Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 7099652, 16 pages
https://doi.org/10.1155/2023/7099652

Research Article

GDENet: Graph Differential Equation Network for Traffic Flow Prediction

Yanming Miao,[1] **Xianghong Tang**,[1] **Qi Wang**,[1] **and Liya Yu**[2]

[1]*The State Key Laboratory of Public Big Data, Guizhou University, Guiyang, Guizhou 550025, China*
[2]*School of Mechanical Engineering, Guizhou University, Guiyang, Guizhou 550025, China*

Correspondence should be addressed to Xianghong Tang; xhtang@gzu.edu.cn, Qi Wang; qiwang@gzu.edu.cn, and Liya Yu; lyyu@gzu.edu.cn

Received 17 January 2023; Revised 3 July 2023; Accepted 9 November 2023; Published 9 December 2023

Academic Editor: Surya Prakash

The accurate prediction of traffic flow is paramount for the advancement of intelligent transportation systems. Despite this, current prediction models only account for either temporal or spatial features in isolation, without considering their interaction, impeding the model's ability to express itself. In light of this, we propose the graph differential equations network (GDENet), an approach that can effectively mine spatiotemporal correlation. Specifically, we propose a spatiotemporal feature integrator (STFI), which alleviates the error caused by the deviation of the sampling distribution from the overall distribution. By incorporating temporal information into the model for training and combining it with spatial features, we thoroughly explore the spatiotemporal intrinsic association. When compared to state-of-the-art methods, our proposed algorithm reduces memory consumption and elevates computational efficiency and the practical value. We conduct experiments with real-world datasets, and our proposed model outperformed advanced prediction models.

1. Introduction

Traffic flow prediction is a prominent example of the spatiotemporal prediction [1, 2] problem and is an important section of the intelligent transportation system (ITS) [3–5]. The study of spatiotemporal prediction involves the analysis of historical data across both spatial and temporal dimensions to extract underlying patterns of change and facilitate the generation of prediction. Compared with traditional temporal sequence prediction, spatiotemporal prediction incorporates geographical space characteristics, yielding more accurate prediction outcomes. Spatiotemporal prediction has extensive applications in various domains such as intelligent transportation [6, 7], urban planning [8, 9], and logistics management [10, 11]. Hence, it is crucial to excavate spatiotemporal data for discerning valuable patterns and making accurate spatiotemporal predictions. Traffic flow prediction refers to the accurate forecasting of traffic flow changes on a particular road in the upcoming minutes or hours through the utilization of historical traffic data. Precise traffic flow prediction is of utmost importance for efficient urban traffic planning [12], traffic management [13], and traffic control [14].

Traffic flow data entail the recording of the count of vehicular entities at a specific point along a roadway over a designated period of time. Traffic flow data directly reflect the status of traffic and enable researchers to describe traffic conditions [15] and conduct traffic-related research. As shown in Figure 1, such data exhibit diverse characteristics across distinct time intervals, manifesting as a high-dimensional feature. Consequently, the inadequacy and imbalance of traffic flow data pose significant challenges, with the periodicity and regularity of the data proving arduous to discern. Moreover, the high number of feature dimensions in traffic flow data increases computational complexity, resulting in prolonged calculation times. Traffic flow data play a vital role in understanding traffic dynamics and devising effective traffic management strategies.

Owing to its nonlinear [16], high-dimensional [17, 18], and noisy feature, the task of scientifically and accurately

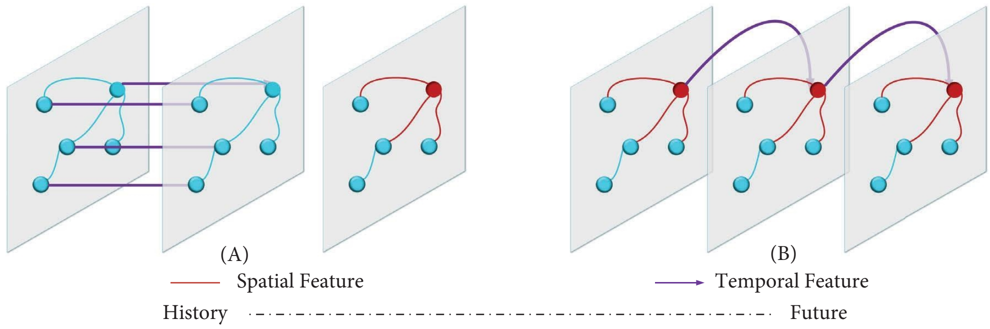

Figure 1: (A) is the traditional method and (B) is our algorithm. The traffic flow information of adjacent times mainly relates to the temporal characteristics. In contrast, the spatial characteristics mainly relate to the traffic flow of adjacent nodes in the road structure. Compared to other methods, we explore the intrinsic properties of temporal and spatial features, which mine intrinsic data relation jointly.

predicting traffic flow has remained a formidable challenge. A majority of existing models tend to extract either temporal or spatial features in isolation [19–21], thereby neglecting the crucial interactions between these features. Consequently, these models' overall learning ability is substantially constrained.

Among them, the statistical model has limited feature extraction abilities. The deep learning model can only handle structured and ruled data, and it is difficult to excavate the temporal and spatial correlation of the data. The graph neural network model takes more into account the spatial position relationship but falls short in temporal feature extraction [22, 23]. In addition, current prediction models commonly employ the uniform sample to generate sample data. The uniform sample is based on equal time intervals, which may not reflect real-world scenarios. In reality, irregular sampling is more common. For example, network traffic observations or sampling intervals are often irregular. To overcome these challenges, there is a need for further research and development of prediction models that can effectively handle irregularly sampled data while extracting both spatial and temporal features.

As is commonly understood, deep learning models rely on the independent identical distribution assumption [24], which means the training samples have a similar distribution to the real distribution [25]. This assumption helps to minimize the impact of special cases in the sample data. In the task of traffic flow prediction, historical flow data are used for training and predicting future flow data. To predict future flow data, it is necessary to extract the latent properties from sample data. Therefore, it is imperative that historical sample data closely resemble the overall distribution and accurately reflect its characteristics. However, if the training samples do not accurately represent the overall, the latent properties may be irregular, particularly when properties are derived from special cases. However, the uniform sample is currently the most common method for generating samples [26]. For example, recurrent neural networks (RNNs), which divide input time data equally and use a fixed time interval. Traffic flow prediction models also rely on uniform sample to generate sample data. This approach of equally dividing input time data at fixed time intervals may not accurately reflect real-world traffic patterns. When estimating at fixed intervals, the sampling distribution of the data may deviate from the overall distribution. Consequently, models trained using such data may fail to accurately capture key characteristics and intrinsic attributes of the flow data overall distribution.

In reality, irregular sampling is general. For instance, the observation intervals for traffic networks may vary widely. With the fixed sampling interval for estimation, the final generated distribution will deviate from the overall distribution. The characteristics of the distribution cannot be described accurately, and the feature extraction is not sufficient. In an effort to mitigate the distribution deviation caused by the uniform sample, Chen et al. [27] have developed a novel modeling approach that incorporates both a neural network and ordinary differential equation (ODE) techniques. The resultant ODENet model is designed to operate in continuous time, which represents a significant improvement over previous methods. In a similar vein, Fang et al. [28] have recently proposed a continuous network model called STGODE, which incorporates the TCN layer and ODENet while also introducing an ODESolver into the middle of the TCN layer. Nevertheless, the TCN layer is discrete itself, so STGODE is still not continuous. Building upon the strengths of ODENet and graph neural networks, we have developed a new spatiotemporal feature integrator (STFI) that integrates the GNN over a continuous interval. By utilizing the time interval as the integral interval and training the model jointly with the original data, we are able to effectively reduce the error resulting from distribution deviation and more accurately exploit spatiotemporal correlations.

In summary, the contributions of this paper include the following:

(1) We propose the GDENet model, which captures spatiotemporal potential relevance to predict traffic flow, which enhances the accuracy of prediction.

(2) We present the spatiotemporal feature integrator (STFI). In addition, we reason out the backpropagation process of STFI, which is called GDESolver. The GDESolver can help the model obtain higher accuracy with lower memory occupancy.

(3) We conduct experiments on real-world datasets to evaluate the effectiveness of our proposed approach.

Our experimental results corroborate the efficacy of the GDENet model, demonstrating higher computational efficiency and a greater practical value than the current state-of-the-art methods for traffic flow prediction.

We organize this paper as follows: Section 2 introduces the current research status. Section 3 presents the proposed GDENet model in detail. Section 4 conducts the experimental results and corresponding analysis. Section 5 summarizes our work.

2. Related Work

In light of recent advances in the field of traffic flow prediction, existing techniques can be broadly classified into two categories: statistical models and artificial intelligence (AI) models. While statistical models rely on probabilistic assumptions to model traffic patterns, AI models leverage machine learning algorithms to learn from raw data and extract meaningful insights.

2.1. Statistical Model. Within the category of statistical models for traffic flow prediction, several approaches have been proposed. One such approach is time series analysis, which involves fitting mathematical models to observed traffic patterns to make predictions. Another commonly employed statistical method is the history average model [29], which finds the average value of historical traffic flow data to predict future patterns. A third approach, the autoregressive integrated moving average [30] model can convert the nonstationary data into a stationary sequence through several differential operations and then approximates the traffic flow data with a mathematical model. These techniques have demonstrated varying degrees of success in predicting traffic flow and continue to be widely studied within the field.

The statistical model is based on the premise of stability assumptions. However, traffic conditions are affected by multiple factors in time and space and have a temporal and spatial correlation. Consequently, only relying on time-series forecasting models cannot fully mine latent attributes. The proposed GDENet employs a gate structure that facilitates the extraction of potential spatiotemporal relevance.

2.2. Artificial Intelligence Model. Artificial intelligence models can be divided into machine learning models, deep learning models, and graph neural network models. Specifically, machine learning models rely on algorithms that enable them to learn from data patterns, whereas deep learning models make use of complex architectures composed of multiple layers and graph neural network models utilize graph structures to model complex relationships between nodes.

2.2.1. Machine Learning Model. Traditional machine learning models mainly include support vector regression (SVR), self-encoding algorithms, and other models. Smola et al. [31] input historical traffic flow data into a support vector regression machine for training to predict future traffic flow data. Lv et al. [32] used the self-encoding algorithm to predict traffic flow.

Although traditional machine learning models can process complex traffic flow data, spatial feature extraction is still insufficient. Therefore, it is difficult to mine the temporal and spatial correlation of traffic data, leading to a lower prediction accuracy rate [33]. The GDENet model we proposed takes advantage of the graph neural network, inputs spatial position relationships during training, sufficiently extracts spatial features, and mines temporal and spatial correlations.

2.2.2. Deep Learning Model. In the deep learning model, Liu et al. [34] combined the CNN and LSTM models and proposed the ConvLSTM module to extract spatiotemporal features of traffic flow data. Liu et al. [35] presented Att-ConvLSTM which leverages a ConvLSTM, CNNs along with an attention mechanism under a sequence-to-sequence learning framework to predict citywide crowd flow. Zhang et al. [36], with the powerful feature extraction of ResNet, mined the temporal and spatial features in the traffic flow data for traffic flow prediction.

Nevertheless, deep learning models such as CNN, LSTM, and ResNet can process structured rule data in most cases, which causes the unstable and nonlinear traffic flow data that cannot be addressed effectively. The STFI we proposed uses the graph neural network model, which can handle unstructured data and can process complex traffic flow data so that feature extraction is more sufficient.

2.2.3. Graph Neural Network Model. In recent years, research on graph neural networks has developed rapidly and become one of the most popular topics in artificial intelligence research [22]. In the task of traffic flow prediction, traffic flow data are complex structured data and the relationship between roads is not only a simple spatial position relationship. The graph structure has a more vital expression ability, which can describe more problem scenarios. The graph structure simulates the transportation network properly and reduces the characteristic loss. The graph neural network can directly perform feature extraction on the graph structure. As a result, traffic flow prediction based on graph neural networks has gradually become an important research direction.

Yu et al. [37] adopted the graph convolutional neural network (GCN) structure in traffic prediction for the first time and used a fully convolutional structure instead of a conventional RNN module, proposing Gated CNN to extract temporal features. Li et al. [38] worked on the basis of the graph convolutional neural network and used diffusion convolution instead of matrix multiplication to mine temporal properties. Zhao et al. [21] introduced the GCN graph convolutional network to take the spatial attributes of the traffic data while using GRU to extract the temporal features, achieving a favorable outcome. Guo et al. [39] used GCN to take spatial attributes and ordinary convolution to extract temporal features and, at the same time, introduced an attention mechanism in the temporal dimension to mine temporal properties in different periods. Furthermore, Rao et al. [40] proposed FOGS, a novel method for traffic flow prediction which divides a day into four time periods. Moreover, the FOGS uses the statistical data (average value, median, and standard deviation) of traffic flow for training, relieving the problem of fitting the irregularly-shaped traffic data.

Most of the studies focus on the spatial correlation of neighbor regions, ignoring the global feature information. Therefore, Zhang et al. [41] proposed the STGDN model. Similar to Guo et al., STGDN divides the input into three parts: hour, day, and week for time feature extraction. Also, STGDN divides the whole area into several small areas and uses the multihead attention mechanism to fuse local spatial contextual information for global representation. Jiang et al. [42], with the help of the transformer, proposed PDFormer to improve the long-range spatial dependency extract effect. At the same time, the spatial self-Wention module is introduced to transform the time delay in the traffic systems into explicit features. The PDFormer achieves state-of-the-art performance on six real-world public traffic datasets. Wu et al. [43] used dilated causal convolution to construct a temporal convolution layer to learn the temporal trend of nodes and take temporal attributes. Song et al. [44] aggregated the dependencies of neighbor nodes on the same node in different time slices and further extracted spatiotemporal features. Li et al. [45] merged the transportation network graphs in different time slices, constructed a spatiotemporal fusion graph, and effectively discovered hidden spatiotemporal relevance. Cao et al. [46] proposed the StemGNN model, which considers both intratime series correlation and interseries correlation. Combining the advantages of graph Fourier transform (GFT) and discrete fourier transform (DFT), the multivariate input is transformed into the same space for feature representation. Ji et al. [47] introduced heterogeneity into the field of traffic prediction and construct the ST-SSL framework which achieves the adaptive spatiotemporal self-supervised learning. Most models uniformly predict the traffic flows in all regions without accounting for spatiotemporal heterogeneity, while the ST-SSL forecasts traffic flow with the help of spatial and temporal heterogeneity-aware augmentation. Gong et al. [48] introduced the online latent space (OLS) strategy to the crowd flow prediction. Specifically, the OLS strategy takes into account the various trending patterns and climate influences so that their models address the online crowd flow prediction problem effectively. Ou et al. [49] proposed STP-TrellisNets + to predict metro station passenger flow. The STP-TrellisNets + adopts a novel transfer flow-based metric to characterize the spatial correlation and employs a closeness TrellisNet to jointly capture the short- and long-term temporal correlation of metro station passenger flow.

Existing models coarsely regard the traffic road network as a static graph. Shao et al. [50] proposed D2STGNN which encompasses a dynamic graph learning module. The D2STGNN decomposes traffic flow data into the diffusion signal and inherent signal to effectively mine dynamic characteristics. Lan et al. [51] replaced the predefined static graph with a dynamic spatial-temporal aware graph and proposed DSTAGNN. The DSTAGNN acquires intrinsic dynamic information and spatial structure properties by excavating historical traffic flow data.

Combining the abovementioned research, the graph network-based model has significant advantages in spatial feature extraction. Therefore, the graph network-based model focuses more on how to excavate temporal properties accurately. The GDENet model combines the gate structure and the ordinary differential equation model to aggregate time information and efficiently mine spatiotemporal correlations. At the same time, our model uses a spatiotemporal feature integrator (STFI), taking the time interval as the integration interval, serving original data for training. The modeling is more in line with the actual situation, reducing the error caused by the deviation of the distribution and achieving a precise prediction effect.

3. Methodology

This section will introduce the specific details of the model structure in detail. First, we provide an overview of the GDENet framework. Then, in two separate subsections, we introduce the two main components, i.e., temporal features extraction and spatial features extraction. The last subsection introduces the spatiotemporal feature integrator.

3.1. Overview. Traffic flow data have temporal features and spatial features, which are interdependent and have spatiotemporal correlation. Therefore, our proposed model explores temporal features and spatial features jointly to excavate potential spatiotemporal relevance more adequately. The model structure is shown in Figure 2. We input the original data into multilayer perceptions to realize feature conversion and information reorganization. Our model first learns temporal properties and then discovers spatial attributes. Besides, we captured potential spatiotemporal relevance by the STFI. Finally, the features are fused and input multilayer perceptron for dimensionality reduction to obtain the prediction result.

3.2. Temporal Features Extraction. As a kind of time-series data, traffic flow data must contain many time properties. In the process of time feature extraction, most models pay attention to the information in the adjacent time slices: short-term time information. Long-term time information with extensive time intervals is often ignored by the model, resulting in low prediction accuracy. In this regard, we use the gate structure to remember valuable time information in model training and fully extract long-term time attributes. In addition, most models generate samples through uniform sampling, equal interval sequence modeling, and actual. The distribution situation is different, and the distribution deviation error will occur when we put models into the application, which will affect the prediction performance. Aiming at the problem of the deviation of the distribution, we use ODE to make the model more in line with the actual situation.

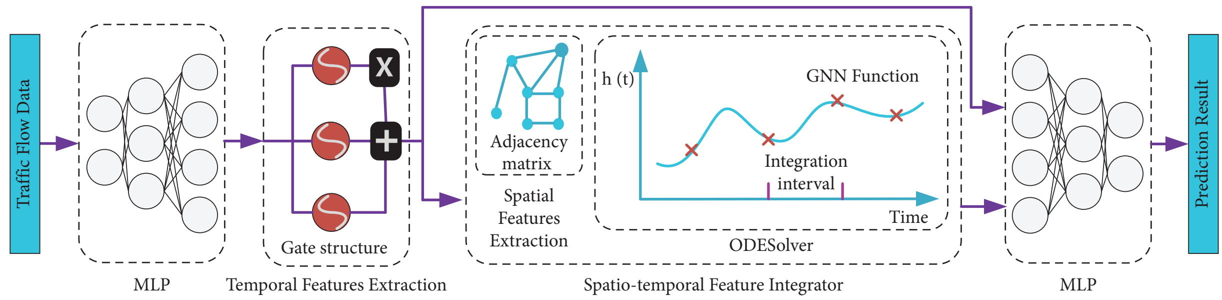

Figure 2: An overview of our proposed framework.

3.2.1. Long-Term Temporal Features Extraction. If a time series is long enough, it will be difficult for the model to transfer feature information from earlier time steps to later time steps. There may even be important information left out from the start. Regarding the abovementioned issue, Hochreiter et al. [52] introduced the gate structure and proposed an LSTM model with the function of saving or forgetting information. The gate structure can memorize essential information from the beginning and retain this critical information in the subsequent learning process. Referred to the RNN model [53], we improve the gate structure to adapt to the temporal features of traffic flow. The specific functions of the gate structure are as follows: the forget gate determines which related information in the previous step needs to be retained, the input gate judges which information in the current state is essential and needs to be learned, and the output gate decides where the next hidden state should be. In addition, for the hidden layer state h_t, the gate structure adds a state C_t to save the long-term state, called the cell state, to learn long-term dependent information, which can be expressed as follows:

$$\begin{aligned} i_t &= \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right), \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \tanh\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right), \end{aligned} \tag{1}$$

where i_t can be seen as the information retained in the new information at time t, h_{t-1} is the output of the previous layer, x_t represents the information input at time t, f_t can be understood as the information retained at time t, C_{t-1} is the cell state at time t, and W and b represent the parameters that the network model needs to learn. Among them, it can be explained as the information retained in the new information at time t.

3.2.2. Reduce Distribution Deviation. Inspired by ODENet, we propose an ordinary differential equation to model the time dimension to reduce distribution deviation errors. The neural networks can be regarded as a complex compound function, whether recurrent neural networks or convolutional neural networks. For example, a network model composed of two fully connected layers can be expressed as follows:

$$Y = f_{\mathrm{fc1}}\left(f_{\mathrm{fc2}}\left(X, \theta_2\right), \theta_1\right), \tag{2}$$

where X represents the input, Y is the output result, f_{fc1} and f_{fc2} represent the fully connected layer, and θ_1 and θ_2 represent the parameters that the network model needs to learn. Therefore, each neural network model is similar to a universal function approximator by constantly updating the network layer. From another perspective, instead of directly modeling the objective function, we might as well try to simulate the function change rate.

$$\frac{\mathrm{d}h(t)}{\mathrm{d}t} = f\left(h(t), \theta(t), t\right). \tag{3}$$

Based on the above, we turn the problem into

$$h\left(t_1\right) = h\left(t_0\right) + \int_{t_0}^{t_1} f\left(h(t), \theta(t), t\right)\mathrm{d}t, \tag{4}$$

where t_0 and t_1 represent time. $h(t_1)$ and $h(t_0)$ represent the hidden state at t_0 and t_1. f is the network layer function, and $\theta(t)$ is the parameters that the hidden state layer needs to learn at time t. In fact, this is an initial value problem of the differential equation. We can obtain $h(t_1)$ by calling the graph differential equation solver (GDESolver). Finding the numerical solution of the differential equation is equivalent to completing the forward propagation of the network model.

In traffic flow prediction tasks, most model samples from the population are uniform, which is not consistent with the actual situation and often leads to deviations between the estimated distribution and the overall distribution and thus cannot achieve better prediction results. In practice, treating time as a continuous variable is a more natural choice. A differential equation can describe the evolution of a specific process over time and have continuity in the time dimension. Hence, using differential equations can obtain more robust properties. We adopt the gate structure and STFI to mine temporal attributes to prepare for the subsequent extraction of the spatiotemporal correlation.

3.3. Spatial Features Extraction. Traffic datasets often contain the spatial structure of traffic roads, which contain many spatial features. In order to extract the spatial features, we transform the traffic road into a graph structure and use the graph neural network to discover the spatial properties. Traditional convolutional neural networks can only handle Euclidean spatial data, such as images, text, and speech, with translation invariance. There are many non-Euclidean spatial data in real life, such as transportation networks, World Wide Web, and social networks. The local structure of each data point in these data is different, making the translation invariance no longer satisfied. Graph structure can naturally express non-Euclidean spatial data in real life, so it is widely applied in data storage, retrieval, and computing applications. Graph neural network (GNN) is a model of building a neural network on the graph structure, applying the deep learning model to the graph structure, and being able to deal with non-Euclidean structures.

Graph attention network (GAT) [54] aggregates the characteristics of neighboring nodes through the attention mechanism to determine the importance of each neighbor node to the central node. The attention mechanism allows the GAT to learn the dependencies between global features better, and the relatedness between node features is better integrated into the model. α_{ij} represents the attention weight between the neighbor nodes j of node i, which is calculated by the following formula:

$$\alpha_{ij} = \text{softmax}\left(\frac{\sigma\left(a^T\left[W^{(l)}\cdot H_i \,\|\, W^{(l)}\cdot H_j\right]\right)}{\sum_{k\in N_i}\sigma\left(a^T\left[W^{(l)}\cdot H_i \,\|\, W^{(l)}\cdot H_j\right]\right)}\right), \quad (5)$$

where σ represents the activation function, a^T is a learnable parameter vector, N_i represents the set of neighbor nodes of node i, $W^{(l)}$ represents the weight coefficient of the l layer, H_i represents the node feature of i, and the softmax function ensures that the sum of the attention weights of all neighbors of node i is 1. The graph neural network model has strong spatial feature extraction capabilities. Bai et al. [55] introduced an attention mechanism in the time dimension to better extract temporal features. The model has achieved good results on the SZ-taxi Shenzhen taxi trajectory datasets and the Los-loop Los Angeles highway datasets. Our proposed model utilizes the attention mechanism as a means to enhance the aggregation of information from neighboring nodes. Figure 3 shows the attention weight of all neighbors of node i. Among them, the sum of α_{i1}, α_{i2}, α_{i3}, and α_{i4} is 1. The attention mechanism efficiently integrates the attributes of neighboring nodes, thereby amplifying the efficacy of spatial feature extraction.

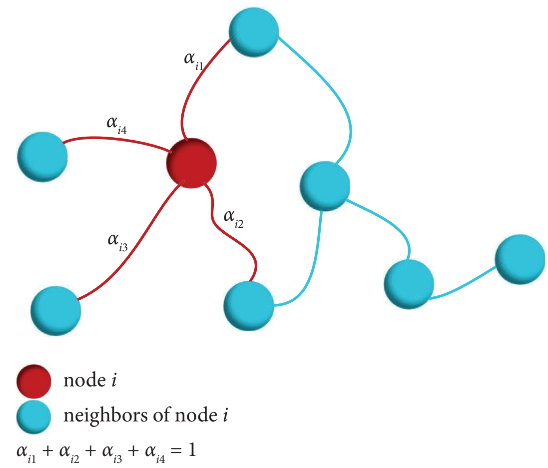

Figure 3: The attention weight of all neighbors of node i.

3.4. Spatiotemporal Feature Integrator. In traffic flow data, time features and spatial features are not independent of each other. However, most models learn temporal features or spatial features separately, ignoring the interaction and association between them, which limits the predictive effect of the model [56]. We combine the graph neural network model with ODENet and propose the STFI to explore the temporal and spatial correlation. We analyzed temporal features with spatial features to further improve the prediction effect.

3.4.1. Spatiotemporal Integral Modeling. The most critical steps of the neural network model are forward propagation and backpropagation. Many neural network models can be seen as discretized forms of differential equations. In other words, it means that neural network models have corresponding numerical solutions. Lu et al. [57] summarized the correspondence between the mainstream models as follows: ResNet [58] corresponds to the forward Euler method [59], PolyNet [60] corresponds to the backward Euler method [61], and FractalNet [62] corresponds to the Runge–Kutta [63] method. Therefore, we can regard the forward propagation process of the neural network model as solving differential equations given an initial value. By calling the differential equation solver, we adopt the forward and backward propagation to finish this solution. Combining the abovementioned methods, we use the GAT model and differential equations to propose the STFI, simulate the rate of change, and extract the temporal and spatial features of the traffic flow data.

In the forward propagation of the conventional neural network model, we map the input X to the output Y_{pred} and then adjust the weight of the network to match a certain Y_{true}. Similarly, in GDEnet, we input the node features of t_0 and convert them into feature vectors X. We treat X as the initial value $h(t_0)$, therefore, the forward propagation process of GDEnet can be equivalent to solving the final value $h(t_1)$ by using f_{GNN}, $h(t_0)$, t_0, and t_1. The specific process is as follows:

$$\begin{aligned}\frac{\mathrm{d}h(t)}{\mathrm{d}t} &= f_{\text{GNN}}(h(t),\theta(t),t),\\ h(t_1) &= h(t_0) + \int_{t_0}^{t_1} f_{\text{GNN}}(h(t),\theta(t),t)\mathrm{d}t,\end{aligned} \quad (6)$$

where t_0 and t_1 represent time, $h(t_0)$ and $h(t_1)$ are the hidden states at t_0 and t_1. f_{GNN} is the graph neural network function, and $\theta(t)$ are the parameters that the hidden state layer needs to learn at t. When we solve an ordinary differential equation, we actually take the process of integral calculus. Integral, a fundamental concept in calculus, serves as the inverse process of derivative and holds significant relevance within this mathematical discipline. Given the derivative, we take the reverse process of the derivative and find the primary function. According to the theorem of continuity of a primitive function [64], if F is a primitive function of f, then F is continuous. In the same way, the spatiotemporal feature integrator treats the GNN as an ordinary differential equation. The obtained prediction function exhibits a state of continuity. Therefore, the STFI can achieve continuous GNNs and provide more accurate traffic flow prediction. The STFI takes the time interval as the integral interval and integrates the GNN over a continuous interval, alleviating the deviation of the distribution caused by uniform sampling. Figure 4 introduces the basic operating principle of the STFI. The STFI employs GDESolver to realize the backpropagation with lower memory consumption.

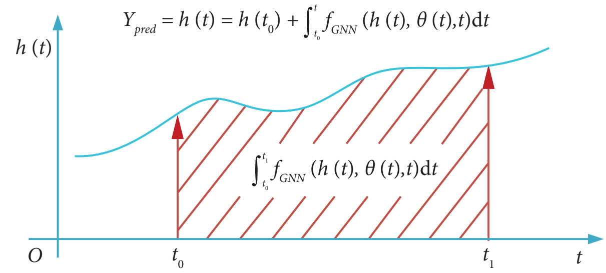

Figure 4: The basic operating principle of STFI.

3.4.2. Loss Function. Since traffic flow prediction is a regression problem, we define the loss function based on the MSE function. We assumed the prediction result is Y_{pred_t} and define the loss function as follows:

$$\mathrm{loss} = MSE\left(Y_{\mathrm{pred}_t}, Y_{\mathrm{true}_t}\right) = \sum_{t=1}^{n}\left(Y_{\mathrm{pred}_t} - Y_{\mathrm{true}_t}\right)^2. \tag{7}$$

3.4.3. Graph Differential Equation Solver. Traditional neural network models often use fully connected layers and convolutional layers. The fully connected layer has an enormous number of parameters and requires a vast amount of memory occupancy. The main technical difficulty in training neural network models lies in the back propagation of gradients. According to the chain rule, the gradient propagates back along the calculation path of the forward propagation so that the weight parameters are updated. When training the convolutional layer, backpropagating requires massive computation, producing an enormous number of intermediate parameters. The intermediate variables take up a large amount of memory. Therefore, most models have relatively high hardware requirements and high economic costs. At the same time, the model training time is extended and cannot satisfy real-time performance, which leads to low practicability.

In our STFI, considering that the gradient back propagation along the forward propagation path will take up a lot of memory resources, we propose GDESolver to realize the back propagation process by using the adjoint method. The adjoint method can calculate the gradient and renew the network model parameters. The adjoint method does not need to propagate from back to forward through the model and not only has less memory occupancy but also has higher computational efficiency than the traditional back propagation process. The specific routing is as follows.

We define the adjoint state as follows:

$$\mathrm{Adj}(t) = \frac{\partial \mathrm{Loss}}{\partial h(t)}. \tag{8}$$

According to the equation, it is actually the gradient of the hidden state and the Loss. Given Loss and $h(t_1)$, the accompanying state at t_1 can be obtained:

$$\mathrm{Adj}(t_1) = \frac{\partial \mathrm{Loss}}{\partial h(t_1)}. \tag{9}$$

We want to find the adjoint state at the previous moment, namely, $\mathrm{Adj}(t_0)$. According to the chain rule, it can be expressed as follows:

$$\begin{aligned}\frac{\partial \mathrm{Loss}}{\partial h(t)} &= \frac{\partial \mathrm{Loss}}{\partial h(t+\varepsilon)}\frac{\partial h(t+\varepsilon)}{\partial h(t)}\\ &= \mathrm{Adj}(t+\varepsilon)\frac{\partial h(t+\varepsilon)}{\partial h(t)},\end{aligned} \tag{10}$$

where ε is an infinitesimal quantity. We express $h(t+\varepsilon)$ as an integral form as follows:

$$h(t+\varepsilon) = h(t) + \int_{t}^{t+\varepsilon} f_{\mathrm{GNN}}(h(T), \theta(T), T)\mathrm{d}T. \tag{11}$$

According to the above two equations,

$$\mathrm{Adj}(t) = \mathrm{Adj}(t+\varepsilon) + \mathrm{Adj}(t+\varepsilon)\frac{\partial}{\partial h(t)}\left(\int_{t}^{t+\varepsilon} f_{\mathrm{GNN}}\mathrm{d}T\right). \tag{12}$$

According to the derivative definition,

$$\frac{d\mathrm{Adj}(t)}{dt} = \lim_{\varepsilon \longrightarrow 0^+}\frac{Adj(t+\varepsilon) - Adj(t)}{\varepsilon}. \tag{13}$$

We can simplify the equation as follows:

$$\frac{d\mathrm{Adj}(t)}{\mathrm{d}t} = -\mathrm{Adj}(t)\frac{\partial f_{\mathrm{GNN}}(h(t), \theta(t), t)}{\partial h(t)}. \tag{14}$$

We have obtained the differential equation about $\mathrm{Adj}(t)$, which can be obtained by integration:

$$\mathrm{Adj}(t_0) = \mathrm{Adj}(t_1) - \int_{t_1}^{t_0} \mathrm{Adj}(t)\frac{\partial f_{\mathrm{GNN}}}{\partial h(t)}\mathrm{d}t. \tag{15}$$

We call GDESolver to solve and realize the back propagation process. We can establish the parameter $\theta(t)$ and the adjoint state of time t and derive the corresponding differential equation. We vectorize the three adjoint states and relate differential equations, which can be denoted by Adj_{Ω}. We integrate three differential equations to simultaneously solve the gradients of all parameters involved in model training using a single GDESolver. Details are as follows:

$$\mathrm{Adj}_{\Omega} = \left[\mathrm{Adj}(t), \mathrm{Adj}_{\theta}(t), \mathrm{Adj}_{t}(t)\right]. \tag{16}$$

Furthermore, we can get

$$\frac{\mathrm{dAdj}_{\Omega}}{\mathrm{d}t} = -\left[\mathrm{Adj}(t), \mathrm{Adj}_{\theta}(t), \mathrm{Adj}_{t}(t)\right] \frac{\partial f_{\mathrm{GNN}}}{\partial[(h(t), \theta(t), t)]}. \tag{17}$$

In summary, when every time the parameters are updated, we only need to solve three ODEs. We calculate $h(t_1)$, $\mathrm{Adj}(t)$, $\mathrm{Adj}_{\theta}(t)$, and $\mathrm{Adj}_{\theta}(t)$ sequentially. This way of calculation can complete the inverse propagation, reduce the number of parameters, and save memory space.

4. Experiments and Results

4.1. Experimental Data

4.1.1. Datasets. We verify the performance of GDENet on four real-world traffic datasets, namely, PeMS03, PeMS04, PeMS07, and PeMS08, which are collected by the California Department of Transportation Performance Measurement System (PeMS). They deploy more than 39,000 detectors on major highways in California. The original flow data are stored in data.npz, and the three-dimensional data features are flow, occupation, and speed. The original adjacency matrix is saved in distance.csv. The details are as follows.

(1) PeMS03 is gathered from the areas of California by 358 detectors. PeMS03 includes traffic data for 91 days from September to November 2018. The traffic data are aggregated into 5 minutes which instructs that there are 12 intervals for each hour.
(2) PeMS04 uses detectors to collect traffic flow data at detection points and collect traffic data for 59 days from January to February 2018. PeMS04 has 307 detectors collected at 5-minute intervals, and 288 sets of data are collected in one day.
(3) PeMS07 collects the traffic flow data through the detector and collects 98 days from May to August 2017. PeMS07 has 883 detectors, collecting 5 minutes as an interval, and 288 data are collected each day.
(4) PeMS08 is sampled from June to August 2016, spanning 62 days. There are 170 detectors, which are set in the region of California highways. In addition, the traffic data generate every 5 minutes, which means there are 288 data for each day. Among them, PeMS03 has the maximum number of nodes, which leads to a most complicated adjacency matrix.

4.1.2. Data Visualization. To enhance the analysis of traffic flow data, we conduct a data visualization experiment on the PEMS04 dataset. No. 224 is selected as the observational target for this experiment, and the traffic flow data and speed data are represented in Figure 5. Figure 5 shows the original traffic flow data in (a), while (b) showcases the original speed data. A comparison of the two data is illustrated in (c). Obvious from the line chart is the substantial fluctuation range exhibited by traffic flow data, which further exhibits intricate spatial-temporal characteristics. Conversely, speed data exhibit a comparatively smaller fluctuation range and thus poses lesser difficulty in prediction.

4.2. Baseline Methods. We compare the GDENet model with other excellent traffic prediction models, including the followings:

(1) LSTM [52]: long short-term memory is one of the classical recurrent neural networks, which is originally used in translation tasks. LSTM is suitable for dealing with some sequence problems.
(2) DCRNN [38]: the diffusion convolutional recurrent neural network treats traffic prediction as a diffusion process. The DCRNN proposes diffuse convolution to discover spatial attributes.
(3) STGCN [37]: the spatiotemporal graph convolutional network constructed the transportation network with graph. The STGCN extracts spatial properties by GCN and mines temporal features with the gated CNN.
(4) ASTGCN [39]: the attention-based spatial-temporal graph convolutional network introduces attention mechanism to traffic prediction. The ASTGCN uses GCN to extract spatial features while using ordinary convolution to learn temporal attributes.
(5) Graph WaveNet [43]: Graph WaveNet constructs a temporal convolution layer for temporal feature extraction. In addition, Graph WaveNet uses dilated causal convolution to increase the receptive field.

4.3. Experimental Settings and Details. Experiments are conducted under the environment with one Intel (R) Core (TM) i7-7700 CPU @ 3.60 GHz and NVIDIA GeForce GTX 1080 GPU card. We split all datasets into the training set, validation set, and test set at a ratio of 3 : 1 : 1. We set the historical window size to 12 and use the data of 12-time slices in the previous 60 minutes to predict the traffic of the 12-time slices in the next 60 minutes. We have selected three commonly used metrics in traffic flow forecasting for evaluation. When calculating these indicators, we exclude missing values. The evaluation metrics are defined as follows:

(1) Mean absolute error:

$$\mathrm{MAE} = \frac{1}{n} \sum_{t=1}^{n} \left| Y_{\mathrm{pred}_t} - Y_{\mathrm{true}_t} \right|. \tag{18}$$

(2) Mean absolute percentage error:

$$\mathrm{MAPE} = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{Y_{\mathrm{pred}_t} - Y_{\mathrm{true}_t}}{Y_{\mathrm{pred}_t}} \right|. \tag{19}$$

(3) Root mean square error:

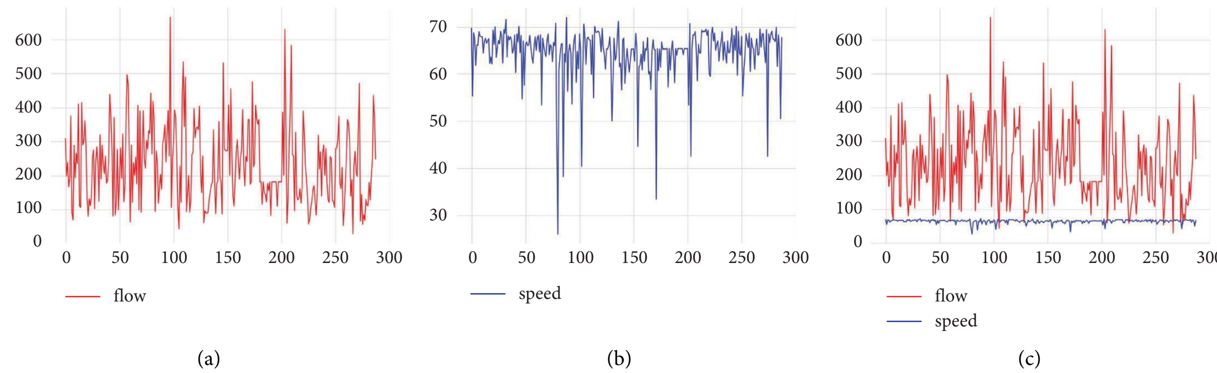

Figure 5: Data visualization of experimental data. (a) Line chart of flow data. (b) Line chart of speed data. (c) Comparison of flow data and speed data.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(Y_{\mathrm{pred}_t} - Y_{\mathrm{true}_t}\right)^2}. \tag{20}$$

4.4. Performance Comparison Experiment. We conduct experiments to verify that our model has lower memory occupancy. We use one-hour historical traffic flow data to predict the following one-hour traffic flow data. We count the parameters of each model when it is running. Also, we count the amount of memory occupation and the time required for the prediction of each model under the fixed condition. The final results are shown in Table 1.

As can be seen, GDENet occupies less memory and has higher computing efficiency, leading to a higher practical value. It demonstrates the effectiveness of the STFI. Because of adopting ODE, GDENet uses a known mathematical model to solve the gradient. Besides, compared with the traditional neural network model, a large number of parameters in the back propagation process are omitted, resulting in lower memory occupancy. Concerning the DCRNN, it replaces the matrix product with a convolution operation, thus a substantial increase in calculation time. Furthermore, we add a mathematical model to the GDENet which reduces the uncertainty of the network model and saves memory occupied. Detailed experiments and analyses disclose the advantages and defects of previous models, which illustrate GDENet's outstanding overall performance. At last, compared with other models, our model has higher computational efficiency and is more in line with the needs of practical applications. It is straightforward to see that GDENet better comprehends the traffic flow data and fits the tasks very well.

4.5. Data Comparison Experiment. We compare the prediction effects of our model with the baseline model by conducting experiments on the public dataset. The experimental conditions are the same as performance comparison experiments. Table 2 shows the comparison of those methods of evaluation metrics, from which we can draw the following conclusions.

First, the MAE is the highest on PeMS07 because it has a complicated adjacency matrix. As the traffic road dataset becomes more complex, the prediction error increases. The adjacency matrix with a large number of nodes brings vast difficulties to the prediction task. The traffic road of PeMS08 is relatively simple; therefore, the prediction effect is excellent. Second, we observe that GNN-based models, including STGCN, DCRNN, ASTGCN, and Graph WaveNet, generally outperform LSTM. We argue that this is because the GNN has a good effect on spatial feature extraction. This phenomenon proves the importance of discovering spatial properties. The GDENet can effectively utilize spatial structure to make more accurate predictions. The intrinsic spatial attributes help GDENet achieve a higher-quality prediction effect and performance improvement. Besides, our GDENet outperforms other GNN-based models, emphasizing that GDENet can learn the latent relevance of temporal and spatial. The spatiotemporal correlation helps our model obtain the best prediction accuracy on the same dataset. Moreover, our GDENet has a lower RMSE, which implies our prediction results conform to reality. Finally, compared with other methods, our GDENet suffers less from the problem of distribution deviation and the prediction effect infers that our model can mine more information and improve performance.

4.6. Qualitative Comparison of Resource Consumption. In this section, we compare the resource consumption of GDENet with other methods based on the PeMS04 dataset. We report the model average test time and GPU memory usage for a more intuitive and efficient comparison, as shown in Figure 6. Specifically, we compare the resource consumption of GDENet, STGODE [28], DCRNN [38], STGCN [37], LSTM [52], ASTGCN [39], and Graph WaveNet [43] intuitively. All experiments are performed under the same conditions. Among them, STGODE also uses ODE and GNN algorithms.

Intuitively, Figure 6 shows that GDENet achieves good prediction results, and our proposed GDENet achieves better performance and higher efficiency than other state-of-the-art baselines. This is mainly because the GDENet focuses on developing a more reasonable model structure rather

Table 1: Performance comparison between different methods (↓).

Metrics	LSTM	DCRNN (ICLR2018)	STGCN (ICLR2018)	ASTGCN (AAAI2019)	GWnet (IJCAI2019)	STGODE (KDD2021)	GDENet (ours)
MAE	27.09	24.68	22.99	23.06	25.40	**20.84**	21.08
Parameter	262977	373313	270104	450031	292248	714504	**251805**
Memory	1.3 G	1.9 G	2.2 G	2.2 G	2.2 G	2.0 G	**1.2 G**
Test time	97 s	1088 s	132 s	151 s	141 s	115 s	**73 s**

Bold values indicate the best performances.

Table 2: Comparison experiments between different methods on the four datasets (↓).

Datasets	Metric	LSTM	DCRNN	STGCN	ASTGCN	GWnet	GDENet
PeMS03	MAE	21.46	18.25	**17.81**	18.68	19.80	19.18
	MAPE	24.17	19.07	**17.15**	20.46	19.55	18.57
	RMSE	35.46	30.33	30.12	29.68	32.75	**28.28**
PeMS04	MAE	27.09	24.68	22.99	23.06	25.40	**21.08**
	MAPE	18.20	17.12	**14.59**	16.56	17.29	19.06
	RMSE	41.59	38.12	35.55	35.22	39.70	**32.83**
PeMS07	MAE	30.48	**24.57**	25.88	30.41	27.19	30.88
	MAPE	13.33	11.31	**11.10**	15.57	12.50	14.91
	RMSE	46.32	**37.94**	38.55	45.79	42.78	44.57
PeMS08	MAE	22.25	**17.80**	18.03	18.41	19.09	18.61
	MAPE	14.34	11.48	11.44	12.64	12.51	**11.25**
	RMSE	33.08	**27.81**	27.99	28.39	31.02	28.35

Bold values indicate the best performances.

than increasing the parameters. Although the STGODE achieves the minimum error, the resource consumption is a vast amount of parameters and longer train-run time. Overall, compared with advanced methods, our method can achieve competitive performance and our resource consumption is the lowest, which indicates the superiority of our method.

4.7. Sample Distribution Experiment. Under the same conditions of Section 4.4, we calculate the divergence between the prediction samples generated by each model and the overall samples and compare the level of deviation between the two samples. We employ JS divergence and KL divergence to express the deviation of the sampling distribution. KL divergence is also called relative entropy or KL distance. For the similarity between the two probability distributions, P and Q, the more similar the two, the smaller the KL divergence. JS divergence is a variant of KL divergence, i.e., similar to KL divergence. The details are as follows:

$$\begin{aligned} D_{\mathrm{KL}}(P|Q) &= \sum_i P(i)\log\frac{P(i)}{Q(i)}, \\ D_{\mathrm{JS}}(P|Q) &= \frac{1}{2}D_{\mathrm{KL}}\left(P\left|\frac{P+Q}{2}\right.\right)+\frac{1}{2}D_{\mathrm{KL}}\left(Q\left|\frac{P+Q}{2}\right.\right), \end{aligned} \tag{21}$$

where P is the actual distribution of the data, Q is the prediction distribution of the data, $P(i)$ represents the probability distribution function of the data, and $Q(i)$ represents the probability distribution function of the prediction distribution. The result is shown in Table 3. The proposed GDENet suffers less from the problem of distribution deviation. The prediction samples generated by GDENet are closer to the real world situation. To gain a more intuitive understanding of the distribution deviation, we present the results in Figure 7. The results demonstrate that GDENet exhibits the best performance across both D_{KL} and D_{JS} indicators. We apply ODE to the model in the time dimension, making the discrete GNN model continuous, which indicates that GDENet utilizes intrinsic spatiotemporal association effectively.

4.8. Ablation Study. To explain the predictive ability of the GDENet more clearly, we conduct ablation experiments on PEMS04 and compare GDENet with the following variants under the same condition.

(1) Gate structure model: this is GDENet without STFI. We replace the STFI with an MLP.
(2) Gate-GAT model: we remove the ODE module from GDENet.
(3) Gate-ODE model: we remove the GAT from GDENet.

The experimental results are shown in Table 4. We can observe that GDENet excels at the other approaches distinctly. The Gate-GAT model performs next to GDENet, emphasizing the importance of reducing distribution deviation errors. The poor performance of the Gate-ODE model demonstrates the effectiveness of spatial feature extraction. The introduction of GAT significantly improves the performance, which provides the function of exploring spatial attributes. Experimental results of the gate structure model infer that STFI has significant advantages in capturing spatiotemporal correlations.

4.9. Prediction Effectiveness Experiment. In order to observe the prediction results of our model more intuitively, we choose node no. 224 as the observation object. We use one-hour historical traffic flow data to predict future one-hour traffic flow data. The comparison line between predicted and actual values is shown in Figure 8. Each figure shows the prediction result at different times on the 49th day of the PeMS04 dataset.

First, we find that the prediction curve of GDENet can align with the ground-truth curve in general. It shows that the fitting effect of our model is superior, which can accommodate the fluctuation of traffic flow data. Then, we speculate that the leading cause is that our GDENet adopted a STFI, which can mine potential spatiotemporal association. The association contributes to predicting traffic congestion and other traffic phenomena, which improves the model's perception of peaks and turning points. In addition, the prediction results in the future 15 minutes are superior. This is because GDENet contained a gate structure that gives the model the capability to capture complicated long-term temporal attributes. However,

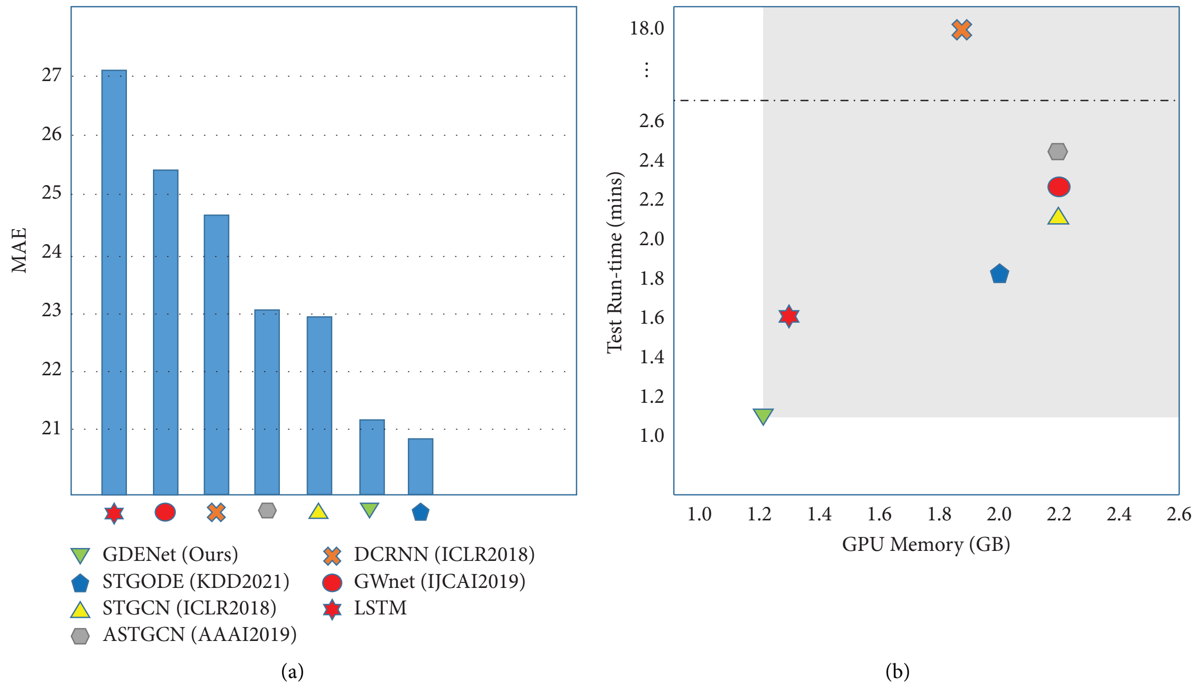

Figure 6: Qualitative comparison of resource consumption on PeMS04 datasets. (a) Comparison of mean absolute error. (b) Comparison of resource consumption.

Table 3: Sample distribution experiment between different methods on the PEMS04 dataset (↓).

Metrics	LSTM	DCRNN	STGCN	ASTGCN	GWnet	GDENet
MAE	27.09	24.68	22.99	23.06	25.40	**21.08**
D_{KL}	0.000802	0.000773	0.000746	0.000702	0.000725	**0.000641**
D_{JS}	0.003329	0.003126	0.003051	0.002799	0.002931	**0.002570**

Bold values indicate the best performances.

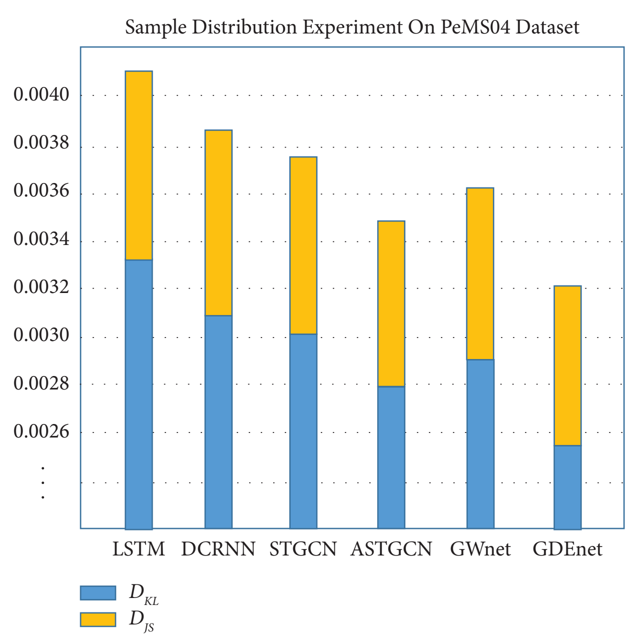

Figure 7: Sample distribution experimental results.

Table 4: Ablation study on the PEMS04 dataset (↓).

Metrics	Gate structure	Gate-GAT	Gate-ODE	GDENet
MAE	28.30	26.89	33.78	**21.08**
MAPE	20.25	25.46	28.34	**19.06**
RMSE	41.44	38.81	47.11	**32.83**

Bold values indicate the best performances.

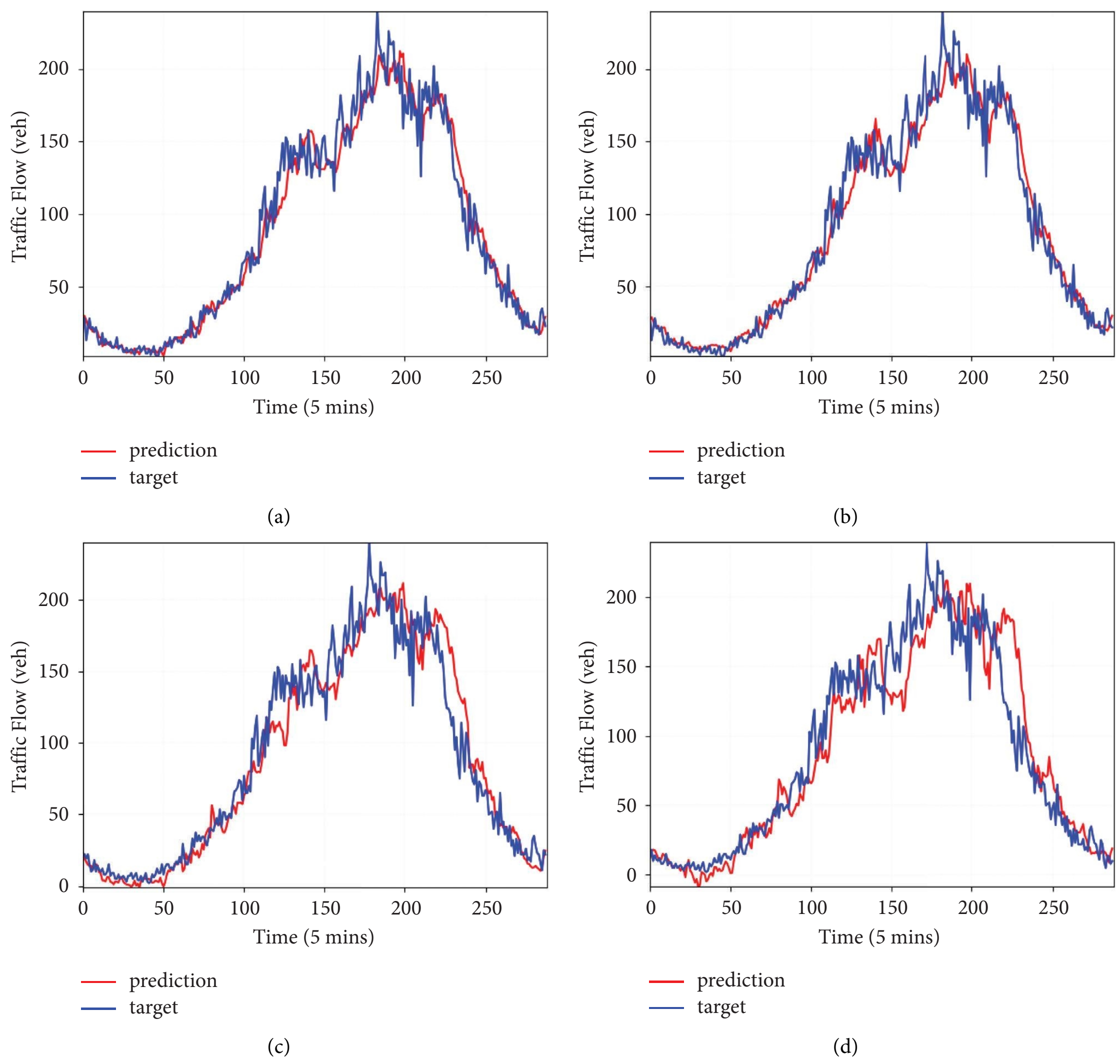

Figure 8: Prediction results of our method at different times. (a) Prediction effectiveness in the next 5 mins. (b) Prediction effectiveness in the next 15 mins. (c) Prediction effectiveness in the next 30 mins. (d) Prediction effectiveness in the next 60 mins.

as the prediction time intervals exceed the future 30 minutes, the error gradually increases. This result from many existing works proves that long-term traffic flow is still a thorny problem because the temporal properties become increasingly nonlinear with the growth of the time intervals.

5. Conclusion and Future Work

In this paper, we propose GDENet to predict traffic flow and achieve excellent performance. First, we learn temporal properties through the gate structure. Then, we mine spatial attributes with the aid of the GNN and propose STFI to explore spatiotemporal correlations. The STFI can alleviate distribution bias and make more accurate predictions with the help of ordinary differential equations. Finally, we execute experiments on real-world datasets, which validate the feasibility and the performance of GDENet. Combining mathematical methods with deep learning methods has become a research hotspot. In future research, we hope to establish a connection between differential equations and neural network models, adopting the differential equations to explain the "black box." We can further increase the interpretability of our method and optimize neural network models in a targeted manner.

Differential equations can also be used to build new neural network models and promote the development of the field of deep learning.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

The views and conclusions expressed in this paper are those of the authors and do not necessarily reflect the views of any of the sponsors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All the authors have approved the manuscript and agreed with submission to the esteemed journal.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (no. 62162008), Innovation and Entrepreneurship Project for Overseas Educated Talents in Guizhou Province ((2022)-04), Guizhou Provincial Science and Technology Projects (ZK[2022]-108), and Guizhou University Cultivation Project (no. 2021-55). This research was also supported by the Natural Science Special Research Fund of Guizhou University (no. 2021-24).

References

[1] Y. Zhang, S. Wang, B. Chen, J. Cao, and Z. Huang, "Trafficgan: network-scale deep traffic prediction with generative adversarial nets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 219–230, 2021b.

[2] S. Wang, J. Cao, and P. S. Yu, "Deep learning for spatio-temporal data mining: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3681–3700, 2022.

[3] C. Hu, S. Xiao, L. Gao, and M. Liu, "Tracing the spatial-temporal evolution dynamics of air traffic systems using graph theories," *International Journal of Intelligent Systems*, vol. 37, no. 10, pp. 8021–8045, 2022.

[4] X. Yu, X. Che, Q. Mao, Z. Gong, W. Fu, and X. Zheng, "Pf-its: intelligent traffic service recommendation based on deepafm model," *International Journal of Intelligent Systems*, vol. 37, no. 10, pp. 7638–7664, 2022.

[5] A. Ali, Y. Zhu, and M. Zakarya, "Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction," *Neural Networks*, vol. 145, pp. 233–247, 2022.

[6] J. Cuena, "Intelligent systems for traffic flow management: a qualitative modeling approach," *International Journal of Intelligent Systems*, vol. 7, no. 2, pp. 133–153, 1992.

[7] B. Du, H. Peng, S. Wang et al., "Deep irregular convolutional residual lstm for urban traffic passenger flows prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 972–985, 2020.

[8] H. Wang, R. Liu, P. Wang, G. Liu, H. Wang, and L. Yan, "Intelligent optimization of dynamic traffic light control via diverse optimization priorities," *International Journal of Intelligent Systems*, vol. 36, no. 11, pp. 6748–6762, 2021.

[9] X. Zhou, Y.-A. Geng, H. Yu et al., "Lightnet+: a dual-source lightning forecasting network with bi-direction spatiotemporal transformation," *Applied Intelligence*, vol. 1, 2022b.

[10] W. Liao, B. Zeng, J. Liu, P. Wei, and X. Cheng, "Taxi demand forecasting based on the temporal multimodal information fusion graph neural network," *Applied Intelligence*, vol. 14, 2022.

[11] Y. Qin, F. Zhao, Y. Fang, H. Luo, and C. Wang, "Memory attention enhanced graph convolution long short-term memory network for traffic forecasting," *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 6555–6576, 2022.

[12] M. Last, G. Avrahami, and A. Kandel, "Using data mining techniques for optimizing traffic signal plans at an urban intersection," *International Journal of Intelligent Systems*, vol. 26, no. 7, pp. 603–620, 2011.

[13] P. Cheng, M. Han, A. Li, and F. Zhang, "Stc-ids: spatial–temporal correlation feature analyzing based intrusion detection system for intelligent connected vehicles," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 9532–9561, 2022.

[14] B. Tu, Y. Zhao, G. Yin, N. Jiang, G. Li, and Y. Zhang, "Research on intelligent calculation method of intelligent traffic flow index based on big data mining," *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1186–1203, 2022.

[15] Z. Yang, Q. Li, H. Xie, Q. Wang, and W. Liu, "Learning representation from multiple media domains for enhanced event discovery," *Pattern Recognition*, vol. 110, Article ID 107640, 2021.

[16] H. Peng, H. Wang, B. Du et al., "Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting," *Information Sciences*, vol. 521, pp. 277–290, 2020.

[17] J. An, L. Guo, W. Liu et al., "IGAGCN: information geometry and attention-based spatiotemporal graph convolutional networks for traffic flow prediction," *Neural Networks*, vol. 143, pp. 355–367, 2021.

[18] L. Yu, B. Du, X. Hu, L. Sun, L. Han, and W. Lv, "Deep spatio-temporal graph convolutional network for traffic accident prediction," *Neurocomputing*, vol. 423, pp. 135–147, 2021.

[19] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[20] R. Huang, C. Huang, Y. Liu, G. Dai, and W. Kong, "Lsgcn: long short-term traffic prediction with graph convolutional networks," *IJCAI*, vol. 7, pp. 2355–2361, 2020.

[21] L. Zhao, Y. Song, C. Zhang et al., "T-GCN: a temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.

[22] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[23] Q. Wang, X. Liu, W. Liu, A. Liu, W. Liu, and T. Mei, "Metasearch: incremental product search via deep meta-learning," *IEEE Transactions on Image Processing*, vol. 29, pp. 7549–7564, 2020a.

[24] Q. Wang, H. Deng, X. Wu et al., "Lcm-captioner: a lightweight text-based image captioning method with collaborative mechanism between vision and text," *Neural Networks*, vol. 162, pp. 318–329, 2023a.

[25] H. Zhou, H. Zhao, Q. Wang et al., "EMO-MVS: error-aware multi-scale iterative variable optimizer for efficient multi-view stereo," *Remote Sensing*, vol. 14, no. 23, p. 6085, 2022a.

[26] Q. Wang, J. Wang, H. Deng, X. Wu, Y. Wang, and G. Hao, "Aa-trans: Core attention aggregating transformer with information entropy selector for fine-grained visual classification," *Pattern Recognition*, vol. 140, Article ID 109547, 2023b.

[27] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," *NIPS 2018: Advances in Neural Information Processing Systems*, NeurIPS, pp. 6572–6583,Montréal, Canada, 2018.

[28] Z. Fang, Q. Long, G. Song, and K. Xie, "Spatial-temporal graph ode networks for traffic flow forecasting," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 364–373, Singapore, August 2021.

[29] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *Journal of Transportation Engineering*, vol. 121, no. 3, pp. 249–254, 1995.

[30] M. S. Ahmed and A. R. Cook, "Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques," 1979, https://onlinepubs.trb.org/Onlinepubs/trr/1979/722/722-001.pdf.

[31] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[32] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 865–873, 2015.

[33] L. Zheng, J. Yang, L. Chen, D. Sun, and W. Liu, "Dynamic spatial-temporal feature optimization with ERI big data for short-term traffic flow prediction," *Neurocomputing*, vol. 412, pp. 339–350, 2020.

[34] Y. Liu, H. Zheng, X. Feng, and Z. Chen, "Short-term traffic flow prediction with conv-lstm," in *Proceedings of the 2017 9th International Conference On Wireless Communications And Signal Processing (WCSP)*, IEEE, Nanjing, China, October 2017.

[35] C. H. Liu, C. Piao, X. Ma et al., "Modeling citywide crowd flows using attentive convolutional LSTM," in *Proceedings of the 37th IEEE International Conference on Data Engineering, ICDE 2021*, IEEE, Chania, Greece, April 2021.

[36] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the Thirty-first AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, February 2017a.

[37] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI 2018*, Stockholm, Sweden, July 2018.

[38] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: data-driven traffic forecasting," in *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*, Vancouver, BC, Canada, April 2018.

[39] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 922–929, Honolulu, HI, USA, February 2019.

[40] X. Rao, H. Wang, L. Zhang, J. Li, S. Shang, and P. Han, "FOGS: first-order gradient supervision with learning-based graph for traffic flow forecasting," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022*, Vienna, Austria, July 2022.

[41] X. Zhang, C. Huang, Y. Xu et al., "Traffic flow forecasting with spatial-temporal graph diffusion network," *AAAI Conference on Artificial Intelligence*, vol. 35, no. 17, pp. 15008–15015, 2021a.

[42] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "Pdformer: propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4365–4373, Washington, DC, USA, February 2023.

[43] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, S. Kraus, Ed., pp. 1907–1913, Macao, China, August 2019.

[44] C. Song, Y. Lin, S. Guo, and 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*, pp. 914–921, British Columbia, Canada, February 2020.

[45] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4189–4196, Palo Alto, CA, USA, February 2021.

[46] D. Cao, Y. Wang, J. Duan et al., "Spectral temporal graph neural network for multivariate time-series forecasting," in *Proceedings of the NeurIPS 2020*, Vancouver BC Canada, December, 2020.

[47] J. Ji, J. Wang, C. Huang et al., "Spatio-temporal self-supervised learning for traffic flow prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 27, pp. 4356–4364, 2023.

[48] Y. Gong, Z. Li, J. Zhang, W. Liu, and Y. Zheng, "Online spatio-temporal crowd flow distribution prediction for complex metro system," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 865–880, 2022.

[49] J. Ou, J. Sun, Y. Zhu et al., "Stp-trellisnets+: spatial-temporal parallel trellisnets for multi-step metro station passenger flow prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, pp. 7526–7540, 2023.

[50] Z. Shao, Z. Zhang, W. Wei et al., "Decoupled dynamic spatial-temporal graph neural network for traffic forecasting," *Proceedings of the VLDB Endowment*, vol. 15, no. 11, pp. 2733–2746, 2022.

[51] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li, "DSTAGNN: dynamic spatial-temporal aware graph neural network for traffic flow forecasting," in *Proceedings of the International Conference on Machine Learning, ICML 2022*, pp. 17–23, Baltimore, ML, USA, July 2022.

[52] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[53] B. Shiva Prakash, K. Sanjeev, R. Prakash, and K. Chandrasekaran, "A survey on recurrent neural network architectures for sequential learning," in *Soft Computing for Problem Solving: SocProS*, pp. 57–66, Springer, Springer, Singapore, 2019.

[54] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *ICLR 2018 Conference Program Chairs*, vol. 1050, p. 20, 2017.

[55] J. Bai, J. Zhu, Y. Song et al., "A3T-GCN: attention temporal graph convolutional network for traffic forecasting," *ISPRS International Journal of Geo-Information*, vol. 10, no. 7, p. 485, 2021.

[56] S. Wang, J. Cao, and P. S. Yu, "Deep learning for spatio-temporal data mining: a survey," *IEEE transactions on knowledge and data engineering*, vol. 34, pp. 3681–3700, 2020.

[57] Y. Lu, A. Zhong, Q. Li, and B. Dong, "Beyond finite layer neural networks: bridging deep architectures and numerical differential equations," in *Proceedings of the International Conference on Machine Learning*, pp. 3276–3285, Vienna, Austria, July 2018.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, June 2016.

[59] F. R. Villatoro and J. I. Ramos, "On the method of modified equations. I: asymptotic analysis of the euler forward difference method," *Applied Mathematics and Computation*, vol. 103, no. 2-3, pp. 111–139, 1999.

[60] X. Zhang, Z. Li, C. Change Loy, and D. Lin, "Polynet: a pursuit of structural diversity in very deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 2017b.

[61] K. B. Fadimba, "A linear backward euler scheme for the saturation equation: regularity results and consistency," *Journal of Computational and Applied Mathematics*, vol. 234, no. 1, pp. 272–282, 2010.

[62] G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: ultra-deep neural networks without residuals," in *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017*, Toulon, France, April 2017.

[63] R. F. Warming and B. Hyett, "The modified equation approach to the stability and accuracy analysis of finite-difference methods," *Journal of Computational Physics*, vol. 14, no. 2, pp. 159–179, 1974.

[64] K. Roy, "The primitive function," in *Technohumanism, Global Crises, and Education: Toward a Posthuman Pedagogy*, pp. 155–171, Springer, Berlin, Germany, 2022.