17. L. Hu, L. Wei, Y. Lin, Decomposition dynamic multi-graph convolutional recurrent network for traffic forecasting, *Appl. Intell.*, **55** (2025), 1–17. <https://doi.org/10.1007/s10489-025-06503-4>
18. C. Song, Y. Lin, S. Guo, H. Wan, Spatial-temporal synchronous graph convolutional networks: a new framework for spatial-temporal network data forecasting, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2020), 914–921. <https://doi.org/10.1609/aaai.v34i01.5438>
19. S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **33** (2019), 922–929. <https://doi.org/10.1609/aaai.v33i01.3301922>
20. H. He, K. Ye, Graph structure neural differential equations on spatio-temporal prediction, in *2022 IEEE International Conference on Big Data (Big Data)*, (2022), 1830–1835. <https://doi.org/10.1109/BigData55660.2022.10020863>
21. X. Zheng, Y. Wang, Y. Liu, M. Li, M. Zhang, D. Jin, et al., Graph neural networks for graphs with heterophily: a survey, preprint, arXiv:2202.07082. <https://doi.org/10.48550/arXiv.2202.07082>
22. J. Zhu, R. A. Rossi, A. Rao, T. Mai, N. Lipka, N. K. Ahmed, et al., Graph neural networks with heterophily, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **35** (2021), 11168–11176. <https://doi.org/10.1609/aaai.v35i12.17332>
23. R. K. Yadav, A. Abhishek, S. Sourav, S. Verma, GCN with clustering coefficients and attention module, in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, (2020), 185–190. <https://doi.org/10.1109/ICMLA51294.2020.00038>
24. L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive graph convolutional recurrent network for traffic forecasting, *Adv. Neural Inf. Process. Syst.*, **33** (2020), 17804–17815. <https://doi.org/10.48550/arXiv.2007.02842>
25. W. Weng, J. Fan, H. Wu, Y. Hu, H. Tian, F. Zhu, et al., A decomposition dynamic graph convolutional recurrent network for traffic forecasting, *Pattern Recognit.*, **142** (2023), 109670. <https://doi.org/10.1016/j.patcog.2023.109670>
26. S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **33** (2019), 922–929. <https://doi.org/10.1609/aaai.v33i01.3301922>
27. Z. Fang, Q. Long, G. Song, K. Xie, Spatial-temporal graph ode networks for traffic flow forecasting, in *KDD '21: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, (2021), 364–373. <https://doi.org/10.1145/3447548.3467430>

28. M. Li, Z. Zhu, Spatial-temporal fusion graph neural networks for traffic flow forecasting, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **35** (2021), 4189–4196. <https://doi.org/10.1609/aaai.v35i5.16542>
29. X. Cai, Y. Zhu, X. Wang, Y. Yao, MambaTS: improved selective state space models for long-term time series forecasting, preprint, arXiv:2405.16440. <https://doi.org/10.48550/arXiv.2405.16440>
30. D. Liang, H. Zhang, D. Yuan, B. Zhang, M. Zhang, Minusformer: improving time series forecasting by progressively learning residuals, preprint, arXiv:2402.02332. <https://doi.org/10.48550/arXiv.2402.02332>
31. W. Lin, Z. Zhang, G. Ren, Y. Zhao, J. Ma, Q. Cao, MGCN: mamba-integrated spatiotemporal graph convolutional network for long-term traffic forecasting, *Knowl.-Based Syst.*, **309** (2025), 112875. <https://doi.org/10.1016/j.knosys.2024.112875>
32. M. Wu, W. Weng, X. Wang, D. Seng, TSHDNet: temporal-spatial heterogeneity decoupling network for multi-mode traffic flow prediction, *Appl. Intell.*, **55** (2025), 1–14. <https://doi.org/10.1007/s10489-024-06218-y>
33. B. A. Dai, N. Lyu, Y. Miao, FasterSTS: a faster spatio-temporal synchronous graph convolutional networks for traffic flow forecasting, preprint, arXiv:2501.00756. <https://doi.org/10.48550/arXiv.2501.00756>
34. T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, preprint, arXiv:1609.02907. <https://doi.org/10.48550/arXiv.1609.02907>



AIMS Press

©2025 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)