Reaction-Diffusion Graph Ordinary Differential Equation Networks: Traffic-Law-Informed Speed Prediction under Mismatched Data

Yue Sun
Lehigh University
Bethlehem, USA
yus516@lehigh.edu

Chao Chen
Lehigh University
Bethlehem, USA
cha01nbox@gmail.com

Yuesheng Xu
New York University
New York, USA
xuyuesheng324@gmail.com

Sihong Xie
Lehigh University
Bethlehem, USA
xiesihong1@gmail.com

Rick S. Blum
Lehigh University
Bethlehem, USA
rb0f@lehigh.edu

Parv Venkitasubramaniam
Lehigh University
Bethlehem, USA
pav309@lehigh.edu

ABSTRACT

Accurate traffic speed prediction is critical to many applications, from routing and urban planning to infrastructure management. With sufficient training data where all spatio-temporal patterns are well-represented, machine learning models such as Spatial-Temporal Graph Convolutional Networks (STGCN), can make reasonably accurate predictions. However, existing methods fail when the training data distribution (e.g., traffic patterns on regular days) is different from test distribution (e.g., traffic patterns on special days). We address this challenge by proposing a traffic-law-informed network called Reaction-Diffusion Graph Ordinary Differential Equation (RDGODE) network, which incorporates a physical model of traffic speed evolution based on a reliable and interpretable reaction-diffusion equation that allows the RDGODE to adapt to unseen traffic patterns. We show that with mismatched training data, RDGODE is more robust than the state-of-the-art machine learning methods in the following cases. (1) When the test dataset exhibits spatio-temporal patterns not represented in the training dataset, the performance of RDGODE is more consistent and reliable. (2) When the test dataset has missing data, RDGODE can maintain its accuracy by intrinsically imputing the missing values.

KEYWORDS

Traffic speed prediction, graph neural networks, spatial-temporal time series prediction

ACM Reference Format:
Yue Sun, Chao Chen, Yuesheng Xu, Sihong Xie, Rick S. Blum, and Parv Venkitasubramaniam. 2023. Reaction-Diffusion Graph Ordinary Differential Equation Networks: Traffic-Law-Informed Speed Prediction under Mismatched Data. In *Proceedings of ACM SIGKDD (UrbComp '23).* ACM, Long Beach, CA, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Traffic speed prediction [1, 26, 39] in road networks based on historical observations has continued to be a topic of great interest, given its myriad uses in the transportation sector. Machine learning approaches [39] have provided the most accurate predictions given sufficient training data that contains comprehensive traffic patterns that are likely to appear in test situations. Among the best-performing machine learning models, graph-based neural networks [11, 27, 32, 36] dominate due to their ability to incorporate spatio-temporal information so that dependent traffic speeds sensed at different locations and times can be modeled and exploited to make more accurate predictions. The predictive models, especially those based on deep learning trained with a large amount of data, tend to work well *only* when the training and test data have similar distributions [37, 38]. However, collecting representative training data is challenging in many practical situations [9, 31] because we can only sample in regular everyday conditions which are limited, while the model is expected to work in exceptional circumstances. For example, natural disasters (e.g., earthquakes or hurricanes) are rare events where traffic patterns can be significantly and abruptly altered. At best extreme situations [12] can be simulated, but these cannot truly capture the patterns in an actual event.

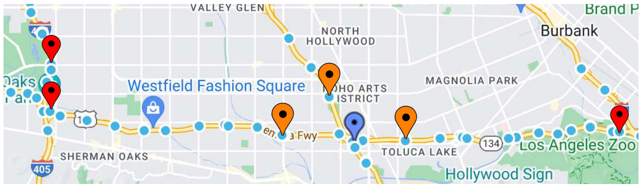


Figure 1: The most important sensors under mismatched data (red markers) for the traffic speed prediction at Sensor 718141 (blue marker) are located far away. However, the most important sensors under matched data (orange markers) are close to the target sensor. Each blue dot is a sensor with available data.

As a motivating example, we train a well-known graph learning approach, Spatial Temporal Graph Convolutional Network (STGCN), using traffic speed data on weekdays and test the model on weekends. To better understand how STGCN makes its prediction, we use GNNExplainer [35] to identify the most influential sensors on the

UrbComp '23, August 06–10, 2023, Long Beach, CA

ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/XXXXXXX.XXXXXXX

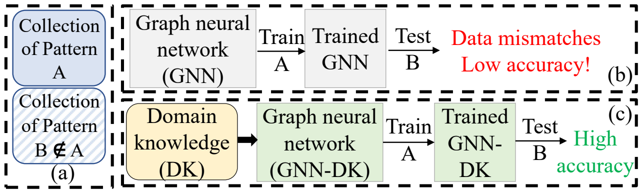


Figure 2: (a) Two collections of patterns (i.e., pattern A (exists in a known dataset) and pattern B (difficult to be collected and only available at test time)) in the training and test datasets have no overlap; (b) Without incorporating a traffic law, testing the model with such mismatched patterns may result in poorer accuracy; (c) With an architecture using a traffic law, the model can still achieve good accuracy when adapting to unseen patterns.

graph that contributed to a particular prediction. When the training and test data are drawn from the same distribution (weekdays), the outcome of GNNExplainer showed that the most influential sensors are very close to the target sensor whose speed measurement is being predicted. However, when the graph learning model is tested on a different distribution (weekend data), the outcome of GNNExplainer showed that the most influential sensors are often physically far from the target sensor, implying that the prediction might not conform to any traffic evolution law. In the example shown in Figure 1, the three most important sensors under mismatched testing are geographically unreachable within the 5 minutes prediction window.

The above-mentioned challenge can be formulated as learning with mismatched training data [29] (Figure 2), a problem that is often encountered in practice in different domains.

We postulate that integrating a known traffic law into the machine learning based predictive model would enable the model to overcome this challenge. In this work, we propose a novel traffic-law-informed neural network called Reaction-Diffusion Graph Ordinary Differential Equation (RDGODE) network, that augments GCN with a differential equation based traffic speed evolution model studied in transportation research [2]. The traffic model, expressed as a reaction-diffusion equation, describes the general rule of the evolution of traffic speed. We develop an ML architecture informed by this traffic law and respect the underlying traffic dynamics. Consequently, even when the traffic patterns are altered between training and test data, the in-built dynamical relationship ensures that the prediction performance is not significantly impacted. Furthermore, the prior knowledge encoded by the traffic-law-informed architecture reduces the number of model parameters, thus requiring less training data. The model computations are better grounded in domain knowledge and are thus more accessible and interpretable to domain experts in transportation management.

The contributions of this work are as follows:

- We study the challenge of traffic speed prediction with mismatched data where the patterns in the training set are not representative of those in the test set. To address the challenge, we derive a novel traffic-law-informed graph-based machine learning model RDGODE that integrates a differential equation based traffic law into a graph convolutional network using Neural ODEs [3].
- Through extensive testing, we demonstrate (*i*) the prediction accuracy of RDGODE is more robust in situations with data mismatches compared to baseline models; (*ii*) RDGODE can react more quickly to rapid short-term variations; (*iii*) RDGODE is highly effective for data imputation, and can handle missing data in traffic speed prediction.

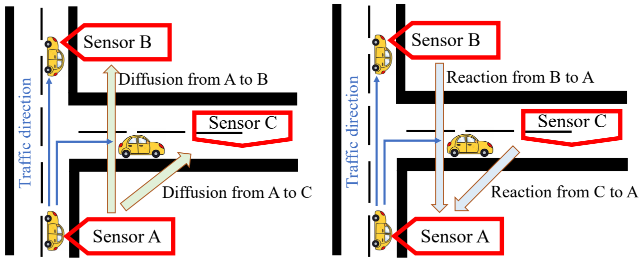
2 RELATED WORK

Graph Neural Networks. Graph Neural Networks (GNNs) have been widely utilized to enable great progress in dealing with graph-structured data [17]. [6, 20, 36] build spatio-temporal blocks to encode the traffic spatio-temporal features. [10, 11, 27, 30, 32] generate dependency graphs, which only focus on "data-based" dependency wherein traffic speed at a sensor can be influenced by a sensor, not in its physical vicinity. Recently, machine learning models [3, 4, 7, 16, 21] incorporating differential equations were proposed, to better capture the continuous spatial-temporal pattern. Specifically, a prediction model [16] using Neural Controlled Differential Equation [4] was proposed to handle irregular time series. The work [21] constructed a Recurrent Neural Network incorporating Reaction Kinetics to improve prediction accuracy by respecting the underlying physics. None of these approaches exploit traffic laws for better generalization and robustness.

Traffic Law and Application on Machine Learning. Modeling traffic with equations developed through physics has a long history [15, 23], These approaches focus on finding conservation laws through mathematics and experiments, and propose models that reflect the most essential relationships in traffic. Our approach relies on a specific network-level model studied recently [2, 19, 22], where they use a reaction-diffusion equation to model the traffic speed. The model specifically addresses the opposing forces that affect traffic flow, namely i) Diffusion which captures the impact on network nodes forward in the direction of travel, and ii) Reaction which captures the impact on nodes behind in the direction of travel owing to congestion. While incorporating a traffic law in machine learning is an area of growing interest [14], integration of traffic models with neural graphical models has not yet been explored and particularly, in the context of prediction with mismatched training data.

Mismatched Data. Meta-learning [8, 28] is often used to augment machine learning with limited data, through additional training processes. Mismatches [25] between the training and test sets are frequently present in practical applications. Robustness to mismatched data is important in designing trustworthy models [29]. [33] studied the optimizations of supervised learning when knowing the data difference between training and test sets. However, our approach addresses the challenge by incorporating a traffic law instead of using extra training processes, or employing additional assumptions on permutation thus works for arbitrarily mismatched scenarios.

Model Explainability. Intrinsically transparent ML models [18, 24] based on simple rules or linear models are useful in that their computation processes can be revealed to domain experts to increase their confidence in the models. In contrast, we incorporate non-linear physical laws in graphical models to promote intrinsic explainability. In graph-based ML, understanding how neighbors collaborate to make predictions. Prior methods, such as [34], use a surrogate model to approximate a graphical model and thus do not reveal the computational process of the traffic prediction model.



(a) Diffusion. **(b) Reaction.**

Figure 3: (a) Diffusion occurs in the direction of a road segment; (b) reaction occurs opposite to the direction of a road segment.

3 PROBLEM DEFINITION

The graphs used in this paper are derived from the highway network, and we call them the "physical graphs". The physical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed unweighted graph with $|\mathcal{V}| = n$ vertices and $|\mathcal{E}|$ edges. Each vertex corresponds to a sensor on the highway network. Each edge, denoted as $(i, j) \in \mathcal{E}$, represents the directional connectivity between two vertices $i, j \in \mathcal{V}$, and corresponds to a road segment where one can travel between the two neighboring sensors along the traffic direction (without passing other sensors). Let $\mathcal{A} \in \mathbb{R}^{n\times n}$ denote the adjacency matrix of the graph $\mathcal{G}$. Since each sensor is placed on one side of a road segment and measures the speed along that specific direction, $\mathcal{A}$ is asymmetric, and in particular, only one of $\mathcal{A}_{i,j}$ and $\mathcal{A}_{j,i}$ can be non zero.

Time is discretized into 5-minute periods in our work to reflect the typical frequency of traffic sensor data collection. Let $X_{t_1:t_2} \in \mathbb{R}^{n\times(t_2-t_1)}$ denote the sequence of traffic speeds at all n sensors from time t_1 to time t_2, where $t_1, t_2 \in \{1, 2, 3, ...\}$. The value of traffic speed at sensor i at time t is denoted $X_{t,i}$, and the vector of speeds at all vertices at time t is denoted X_t.

The objective is to train the models with limited data sampled from the source domain (e.g., normal days) that does not contain all possible traffic patterns, while the test set sampled from the target domain (e.g., during a disaster) has mismatched patterns, including different periods, high temporal variations, and missing data. These pattern differences may change the labeling function based on past data. For example, traffic congestions during rush hours usually get worse, while traffic congestions during free flow hours typically disappear quickly. Mathematically, we define the source domain as

$$\mathcal{X}_s = \{(X_{t-T:t}, X_{t+1}) : X_{t+1} = l_s(X_{t-T:t}), X_{t-T:t} \sim \mathcal{D}\},$$

where l_s is the labeling function in the source domain. The target domain can be defined in a similar way, but the difference is that the labeling function in the target domain is l_τ instead of l_s, where $l_s \neq l_\tau$. We note that T is the length of the time sequence required by the "ground truth" labeling function, which is not fully known. Thus, each forecasting algorithm can potentially use a time sequence of length different from T. Our proposed RDGODE only requires 1 time point for prediction since it is built out of the spatio-temporal difference equation that connects the speed at successive time points. We allow baseline models to use 12 time points for prediction.

Given the past traffic speed observations denoted as $(X^s_{t-T:t}, X^s_{t+1}) \in \mathcal{X}_s$ on the graph $\mathcal{G}$, we aim to train a predictive model F that can predict the traffic speeds at time $t+1$ for all vertices (denoted as $\hat{X}^s_{t+1} = F(X^s_{t-T:t}, \theta)$ where θ is the parameter, $\hat{X}^s_{t+1} \in \mathbb{R}^n$), such that the model (trained on data from the source domain) makes good predictions using the samples $(X^\tau_{t-T:t}, X^\tau_{t+1}) \in \mathcal{X}_\tau$ in the target domain (i.e., $\hat{X}^\tau_{t+1} = F(X^\tau_{t-T:t}, \theta)$ is also a reasonable prediction) **without extra training**. The optimization objective is defined as

$$\min_{\theta\in\{\theta_1\}} \mathcal{L}(F(X^\tau_{t-T:t}, \theta), X^\tau_{t+1}), \tag{1}$$

where $\{\theta_1\} = \text{argmin}_\theta\, \mathcal{L}(F(X^s_{t-T:t}, \theta), X^s_{t+1})$ and $\mathcal{L}(*,*)$ is the loss function, which we will define formally in Section 5.2. We assume that T is identical in source and target domains.

4 METHODOLOGY

In the following, we describe the underlying domain model for traffic speed (local reaction-diffusion) and describe how we build our novel GCN architecture using that model.

4.1 Local Reaction-Diffusion Equation

Reaction-Diffusion systems were first proposed in the context of chemical systems to describe spatio-temporal changes which involve both a local chemical reaction and diffusion simultaneously in dynamic changes. The basis of a reaction-diffusion model is to express the rate of change of a quantity as a sum of two, generally opposing processes. Pivot to the traffic, Bellocchi in [2] proposed the reaction-diffusion approach to reproduce transportation network characteristics such as speed and congestion using few observations.

Consider sensor i, let $\mathcal{N}^d$ denote the set of sensor i's neighbors in the road segment direction, and let $\mathcal{N}^r$ denote the set of the neighbors in the opposite direction of the sensor i. Let $u_i(t)$ denote speed as a function of time at vertex i, the local reaction-diffusion equation at vertex i can be formulated as

$$\begin{aligned}\frac{du_i(t)}{dt} &= \sum_{j\in\mathcal{N}^d} \rho_{(i,j)} \left(u_j(t) - u_i(t)\right) + b^d_i \\ &\quad + \tanh\left(\sum_{j\in\mathcal{N}^r} \sigma_{(i,j)} (u_j(t) - u_i(t)) + b^r_i\right),\end{aligned} \tag{2}$$

where $\rho_{(i,j)}$ and $\sigma_{(i,j)}$ are the diffusion and reaction parameters respectively; b^d_i and b^r_i are biases to correct the average traffic speed at vertex i in the diffusion and reaction terms.

Eq. (2) expresses the change in speed as a sum of two terms. The first term is the *diffusion* term, a monotone linear function of speed change in the direction of traffic, and it relies on the empirical fact that in the event of congestion, drivers prefer to bypass the congestion by following one of the neighboring links (Figure 3a). The second term is the *reaction* term, a non-linear monotone function (tanh activation) of speed change opposite to the direction of traffic, and it relies on the empirical fact that a road surrounded by congested roads is highly likely to be congested as well (Figure 3b).

4.2 RDGODE

We incorporate the reaction-diffusion approach to building a novel graph-based machine-learning model for accurate and interpretable prediction of traffic speed. Specifically, we consider a continuous-time model for predictions using Eq. (2):

$$\hat{X}_{t_1,i} = X_{t_0,i} + \int_{t_0}^{t_1} \frac{du_i(t)}{dt} dt \tag{3}$$

where $X_{t,i}$, as mentioned earlier, is the speed of $u_i(t)$ measured by the sensor at vertex i at time t.

As shown in Figure 4, there are four key steps to use Eq. (3) to build a traffic-law-informed reaction-diffusion graph convolutional network (RDGODE).

❶ **Derive the Adjacency Matrices from the Physical Graph for the Reaction and Diffusion Process.** We define a diffusion graph $\mathcal{G}^d = (\mathcal{V}, \mathcal{E}^d)$ and a reaction graph $\mathcal{G}^r = (\mathcal{V}, \mathcal{E}^r)$ derived from the physical graph $\mathcal{G}$. The diffusion graph represents whether two vertices are direct neighbors in the road direction, i.e., $\mathcal{E}^d = \mathcal{E}$ and $\mathcal{A}^d = \mathcal{A}$; the reaction graph represents whether two vertices are direct neighbors in the opposite direction of a road segment, i.e., $\mathcal{E}^r = \{(i,j) : (j,i) \in \mathcal{E}\}$ and $\mathcal{A}^r = \mathcal{A}^\top$, and $\top$ is matrix transpose.

❷ **Define Model Weights for Reaction and Diffusion Networks based on the Physical Equation.** Define $\rho = \{\rho_{(i,j)} \in \mathbb{R} | (i,j) \in \mathcal{E}^d\}$, $\sigma = \{\sigma_{(i,j)} \in \mathbb{R} | (i,j) \in \mathcal{E}^r\}$, $b^d \in \mathbb{R}^n$, $b^r \in \mathbb{R}^n$. Each parameter $\rho_{(i,j)}$ (resp. $\sigma_{(i,j)}$) is a diffusion weight (resp. reaction weight) for edge (i,j). Each parameter in ρ and σ corresponds to a directed edge (i,j) in $\mathcal{E}^d$ and $\mathcal{E}^r$. Let $\mathbf{W}^d \in \mathbb{R}^{n \times n}$ and $\mathbf{W}^r \in \mathbb{R}^{n \times n}$ denote the sparse weight matrices for diffusion graph $\mathcal{G}^d$:

$$\mathbf{W}^d_{i,j} = \begin{cases} \rho_{(i,j)} & \forall (i,j) \in \mathcal{E}^d , \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

and $\mathbf{W}^r_{i,j}$ for reaction graph $\mathcal{G}^r$ is defined in a similar way but the non-zero element at i, j is $\sigma_{(i,j)}$ where $\forall (i,j) \in \mathcal{E}^r$.

❸ **Characterize the Graph Laplacian by Combining the Adjacency Matrices and the Defined Parameters for the Reaction and Diffusion Terms.** Given any weighted adjacency matrix $\mathbf{A}$, the corresponding Laplacian matrix $\mathbf{L}$ can be calculated by the function $Lap(\mathbf{A})$ which is defined as

$$\mathbf{L} = Lap(\mathbf{A}) = Degree(\mathbf{A}) - \mathbf{A}, \tag{5}$$

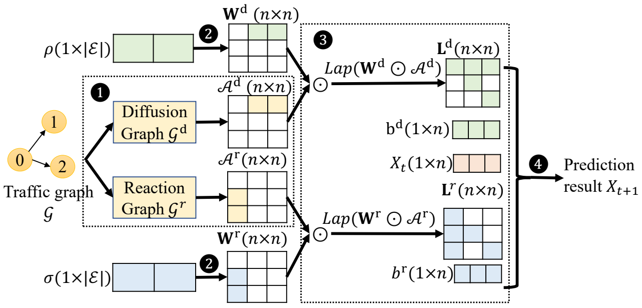
where $Degree(*)$ is the degree matrix of an adjacency matrix. We use a common variation measure in graph signal processing [5] to express the action of the Laplacian on sensor i and X_t:

$$(\mathbf{L}X_t)_i = \sum_{j:(i,j) \in \mathcal{E}} \mathbf{A}_{i,j}(X_{t,i} - X_{t,j}). \tag{6}$$

Let $\mathbf{L}^d$ (resp. $\mathbf{L}^r$) be the corresponding Laplacian of the combination of diffusion (resp. reaction) weight tensor $\mathbf{W}^d$ (resp. $\mathbf{W}^r$) and diffusion (resp. reaction) adjacency matrices $\mathcal{A}^d$ (resp. $\mathcal{A}^r$), then

$$(\mathbf{L}^d X_t)_i = \sum_{j:(i,j) \in \mathcal{E}^d} (\mathbf{W}^d \odot \mathcal{A}^d)_{i,j}(X_{t,i} - X_{t,j}), \tag{7}$$

where $\odot$ is the Hadamard product, and $(\mathbf{L}^r X_t)_i$ is defined similarly with different weight tensor $\mathbf{W}^r$ and adjacency matrix is $\mathcal{A}^r$.



$\mathbf{W}^{\mathrm{d}}\ (n\times n)$
$\rho(1\times|\mathcal{E}|)$
❷
❸
$\mathbf{L}^{\mathrm{d}}(n\times n)$
❶
$Lap(\mathbf{W}^{\mathrm{d}} \odot \mathcal{A}^{\mathrm{d}})$
$\mathcal{A}^{\mathrm{d}}\ (n\times n)$
Diffusion Graph $\mathcal{G}^{\mathrm{d}}$
1
0
2
$\mathrm{b}^{\mathrm{d}}(1\times n)$
Traffic graph $\mathcal{G}$
$\mathcal{A}^{\mathrm{r}}(n\times n)$
Reaction Graph $\mathcal{G}^r$
$X_t(1\times n)$
❹
Prediction result X_{t+1}
$\mathbf{L}^r(n\times n)$
$Lap(\mathbf{W}^{\mathrm{r}} \odot \mathcal{A}^{\mathrm{r}})$
$\mathbf{W}^{\mathrm{r}}(n\times n)$
❷
$\sigma(1\times|\mathcal{E}|)$
$b^{\mathrm{r}}(1\times n)$

Figure 4: Reaction-diffusion GCN architecture for graph with $|\mathcal{V}| = 3$ and $|\mathcal{E}| = 2$. ❶ derives the diffusion and reaction adjacency matrices A^d and A^r; ❷ defines model weights ρ and σ for the reaction and diffusion networks, and maps them to W^d and W^r by Eq. (4) with weights ρ and σ; ❸ characterizes the Graph Laplacian L^d and L^r; ❹ defines the network prediction function Eq. (9).

❹ **Define the Network Prediction Function Using a Graph Neural Network Approach with the Derived Laplacian.** By Eq. (7), let $U(t)$ denote the network-level differential equation:

$$\begin{aligned} \frac{dU(t)}{dt} &= (\mathbf{L}^d U(t) + b^d) + \tanh(\mathbf{L}^r U(t) + b^r) \\ &\approx (\mathbf{L}^d X_t + b^d) + \tanh(\mathbf{L}^r X_t + b^r). \end{aligned} \tag{8}$$

Then we calculate the integral of Eq. (8) from t_0 to t_1 using Neural ODE [3], given the differential equation, initial state, and the time sequence. Following Neural ODE [3] and combining Eq. (3) with Eq. (8), the prediction of the traffic speed at all vertices is

$$\hat{X}_{t_1} = ODESolve(\frac{dU(t)}{dt}, X_{t_0}, [t_0, t_1]), \tag{9}$$

where $ODESolve$ is to solve the differential equation in Eq. (8). In our context, the time difference between t_0 and t_1 is 5 minutes, which we denote by one unit of time, $t_1 - t_0 = 1$.

5 EVALUATION

In this section, we evaluate the performance of RDGODE on real-world datasets to answer the following research questions.

(1) What is the overall prediction performance of RDGODE under *mismatched data* compared to other machine learning baselines?
(2) How well can RDGODE *track rapid temporal variations in speed* in comparison with existing baseline models?
(3) How well can RDGODE *impute missing data* in comparison with existing baseline models?

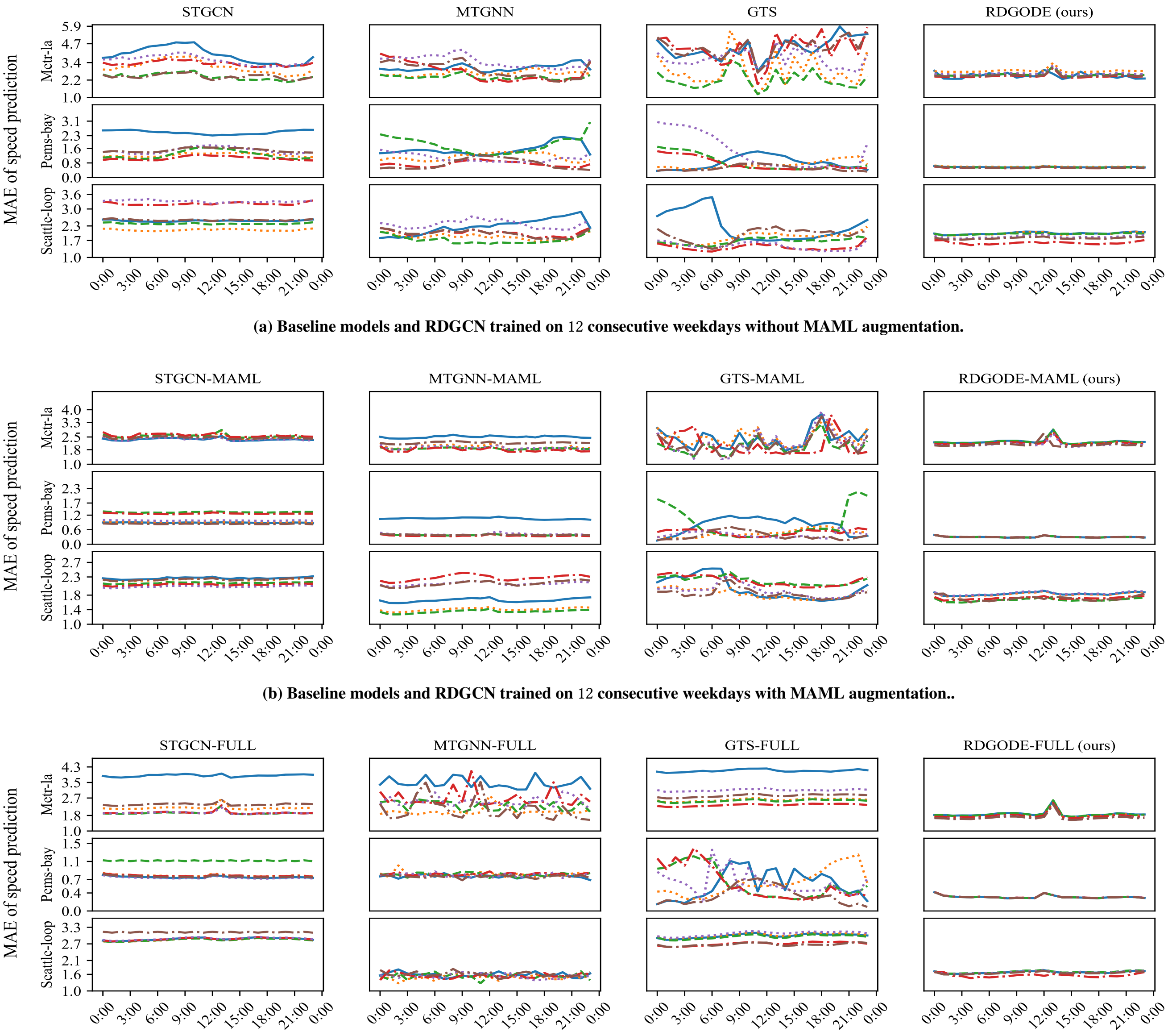
5.1 Datasets

Our experiments are conducted on the following three real-world datasets with a sample speed every 5 minutes. We label the physical graph according to the road network.

Metr-la. The traffic speed on Los Angeles County highways from 03/01/2012 to 06/30/2012 was collected by 207 sensors [13].

Pems-bay. The traffic speed in the Bay area from 01/01/2017 to 05/31/2017 was collected by 325 sensors [20]. We select 281 of 325 sensors, where road directions are clear.

Seattle-loop. The traffic speed from 01/01/2015 to 12/31/2015 was collected by 323 sensors on Great Seattle highways [6].



(a) Baseline models and RDGCN trained on 12 consecutive weekdays without MAML augmentation.

(b) Baseline models and RDGCN trained on 12 consecutive weekdays with MAML augmentation..

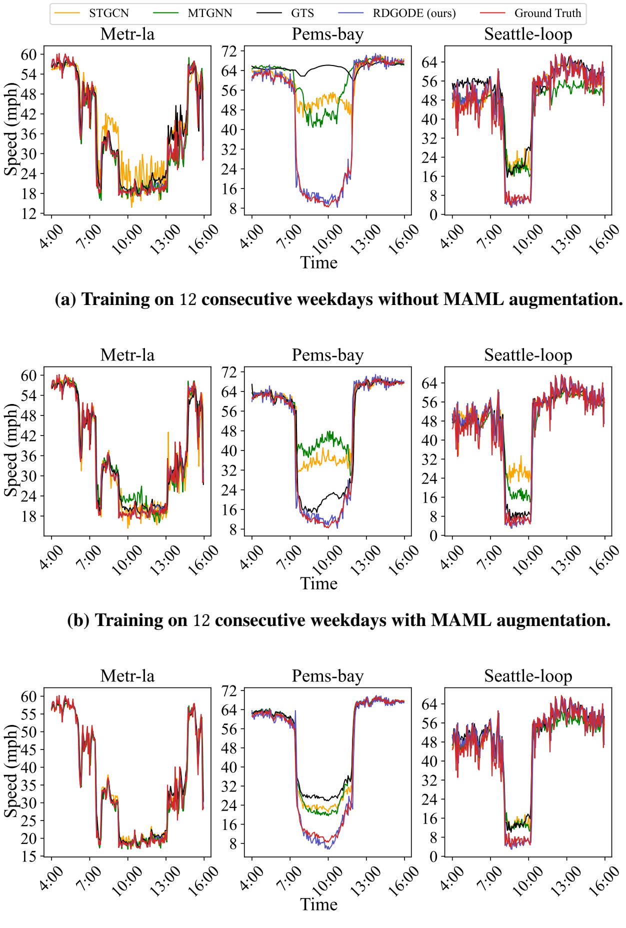
(c) Baseline models and RDGCN trained with data more than half of a year.

Figure 5: (a) The results of RDGODE are very close regardless of the period of the training set. (b) MAML significantly augments the performance of baseline models. However, the results of RDGODE are still closer regardless of the period, compared to baseline models. (c) More training data augments the performance of baseline models and RDGODE. However, the results of RDGCOD are still closer.

5.2 Experiment Settings

Evaluation Metric. The loss function we use is the mean absolute error $MAE(X_t, \hat{X}_t) = \frac{1}{n}\sum_{i=1}^{n}|X_{t,i} - \hat{X}_{t,i}|$, and the root mean square error $RMSE(X_t, \hat{X}_t) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_{t,i} - \hat{X}_{t,i})^2}$.

Baselines. We compare RDGODE with three graph learning models, STGCN [36], MTGNN [32], and GTS [27]. These are influential and best-performing graph learning models for predicting future traffic speed using historical traffic speed alone. Moreover, we use Model-Agnostic Meta-Learning (MAML) [8] to augment baseline models and our approach. Specifically, (1)We randomly select sequences of 12 consecutive weekdays, and sample 4-hour data as the training set. We evaluate the model with hourly data on weekends. (2) We divide the training set into two equal parts: the support set and the query set. (3) The support set is used to compute adapted parameters. (4) We use the adapted parameters to update the MAML parameters on the query set. (5) We repeat it 200 times to obtain initial parameters for



(a) Training on 12 consecutive weekdays without MAML augmentation.

(b) Training on 12 consecutive weekdays with MAML augmentation.

(c) Training with more than half of a year data.

Figure 6: RDGODE makes accurate predictions in the congestion, compared with (a) baselines which make bad predictions when they encounter rapid changes in traffic. (b) baselines being augmented by MAML. (c) baselines being trained with all available weekday data.

baselines. (6) We train baselines using the obtained initial parameters. MAML is trained for 200 epochs.

Hyperparameter Settings. RDGODE is optimized via Adam. The batch size is 64. The learning rate is 0.001, and the early stopping strategy is used with the patience of 30 epochs. The training and validation set are split by a ratio of 3:1 from the weekday subset, and the test data is sampled from the weekend subset with different patterns. As for baselines, we follow the settings in their works.

Evaluation. We assume that all zeros in the datasets are missing values, and we remove the predicted speed when the ground truth is 0, or when the last speed recorded is 0.

5.3 Results and Analysis

Mismatched Data Exploration. We first explore the performance of the models when they are trained using mismatched data. Specifically, the models are trained with four-hour data on weekdays (e.g.,16:00-20:00 on weekdays) and evaluated with hourly data on weekends (e.g., 13:00-14:00 on weekends). The training set consists of data from five different sequences of 12 consecutive weekdays selected randomly from the available data (to limit patterns in the training set) or all available weekdays (contains all possible patterns in the source domain). The results are shown in Figure 5, where each curve denotes the average test prediction MAE of the model trained on the aforementioned five sequences.

Figure 5a plots the prediction MAE of STGCN, MTGNN, GTS, and RDGODE over time. RDGODE has nearly identical performance regardless of which time window of data is used for training. The RDGODE consistently has low MAE (i.e., small y-axis values) and low variance across different time windows (i.e., the difference of curves with the highest MAE and lowest MAE is small). However, the performances of STGCN, MTGNN, and GTS are significantly different depending on the training set, and some may have a relatively high MAE (e.g., the curve of STGCN trained with data in 0:00-4:00 on Pems-bay dataset has much higher MAE values than the one of RDGODE over time). In Figure 5b, we compare the performance of RDGODE with the baseline models being augmented with MAML. Even when the training process is augmented by MAML, RDGODE outperforms the baseline models wherein the variance across time and models is very low. While MAML brings some gain to baseline models, its impact on RDGODE is fairly limited, indicating that RDGODE performs well in different testing domains without needing additional expensive training. In Figure 5c, we compare RDGODE with baselines trained by all available weekday data. Note that identical training and test data are used for all the results in Figure 5a and Figure 5b, indicating that even when the baseline models are trained using all available weekday data, RDGODE's prediction is better with low MAE and low variance.

As evidence, incorporating the traffic dynamics into the learning model is highly beneficial to dealing with mismatched data between training and test data. We speculate that this is a consequence of our model capturing the relative changes in speed through the dynamical equations, whereas black box models that derive complex functions of the absolute values of speed across time might not always capture the immediate dynamics but rely more on long term patterns. In effect, when there is a mismatch, the underlying nature of traffic dynamics is less likely to be impacted whereas the complex patterns of absolute speed values might vary significantly across domains. This is particularly true when dealing with limited data that does not contain all possible patterns. At the same time, RDGODE is designed to predict based on neighboring vertices, so even if the speed patterns of a distant sensor and a close sensor are similar (e.g., both are free flow), it does not factor into the model's predictions. We note that the prediction of RDGODE is not uniformly better than the prediction of the STGCN, MTGNN, and GTS (e.g., the prediction of MTGNN trained by weekday data from 8:00 to 12:00 is better than the prediction of RDGODE), and one possible reason is that speed pattern mismatches between weekdays and weekends are not always significant (e.g., when the training weekday is a holiday). Although real-world data under situations such as disasters or events are hard to obtain, our approach of splitting the dataset emulates test scenarios that are sufficiently different from the training dataset. The Mean and STD of prediction MAE (resp. RMSE) of each model are shown in Table 1 (i.e., the Mean and STD of all points on each subfigure in Figure 5a, Figure 5b and the corresponding results using

Table 1: Numerical result of Figure 5: the Mean and STD of prediction MAE of RDGODE and baselines on three real-world datasets.

	MAE				RMSE			
Without MAML	STGCN	MTGNN	GTS	RDGODE	STGCN	MTGNN	GTS	RDGODE
Metr-la	3.31 ± 0.64	2.94 ± 0.50	3.70 ± 1.13	2.36 ± 0.12	6.31 ± 1.34	5.17 ± 1.16	6.97 ± 1.33	5.11 ± 0.89
Pems-bay	1.50 ± 0.49	1.38 ± 0.44	1.07 ± 0.52	0.81 ± 0.08	1.83 ± 0.35	2.88 ± 1.01	3.18 ± 0.99	1.48 ± 0.05
Seattle-loop	2.68 ± 0.43	2.50 ± 0.22	2.28 ± 0.32	2.12 ± 0.13	6.33 ± 0.44	4.89 ± 0.36	6.09 ± 1.98	3.46 ± 0.48
With MAML	STGCN	MTGNN	GTS	RDGODE	STGCN	MTGNN	GTS	RDGODE
Metr-la	2.47 ± 0.11	2.41 ± 0.22	2.55 ± 0.48	2.38 ± 0.08	5.28 ± 0.94	5.17 ± 1.16	7.55 ± 0.91	5.01 ± 0.82
Pems-bay	1.03 ± 0.19	0.91 ± 0.21	0.96 ± 0.03	0.81 ± 0.03	1.41 ± 0.05	2.86 ± 1.11	2.85 ± 0.84	1.43 ± 0.05
Seattle-loop	2.20 ± 0.08	2.23 ± 0.24	2.34 ± 0.15	2.12 ± 0.04	5.94 ± 0.14	3.92 ± 0.37	5.80 ± 0.60	3.44 ± 0.22
FULL	STGCN	MTGNN	GTS	RDGODE	STGCN	MTGNN	GTS	RDGODE
Metr-la	2.57 ± 0.68	3.11 ± 0.48	3.44 ± 0.47	2.37 ± 0.13	5.31 ± 0.92	4.02 ± 0.31	7.04 ± 1.20	3.96 ± 0.13
Pems-bay	1.38 ± 0.06	1.85 ± 0.38	2.08 ± 0.51	0.83 ± 0.07	1.37 ± 0.06	1.85 ± 0.38	2.08 ± 0.53	1.45 ± 0.03
Seattle-loop	2.90 ± 0.10	2.18 ± 0.06	3.11 ± 0.11	2.16 ± 0.06	3.91 ± 0.45	3.81 ± 0.65	5.33 ± 0.74	3.66 ± 0.07

RMSE), respectively. Table 1 shows that RDGODE has a lower MAE (resp. RMSE) and lower variance compared with baselines under limited training set with or without MAML augmentation, and the gain of adding more data on RDGCN is limited, which is consistent with our observation in Figure 5.

Tracking Rapid Temporal Variations in Speed. Rapid speed change patterns often have a very limited presence in traffic datasets, as they usually occur in the case of sudden events, or rapidly degrading infrastructure which is rare. Consequently, even well-trained models are not equipped to predict well under these circumstances. Here, we investigate the performance of models in rapid temporal variation scenarios, where the traffic speed changes rapidly in the given time window. On each dataset, we first train STGCN, MTGNN, GTS, and RDGODE using data from 4:00 to 8:00 in a randomly selected sequence of 12 consecutive weekdays and then test different models with data on vertices where speed patterns are not covered in the training set (e.g., the step-like speed variation is not in the selected 12 weekdays). We show curves of the predicted speeds of various models and ground truth in Figure 6a, of predicted speeds augmented by MAML in Figure 6b, and of predicted speed training using all available weekday data in Figure 6c.

Figure 6 indicates that RDGODE produces predictions that better track ground truth speeds, with or without MAML augmentation, trained by limited data or all available data. On the other hand, the predicted speed from STGCN has excessive oscillations that are not present in the actual speed on Metr-la. The predicted speed from STGCN, MTGNN, and GTS is higher than the ground truth speed in the period of congestion, especially when the traffic speed drops rapidly, as shown in Figure 6a. Since STGCN, MTGNN and GTS have very limited training data that exhibits such rapid variation, the prediction is very different from the ground truth speeds. Figure 6b shows that MAML can help the baselines learn better (e.g., fewer oscillations of STGCN in Metr-la and the speed prediction of GTS are closer to ground truth during the congestion in Pems-bay). However, the difference between the prediction of baselines and ground truth is still large, which implies that MAML cannot completely address the challenge under rapid changes. Figure 6c indicates that more mismatched training data can make the baselines' prediction better, but it cannot address the challenging speed prediction under rapid changes. On the other hand, we conjecture that the short-term dynamics incorporated in the RDGODE through the reaction-diffusion equation allow the RDGODE to better track instant variations very quickly and the lack of relevant training data does not limit its ability to predict traffic speed under these circumstances.

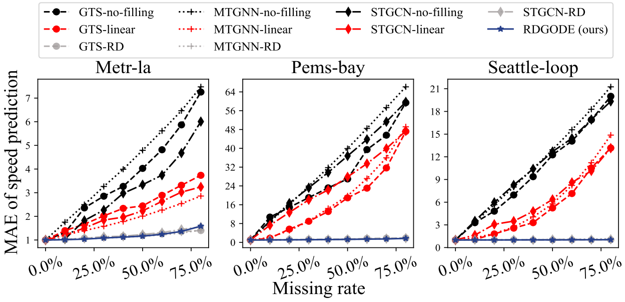


Figure 7: STGCN-RD (resp. MTGNN-RD, GTS-RD) represents STGCN (resp. MTGNN, GTS) using RD imputation. The MAE of baselines with RD imputation and RDGODE does not increase much when the missing rate increases.

Missing Data and Imputation. We investigate the prediction performance of STGCN, MTGNN, GTS, and RDGODE under scenarios where data has random missing values. This can frequently occur when there are intermittent sensor or communication failures. We first train baseline models and RDGODE using the full training set (does not contain mismatched data), then we simulate the missing data situation by setting a percentage of values (missing rate) on a sensor to 0 and test the models using the data generated above. Since the existing baseline models, such as STGCN, MTGNN, and GTS, do not employ a physical model for traffic speed, there is no intrinsic mechanism to fill in missing values. Whereas, in our domain

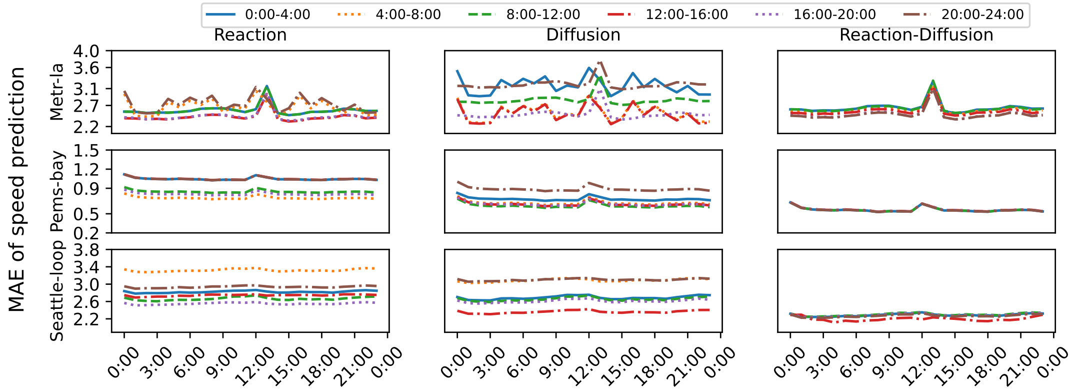


Figure 8: MAE of speed predictions on models incorporating reaction equation, diffusion equation, and reaction-diffusion equation.

informed approach, one can use the physical model given in Eq. (3) to impute the missing values (RD imputation) using the data of only neighbor sensors (i.e., the speed at neighbor sensors $X_{t,i}$, $X_{t,j}$ are known and $\rho_{(i,j)}$ and $\sigma_{(i,j)}$ are in the trained RDGODE model) to improve the performance. In our experiments, we compare the MAE of our approach with that of the raw STGCN, MTGNN, and GTS without any imputation, as well as when these baseline methods are supplemented with linear imputation and a domain informed Reaction Diffusion (RD) based imputation.

Figure 7 indicates that the RDGODE and the models with RD imputation are more robust to missing sensor data since the MAE loss does not increase much with the missing rate. Even when the missing rate of data reaches 80%, the average MAE on the three datasets is 2.9723 mph and the maximum MAE does not reflect a speed difference beyond 5 mph. In contrast, the MAE of the models without RD imputation increases significantly with the increase of the missing rate. This is a particularly important result since the stable prediction performance of models using RD imputation under missing data demonstrates the soundness of the underlying reaction-diffusion model employed for traffic speed evolution. Not only does this make our approach inherently explainable to domain specialists, but it also provides insights into why RDGODE performs better when test data has missing data.

6 ABLATION STUDIES

In this section, we investigate the prediction models incorporating the reaction equation and the diffusion equation, independently, under mismatched data, to understand whether both the reaction and diffusion processes are essential. We use the same training set (i.e., 12 consecutive working days selected randomly) and test set (i.e., hourly weekend data) in Section 5.3. The curves of MAE versus time using the model incorporating the reaction equation, the diffusion equation, and the reaction-diffusion equation are shown in Figure 8.

Figure 8 indicates that the predictions of all models with the reaction-diffusion equation provide low MAE with low variance (i.e., the difference between curves with the highest MAE and lowest MAE is small) over time. However, the predictions of the reaction models and the diffusion models have weaker performance in at least one time period. We speculate that using only the reaction (or diffusion) equation is not sufficient to capture the pattern of the traffic speed change accurately. Furthermore, the prediction of the model incorporating the reaction-diffusion equation is not uniformly better than the prediction of the model incorporating only the reaction or diffusion equation. One possible reason is that the reaction or diffusion process does not always exist in a specific period (e.g., if two neighboring road segments are in free flow during the test period, the traffic speeds at the two segments do not affect each other. Thus there is neither diffusion nor reaction between these two road segments). These observations further emphasize the necessity of both the reaction and diffusion processes for reliable predictions.

7 MODEL EFFICIENCY

Table 2 shows the training and inference times of baselines and RDGODE on the Metr-la dataset using two NVIDIA-2080ti graphic cards. It's observed that RDGODE takes less time in both training and inference than the other models.

	# Parameters	Training (s/epoch)	Inference (s)
STGCN	458865	0.5649	0.0232
MTGNN	405452	0.5621	0.0607
GTS	38377299	1.0632	0.1641
RDGODE	872	0.3551	0.0173

Table 2: The computation time on the Metr-la dataset.

8 CONCLUSION

In this paper, we investigate traffic speed prediction under mismatched data. Specifically, we propose a traffic-law-informed graph learning model, RDGODE, by incorporating the traffic reaction-diffusion model into GCNs. The new model shows strong robustness under mismatched data with the help of traffic law. We intentionally introduce two types of mismatches by utilizing data from different time periods during the training and testing phases, and demonstrate the robustness of RDGODE. We also explore the reasons for the enhanced robustness of our traffic-law-informed model. The presented results show the potential of the reaction-diffusion model as a new architecture in GCNs in traffic speed prediction. We believe reaction-diffusion models are not limited to traffic speed prediction, even though this is our focus here. We will extend graph reaction-diffusion architectures to other problems in the future.

9 ACKNOWLEDGEMENTS

This work was partly funded through a Lehigh internal Accelerator Grant and the CCF-1617889 grant from the National Science Foundation. Sihong was supported in part by the National Science Foundation under NSF Grants IIS-1909879, CNS-1931042, IIS-2008155, and IIS-2145922. Rick S. Blum was supported by the U.S. Office of Naval Research under Grant N00014-22-1-2626.

REFERENCES

[1] Joaquim Barros, Miguel Araujo, and Rosaldo JF Rossetti. 2015. Short-term real-time traffic prediction methods: A survey. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, 132–139.

[2] Leonardo Bellocchi and Nikolas Geroliminis. 2020. Unraveling reaction-diffusion-like dynamics in urban congestion propagation: Insights from a large-scale road network. *Scientific reports* 10, 1 (2020), 1–11.

[3] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural ordinary differential equations. *Advances in neural information processing systems* 31 (2018).

[4] Jeongwhan Choi, Hwangyong Choi, Jeehyun Hwang, and Noseong Park. 2022. Graph neural controlled differential equations for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 6367–6374.

[5] Fan RK Chung and Fan Chung Graham. 1997. *Spectral graph theory*. Vol. 92. American Mathematical Soc.

[6] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, Xiaolei Ma, and Yinhai Wang. 2020. Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction. *Transportation Research Part C: Emerging Technologies* 115 (2020), 102620.

[7] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. 2021. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 364–373.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*. PMLR, 1126–1135.

[9] Victor Garcia and Joan Bruna. 2017. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043* (2017).

[10] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 922–929.

[11] Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, and Hui Xiong. 2021. Dynamic and Multi-faceted Spatio-temporal Deep Learning for Traffic Speed Forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 547–555.

[12] Dedy Hartama, Herman Mawengkang, Muhammad Zarlis, Rahmat Widia Sembiring, Mhd Furqan, Dahlan Abdullah, and Robbi Rahim. 2017. A research framework of disaster traffic management to Smart City. In *2017 Second International Conference on Informatics and Computing (ICIC)*. IEEE, 1–5.

[13] Hosagrahar V Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. 2014. Big data and its technical challenges. *Commun. ACM* 57, 7 (2014), 86–94.

[14] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. 2021. Physics-informed machine learning. *Nature Reviews Physics* 3, 6 (2021), 422–440.

[15] Femke Kessels, Kessels, and Rauscher. 2019. *Traffic flow modelling*. Springer.

[16] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. 2020. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems* 33 (2020), 6696–6707.

[17] Thomas N Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

[18] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. 2016. Interpretable Decision Sets: A Joint Framework for Description and Prediction. In *SIGKDD*.

[19] Daqing Li, Bowen Fu, Yunpeng Wang, Guangquan Lu, Yehiel Berezin, H Eugene Stanley, and Shlomo Havlin. 2015. Percolation transition in dynamical traffic network with evolving critical bottlenecks. *Proceedings of the National Academy of Sciences* 112, 3 (2015), 669–672.

[20] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).

[21] Yuxuan Liang, Kun Ouyang, Yiwei Wang, Zheyi Pan, Yifang Yin, Hongyang Chen, Junbo Zhang, Yu Zheng, David S. Rosenblum, and Roger Zimmermann. 2022. Mixed-Order Relation-Aware Recurrent Neural Networks for Spatio-Temporal Forecasting. *IEEE Transactions on Knowledge and Data Engineering* (2022), 1–15. https://doi.org/10.1109/TKDE.2022.3222373

[22] Gy Lipták, Mike Pereira, Balázs Kulcsár, Mihály Kovács, and Gábor Szederkényi. 2021. Traffic reaction model. *arXiv preprint arXiv:2101.10190* (2021).

[23] Allister Loder, Lukas Ambühl, Monica Menendez, and Kay W Axhausen. 2019. Understanding traffic capacity of urban networks. *Scientific reports* 9, 1 (2019), 1–10.

[24] Yin Lou, Rich Caruana, and Johannes Gehrke. 2012. Intelligible Models for Classification and Regression. In *SIGKDD*.

[25] Hayden C Metsky, Nicole L Welch, Priya P Pillai, Nicholas J Haradhvala, Laurie Rumker, Sreekar Mantena, Yibin B Zhang, David K Yang, Cheri M Ackerman, Juliane Weller, et al. 2022. Designing sensitive viral diagnostics with machine learning. *Nature biotechnology* 40, 7 (2022), 1123–1131.

[26] Attila M Nagy and Vilmos Simon. 2018. Survey on traffic prediction in smart cities. *Pervasive and Mobile Computing* 50 (2018), 148–163.

[27] Chao Shang, Jie Chen, and Jinbo Bi. 2021. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861* (2021).

[28] Joaquin Vanschoren. 2018. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548* (2018).

[29] Kush R. Varshney. 2020. On Mismatched Detection and Safe, Trustworthy Machine Learning. In *2020 54th Annual Conference on Information Sciences and Systems (CISS)*. 1–4. https://doi.org/10.1109/CISS48834.2020.1570627767

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[31] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)* 53, 3 (2020), 1–34.

[32] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 753–763.

[33] Xun Xian, Mingyi Hong, and Jie Ding. 2022. Mismatched Supervised Learning. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4228–4232.

[34] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnn explainer: A tool for post-hoc explanation of graph neural networks. *arXiv preprint arXiv:1903.03894* (2019).

[35] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. In *NeurIPS*. 9244–9255.

[36] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875* (2017).

[37] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. 2021. Domain generalization in vision: A survey. *arXiv preprint arXiv:2103.02503* (2021).

[38] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. 2022. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[39] Zewei Zhou, Ziru Yang, Yuanjian Zhang, Yanjun Huang, Hong Chen, and Zhuoping Yu. 2022. A comprehensive study of speed prediction in transportation system: From vehicle to traffic. *Iscience* (2022), 103909.