Spatial-Temporal PDE Networks for Traffic Flow Forecasting

Tianshu Bao[1], Hua Wei[2], Junyi Ji[1], Daniel Work[1], and Taylor Thomas Johnson[1]

[1] Vanderbilt University, Nashville, TN 37235, US
{tianshu.bao, junyi.ji, dan.work, taylor.johnson}@vanderbilt.edu
[2] Arizona State University, Tempe, AZ 85281, US
hua.wei@asu.edu

Abstract. Spatial-temporal forecasting is crucial in various domains, including traffic flow prediction for Intelligent Transportation Systems (ITS). Despite the challenges posed by complex spatial-temporal dependencies in traffic networks, Partial Differential Equations (PDEs) have proven effective for capturing traffic dynamics. However, recent trends favor data-driven approaches like Graph Neural Networks (GNNs) for traffic forecasting, often overlooking the principles described by PDEs. In this paper, we propose a Graph Partial Differential Equation Network (GPDE) that integrates PDE principles with GNNs to enhance traffic flow forecasting. Our approach leverages dynamic graph structures based on PDE flux functions, incorporating residual connections and learnable rates for improved model performance. Extensive experiments on real-world traffic datasets demonstrate the superiority of GPDE over existing methods in both short-term and long-term traffic speed prediction tasks.

1 Introduction

Spatial-temporal forecasting has garnered significant attention in recent years due to its widespread applications in various domains such as traffic flow forecasting [11, 28], climate prediction [13], and more. Accurate spatial-temporal forecasting plays a crucial role in enhancing the service quality of these applications. In this paper, we focus on traffic flow forecasting, an essential component of Intelligent Transportation Systems (ITS), which aims to predict future traffic flow based on historical traffic conditions and underlying road networks.

Predicting traffic flow is particularly challenging due to the complex and long-range spatial-temporal dependencies inherent in traffic networks. The travel distances of different individuals vary significantly [21], resulting in both nearby and distant spatial dependencies coexisting simultaneously. Moreover, the interaction between spatial attributes and temporal patterns poses a formidable challenge for traffic flow forecasting.

Partial Differential Equations (PDEs) have emerged as powerful tools for capturing the spatial and temporal variations in traffic density and flow, enabling a dynamic representation of traffic behavior. Notably, in traffic flow modeling,

well-known PDEs such as the first-order Lighthill-Whitham-Richards (LWR) model [6] and the second-order Aw-Rascle-Zhang (ARZ) model [32] are commonly employed to describe the conservation of vehicles within roadway segments. These PDEs facilitate the understanding of complex traffic phenomena including congestion, shockwaves, and traffic propagation.

While PDEs provide a solid framework for traffic flow modeling, recent years have seen a surge of interest in data-driven methods, such as Graph Neural Networks (GNNs), for traffic forecasting and other applications [3,16,30]. GNNs excel in handling graph-structured data, enabling effective aggregation of node representations from their neighbors. Although many GNN-based models, like STGCN [30] and DCRNN [16], have been proposed to extract spatial features in traffic networks, they often overlook the underlying traffic flow laws described by PDEs.

In this paper, we propose a novel approach termed Graph Partial Differential Equation Network (GPDE) to address the challenges in traffic flow forecasting. Our method leverages the knowledge from PDEs to guide the design of a dynamic graph neural network layer, enabling the modeling of interacting traffic flows within a traffic network. Specifically, we introduce a dynamic adjacency matrix based on the PDE flux function to capture spatial correlations from both geographical and semantic perspectives. Additionally, we incorporate residual connections inspired by residual networks [12] to mitigate the over-smoothing problem often encountered in GNNs. Furthermore, we introduce weighted learnable rates to control parameter updates and a non-homogeneous term to account for unobserved flows in traffic networks. We integrate these components into a PDE layer and augment existing baselines to demonstrate the superiority of our model using real-world traffic datasets.

The main contributions of this work are as follows:

- We propose a novel dynamic graph neural network layer, GPDE, to model traffic networks with interacting traffic flows, improving the performance of existing baselines.
- We leverage the knowledge from PDEs to guide the design of our evolving graph structure, incorporating residual connections and learnable rates to enhance model effectiveness.
- We conduct extensive experiments to evaluate our method, demonstrating its superiority in both short-term and long-term traffic speed prediction tasks compared to existing methods.

2   Related Work

Recent advancements in traffic prediction techniques have seen the application of various neural network architectures to capture the complex spatial and temporal dependencies inherent in traffic data. Traditional approaches have utilized RNNs and CNNs to encode temporal and spatial dependencies, respectively [17, 28, 29, 31, 33]. More recently, the adoption of Graph Neural Net-

works (GNNs) has been proposed to more effectively learn traffic network patterns [14,18,34]. Notable contributions include the Graph Recurrent Neural Network (GRNN) [26] and the Diffusion Convolutional Recurrent Neural Network (DCRNN) [16], which offer sophisticated models for understanding traffic flow and spatial-temporal dynamics. Furthermore, advancements like STGCN [30], GraphWaveNet [27], and ASTGCN [11] have separately addressed the spatial and temporal dependencies in traffic prediction. The STSGCN model [24] represents a significant step forward by integrating spatial and temporal blocks to model these dependencies synchronously. However, the challenge of over-smoothing in deep GNNs limits their ability to capture long-range spatial-temporal relationships [15]. In contrast, the STGODE model [8] utilizes a tensor-based ordinary differential equation approach to overcome these limitations, though it primarily addresses temporal dynamics without incorporating spatial information. [5] proposes the spatio-temporal graph neural controlled differential equation (STGNCDE), which designs two neural controlled differential equations: one for the temporal processing and the other for the spatial processing.

Integrating physics with machine learning models has shown promise in enhancing predictive performance and generalizability in scientific domains [2]. Physics-based ML models have been explored for traffic state estimation [7,20], offering a novel perspective by combining deep learning with physics information, though these approaches have not fully leveraged graph structures [23]. Our work seeks to distinguish itself by simulating spatial dependencies through PDE knowledge within a GNN framework, thereby addressing the spatial aspect more effectively than the ODE-based STGODE model [8].

3 Preliminaries

3.1 Problem Formulation

We aim to model traffic flow dynamics within a connected road network, representable by a graph structure $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$, where $\mathcal{V}$ denotes the set of N nodes (road segments), $\mathcal{E}$ the connections among these nodes, and $\mathbf{A}$ the adjacency matrix. Each node i at time t has an observation $\mathbf{x}_i^t \in \mathbb{R}^F$, with F being the feature length. The full observation set at time t is $X^t = (\mathbf{x}_1^t, \mathbf{x}_2^t, ..., \mathbf{x}_N^t) \in \mathbb{R}^{N \times F}$, and the entire series of observations over time is $\mathcal{X} = (X^1, X^2, ..., X^T) \in \mathbb{R}^{T \times N \times F}$. The task is to predict future traffic observations T' based on past observations T, formally defined as:

$$[X^{t-T+1}, X^{t-T+2}, ..., X^t; \mathcal{G}] \xrightarrow{f} [X^{t+1}, X^{t+2}, ..., X^{t+T'}] \tag{1}$$

3.2 Network Architectures

GRNN The GRNN [26] incorporates graph convolution into an LSTM mechanism as follows:

$$\begin{aligned}
\mathbf{h}^{t+1} &= \sigma(\mathbf{W}_h^h \mathbf{A} \mathbf{h}^t + \mathbf{b}_h), \\
\mathbf{c}^{t+1} &= \sigma(\mathbf{W}_c^h \mathbf{A} \mathbf{c}^t + \mathbf{b}_c),
\end{aligned} \tag{2}$$

where $\mathbf{A}$ is the adjacency matrix, $\mathbf{h}^t$ and $\mathbf{c}^t$ are the hidden state and cell state at time t, respectively, and σ denotes the activation function. Model parameters include weight matrices $\mathbf{W}$ and bias vectors $\mathbf{b}$.

DCRNN The DCRNN model [16] integrates diffusion convolution with recurrent neural network mechanisms to capture the dynamics of traffic flow. The diffusion convolution operation at each time step is formulated as follows:

$$\mathbf{H}^{(l+1)} = \sigma(\sum_{k=0}^{K-1} \mathbf{W}_k^{(l)} (\mathbf{D}^{-1}\mathbf{A})^k \mathbf{H}^{(l)} + \mathbf{b}^{(l)}) \tag{3}$$

where $\mathbf{H}^{(l)} \in \mathbb{R}^{N \times F_l}$ is the hidden state matrix at layer l, $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix with its degree matrix $\mathbf{D}$, $\mathbf{W}_k^{(l)}$ is the weight matrix for the k-th power of the normalized adjacency matrix at layer l, $\mathbf{b}^{(l)}$ is the bias term, σ denotes the activation function, and K is the maximum diffusion step. This equation captures the spatial dependency through diffusion convolution, modeling traffic flow as a diffusion process across the network.

STGCN The STGCN [30] architecture operates through a series of spatio-temporal convolutional blocks, where each block is designed to capture both spatial and temporal dependencies. The key operation in a block is defined as:

$$\mathbf{Z} = \sigma(\mathbf{W}_t * (\sigma(\mathbf{W}_s\mathbf{X}\mathbf{A} + \mathbf{b}_s)) + \mathbf{b}_t) \tag{4}$$

where $\mathbf{X} \in \mathbb{R}^{N \times F}$ is the input feature matrix, $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix, $\mathbf{W}_s$ and $\mathbf{W}_t$ are the spatial and temporal convolutional weight matrices, respectively, $\mathbf{b}_s$ and $\mathbf{b}_t$ are bias terms, σ denotes the activation function, and $*$ represents the convolution operation in the temporal dimension.

4   Methods

In the Methods section, we introduce the GPDE, a novel framework designed to enhance traffic flow forecasting by integrating the principles of PDEs into different spatio-temporal prediction models like GRNN, STGCN, and DCRNN. This section outlines the underlying theory of our approach, the formulation of the PDE layer, and its integration with established GNN architectures, offering a comprehensive overview of how GPDE systematically addresses the complexities of traffic flow forecasting.

4.1   Vanilla PDE for Traffic Flow

In tackling the complex challenge of traffic flow forecasting, our approach hinges on the foundational use of PDEs to model the continuous dynamics of traffic movement. PDEs offer a robust framework for encapsulating the spatial and

temporal variations observed in traffic flow, making them indispensable for understanding and predicting traffic behavior. Specifically, we consider the ARZ model [1, 32] for traffic flow, derived from the principles of fluid dynamics as a cornerstone for our method, bridging the gap between theoretical physics and practical traffic forecasting. Other PDE methods like [22] could also be integrated in a similar way.

ARZ model The ARZ model is a well-acknowledged representation in traffic flow theory to capture the non-linear dynamics of traffic flows, including the crucial aspects of vehicle conservation and momentum, which are essential for accurate traffic prediction. Specifically, we consider the Zhang model [32]. Our implementation of the model incorporates source functions to account for external influences on traffic flow, such as ramps or intersections, not directly observable through data:

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial (\rho v)}{\partial s} &= H_1, \\
\frac{\partial v}{\partial t} + (v + \rho V^{'}(\rho))\frac{\partial v}{\partial s} &= H_2,
\end{aligned}
\tag{5}
$$

where $\rho(s,t)$ denotes the traffic density, $v(s,t)$ the traffic speed, and $H_1(s,t), H_2(s,t)$ are non-homogeneous source functions representing the traffic flow variations. The equilibrium traffic speed profile, $V(\rho)$, is described using Greenshield's linear model [10], which simplifies the relationship between traffic density and speed:

$$
V(\rho) = v_f(1 - (\frac{\rho}{\rho_{max}})^{\gamma}),
\tag{6}
$$

where v_f is the free flow speed, ρ_{max} the maximum traffic density, and γ a parameter governing the density-speed relationship. For our study, we select $\gamma = 1$, yielding a linear model that simplifies the subsequent computations.

With the assumption of constant traffic density over short forecasting horizons, we can reformulate the Zhang model to focus on predicting traffic speed. Essentially, we reduce the model back to the Lighthill-Whitham-Richards (LWR) [6] model and write it in a velocity format. This simplification leads to a more tractable PDE for integration within GNN architectures:

$$
\frac{\partial v}{\partial t} + \frac{\partial f(v)}{\partial s} = H,
\tag{7}
$$

where $f(v) = \frac{v^2}{2} - v_f\frac{\rho}{\rho_{max}}v$ represents the **flux function** capturing the movement of traffic through the network, and $H = H_2$ embodies external influences on the traffic flow. This equation lays the foundation for our proposed GPDE layer, allowing the dynamic modeling of traffic flow within the structured framework of GNNs, thus bridging theoretical models with practical forecasting needs.

4.2   Discrete-time PDE Solutions

In tackling the challenge of traffic flow forecasting, our approach necessitates transitioning from the continuous dynamics captured by PDEs to a discrete-time computational model that aligns with the practicalities of traffic networks. This transition is vital for bringing theoretical models to bear on real-world forecasting tasks. To achieve this, we employ numerical methods to discretize the PDE presented in Eq. 7, focusing on finite difference schemes that provide a practical means to approximate traffic speed changes over discrete intervals in both time and space. This section delves into the discretization process for two distinct scenarios: traffic flow on a linear roadway and within a more complex network topology.

Linear roadway discretization Our first discretization scheme is tailored for linear roadway scenarios, such as highways, where traffic flow can be approximated as moving along a straight path. The scheme is formulated as follows:

$$v_i^{t+1} = v_i^t - \frac{\Delta t}{\Delta s_i}(f_i^t - f_{i-1}^t) + H_i^t \Delta t, \tag{8}$$

where v_i^t denotes the speed at node i and time t, $f_i^t - f_{i-1}^t$ represents the flux difference at node i between two discrete time steps, and H_i^t encompasses external factors affecting traffic flow. Δs_i indicates the spatial difference between consecutive nodes, derived from the adjacency matrix $\mathbf{A}$, and Δt specifies the time step. This equation updates traffic states along a linear path, effectively capturing traffic dynamics as observed in datasets like the I-24 MOTION project [9].

Network discretization Extending the discretization to accommodate complex traffic networks involves adapting the scheme to account for the flow at intersections and merges, characteristic of urban and suburban environments:

$$v_i^{t+1} = v_i^t - \frac{\Delta t}{\Delta s_{i+1} + \Delta s_i}(f_{i+1}^t - f_{i-1}^t) + H_i^t \Delta t, \tag{9}$$

where Eq. 9 uses a central difference scheme to approximate $\frac{\partial f(v)}{\partial s}$. $\Delta s_{i+1}, \Delta s_i$ is the distance of node i to downstream node $i+1$ and upstream node $i-1$.

Since we consider node i in a network, it must have multiple upstream nodes and downstream nodes. In this case, the flux difference becomes the difference between all incoming flows and outgoing flows. Also, we double the denominator number because the distance is asymmetric in the graph using the central difference scheme. This approach is refined to reflect the directional nature of traffic flow through network nodes:

$$v_i^{t+1} = v_i^t - \left(\frac{\Delta t}{2\Delta s_j} \sum_j^{n_{dn}} f_j^t - \frac{\Delta t}{2\Delta s_k} \sum_k^{n_{up}} f_k^t \right) + H_i^t \Delta t, \tag{10}$$

where n_{up} and n_{dn} represent the numbers of upstream and downstream nodes connected to node i, respectively. This detailed approach captures the essence of network-based traffic dynamics, where each node's traffic state is influenced by its immediate neighborhood.

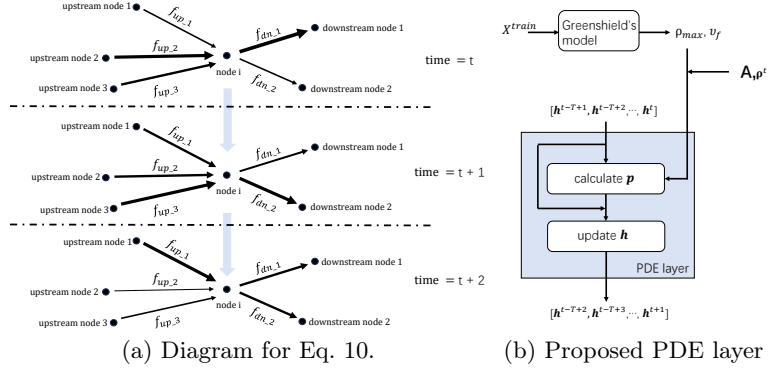(a) Diagram for Eq. 10.   (b) Proposed PDE layer

Fig. 1: (a) Flux functions f_{dn}, f_{up} evolve accordingly. (b) Visualization of the Graph PDE Layer's architecture. It includes how the layer processes input states through the combined mechanisms of flux difference calculation, non-homogeneous term adjustment, and weighted residual updating to produce the next state in the traffic prediction sequence.

Boundary conditions Accurately simulating the dynamics of traffic flow systems, such as those represented by our model, necessitates the careful consideration of boundary conditions. These conditions are essential for ensuring that the solution to the PDE is well-defined and physically plausible. For traffic flow forecasting within a network, boundary conditions consider the traffic behavior at edge road segments in a network or linear roadways, which can significantly differ from internal network dynamics.

We apply the Neumann boundary condition [19], which specifies the spatial derivative of the traffic speed at the network's boundaries. Specifically, we set the derivative to zero: $v^{'}(L, t) = 0$, where L denotes the boundary points of the network. This reflects the understanding that the traffic speed at the outermost points of a network does not experience direct changes due to external traffic entering or exiting the system. The Neumann condition aligns with the physical intuition that the outer edges of a traffic network are insulated from external speed variations, thus stabilizing the model at the network's extremes.

Fig. 1(a) conceptually illustrates the application of our discretization schemes within a traffic network. It highlights the dynamic interplay between different network nodes and underscores the continuous updates of traffic states based on calculated flux differences, laying a methodological foundation for incorpo-

rating PDE dynamics into GNN architectures for enhanced traffic forecasting capabilities.

4.3   Graph PDE Layer

Having established the foundation for applying PDEs to graph structures, in this section, we introduce the Graph PDE Layer which integrates PDEs into GNN architectures.

State update of PDE output $\mathbf{p}^t$ In typical GNN operations, the hidden states ($\mathbf{h}^t$) represent the traffic state at each node within the network at time t. The evolution of these states is influenced by two primary factors: the flux differences F arising from the traffic flow dynamics and the non-homogeneous terms $\mathbf{H}$ accounting for external traffic influences.

To adeptly manage the contribution of these factors to the state updates, we introduce two learnable rates, r_1 and r_2 to modulate the impact of flux differences and the non-homogeneous term, respectively. This approach leads to the following formulation for the state update within the Graph PDE Layer:

$$\mathbf{p}^t = \phi(\mathbf{h}^t + r_1 \cdot F(\mathbf{h}^t, \boldsymbol{\rho}^t, \rho_{max}, v_f, \mathbf{A}) + r_2 \cdot \mathbf{H}), \tag{11}$$

where $\phi(\cdot)$ represents the activation function, chosen to be the sigmoid function for its properties conducive to modeling nonlinearities inherent in traffic dynamics. The term $F(\mathbf{h}^t, \boldsymbol{\rho}^t, \rho_{max}, v_f, \mathbf{A})$ computes the flux differences based on the current state, traffic densities ($\boldsymbol{\rho}^t$), the maximum density (ρ_{max}), free flow speed (v_f), and the spatial relationships as delineated by the adjacency matrix ($\mathbf{A}$). The non-homogeneous term ($\mathbf{H}$), treated as an external influence, is modeled as a learnable and time-invariant tensor, providing a flexible mechanism to incorporate external traffic influences not directly observable from the data.

Final update of state $\mathbf{h}^t$ To further refine the model and incorporate the concept of residual learning, we propose the following weighted combination of the updated state ($\mathbf{p}^t$) and the previous state ($\mathbf{h}^t$):

$$\mathbf{h}^{t+1} = \alpha \cdot \mathbf{p}^t + \beta \cdot \mathbf{h}^t, \tag{12}$$

The weights α and β are learnable parameters that allow the network to balance the influence of the PDE-based updates with the preservation of information from the previous state. This mechanism not only facilitates the integration of the dynamic traffic flow information encoded by the PDEs but also ensures that the model can adaptively learn the importance of historical versus newly computed states for accurate traffic flow prediction.

Overall flow of a PDE layer The flow of operations within the proposed PDE layer is summarized in Algorithm 1, outlining the sequential steps undertaken for each epoch of model training. This includes the retrieval and update of hidden states, the application of the PDE layer for state evolution, and the iterative adjustment of both network and physical parameters to refine the model's predictive performance.

Algorithm 1 The flow of the proposed PDE layer

Input: hidden states $\mathbf{h}^t$, maximum traffic flow density ρ_{max}, current traffic flow density $\boldsymbol{\rho}^t$, free flow speed v_{max}, adjacent matrix $\mathbf{A}$, trainable non-homogeneous term $\mathbf{H}$, trainable weights r_1, r_2, α, β.

for $epoch = 1$: number of iterations **do**
    for $t = 1$: T **do**
        Get hidden states $\mathbf{h}^t$ from the previous steps
        Get $\mathbf{p}^t$ through Eq. 11
        Get $\mathbf{h}^{t+1}$ through Eq. 12
    end for
    Update model parameters (i.e., networks weights) and physical parameters (i.e., $\mathbf{H}$ and α)
end for

4.4 Integrating PDE layer with GNNs

With the development of the GPDE layer, we aim to augment traditional GNN architectures with the capability to simulate traffic dynamics informed by PDEs. This integration enables GNNs to not only leverage the structural information present in traffic networks but also incorporate the underlying physical principles governing traffic flow. Here, we detail the integration of the GPDE layer with specific GNN architectures, including Graph Recurrent Neural Networks (GRNN), Spatio-Temporal Graph Convolutional Networks (STGCN), and Diffusion Convolutional Recurrent Neural Networks (DCRNN).

GRNN For GRNN [26], which utilizes LSTM mechanisms to capture spatial-temporal dependencies, integrating the GPDE layer involves substituting the adjacency matrix $\mathbf{A}$ with dynamics encoded by the GPDE layer. This adjustment enhances the model's capacity to process spatial-temporal traffic patterns effectively: $\mathbf{h}^{t+1} = \mathrm{PDE}(\mathbf{h}^t, \boldsymbol{\rho}^t, \rho_{max}, v_f, \mathbf{A}, \mathbf{H}), \mathbf{c}^{t+1} = \mathrm{PDE}(\mathbf{c}^t, \boldsymbol{\rho}^t, \rho_{max}, v_f, \mathbf{A}, \mathbf{H})$.

DCRNN DCRNN [16] utilizes a sequence-to-sequence framework with diffusion convolution for modeling traffic flow dynamics. Integrating the GPDE layer into DCRNN entails placing it before the DCGRU layers in both the encoder and decoder components. This configuration empowers DCRNN with enhanced

Fig. 2: Illustration for the I-24 RDS dataset on the linear roadways [35].

predictive capabilities grounded in the physical behaviors captured by the GPDE layer: $\mathbf{h}^{t+1}_{\text{enc/dec}} = \text{DCGRU}_{\text{enc/dec}}(\text{GPDE}(\mathbf{h}^t, \boldsymbol{\rho}^t, \rho_{max}, v_f, \mathbf{A}, \mathbf{H}))$, where $\mathbf{h}^{t+1}_{\text{enc/dec}}$ are the updated hidden states within the encoder/decoder processed by the DC-GRU layer, which now incorporates inputs processed by the GPDE layer.

STGCN STGCN's [31] design is geared towards analyzing traffic data through spatial-temporal convolution blocks. To incorporate the GPDE layer within STGCN, it is positioned immediately before the spatial graph-convolution layer within each spatial-temporal convolution block. This positioning allows the network to preprocess input data through the lens of the GPDE layer, integrating physical traffic dynamics before spatial convolutional processing. The operation can be formalized as $\mathbf{X}^t_{\text{GPDE}} = \text{GPDE}(\mathbf{X}^t, \boldsymbol{\rho}^t, \rho_{max}, v_f, \mathbf{A}, \mathbf{H})$, where $\mathbf{X}^t_{\text{GPDE}}$ denotes the output of the GPDE layer, serving as the input to the subsequent spatial graph-convolution layer.

5   Experiments

In this section, we describe the experiment details and results. All experiments are conducted using TensorFlow and PyTorch on a computer with the following configuration: Intel Core i7-8750H CPU @2.20GHz × 6 Processor, 16 GiB Memory, GeForce GTX 1060, 64-bit Win10 OS. [3]

5.1   Datasets and Baselines

Datasets

Datasets We evaluate the proposed method on two different traffic datasets, a linear roadway dataset on the Nashville I-24 Radar Detector System (I-24 RDS) [35] and a network dataset on the Caltrans Performance Measurement System (PeMSD8) [4].

---

[3] The code for this paper can be found at `https://drive.google.com/drive/folders/1FzKPIfORu54vQ2oWDSFjh4tvepIFIEVW?usp=drive_link`

• **I-24 RDS dataset** This dataset contains the radar detection traffic data in Nashville from April 1 to April 30 in 2023, with 44 sensors on the I-24 road with a time interval of 30 seconds. Every observation contains 5 features: speed, occupancy, volume, smooth speed, and smooth occupancy. Since occupancy is linearly related to density and the GPDE contains linear transformations, we simply use occupancy as the GPDE input. We use the first 20 days of data to train the models and use the remaining 10 days of data to test the models. A description of the sensor layout for the road section can be found in [35].

• **PeMSD8 dataset** This dataset contains the traffic data in San Bernardino from July to August in 2016, with 170 detectors on 8 roads with a time interval of 5 minutes. Three kinds of traffic measurements were considered, including total flow, average speed, and average occupancy. We use the first 40 days of data to train the models and use the remaining 22 days of data to test the models.

Baselines We compare our proposed method with existing state-of-the-art methods, including GRNN [26], STGCN [31], and DCRNN [16]. Their implementations are supported by LibCity [25] toolbox, which is a comprehensive and extensible library for traffic prediction. For each model and its GPDE-enhanced variant, we apply distinct learning rates for experiments on the I-24 RDS and PeMSD8 datasets, with a consistent epoch count of 50 across all setups. Specifically, GRNN and GPDE(GRNN) use learning rates of 0.003 and 0.01 for I-24 RDS and PeMSD8 respectively, maintaining a hidden state size of 20. STGCN and GPDE(STGCN) employ learning rates of 0.001 and 0.01 for the same datasets, with block sizes set to [[1, 8, 16], [16, 8, 16]], and other parameters at default values. Similarly, DCRNN and GPDE(DCRNN) are configured with learning rates of 0.001 and 0.01 for I-24 RDS and PeMSD8 respectively, featuring an RNN unit size of 16, while adhering to default settings for additional parameters as specified by LibCity. The hyperparameters are selected by experience. The instructions for running the experiments can be found on LibCity's website.

5.2 Performance Evaluation

We report the short-term and long-term testing performance of different methods for traffic speed prediction in two datasets. For evaluation metrics, each experiment is conducted five times with random model initialization, and the mean of the Mean Square Error (MSE), Mean Absolute Error (MAE), and Rooted Mean Square Error (RMSE) are reported.

Short-term prediction Table 1 demonstrates the performance for short-term prediction on two datasets. We have the following observations:

• Our proposed GPDE performs consistently better than the backbone models without GPDE layers. This is because the PDE layers simulate the traffic in-and-out flows in the graph. The PDE flux works as a regularization term to force the model to follow the traffic rules for both short-term and long-term predictions.

Table 1: Predictive performance of short-term speed prediction in I-24 RDS dataset and PeMSD8 dataset. GDPE with different backbone models consistently performs better with the lower MSE, MAE, and RMSE.

Method	Metric	I-20 MOTION Dataset					PeMSD8 Dataset				
		30 sec	60 sec	90 sec	120 sec	150 sec	5 min	10 min	15 min	20 min	25 min
GRNN	MSE	29.747	34.63	38.159	42.747	48.263	12.263	16.662	18.317	17.152	19.377
	RMSE	5.454	5.884	6.177	6.538	6.947	3.501	4.081	4.279	4.141	4.401
	MAE	3.779	4.031	4.190	4.364	4.576	2.337	2.517	2.700	2.531	2.646
GPDE(GRNN)	MSE	22.877	28.475	35.854	39.077	46.821	2.268	4.545	6.557	8.536	10.424
	RMSE	4.782	5.336	5.987	6.251	6.842	1.505	2.131	2.561	2.921	3.228
	MAE	3.338	3.586	3.897	4.233	4.524	0.818	1.084	1.249	1.389	1.522
STGCN	MSE	18.858	18.969	19.055	19.290	19.324	3.043	3.407	3.696	4.011	4.176
	RMSE	4.342	4.355	4.365	4.392	4.395	1.744	1.845	1.922	2.002	2.043
	MAE	2.907	2.918	2.927	2.946	2.951	0.975	**1.028**	**1.071**	**1.112**	**1.138**
GPDE(STGCN)	MSE	17.499	**17.604**	**17.622**	**17.657**	**17.782**	2.953	**3.238**	**3.462**	**3.778**	**3.959**
	RMSE	4.183	**4.195**	**4.197**	**4.202**	**4.216**	1.718	**1.799**	**1.860**	**1.943**	**1.989**
	MAE	2.781	**2.787**	**2.788**	**2.792**	**2.804**	0.994	1.037	1.074	1.117	1.145
DCRNN	MSE	16.184	21.114	25.118	28.619	31.799	2.477	4.673	6.764	8.738	10.550
	RMSE	4.022	4.594	5.011	5.349	5.639	1.573	2.161	2.601	2.956	3.248
	MAE	**2.742**	3.054	3.243	3.382	3.508	0.846	1.095	1.267	1.404	1.516
GPDE(DCRNN)	MSE	**16.029**	20.644	24.393	27.849	30.913	**2.155**	4.339	6.410	8.352	10.069
	RMSE	**4.003**	4.543	4.938	5.277	5.559	**1.467**	2.083	2.531	2.889	3.173
	MAE	2.749	3.068	3.247	3.385	3.504	**0.802**	1.059	1.231	1.366	1.472

Table 2: Predictive performance of long-term speed prediction in I-24 RDS dataset and PeMSD8 dataset. GPDE(STGCN) consistently outperforms the other baseline methods.

Methods	Metric	I-20 MOTION Dataset			PeMSD8 Dataset		
		3 min	4.5 min	6 min	30 min	45 min	60 min
STGCN	MSE	19.413	19.644	19.828	4.343	4.724	4.992
	RMSE	4.406	4.432	4.452	2.083	2.173	2.234
	MAE	2.961	2.988	3.009	**1.161**	**1.219**	**1.257**
GPDE(STGCN)	MSE	**17.829**	**18.006**	**18.153**	**4.149**	**4.643**	**4.986**
	RMSE	**4.222**	**4.243**	**4.260**	**2.036**	**2.154**	**2.232**
	MAE	**2.811**	**2.839**	**2.858**	1.170	1.236	1.275
DCRNN	MSE	34.907	43.504	50.922	12.263	16.6159	19.919
	RMSE	5.908	6.595	7.135	3.501	4.076	4.463
	MAE	3.622	3.911	4.118	1.616	1.856	2.042
GPDE(DCRNN)	MSE	33.837	41.982	48.919	11.619	15.378	18.189
	RMSE	5.816	6.479	6.994	3.408	3.921	4.264
	MAE	3.607	3.86	4.065	1.564	1.78	1.944

Especially in the I-24 RDS dataset, the traffic propagation characteristic lines are better preserved by our model.

• The GPDE(STGCN) method performs the best over the other 2 types of approaches and preserves the accuracy in multistep time predictions. While GRNN does not perform well due to the structure simplicity with only one layer of LSTM, it shows great potential after adding the PDE layer in the PeMSD8 dataset. DCRNN does not perform well because of the diffusion process, which over-smooths the pattern learned by the neural network, and oversimplifies the traffic flows in the network.

Table 3: Ablation experiments of GPDE(STGCN) on PeMSD8 dataset, evaluated for 30-min prediction. Other time steps have similar performances.

Metric	GPDE	GPDE wo. source function	GPDE wo. weighted rates	GPDE wo. residual
MSE	**4.148**	4.215	5.141	6.320
RMSE	**2.036**	2.053	2.267	2.514
MAE	1.170	**1.161**	1.269	1.358

Long-term prediction Table 2 summarizes the performance of long-term predictions for all the methods. We have the following observations:

• The GPDE models show significant improvement in long-term predictions. This improvement is attributed to the PDE regularization effect, which presumably helps in modeling complex spatial-temporal dynamics more accurately.

• For STGCN and DCRNN models, there is a notable reduction in prediction error when transitioning from short-term to long-term forecasts. Specifically, the Mean Squared Error (MSE) reduces from 0.2 in short-term predictions to approximately 2 in long-term predictions. This error reduction indicates the effectiveness of GPDE in enhancing the model's predictive capability over longer horizons. The improvement is partly due to the innovative use of PDE network convolution with weighted residual connections. This technique enhances the model's ability to capture long-term temporal and spatial dependencies, which are crucial for accurate long-term forecasting in complex systems.

5.3   Model Analysis

Ablation study In order to assess the impact of various components within the GPDE framework, we undertook a series of ablation studies using the PeMSD8 dataset. These studies were designed to dissect the GPDE model by systematically removing specific elements, thereby creating three distinct variants on the base model, STGCN:

- GPDE without source function: This variant omits the non-homogeneous function $\mathbf{H}$ in Eq. 11.
- GPDE without learnable rates: This model excludes the weighted learnable rates α, β in Eq. 12.
- GPDE without residual correction: This configuration removes $\mathbf{h}^t$ in Eq. 12.

The results in Table 3 show that the complete GPDE (STGCN) model achieved superior performance among the evaluated variants, highlighting the efficacy of its novel spatial convolution and residual correction mechanisms. Specifically, the absence of residual correction in one of the variants significantly hindered performance, underscoring the crucial role of this component in the model. Additionally, the variant without weighted rates showcased the significance of controlled updates in the model's performance.

Table 4: The performance of GPDE(DCRNN) and DCRNN as the network depth increases. GPDE(DCRNN) is more robust than DCRNN in all aspects.

Methods	Metric	1 layer	2 layers	3 layers
DCRNN	MSE	2.477	4.843	3.662
	RMSE	1.573	2.201	1.913
	MAE	0.846	1.208	1.015
GPDE(DCRNN)	MSE	**2.155**	2.322	2.446
	RMSE	**1.467**	1.523	1.563
	MAE	**0.802**	0.835	0.868

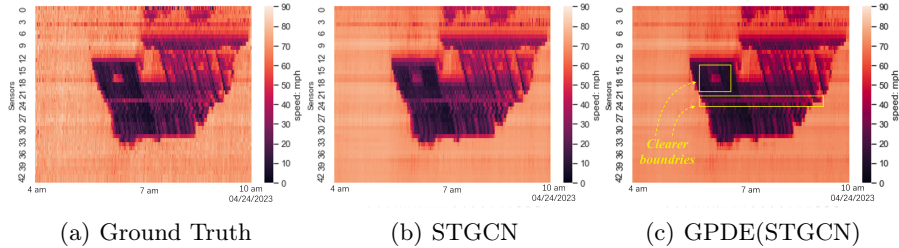(a) Ground Truth   (b) STGCN   (c) GPDE(STGCN)

Fig. 3: Visualization of the long-term 30-min prediction made by (b) STGCN, (c) GPDE(STGCN) on the I-24 RDS dataset on 04/24/2023. The x-axis presents the time, the y-axis represents the sensor location. The dark parts represent the low-speed area and traffic congestion. The congestion propagation can be observed. In (c), the visualization is clearer around traffic changes, indicating a better performance of GPDE(STGCN) over STGCN.

Parameter analysis One major advantage of our GPDE model over other existing methods is performance stability to over-smoothing and thus can construct deeper network structures. In Table 4, we represent the performance of the DCRNN-based models under different depths, which is the number of RNN layers. We can see that as the network depth increases, the performance of DCRNN drops dramatically while the performance of GPDE(DCRNN) is stable, which clearly shows the strong robustness of our model towards over-smoothing.

5.4 Case Study

We present the heatmaps for the I-24 RDS dataset because it is a straight freeway. Fig. 3 shows the 6 time steps predictions. In Fig. 3(b), the traffic propagation lines are better preserved than (c) because PDE is better at capturing such traffic flow property in long-term temporal and spatial dependencies. DCRNN utilizes a diffusion process that is susceptible to few nearby neighbors and thus performs unstably.

For the PeMSD8 dataset, we visualize the predictions of different methods on two locations from the road network. As Fig. 4 shows, the prediction results also show that GPDE methods are better at capturing traffic oscillations.

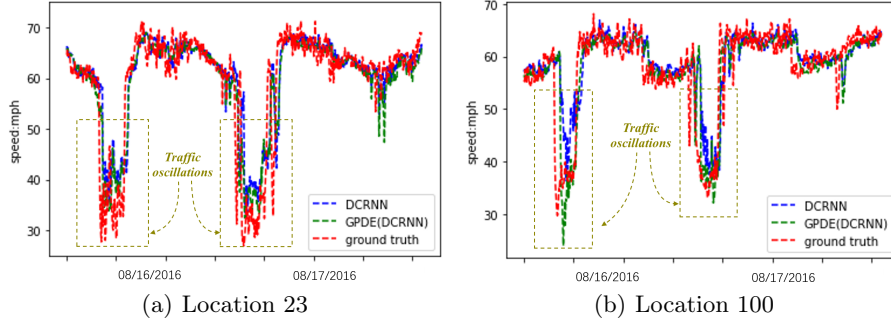(a) Location 23   (b) Location 100

Fig. 4: Comparisons on predictions made by DCRNN and GPDE(DCRNN) with ground truth for two locations (a) location 23, and (b) location 100 on PeMSD8 dataset on 08/16/2016 - 08/17/2016. The green line is closer to the red line (ground truth) than the blue line (DCRNN) especially under oscillations, indicating GPDE(DCRNN) performs better than DCRNN.

6   Conclusion

A tremendous number of works have been proposed to tackle complex spatial-temporal problems, but none of them focus on utilizing physical knowledge to model dynamic traffic flows on the network. In this paper, we present a novel PDE-based spatial-temporal forecasting model named GPDE. To the best of our knowledge, this is the first attempt to bridge physics equations to the node representations of road networks in the area of traffic, which enables us to construct deeper networks and leverage wider-range dependencies. Furthermore, the participation of residual correction and weighted learnable rates largely enhance the model's performance. Extensive experiments prove the effectiveness of GPDE over existing methods on different time spans and road networks. Since this work relies on numerical experiments, future work can include verifying the theoretical correctness of the model.

Acknowledgments

References

1. Aw, A., Rascle, M.: Resurrection of" second order" models of traffic flow. SIAM journal on applied mathematics **60**(3), 916–938 (2000)
2. Bao, T., Chen, S., Johnson, T.T., Givi, P., Sammak, S., Jia, X.: Physics guided neural networks for spatio-temporal super-resolution of turbulent flows. In: Uncertainty in Artificial Intelligence. pp. 118–128. PMLR (2022)

3. Bao, T., Jia, X., Zwart, J., Sadler, J., Appling, A., Oliver, S., Johnson, T.T.: Partial differential equation driven dynamic graph networks for predicting stream water temperature. In: 2021 IEEE International Conference on Data Mining (ICDM). pp. 11–20. IEEE (2021)
4. Chen, C., Petty, K., Skabardonis, A., Varaiya, P., Jia, Z.: Freeway performance measurement system: mining loop detector data. Transportation Research Record **1748**(1), 96–102 (2001)
5. Choi, J., Choi, H., Hwang, J., Park, N.: Graph neural controlled differential equations for traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 6367–6374 (2022)
6. Claudel, C.G., Bayen, A.M.: Guaranteed bounds for traffic flow parameters estimation using mixed lagrangian-eulerian sensing. In: 2008 46th Annual Allerton Conference on Communication, Control, and Computing. pp. 636–645. IEEE (2008)
7. Di, X., Shi, R., Mo, Z., Fu, Y.: Physics-informed deep learning for traffic state estimation: A survey and the outlook. Algorithms **16**(6), 305 (2023)
8. Fang, Z., Long, Q., Song, G., Xie, K.: 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 (2021)
9. Gloudemans, D., Wang, Y., Ji, J., Zachar, G., Barbour, W., Hall, E., Cebelak, M., Smith, L., Work, D.B.: I-24 motion: An instrument for freeway traffic science. Transportation Research Part C: Emerging Technologies **155**, 104311 (2023)
10. Greenshields, B.D., Bibbins, J., Channing, W., Miller, H.: A study of traffic capacity. In: Highway research board proceedings. vol. 14, pp. 448–477. Washington, DC (1935)
11. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 922–929 (2019)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Jones, N.: How machine learning could help to improve climate forecasts. Nature **548**(7668) (2017)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
15. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
16. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
17. Liu, Q., Wu, S., Wang, L., Tan, T.: Predicting the next location: A recurrent model with spatial and temporal contexts. In: AAAI'16 (2016)
18. Long, Q., Jin, Y., Wu, Y., Song, G.: Theoretically improving graph neural networks via anonymous walk graph kernels. In: Proceedings of the Web Conference 2021. pp. 1204–1214 (2021)
19. Ng, M.K., Chan, R.H., Tang, W.C.: A fast algorithm for deblurring models with neumann boundary conditions. SIAM Journal on Scientific Computing **21**(3), 851–866 (1999)
20. Pan, Y.A., Guo, J., Chen, Y., Li, S., Li, W., et al.: Incorporating traffic flow model into a deep learning method for traffic state estimation: A hybrid stepwise modeling framework. Journal of Advanced Transportation **2022** (2022)

21. Plötz, P., Jakobsson, N., Sprei, F.: On the distribution of individual daily driving distances. Transportation research part B: methodological **101**, 213–227 (2017)
22. Rascle, M.: An improved macroscopic model of traffic flow: derivation and links with the lighthill-whitham model. Mathematical and computer modelling **35**(5-6), 581–590 (2002)
23. Shi, R., Mo, Z., Di, X.: Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 540–547 (2021)
24. Song, C., Lin, Y., Guo, S., Wan, H.: Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 914–921 (2020)
25. Wang, J., Jiang, J., Jiang, W., Li, C., Zhao, W.X.: Libcity: An open library for traffic prediction. In: Proceedings of the 29th international conference on advances in geographic information systems. pp. 145–148 (2021)
26. Wang, X., Chen, C., Min, Y., He, J., Yang, B., Zhang, Y.: Efficient metropolitan traffic prediction based on graph recurrent neural network. arXiv preprint arXiv:1811.00740 (2018)
27. Wu, Z., Pan, S., Long, G., et al.: Graph wavenet for deep spatial-temporal graph modeling. arXiv:1906.00121 (2019)
28. Yao, H., Tang, X., Wei, H., Zheng, G., Li, Z.: Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 5668–5675 (2019)
29. Yao, H., Wu, F., Ke, J., et al.: Deep multi-view spatial-temporal network for taxi demand prediction. In: AAAI (2018)
30. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875 (2017)
31. Yu, R., Li, Y., Shahabi, C., Demiryurek, U., Liu, Y.: Deep learning: A generic approach for extreme condition traffic forecasting. In: SDM'17 (2017)
32. Zhang, H.M.: A non-equilibrium traffic model devoid of gas-like behavior. Transportation Research Part B: Methodological **36**(3), 275–290 (2002)
33. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017)
34. Zhang, X., Huang, C., et al.: Spatial-temporal convolutional graph attention networks for citywide traffic flow forecasting. In: CIKM'18 (2020)
35. Zhang, Y., Quinones-Grueiro, M., Zhang, Z., Wang, Y., Barbour, W., Biswas, G., Work, D.: Marvel: Multi-agent reinforcement-learning for large-scale variable speed limits. arXiv preprint arXiv:2310.12359 (2023)