Incorporating Domain Differential Equations into Graph Convolutional Networks to Lower Generalization Discrepancy

[1]Yue Sun, [1]Chao Chen, [1]Yuesheng Xu, [2]Sihong Xie, [1]Rick S. Blum, [1]Parv Venkitasubramaniam

[1]Lehigh University, [2]Hong Kong University of Science and Technology (Guangzhou)
{yus516, rb0f, pav309}@lehigh.edu, {cha01nbox, xuyuesheng324, xiesihong1}@gmail.com

Abstract

Ensuring both accuracy and robustness in time series prediction is critical to many applications, ranging from urban planning to pandemic management. With sufficient training data where all spatiotemporal patterns are well-represented, existing deep-learning models can make reasonably accurate predictions. However, existing methods fail when the training data are drawn from different circumstances (e.g., traffic patterns on regular days) compared to test data (e.g., traffic patterns after a natural disaster). Such challenges are usually classified under domain generalization. In this work, we show that one way to address this challenge in the context of spatiotemporal prediction is by incorporating domain differential equations into Graph Convolutional Networks (GCNs). We theoretically derive conditions where GCNs incorporating such domain differential equations are robust to mismatched training and testing data compared to baseline domain agnostic models. To support our theory, we propose two domain-differential-equation-informed networks called Reaction-Diffusion Graph Convolutional Network (RDGCN), which incorporates differential equations for traffic speed evolution, and Susceptible-Infectious-Recovered Graph Convolutional Network (SIRGCN), which incorporates a disease propagation model. Both RDGCN and SIRGCN are based on reliable and interpretable domain differential equations that allows the models to generalize to unseen patterns. We experimentally show that RDGCN and SIRGCN are more robust with mismatched testing data than the state-of-the-art deep learning methods.

Introduction

Robustness to domain generalization is a crucial aspect in the realm of time series prediction, which has continued to be a topic of great interest, given its myriad uses in many sectors such as transportation (Bui, Cho, and Yi 2022), weather forecasting (Longa et al. 2023), and disease control (Jayatilaka et al. 2020). When sufficient training data is available where all patterns likely to appear in test situations are represented, deep learning approaches have provided the most accurate predictions. Among the best-performing deep learning models, graph-based deep neural networks (Yu, Yin, and Zhu 2018; Wu et al. 2020; Han et al. 2021; Shang, Chen, and Bi 2021; Ji et al. 2022) dominate due to their ability to incorporate spatiotemporal information so that dependent information at different locations and times can be captured and exploited to make more accurate predictions. However, when collecting representative training data is challenging, as it is in many practical situations since we can only sample in limited conditions, the model trained on such limited data is expected to work in exceptional circumstances. For example, in the traffic speed prediction problem, natural disasters (e.g., earthquakes or hurricanes) are rare events where traffic patterns can be significantly and abruptly altered. At best extreme situations can be simulated, but these cannot truly capture the patterns in an actual event. Under these circumstances, existing predictive models, especially those deep learning models trained with a large amount of data that are not representative of the testing circumstance, do not work well, i.e., they are not robust to mismatched patterns.

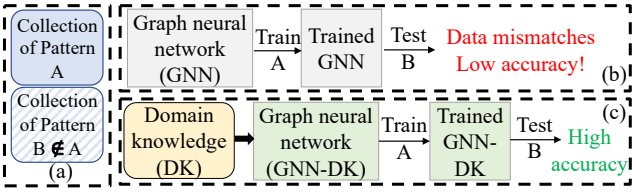


Figure 1: (a) Two collections of patterns (i.e., pattern A (exists in a known dataset) and pattern B (difficult to be collected and only available at test time) in the training and test datasets have no overlap; (b) Without incorporating a traffic ODE, testing the model with such mismatched patterns may result in poorer accuracy; (c) With an architecture using a traffic ODE, we expect the model to achieve good accuracy when adapting to unseen patterns.

Such challenges are usually classified under domain generalization (Figure 1), where a model is trained on a source domain but evaluated on a target domain with different characteristics (mismatches). Consider traffic speed prediction as a motivating example. It is well known that prediction algorithms perform poorly when traffic patterns are unexpectedly disrupted, for instance, due to extreme weather, natural disasters, or even special events. In our evaluation section, we will demonstrate this phenomenon more concretely, where state-of-the-art deep learning methods do not generalize well when dataset patterns are split between training (weekday) and test patterns (weekend). We also provide additional experiments in the appendix to probe further this phenomenon wherein we train a well-known deep learning

approach, Spatial-Temporal Graph Convolutional Network (STGCN) (Yu, Yin, and Zhu 2018), and use sensitivity analysis to identify the most influential sensors on the graph that contributed to a particular prediction. We note that the most influential sensors are geographically nearby when the test dataset is drawn from the same distribution (weekday) as the training data. In contrast, when the test data is drawn from a different distribution (weekend), the three most important sensors identified by the sensitivity analysis are far away. The challenge mentioned above can be formulated as learning with mismatched training data (Varshney 2020), a problem that is often encountered in practice.

Our work's driving hypothesis is that if domain equations can capture spatiotemporal dynamics that stay consistent even under data mismatches, then incorporating those equations into the learning methodology can lower the generalization discrepancy of the learned model. Consequently, the proposed model exhibits robustness to such domain generalization. To that end, we propose an approach to incorporate domain ordinary differential equations (ODE) into Graph Convolutional Networks (GCNs) for spatiotemporal forecasting, study the generalization discrepancy theoretically, and apply our approach to develop novel domain-informed GCNs for two practical applications, namely traffic speed prediction and influenza-like illness (ILI) prediction. Our experimental results demonstrate that even when the 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 domain-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. Our contributions are as follows:

- We study the challenge of graph time series prediction with mismatched data where the patterns in the training set are not representative of that in the test set.
- We prove theoretically the robustness of domain-ODE-informed GCNs to a particular form of domain generalization when the labeling function differs between the source and target domains. Specifically, we show that the generalization discrepancy is lower for the domain-ODE-informed learning model under certain conditions compared to a domain-independent learning model.
- We develop two novel domain-ODE-informed neural networks called Reaction-Diffusion Graph Convolutional Network (RDGCN), and Susceptible-Infectious-Recovered Graph Convolutional Network (SIRGCN) that augments GCNs with domain ODEs studied in transportation research (Bellocchi and Geroliminis 2020) and disease epidemics (Stolerman, Coombs, and Boatto 2015).
- Through experimental evaluation on real datasets, we demonstrate that our novel domain-informed GCNs are more robust in situations with data mismatches when compared to baseline models in traffic speed prediction and influenza-like illness prediction.

Related Work

Graph Neural Networks on Time Series Predictions. Graph Neural Networks (GNNs) have been widely utilized to enable great progress in dealing with graph-structured data (Kipf and Welling 2017). (Yu, Yin, and Zhu 2018; Li et al. 2018; Cui et al. 2020) build spatiotemporal blocks to encode the spatiotemporal features. (Wu et al. 2020; Shang, Chen, and Bi 2021; Han et al. 2021; Veličković et al. 2018; Guo et al. 2019) generate dependency graphs, which only focus on "data-based" dependency wherein features at a vertex can be influenced by a vertex, not in its physical vicinity. None of these approaches exploit domain ODEs for better generalization and robustness.

Domain generalization. Domain generalization is getting increasing attention recently (Wang et al. 2022; Zhou et al. 2022; Robey, Pappas, and Hassani 2021; Zhou et al. 2021), and robustness to domain data with mismatched patterns is important in designing trustworthy models (Varshney 2020). The goal is to learn a model that can generalize to unseen domains. Many works (Robey, Pappas, and Hassani 2021) assume that there exists an underlying transformation between source and target domain, and use an extra model to learn the transformations (Xian, Hong, and Ding 2022), therefore the training data must be sampled under at least two individual distributions. However, our approach addresses the challenge by incorporating a domain-specific ODE instead of using extra training processes learning from the data from two individual domains, or employing additional assumptions on transformations, thus works for arbitrarily domain scenarios.

Domain differential equations and Neural ODEs. Time series are modeled using differential equations in many areas such as chemistry (Scholz and Scholz 2015) and transportation (van Wageningen-Kessels et al. 2015; Loder et al. 2019; Kessels, Kessels, and Rauscher 2019). These approaches focus on equations that reflect most essential relationships. To incorporate differential equations in machine learning, many deep learning models based on Neural ODEs (Chen et al. 2018; Jia and Benson 2019; Asikis, Böttcher, and Antulov-Fantulin 2022) have been proposed. Advancements extend to Graph ODE networks (Ji et al. 2022; Choi et al. 2022; Jin et al. 2022), which employ black-box differential equations to simulate continuous traffic pattern evolution. However, the potential of domain knowledge to fortify algorithmic robustness against domain generalization has yet to be explored.

Problem Definition

Notations. Given an unweighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ vertices and $|\mathcal{E}|$ edges, each vertex $i \in \mathcal{V}$ corresponds to a physical location, and each edge $(i, j) \in \mathcal{E}$ represents the neighboring connectivity between two vertices. Let $\mathcal{N}_i$ denote the set of neighbors of vertex i, and $\mathcal{A} \in \mathbb{R}^{n \times n}$ denote the adjacency matrix of the graph $\mathcal{G}$. The value of the feature at vertices i at time t is denoted $x_i(t)$, and the vector of features at all vertices at time t is denoted $X(t)$. Let $X_{t_1:t_2} \in \mathbb{R}^{n \times (t_2 - t_1)}$ be the sequence of features $X(t_1), X(t_1 + 1), \cdots, X(t_2)$ at all vertices in the interval $[t_1, t_2]$. Assume that the training data and test data are sampled from the source domain $\mathcal{X}_s$ and target domain $\mathcal{X}_\tau$, respectively. Data from different domains exhibit differ-

ent patterns, which we explicitly capture through labeling functions in each domain. Formally

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

where l_s is the labeling function in the source domain and $\mathcal{D}$ is the distribution of inputs. The target domain $\mathcal{X}_\tau$ can be defined similarly but with a different labeling function l_τ. Note that T is the length of the time sequence that defines the "ground truth" labeling function, which is usually unknown. We assume that T is identical in the source and target domains.

Problem definition. We aim to solve the problem of single domain generalization (Qiao, Zhao, and Peng 2020; Wang et al. 2021; Fan et al. 2021). Given the past feature observations denoted as $(X^s_{t-T:t}, X^s_{t+1}) \in \mathcal{X}_s$ on the graph $\mathcal{G}$ on only one source domain s, we aim to train a predictive hypothesis h that can predict the feature at time $t+1$ for all vertices (denoted as $\hat{X}(t+1) \in \mathbb{R}^n$) on the unseen target domain τ **without extra training**. We use L to denote a loss function to evaluate the distance between the prediction and ground truth. Let h denote a hypothesis, and let l denote the labeling function in the corresponding domain. The expectation of the loss is: $\mathcal{L}_{(\mathcal{D},l)}(h) = \mathbb{E}_{X_{t-T:t}\sim\mathcal{D}}(L(h(X_{t-T:t}), l(X_{t-T:t})))$. The hypothesis returned by the learning algorithm is

$$h^* = \underset{h\in\mathcal{H}}{\operatorname{argmin}}\, \mathcal{L}_{(\mathcal{D},l_s)}(h), \tag{1}$$

where $\mathcal{H}$ is any hypotheses set. Let $\mathcal{H}^*$ denote the set of hypotheses returned by the algorithm, i.e., $\mathcal{H}^* = \{h^* : \mathcal{L}_{(\mathcal{D},l_s)}(h^*) < \epsilon\}$, and define the discrepancy measure that quantifies the divergence between the source and target domain as (Kuznetsov and Mohri 2016):

$$disc(\mathcal{H}^*) = \sup_{h\in\mathcal{H}^*} |\mathcal{L}_{(\mathcal{D},l_s)}(h) - \mathcal{L}_{(\mathcal{D},l_\tau)}(h)|. \tag{2}$$

The objective is to train a hypothesis in the source domain with a low discrepancy to domain generalization. We address the challenge by developing GCNs that incorporate domain ODEs, and our methodology is described as follows.

Methodology

Constructing domain-informed GCNs involves three steps:

•Define the domain-specific graph. The unweighted graph $\mathcal{G}$ defined earlier should correspond to the real-world network. Each vertex is associated with a time sequence of data, and edges connect nodes to their neighboring nodes such that the domain equations define the evolution of data at a vertex as a function of the data at 1-hop neighbors.

•Construct the domain-informed feature encoding function. Let $x_i(t)$ denote a feature at vertex i at time t ,and $H^i_{t,T}$ denote the length T history of data prior to time t, and set $\mathcal{N}_i$ of 1-hop neighbors of vertex i. The ODE models the feature dynamics at vertex i is given by

$$\frac{dx_i(t)}{dt} = f_i(x_i(t), \{x_j(t)|j \in \mathcal{N}_i\}) + g_i(H^i_{t,T}), \tag{3}$$

where $f_i(x_i(t), \{x_j(t)|j \in \mathcal{N}_i\})$ models the evolution of feature (Asikis, Böttcher, and Antulov-Fantulin 2022; Xhonneux, Qu, and Tang 2020) at vertex i as a dynamic system related only to the feature at vertex i and the neighboring vertices at current time. Among other things, f encapsulates the invariant physical properties of the network. For example, in transportation networks, demand patterns might change, but traffic flow dynamics would not. In disease transmission, travel patterns might change, but the dynamics of infection transmission would not. In Eq. (3), the influence of the temporal patterns on the measurement cannot be captured by the immediate dynamics is captured by the function g_i, which takes the feature history in a $T-$length window as input. In Eq. (3), $g_i(H^i_{t,T})$ is the function of the feature history in a $T-$length window prior to time t and data from non-adjacent vertices at time t. The pattern-specific function g_i is used to capture some impact of the past data[1] and the impact from distant vertices[2]. The ODE models immediate dynamics are widely studied in many domains. Thus, the function f_i is usually considered a known function (Maier et al. 2019), while the pattern-specific function g_i is difficult to capture, and is generally considered an unknown function. Due to an unknown function, most of the existing deep learning models design complex architectures to approximate g_i. However, in our context, we assume mismatched patterns between domains mentioned earlier lead to the difference between the pattern-specific function g_i in the source and target domain, i.e., let $g_{s,i}$ and $g_{\tau,i}$ denote the pattern-specific function at vertex i in source and target domain respectively. The difference between the labeling function in the respective domain (i.e., l_s and l_τ) is caused by

$$g_{s,i}(H^i_{t,T}) \neq g_{\tau,i}(H^i_{t,T}). \tag{4}$$

The GCNs incorporating domain ODEs is a family of GCNs that incorporate the domain equations f_i to learn only the immediate dynamics to be robust to the domain generalization. We employ a feature extraction function, O, to encode inputs by selecting the relevant input by utilizing a domain graph:

$$O(X(t), \mathcal{A}) = \mathcal{A} \otimes X(t), \tag{5}$$

where $\otimes$ is the Kronecker product, and $\mathcal{A}$ is the adjacency matrix of graph $\mathcal{G}$. We then generalize the local domain Eq. (3) to a graph-level representation:

$$\frac{dX(t)}{dt} = F(O(X(t), \mathcal{A}); \Theta_1) + G(O(X(t), \mathbb{1}-\mathcal{A}), X_{t-T:t-1}; \Theta_2), \tag{6}$$

where $\mathbb{1}$ is the all-one matrix, F (resp. G) with parameters Θ_1 (resp. Θ_2) is a collection of $\{f_1, ...f_n\}$ (resp. $\{g_1, ..., g_n\}$) of the encoded domain-specific features.

•Define the Network Prediction Function. The domain-ODE-informed GCNs only learn F. Thus a network-level prediction using the finite difference method is:

$$\begin{aligned}\hat{X}(t+1) &= X(t) + \int_t^{t+1} F(O(X(t), \mathcal{A}); \Theta_1) dt \\ &\approx X(t) + F(O(X(t), \mathcal{A}); \Theta_1).\end{aligned} \tag{7}$$

Proof of Robustness to Domain Generalization

Without incorporating domain ODEs, most GNNs need longer data streams to make accurate predictions. For instance, black-box predictors in the traffic domain require

[1] E.g., the congestion is caused by the increasing traffic demand.

[2] E.g., temporary change of travel demand.

12 time points to predict traffic speeds, whereas the domain informed GCN we develop requires only 1 time point as it explicitly incorporates the immediate dynamics instead of learning arbitrary functions. (see Eq.(12)). We will discuss the application-specific GCNs in the subsequent section. In the following, we will prove that when the underlying dynamics connect the features at consecutive time points, the approach that incorporates the dynamics is more robust to the domain generalization problem defined by the discrepancy equation in Eq. (2). Similar to the approach in (Redko et al. 2020), we assume the training set is sampled from the source domain, and the test data is sampled from the target domain. In this work, we formulate the mismatch problem as a difference between labeling functions in the source and target domains where the immediate time and nearest neighbor dynamics (function F) are unchanging across domains. In contrast, the impact of long-term and distant neighbor patterns (function G) varies between source and target domains. We observe that although both G_s (resp. G_τ) and F utilize $X(t)$ as part of their input, they consistently select features from distinct nodes. Thus there is no overlap between inputs of G_s (resp. G_τ) and F.

Under such a mismatch scenario, we will show that using long-term patterns and data from nodes outside the neighborhood will have worse generalization as measured by a discrepancy function. We use $\mathcal{H}_1$ to denote the hypothesis set mentioned earlier that predicts the data at time $t+1$ based on a T-length history (from $t-T$ to t, where $T>1$), and $\mathcal{H}_2$ denotes the hypothesis set that uses the data only at time t to predict the speed at $t+1$. In other words, baseline algorithms that use several time points and data from nodes outside the 1-hop neighborhood would fall into $\mathcal{H}_1$. In contrast, algorithms such as ours, which use domain ODEs to incorporate the known functional form F, which requires only immediate and nearest neighbor data, would belong to $\mathcal{H}_2$. We make the following two assumptions:

Assumption 1. There exists $h_1^* \in \mathcal{H}_1$ s.t. $\mathcal{L}_{(\mathcal{D},F+G_s)}(h_1^*) = 0$. There exists $h_2^* \in \mathcal{H}_2$ s.t. $\mathcal{L}_{(\mathcal{D},F)}(h_2^*) = 0$.

Assumption 2. Let $U = G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2)$ be a random variable where $X_{t-T:t} \sim \mathcal{D}$ and $P_U(G)$ be the probability distribution function (PDF) of U. The PDF $P_U(G)$ is symmetric about 0.

Assumption 1 ensures the learnability of the hypotheses. Assumption 2 ensures that the statistical impact of the long-term pattern is unbiased and symmetric. In the appendix, we show the used datasets satisfy these assumptions. The above assumptions lead to the following Lemmas about optimal hypotheses learned by domain-agnostic methods, such as the baselines, and those learned by our domain-informed methods, such as ours.

Lemma 1. $h_1^*(X_{t-T:t}) = F(O(X(t), \mathcal{A}); \Theta_1) + G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2)$.

Proof. Follows Assumption 1 when $\mathcal{L}_{(\mathcal{D},l_s)}(h_1^*) = 0$. □

Lemma 2. If (1) h_2 is trained with data sampled from $\mathcal{X}_s$ such that assumption 2 is true, (2) the loss function L is the L1-norm or MSE, then $h_2^* = F$.

To theoretically establish the enhanced robustness of our approach, we assume the PAC learnability of $\mathcal{H}_1$ and $\mathcal{H}_2$. In detail, with sufficient data, for every $\epsilon_1, \epsilon_2, \delta \in (0,1)$, if Assumption 1 holds with respect to $\mathcal{H}_1, \mathcal{H}_2$, then when running the learning algorithm using data generated by distribution $\mathcal{D}$ and labeled by $F + G_s$, with the probability of at least $1-\delta$, the hypothesis h_1^* is in the set

$$\mathcal{H}_1^* = \{h_1^* : \mathcal{L}_{(\mathcal{D},F+G_s)}(h_1^*) < \epsilon_1\}, \tag{8}$$

and with the probability of at least $1-\delta$, h_2^* is in the set

$$\mathcal{H}_2^* = \{h_2^* : \mathcal{L}_{(\mathcal{D},F)}(h_2^*) < \epsilon_2\}. \tag{9}$$

We will now demonstrate that $\mathcal{H}_2^*$ is more robust to the domain generalization than $\mathcal{H}_1^*$ using the discrepancy measure defined in Eq. (2). For our theoretical result, we require that the loss function $L(h,l)$ satisfy triangle inequality:

$$|L(h,h') - L(h',l)| \leq L(h,l) \leq L(h,h') + L(h',l), \tag{10}$$

where h' is any other hypothesis. The following theorem proves our result.

Theorem 3. *If (1) the training data is sampled from the source domain where assumption 2 is true, (2) the loss function $L(h,l)$ obeys the triangular inequality, then the discrepancy should satisfy*

$$disc(\mathcal{H}_2^*) \leq disc(\mathcal{H}_1^*). \tag{11}$$

Theorem 3 illustrates that models trained using lengthy time sequences and distant nodes are not reliable when there are mismatches between the labeling functions in the source domain and target domain. Loss functions that include mean absolute error and root mean squared error satisfy the triangle inequality assumption. We note that this assumption is not satisfied by the mean-squared error (MSE) loss function. In the Appendix, we prove an extension of this theorem for MSE by making an additional assumption on the data.

Application of Domain-ODE informed GCNs

In the following part of this section, we will use the Reaction-Diffusion equation and SIR-network differential equation as examples to develop practical domain-informed GCNs.

Reaction Diffusion GCN for Traffic Speed Prediction. Bellocchi in (Bellocchi and Geroliminis 2020) proposed the reaction-diffusion approach to reproduce traffic measurements such as speed and congestion using few observations. The domain differential equations included a *Diffusion* term that tracks the influence in the direction of a road segment, while the *Reaction* term captures the influence opposite to the road direction. 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. Consider sensor i, let $\mathcal{N}_i^d$ denote the set of sensor i's neighbors in the road segment direction, and let $\mathcal{N}_i^r$ denote the set of the neighbors in the opposite direction of the sensor i. If $x_i(t)$ denotes the speed observed at node i at time t, the local reaction-diffusion equation[3] at

[3] See the appendix for more details.

vertex i can be formulated as

$$\frac{dx_i(t)}{dt} = \sum_{j \in \mathcal{N}^d} \rho_{(i,j)} \left(x_j(t) - x_i(t)\right) + b_i^d + \tanh\left(\sum_{j \in \mathcal{N}^r} \sigma_{(i,j)}(x_j(t) - x_i(t)) + b_i^r\right), \tag{12}$$

where $\rho_{(i,j)}$ and $\sigma_{(i,j)}$ are the diffusion parameter and reaction parameter, respectively; b_i^d and b_i^r are biases to correct the average traffic speed at vertex i in diffusion and reaction.

In the following, we incorporate this reaction-diffusion (RD) equation using the steps outlined in Section Methodology to build a novel GCN model for domain-informed prediction of traffic speed.

Step 1: Define reaction and diffusion parameters. 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 $\mathcal{A}^r = \mathcal{A}^\top$, where $\top$ denotes matrix transpose. 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 $\mathcal{E}^r$, respectively. $\mathbf{W}^d \in \mathbb{R}^{n \times n}$ is a sparse weight matrix for diffusion graph $\mathcal{G}^d$, where $\mathbf{W}^d_{i,j} = \rho_{(i,j)}, \forall (i,j) \in \mathcal{E}^d$, otherwise $\mathbf{W}^d_{i,j} = 0$. $\mathbf{W}^r$ for reaction graph $\mathcal{G}^r$ is defined in a similar way, but the non-zero element at $(i,j) \in \mathcal{E}^r$ is $\sigma_{(i,j)}$.

Step 2: Construct RD feature encoding function. 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

$$\begin{aligned}(\mathbf{L}^d X(t))_i &= \sum_{(i,j) \in \mathcal{E}^d} (\mathbf{W}^d \odot \mathcal{A}^d)_{i,j} (X_j(t) - X_i(t)) \\ &= ((Degree(\mathbf{W}^d \odot \mathcal{A}^d) - \mathbf{W}^d \odot \mathcal{A}^d) X(t))_i,\end{aligned} \tag{13}$$

where $\odot$ denotes the Hadamard product, $Degree(*)$ is to calculate the degree matrix of an input adjacency matrix, and $(\mathbf{L}^r X_t)_i$ represents a similar reaction process but the weight tensor is $\mathbf{W}^r$ and Adjacency matrix is $\mathcal{A}^r$. Specifically, the reaction and diffusion laplacian $\mathbf{L}^r$ and $\mathbf{L}^d$ is the RD-informed feature encoding function O extracting speed differences between neighboring vertices.

Step 3: Using Eq (7) we can define a prediction:

$$\hat{X}(t+1) = X(t) + (\mathbf{L}^d X(t) + b^d) + \tanh\left(\mathbf{L}^r X(t) + b^r\right), \tag{14}$$

where $\mathbf{L}^d$ and $\mathbf{L}^r$ is the reaction and diffusion functions constructed earlier, corresponds to the function $F = (\mathbf{L}^d X_t + b^d) + \tanh\left(\mathbf{L}^r X_t + b^r\right)$ predicting the traffic speed using the reaction parameters ρ and the diffusion parameters σ.

Susceptible-Infected-Recovered (SIR)-GCN for Infectious disease prediction. The SIR model is a typical model describing the temporal dynamics of an infectious disease by dividing the population into three categories: Susceptible to the disease, Infectious, and Recovered with immunity. The SIR model is widely used in the study of diseases such as influenza and Covid (Cooper, Mondal, and Antonopoulos 2020). Our approach is based on the SIR-Network Model proposed to model the spread of Dengue Fever (Stolerman, Coombs, and Boatto 2015), which we describe as follows. Let $S_i(t)$, $I_i(t)$, $R_i(t)$ denote the number of Susceptible, Infectious, and Recovered at vertex $i \in \mathcal{V}$ at time t respectively and the total population at vertex i is assumed to be a constant, i.e., $N_i = S_i(t) + I_i(t) + R_i(t)$.

Step 1: Define the travel matrices. The spread of infection between nodes is modeled using sparse travel matrices $\mathbf{\Phi} \in [0,1]^{n \times n}$ as $\phi(i,j), \forall (i,j) \in \mathcal{E}^d$; otherwise $\phi(i,j) = 0$, where $\phi_{(i,j)} \in [0,1]$ is a parameter representing the fraction of resident population travel from i to j, therefore we require the fractions satisfy $\sum_{k=1}^n \phi_{(i,j)} = 1, \forall i \in \mathcal{V}$. The SIR-network model at vertex i is defined as:

$$\begin{aligned}\frac{dS_i(t)}{dt} &= -\sum_{j=1}^M \sum_{k=1}^M \beta_j \phi_{(i,j)} S_i(t) \frac{\phi_{(k,j)} I_k(t)}{N_j^p}, \\ \frac{dI_i(t)}{dt} &= \sum_{j=1}^M \sum_{k=1}^M \beta_j \phi_{(i,j)} S_i(t) \frac{\phi_{(k,j)} I_k(t)}{N_j^p} - \gamma I_i, \\ \frac{dR_i(t)}{dt} &= \gamma I_i(t),\end{aligned} \tag{15}$$

where β_i is the infection rate at vertex i, representing the probability that a susceptible population is infected at vertex i, and γ is the recovery rate, representing the probability that an infected population is recovered, $N_i^p = \sum_{k=1}^n \phi_{(i,j)} N_k$ is the total population travel from all vertices to vertex i. We assume the recovery rates at all vertices are the same.

Step 2: Construct the SIR function. The differential equation system (15) is equivalent to:

$$\frac{dI(t)}{dt} = (\mathcal{K} - \gamma) I(t), \tag{16}$$

where $I(t)$ is the feature ($X(t)$ mentioned earlier) representing the number of Infectious people. Then the transformation matrix $\mathcal{K}$ connecting $I(t)$ and $I(t+1)$ at neighboring time is:

$$\mathcal{K}_{i,j} = \sum_{j=1}^n \beta_j \phi_{(i,j)} \phi_{(k,j)} \frac{S_i}{N_j^p}, \tag{17}$$

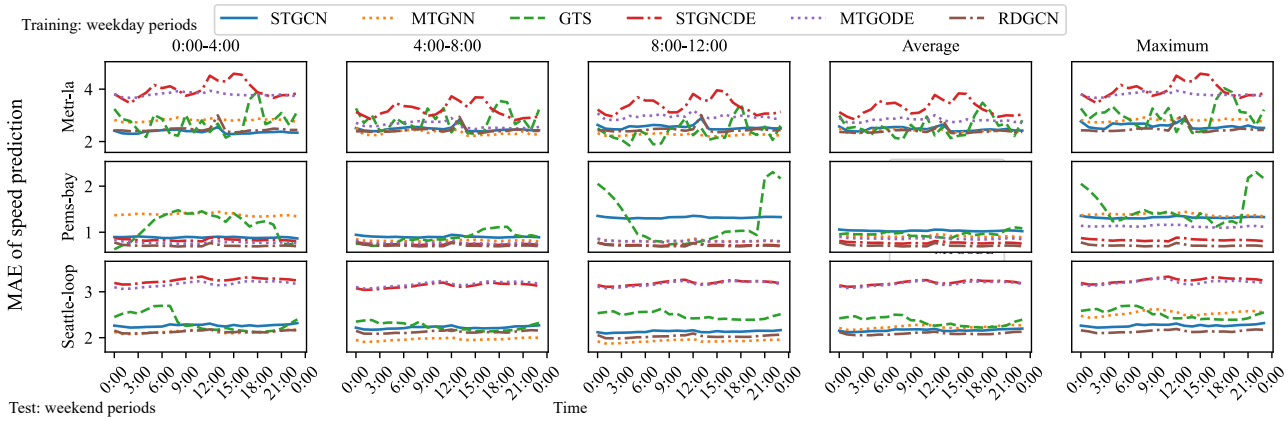
where $S_i(t) = N_i - I_i(t) - R_i(t)$ and $R_i(t) = \gamma \int_{t_0}^t I(t) = \gamma \sum_{t_0}^t I(t)$, t_0 is the starting time of the current epidemic. The domain-informed feature encoding function O is utilized to approximate the counts of susceptible and recovered populations and estimate the infectious people likely to travel, approximated by the flight data.

Step 3: Using Eq. (7), (16), and (17), the prediction is

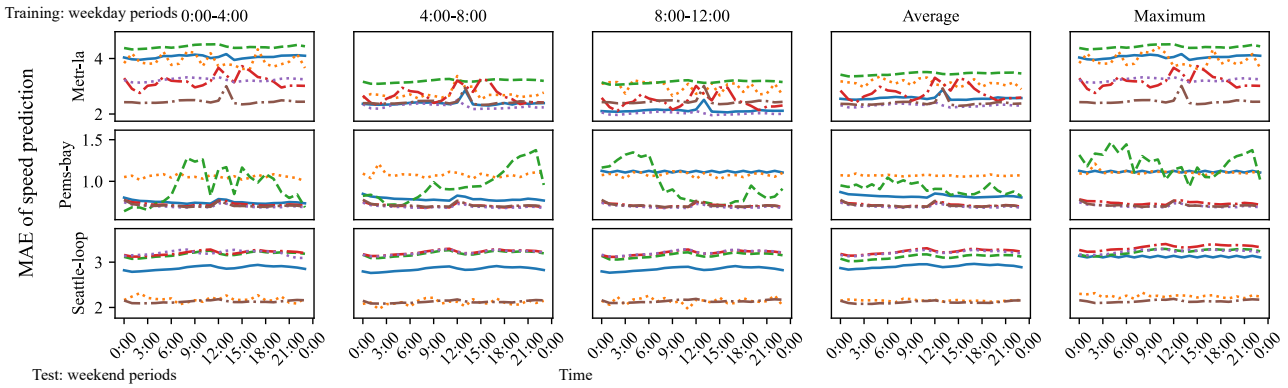
$$\hat{I}(t+1) = I(t) + (\mathcal{K} - \gamma) I(t). \tag{18}$$

Evaluation

In this section, we compare the performance of these domain-ODE-informed GCNs with baselines when tested with mismatched data and demonstrate that our approach is more robust to such mismatched scenarios.



(a) Baseline models and RDGCN trained on 12 consecutive weekdays and all models are augmented by MAML



(b) Baseline models and RDGCN trained on more than half a year of weekdays

Figure 2: (a) The results of RDGCN are very close regardless of the period of the training set. (b) Even though all the models are trained using all available weekdays, the results of RDGCN are still closer regardless of the period, compared to baseline models. The numerical result, the plot in the other three time window, and the corresponding result for RMSE are in Ablation Study in the appendix.

Experiment Settings

Datasets. Our experiments are conducted on three real-world datasets (Metra-la, Pems-bay and Seattle-loop) for traffic prediction, and two real-world datasets (in Japan and US) for disease prediction. The details are shown in Table 1.

Table 1: Dataset statistics

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	resolution	period
Metr-la (Jagadish et al. 2014)	207	233	5 mins	122 days
Pems-bay (Li et al. 2018)	281	315	5 mins	151 days
Seattle-loop (Cui et al. 2020)	323	660	5 mins	365 days
Japan-Prefectures (Deng et al. 2020)	47	133	weekly	347 weeks
US-States (Deng et al. 2020)	49	152	weekly	834 weeks

Evaluation Metric. The loss function we use is the mean absolute error and the root mean square error: $MAE(X(t), \hat{X}(t)) = \frac{1}{n}\sum_{i=1}^{n}|x_i(t) - \hat{x}_i(t)|$, $RMSE(X(t), \hat{X}(t)) = (\frac{1}{n}\sum_{i=1}^{n}(x_i(t) - \hat{x}_i(t)))^2)^{\frac{1}{2}}$. We also use MAE and RMSE to evaluate models. We note that both these loss functions satisfy the triangle inequality.

Baselines. For traffic prediction tasks, we compare RDGCN with STGCN (Yu, Yin, and Zhu 2018), MTGNN (Wu et al. 2020), GTS (Shang, Chen, and Bi 2021), STGNCDE (Choi et al. 2022), and MTGODE (Jin et al. 2022). They are influential and best-performing deep learning models for predicting traffic speed using historical speed alone. We also use Model-Agnostic Meta-Learning (MAML) (Finn, Abbeel, and Levine 2017) to help baseline models, and our approach adapts quickly to tasks using good initial weights generated by MAML. For disease prediction, we compare SIRGCN with two state-of-the-art models for infection prediction, CoLaGNN (Deng et al. 2020) and EpiGNN (Xie et al. 2023).

Results and analysis

Mismatched Data Experiments for RDGCN. We first explore the performance of the models when they are trained using mismatched data from certain conditions and tested using alternate, mismatched conditions. Specifically, the models are trained with four-hour data on weekdays (e.g.,8:00-12:00 on weekdays) selected and evaluated with hourly data on weekends (e.g., 13:00-14:00 on weekends). In limited data and mismatched conditions (Figure 2a), the training set consists of data from five different sequences of 12 consecutive weekdays selected randomly from the available data. This experiment aims to replicate scenarios where data collection is challenging, and traffic patterns undergo rapid changes. In mismatched conditions without data limitations (Figure 2b), the training set consists of data from all available weekdays. This captures instances where data collection is compara-

tively less arduous, although the traffic pattern retains the potential to shift swiftly. The results are shown in Figure 2, where each curve in the first three columns denotes the average test prediction MAE of models (resp. the average test prediction MSE of models in the appendix). We generate two summary figures illustrating the average and maximum MAE across all six training sets. In Figure 2a, we compare the performance of our approach with the STGCN, MTGNN, GTS, MTGODE, STGNCDE, and RDGCN in the mismatched data, when the training process is augmented with MAML. Figure 2b plots the prediction MAE of baseline models and RDGCN over time, given all available weekday data.

In Figure 2a, all RDGCN models have nearly identical performance regardless of which time window of data is used for training. The MAE of all the RDGCN models is uniformly low (i.e., small y-axis values), and there is very low variance in performance across RDGCN models trained with different time windows (i.e., the curves of average MAE is close to the curves of maximum MAE). However, the performance of baseline models are significantly different depending on the training set, and some can have a relatively high MAE (e.g., the curve of STGCN on Pems-bay dataset has much higher MAE values than the one of RDGCN over time). From Figure 2b, we can see that even when the model is trained using all available weekday data, RDGCN outperforms the baseline models wherein the variance across time, and across models is very low. While more data brings some gain to baseline models, its impact on RDGCN is fairly limited, indicating that RDGCN performs well in different testing domains without needing additional training data.

These test results support our hypothesis that incorporating traffic dynamics into the learning model makes it more robust to this kind of domain generalization (data from mismatched training and testing conditions). We speculate that this is a consequence of our model capturing the relative changes in speed through the dynamical equations, whereas existing baseline models are black box models that derive complex functions of the absolute speed values across time. 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, RDGCN 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), the model uses close sensors to make predictions. We note that the prediction of RDGCN is not uniformly better than the prediction of baselines (e.g., the prediction of MTGNN trained by Seattle weekday data from 8:00 to 12:00 is better than the prediction of RDGCN), 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). Furthermore, the predictions of MTGNN and MTGODE exhibit a slight superiority over RDGCN in Metr-la dataset in certain windows. Our conjecture is that the mix-hop layers enable these models to assign higher significance to learn short-term patterns, which likely does not change much between the training and test data. 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 to demonstrate the robustness of our approach.

Table 2: Evaluation of models under mismatched data

	Dataset	ColaGNN	EpiGNN	SIRGCN
MAE	Japan-Prefectures	356 ± 21	466 ± 24	342 ± 22
	US-States	46 ± 3	66 ± 6	41 ± 4
RMSE	Japan-Prefectures	901 ± 53	922 ± 69	863 ± 44
	US-States	130 ± 12	178 ± 16	121 ± 10

Mismatched Data Experiments for SIRGCN. We explore the performance of SIRGCN under mismatched situations. Since infection spread and travel patterns vary from season to season, we train our model and the baseline models with ILI data recorded in Summer and Winter, and test the predictions on data in Spring and Fall. The result is shown in Table 2, where each element denotes the MAE and MSE under different seasons.

The results demonstrate that SIRGCN performs consistently well under the mismatched data scenario with low MAE and RMSE compared to the baseline models. Although SIRGCN does not significantly outperform the deep-learning-based ColaGNN model, we note that SIRGCN makes predictions using only the latest observation at 1 time point augmented by approximating the total susceptible and recovered populations as specified by the domain equations, whereas the baselines which consider the disease propagation as a black-box model, require more than 7 years data to train, and twenty weeks worth data to make their predictions.

The two datasets are used for testing, but the theory can also apply to other applications, such as air quality forecasting, molecular simulation, and others where there are underlying graphical models and the domain ODE is well developed. Overall these evaluations validate the the main hypothesis of this paper where integrating domain differential equations into GCN allows for better robustness.

Conclusion

In this paper, we investigate the challenging problem of graph time series prediction when training and test data are drawn from different or mismatched scenarios. To address the challenge, we proposed a methodological approach to integrate domain differential equations in graph convolutional networks to capture the common data behavior across data distributions. We theoretically justify the robustness of this approach under certain conditions on the underlying domain and data. By operationalizing our approach, we gave rise to two novel domain-informed GCNs: RDGCN and SIRGCN. These architectures fuse traffic speed reaction-diffusion equations, and Susceptible-Infected-Recovered infectious disease spread equations, respectively. Through rigorous numerical evaluation, we demonstrate the robustness of our models in mismatched data scenarios. The findings showcased in this work underscore the transformative potential of domain-ODE-informed models as a burgeoning category within the domain of graph neural networks. This framework paves the way for future exploration in addressing the challenges of domain generalization in other contexts.

References

Asikis, T.; Böttcher, L.; and Antulov-Fantulin, N. 2022. Neural ordinary differential equation control of dynamics on graphs. *Physical Review Research*.

Bellocchi, L.; and Geroliminis, N. 2020. Unraveling reaction-diffusion-like dynamics in urban congestion propagation: Insights from a large-scale road network. *Scientific reports*.

Bui, K.-H. N.; Cho, J.; and Yi, H. 2022. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Applied Intelligence*, 52(3): 2763–2774.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Choi, J.; Choi, H.; Hwang, J.; and Park, N. 2022. Graph neural controlled differential equations for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6367–6374.

Cooper, I.; Mondal, A.; and Antonopoulos, C. G. 2020. A SIR model assumption for the spread of COVID-19 in different communities. *Chaos, Solitons & Fractals*.

Cui, Z.; Ke, R.; Pu, Z.; Ma, X.; and Wang, Y. 2020. Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction. *Transportation Research Part C: Emerging Technologies*.

Deng, S.; Wang, S.; Rangwala, H.; Wang, L.; and Ning, Y. 2020. Cola-GNN: Cross-location attention based graph neural networks for long-term ILI prediction. In *CIKM*.

Fan, X.; Wang, Q.; Ke, J.; Yang, F.; Gong, B.; and Zhou, M. 2021. Adversarially adaptive normalization for single domain generalization. In *CVPR*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Guo, S.; Lin, Y.; Feng, N.; Song, C.; and Wan, H. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI*.

Han, L.; Du, B.; Sun, L.; Fu, Y.; Lv, Y.; and Xiong, H. 2021. Dynamic and Multi-faceted Spatio-temporal Deep Learning for Traffic Speed Forecasting. In *SIGKDD*.

Jagadish, H. V.; Gehrke, J.; Labrinidis, A.; Papakonstantinou, Y.; Patel, J. M.; Ramakrishnan, R.; and Shahabi, C. 2014. Big data and its technical challenges. *Communications of the ACM*.

Jayatilaka, G.; Hassan, J.; Marikkar, U.; Perera, R.; Sritharan, S.; Weligampola, H.; Ekanayake, M.; Godaliyadda, R.; Ekanayake, P.; Herath, V.; et al. 2020. Use of Artificial Intelligence on spatio-temporal data to generate insights during COVID-19 pandemic: A Review. *MedRxiv*, 2020–11.

Ji, J.; Wang, J.; Jiang, Z.; Jiang, J.; and Zhang, H. 2022. STDEN: Towards physics-guided neural networks for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 4048–4056.

Jia, J.; and Benson, A. R. 2019. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems*, 32.

Jin, M.; Zheng, Y.; Li, Y.-F.; Chen, S.; Yang, B.; and Pan, S. 2022. Multivariate time series forecasting with dynamic graph neural odes. *IEEE Transactions on Knowledge and Data Engineering*.

Kessels, F.; Kessels; and Rauscher. 2019. *Traffic flow modelling*. Springer.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. *ICLR*.

Kuznetsov, V.; and Mohri, M. 2016. Time series prediction and online learning. In *COLT*.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *ICLR*.

Loder, A.; Ambühl, L.; Menendez, M.; and Axhausen, K. W. 2019. Understanding traffic capacity of urban networks. *Scientific reports*, 9(1): 1–10.

Longa, A.; Lachi, V.; Santin, G.; Bianchini, M.; Lepri, B.; Lio, P.; Scarselli, F.; and Passerini, A. 2023. Graph Neural Networks for temporal graphs: State of the art, open challenges, and opportunities. *arXiv preprint arXiv:2302.01018*.

Maier, A. K.; Syben, C.; Stimpel, B.; Würfl, T.; Hoffmann, M.; Schebesch, F.; Fu, W.; Mill, L.; Kling, L.; and Christiansen, S. 2019. Learning with known operators reduces maximum error bounds. *Nature machine intelligence*.

Qiao, F.; Zhao, L.; and Peng, X. 2020. Learning to learn single domain generalization. In *CVPR*.

Redko, I.; Morvant, E.; Habrard, A.; Sebban, M.; and Bennani, Y. 2020. A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv preprint arXiv:2004.11829*.

Robey, A.; Pappas, G. J.; and Hassani, H. 2021. Model-based domain generalization. *NeurIPS*.

Scholz, G.; and Scholz, F. 2015. First-order differential equations in chemistry. *ChemTexts*, 1: 1–12.

Shang, C.; Chen, J.; and Bi, J. 2021. Discrete graph structure learning for forecasting multiple time series. *ICLR*.

Stolerman, L. M.; Coombs, D.; and Boatto, S. 2015. SIR-network model and its application to dengue fever. *SIAM Journal on Applied Mathematics*.

van Wageningen-Kessels, F.; Van Lint, H.; Vuik, K.; and Hoogendoorn, S. 2015. Genealogy of traffic flow models. *EURO Journal on Transportation and Logistics*, 4(4): 445–473.

Varshney, K. R. 2020. On mismatched detection and safe, trustworthy machine learning. In *CISS*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. *ICLR*.

Wang, J.; Lan, C.; Liu, C.; Ouyang, Y.; Qin, T.; Lu, W.; Chen, Y.; Zeng, W.; and Yu, P. 2022. Generalizing to unseen domains: A survey on domain generalization. *TKDE*.

Wang, Z.; Luo, Y.; Qiu, R.; Huang, Z.; and Baktashmotlagh, M. 2021. Learning to diversify for single domain generalization. In *ICCV*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *SIGKDD*.

Xhonneux, L.-P.; Qu, M.; and Tang, J. 2020. Continuous graph neural networks. In *ICML*.

Xian, X.; Hong, M.; and Ding, J. 2022. Mismatched Supervised Learning. In *ICASSP*.

Xie, F.; Zhang, Z.; Li, L.; Zhou, B.; and Tan, Y. 2023. EpiGNN: Exploring spatial transmission with graph neural network for regional epidemic forecasting. In *ECML PKDD*.

Ying, Z.; Bourgeois, D.; You, J.; Zitnik, M.; and Leskovec, J. 2019. Gnnexplainer: Generating explanations for graph neural networks. In *NeurIPS*.

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *IJCAI*.

Zhou, K.; Liu, Z.; Qiao, Y.; Xiang, T.; and Loy, C. C. 2021. Domain generalization in vision: A survey. *arXiv preprint arXiv:2103.02503*.

Zhou, K.; Liu, Z.; Qiao, Y.; Xiang, T.; and Loy, C. C. 2022. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

APPENDIX

Proofs

Proof of Lemma 2

Lemma 2. If (1) h_2 is trained with data sampled from $\mathcal{X}_s$ such that assumption 2 is true, (2) the loss function L is the L1-norm or MSE, then $h_2^* = F$.

Proof. We prove this by contradiction. If $h_2^* \neq F$, there must exist $\hat{h}_2^*(X_t) \neq 0$ such that $h_2^*(X_t) = F(O(X(t), \mathcal{A}); \Theta_1) + \hat{h}_2^*(X_t)$ and $\hat{h}_2^*$ minimizes the following expectation:

$$\hat{h}_2^* = \min_{\hat{h}_2 : h_2 - F} \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[L(\hat{h}_2(X(t)), G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))]. \tag{19}$$

If the loss function is the L1-norm, Problem (19) is minimized when $\hat{h}_2^*(X_t)$ equals the median of $G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2)$. Assumption 2 in Section 5 implies

$$Median_{X_{t-T:t} \sim \mathcal{D}}[G] = \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[U] = 0. \tag{20}$$

Thus $\hat{h}_2^*(X_t) = 0$ is the optimal solution of Problem (19), which contradicts the fact that $\hat{h}_2^*(X_t) \neq 0$.
If the loss function is the MSE, there must exist $\hat{h}_2(X_t) \neq 0$ such that $h_2^*(X_t) = F(O(X(t), \mathcal{A}); \Theta_1)$ and $\hat{h}_2(X_t)$ minimize the following expectation

$$\mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[(\hat{h}_2(X_t) - G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))^2]. \tag{21}$$

Since Assumption 2 in Section 5 implies

$$\mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2)] = 0, \tag{22}$$

the expectation $\mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[(\hat{h}_2(X_t) - G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))^2]$ is minimized when the derivative $2(\hat{h}_2(X(t)) - \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2)]) = 0$, hence $\hat{h}_2(X_t) = 0$ must minimize the expectation in Eq. (21), which contradicts the fact that $\hat{h}_2(X_t) \neq 0$. Therefore $h_2^* = F$. □

Proof of Theorem 3

Theorem 3. If (1) the training data is sampled from the source domain where assumption 2 is true, (2) the loss function $L(h, l)$ obeys the triangular equality, then the discrepancy with any triangular equality loss should satisfy

$$disc(\mathcal{H}_2^*) \leq disc(\mathcal{H}_1^*). \tag{23}$$

Proof. By the definition of discrepancy in Eq.(2), we know

$$\begin{aligned}
disc(\mathcal{H}_1^*) &= \sup_{h_1 \in \mathcal{H}_1^*} |\mathcal{L}_{(\mathcal{D}, F+G_s)}(h_1) - \mathcal{L}_{(\mathcal{D}, F+G_\tau)}(h_1)| \\
&= \sup_{h_1 \in \mathcal{H}_1^*} |\mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[L(h_1(X_{t-T:t}), F(X(t)) \\
&\qquad + G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2)) \\
&\qquad - L(h_1(X_{t-T:t}), F(X(t)) \\
&\qquad + G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))]| \\
&\overset{(a)}{\leq} \sup_{h_1 \in \mathcal{H}_1^*} \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[|L(h_1(X_{t-T:t}), F(X(t)) \\
&\quad + G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2)) \\
&\qquad - L(h_1(X_{t-T:t}), F(X(t)) \\
&\qquad + G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))|] \\
&\overset{(b)}{\leq} \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[L(G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2), \\
&\qquad G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))],
\end{aligned} \tag{24}$$

where (a) follows from Jensen's equality ($|\cdot|$ is convex) and (b) follows from the triangle inequality (which implies $|L(x, y)| \geq |L(x, z) - L(y, z)|$, for any $x, y, z \in \mathbb{R}$). By Assumption 1 in Section 5, we can set $h_1^* = F + G_s$ where $\mathcal{L}_{(\mathcal{D}, F+G_s)}(h_1^*) = 0$. Then the discrepancy of $\mathcal{H}_1$ is

$$\begin{aligned}
disc(\mathcal{H}_1^*) &\overset{(c)}{\geq} \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[L(F(X(t)) \\
&\qquad + G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2), F(X(t)) \\
&\qquad + G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))] \\
&= \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[L(G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2), \\
&\qquad G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))],
\end{aligned} \tag{25}$$

where (c) follows from the definition that the supremum (the least element that is greater than or equal to each element in the set). Thus from (24) and (25) together

$$\begin{aligned}
disc(\mathcal{H}_1^*) = \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[&L(G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2), \\
&G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))].
\end{aligned} \tag{26}$$

For $\mathcal{H}_2$, by the triangle inequality,

$$\begin{aligned}
disc(\mathcal{H}_2^*) &= \sup_{h_2 \in \mathcal{H}_2^*} |\mathcal{L}_{(\mathcal{D}, F+G_s)}(h_2) - \mathcal{L}_{(\mathcal{D}, F+G_\tau)}(h_2)| \\
&\leq \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[L(G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2), \\
&\qquad G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))].
\end{aligned} \tag{27}$$

Hence we have shown that $disc(\mathcal{H}_2^*) \leq disc(\mathcal{H}_1^*)$. □

Discrepancy using MSE

Assumption 3. Let $U' = G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2) G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2)$ be a random variable where $X_{t-T:t} \sim \mathcal{D}$, $\mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[U'] \leq 0$.

Corollary 1. If (1) the training data is sampled from the source domain where assumption 2 is true, (2) the labeling function in the source and target domain satisfy Assumption 3, (3) the loss function $L(h, l)$ is MSE, (4) $\mathcal{L}_{(\mathcal{D}, F)}(h_2^*) = 0$, then then $disc(\mathcal{H}_2^*) \leq disc(\mathcal{H}_1^*)$.

Proof. By Assumption 1, we can set $h_1^* = F + G_s$ where

$\mathcal{L}_{(\mathcal{D},F+G_s)}(h_1^*) = 0$, then the discrepancy of $\mathcal{H}_1$ is

$$
\begin{aligned}
disc(\mathcal{H}_1^*) = \sup_{h_1 \in \mathcal{H}_1^*} & |\mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[L(h_1(X_{t-T:t}), F(X(t)) \\
& + G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2)) \\
& - L(h_1(X_{t-T:t}), F(X(t)) \\
& + G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))]| \\
\overset{(d)}{\geq} & \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[L(F(X(t)) + G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2), \\
& \qquad F(X(t)) + G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))] \\
= & \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[(G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2) \\
& - G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))^2],
\end{aligned}
\tag{28}
$$

where (d) follows from the definition of the supremum (the least element that is greater than or equal to each element in the set). We note that the triangular equality is not necessarily true in this case thus we cannot find the upper bound of $disc(\mathcal{H}_1^*)$.

Since $\mathcal{L}_{(\mathcal{D},F)}(h_2^*) = \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[((h_2^*(X(t)) - F(O(X(t), \mathcal{A}); \Theta_1))^2] = 0$ implies that $h_2^* = F(O(X(t), \mathcal{A}); \Theta_1)$.

Then the discrepancy of $\mathcal{H}_2^*$ is:

$$
\begin{aligned}
disc(\mathcal{H}_2^*) = \sup_{h_2^* \in \mathcal{H}_2^*} & |\mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[(h_2^*(X(t)) - F(O(X(t), \mathcal{A}); \Theta_1) \\
& - G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))^2 \\
& - (h_2^*(X(t)) - F(O(X(t), \mathcal{A}); \Theta_1) \\
& - G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))^2]| \\
\overset{(e)}{\leq} & \sup_{h_2^* \in \mathcal{H}_2^*} \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[|(h_2^*(X(t)) - F(O(X(t), \mathcal{A}); \Theta_1) \\
& - G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))^2 - \\
& (h_2^*(X(t)) - F(O(X(t), \mathcal{A}); \Theta_1) \\
& - G_\tau(O_2(H_{t,T}, \mathcal{G}); \Theta_2))^2|] \\
= & \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[|(G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))^2 \\
& - (G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))^2|] \\
\leq & \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[(G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))^2 \\
& + (G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))^2] \\
\overset{(f)}{\leq} & \mathbb{E}_{X_{t-T:t} \sim \mathcal{D}}[|(G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2) \\
& - G_\tau(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2))^2|],
\end{aligned}
\tag{29}
$$

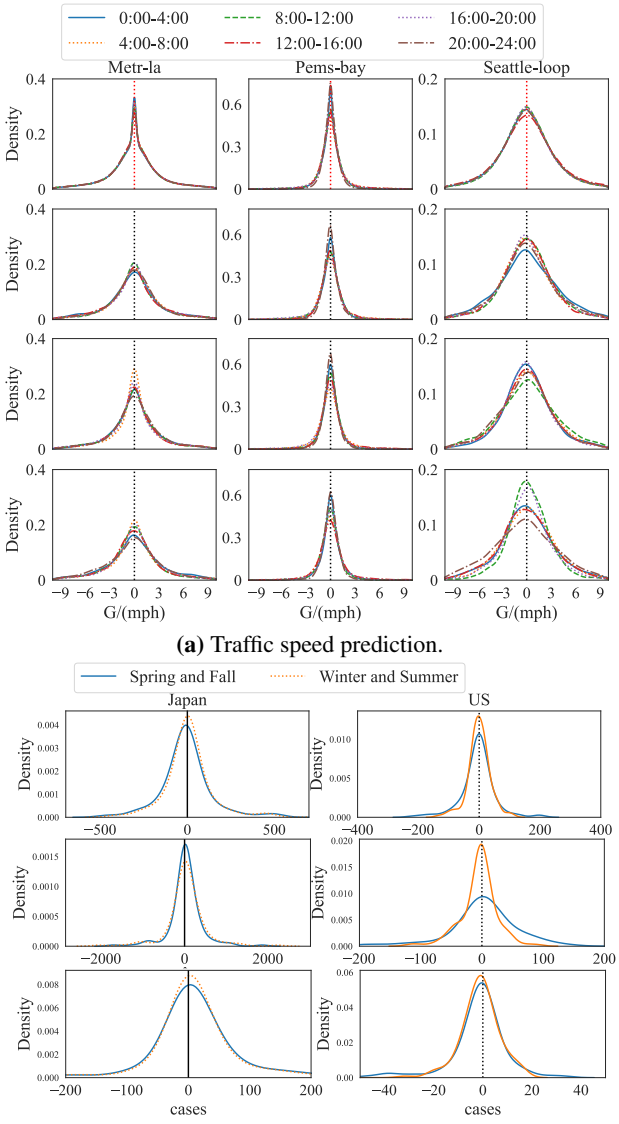
where (e) follows from Jensen's inequality, (f) follows from Assumption 3. Hence by Eq. (28)(29), we know $disc(\mathcal{H}_2^*) \leq disc(\mathcal{H}_1^*)$

□

Data Support

We empirically verify the condition in Assumption 2, in the scenario that the Reaction-Diffusion traffic model is the underlying physical law, and consequently, RDGCN, when trained well, perfectly models function f. Then,

$$
\begin{aligned}
& G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2) \\
= & X_{t+1} - F(O(X(t), \mathcal{A}); \Theta_1)) \\
\approx & X_{t+1} - RDGCN(X(t)).
\end{aligned}
\tag{30}
$$



(a) Traffic speed prediction.

(b) ILI prediction.

Figure 3: (a) The pdf of the random variable, G is symmetric about 0 for all the time periods. Figures in the first row are the mixed distribution of all sensors. Figures in the following three rows are the distribution of three randomly selected sensors in each dataset. (b) The pdf of the random variable, G is symmetric about 0 for all seasons. We randomly select 3 vertices in each data set.

The probability density function (pdf) of the random variable $G = G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2)$ in all the six periods are shown in Figure 3a, which plots the empirical density of the variable G in each dataset. As can be observed, the empirical density adheres to the condition in Assumption 2. For SIRGCN, we approximate G_s by

$$
\begin{aligned}
& G_s(O(X(t), \mathbb{1} - \mathcal{A}), X_{t-T:t-1}; \Theta_2) \\
= & X_{t+1} - F(O(X(t), \mathcal{A}); \Theta_1)) \\
\approx & X_{t+1} - SIRGCN(X(t)).
\end{aligned}
\tag{31}
$$

The corresponding pdf plot of the random variable G in ILI

prediction is in Figure 3b.

Most Important Sensors under Mismatches

In this section, we provide a motivation for going beyond domain-agnostic deep learning models by illustrating a possible weakness of such a model under mismatched data. Specifically, we apply a post-hoc explanation tool GNNExplainer (Ying et al. 2019) to identify the most influential sensors contributing to a model's prediction at the target sensor. We choose the Spatio-Temporal GCN (STGCN) model which has a good performance in graph time series prediction, particularly in traffic speed prediction. The STGCN model is trained by four-hour data in a sequence of 12 consecutive weekdays, while the GNNExplainer is used to identify the 3 most influential sensors on the weekend data. We show the location of the 3 most influential sensors under matched data (train by weekday data and test by weekday data), and mismatched data (trained by weekday data and test by weekend data) in Figure 4.

Figure 4 shows that when the test distribution is mismatched with the training distribution, the most influential sensors identified by GNNExplainer are too far to drive within the prediction window, and the distances change significantly. In other words, speed measurements from vertices that are too far to influence the target vertex, and suggests a violation of domain traffic law. This forms the motivation for our approach.

Reaction-diffusion Equation

As seen in Eq. (12), the change in speed is a function of two terms. The diffusion term is 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 5a). The reaction term is 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 5b). The architecture of RDGCN is shown in Figure 6.

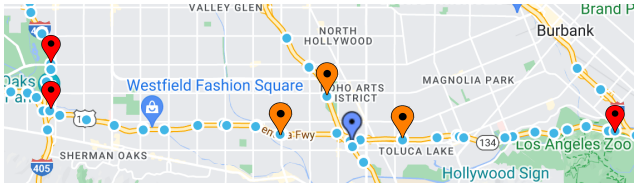
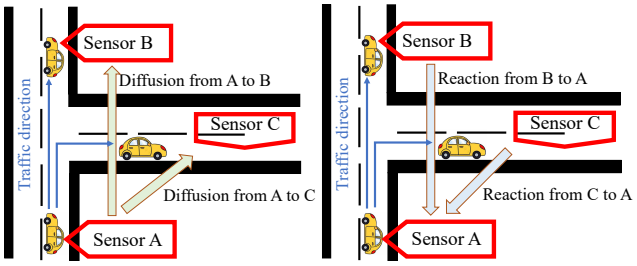


Figure 4: When an STGCN is tested on dataset from a matching distribution, the most important sensors (orange markers) are near the target sesnor, whereas the most important sensors under mismatched data (red markers) for the traffic speed prediction at target sensor (blue marker) are located far away. However, under matched data, the most important sensors are often close to the target sensor.



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

Figure 5: (a) Diffusion occurs in the direction of a road segment; (b) reaction occurs opposite to the direction of a road segment.

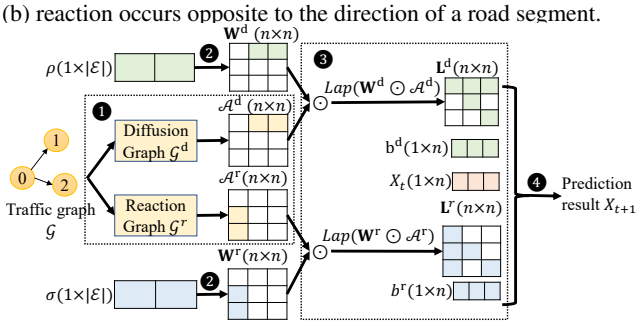


Figure 6: Reaction-diffusion GCN architecture for graph with $|\mathcal{V}| = 3$ and $|\mathcal{E}| = 2$. ❶ derives the diffusion and reaction adjacency matrices A^d and A^r; ❷ defines model weights ρ and σ for the reaction and diffusion networks, and map them to $\mathbf{W}^d$ and $\mathbf{W}^r$ with weights ρ and σ; ❸ characterizes the Graph Laplacian $\mathbf{L}^d$ and $\mathbf{L}^r$; ❹ defines the network prediction function Eq. (14).

Ablation Study

Analysis of RDGCN in Traffic Speed Prediction

Are reaction and diffusion processes essential? In this section, we investigate the prediction models incorporating the reaction equation and the diffusion equation, independently, under limited and 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 . 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 7.

Figure 7 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 only and the diffusion models only have weaker performance in at least one time period. We speculate that using only the reaction equation or the diffusion equation is not sufficient to capture the dynamics of the traffic speed change completely. 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 neigh-

boring 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 strengthen that both the reaction and diffusion processes are necessary for a reliable prediction.

What is the performance under RMSE? We plot the corresponding result under RMSE loss in Figure 8, and the conclusion is consistent with the result using MAE. The RMSE of RDGCN are with low variance regardless of the peroid of the training set. We acknowledge that RDGCN is not always better than baselines under RMSE, for example, when STGCN is trained with weekday data from 16:00-20:00 in Metr-la. One possible reason is that the mismatches between the training data and test data are not significant during the corresponding time period. The prediction results of RDGCN in terms of RMSE may not always be stable. For instance, when considering the models for the 4:00 to 8:00 time period in Metr-la, we observe distinct prediction outcomes. This variation could be due to the difference between the pattern of the morning rush hour during selected weekdays and the pattern during weekends. When the training set includes all available weekday data, the predictions of RDGCN demonstrate stability.

Experimental results. The Mean and STD of prediction MAE (resp. RMSE) of each model with MAML augmentation and with full weekday training set are shown in Table 3 (i.e., the Mean and STD of all points on each subfigure in Figure 2a, Figure 2b, Figure 8b and Figure 8d), respectively. Table 3 shows that RDGCN has lower MAE (resp. RMSE) and lower variance compared with baselines under limited training set with MAML augmentation, and the gain of adding more data on RDGCN is limited, which are consistent with our observation in Figure 2 in Section .

Impact of data volume. We further investigate the influence of training data volume on the performance of baseline models and RDGCN under a mismatched setting. We focus on assessing the adequacy of training data for both morning rush hour (8:00-12:00) and evening rush hour (16:00-20:00) scenarios using the Metr-la dataset. These periods exhibit considerable patterns and exhibit relatively minor mismatches between training and test datasets. To this end, we randomly select contiguous weekdays, ranging from 20% to the entire dataset, for training the models. The MAE of speed prediction across varying quantities of training data is shown in Figure 9.

Figure 9 showcases the performance characteristics of RDGCN and baseline models over the specified time intervals. Remarkably, the performance of RDGCN remains consistent irrespective of the training dataset size. Conversely, the predictive capabilities of STGNCDE and MTGODE are notably contingent upon the amount of training data employed. The observed trend underscores increased training data volume directly correlates with enhanced prediction accuracy. In the morning rush hour, MTGODE achieves optimal performance with approximately 75% of training data (equivalent to 60 weekdays), while STGNCDE demonstrates comparable performance when trained on the entire weekday dataset. We note that the superiority of RDGCN over baseline models is not universally consistent, as elucidated earlier. Notably, integrating domain differential equations drastically reduces the hypothesis class's size, thereby filtering out erroneous hypotheses often prevalent in conventional black-box graph learning models. Consequently, domain-differential-equation-informed GCNs exhibit remarkable robustness on relatively smaller training datasets.

Analysis of SIRGCN in ILI Prediction

Do the infection rates vary among different vertices? In this section, we delve into the question of whether we require an individual infection rate for each vertex in ILI prediction. We specifically examine two approaches: one where we assign a unique infection rate, denoted as β_i, to each vertex i, resulting in a SIRGCN with n infection rates (SIRGCN-n), and another approach where we assign a single infection rate, denoted as β, to all vertices (SIRGCN-1). We report the MAE and RMSE of the prediction under mismatched data (train using Winter-Summer data and test using Spring-Fall data) in Table 4.

Table 4 shows that employing multiple infection rates leads to more accurate predictions, particularly in the case of the US-state dataset. By assigning individual infection rates to each vertex, we achieve a reduction of 2.4% in MAE (and 1.6% in RMSE). However, the advantage of utilizing multiple infection rates is less pronounced ($< 1\%$) in the ILI prediction of Japan. There could be two potential explanations for this phenomenon. First, the size of Japan's prefectures is not as substantial as that of the states in the United States. Second, the climate across Japan is relatively homogeneous, whereas the climate across different states of the United States exhibits significant variations, such as wet coastal areas and dry inland areas.

Predictions in Different Seasons. Learning patterns across different trends becomes challenging when baseline models are not trained using the same trend. For example, during Winter, the infectious number shows an increasing trend, whereas during Spring, it exhibits a decreasing trend. Figure 10 shows the predicted number of infectious cases alongside the ground truth data, revealing that SIRGCN's prediction aligns better with the ground truth. Conversely, EpiGNN's prediction performs poorly during the decline phase and when the number of infections approaches 0.

In the case of US-State ILI prediction in May 2014, both COLAGNN and EPIGNN fail to make accurate predictions around the peak, while SIRGCN demonstrates its effectiveness during the corresponding period, with the help of SIR-network model.

Model Efficiency in Computation Time

The training time and inference time (on two NVIDIA-2080ti graphic cards) of STGCN, MTGNN, GTS, and RDGCN on the Metr-la dataset are demonstrated in Table 5. It's observed that RDGCN takes less time in both training and inference than the other models, since the RDGCN contains significantly less number of parameters than the baseline models.

The training and inference time of ColaGNN, EpiGNN, and SIRGCN are shown in Table 5. SIRGCN has significantly less number of parameters than the baseline models.

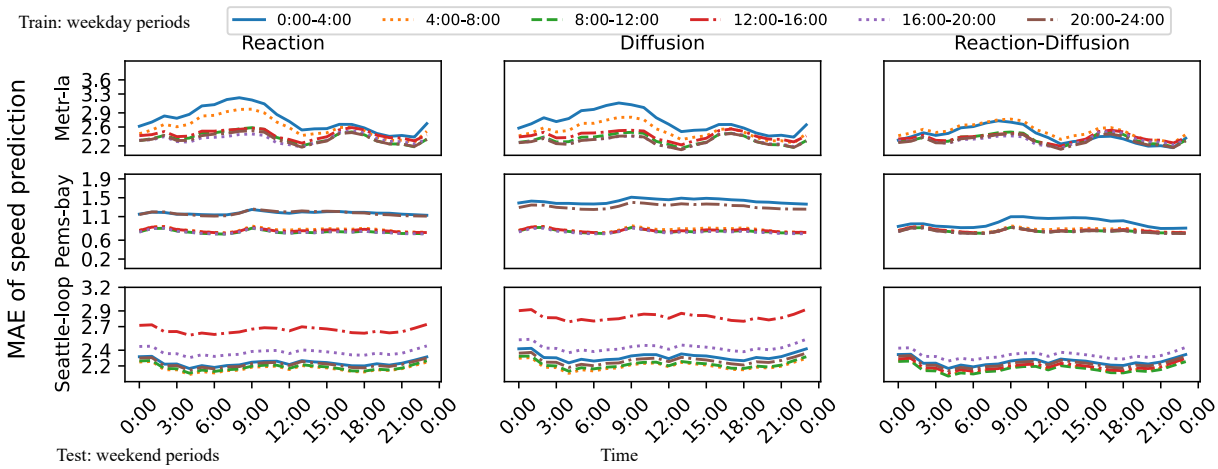


Figure 7: MAE of speed predictions on models incorporating reaction equation, diffusion equation, and reaction-diffusion equation.

Table 3: Numerical result of Figure 2: the Mean and STD of prediction MAE and RMSE of RDGCN and baselines on three real-world datasets.

	MAE						RMSE					
With MAML	STGCN	MTGNN	GTS	STGNCDE	MTGODE	RDGCN	STGCN	MTGNN	GTS	STGNCDE	MTGODE	RDGCN
Metr-la	2.47 ± 0.11	2.41 ± 0.22	2.55 ± 0.48	3.27 ± 0.47	2.82 ± 0.49	2.39 ± 0.08	5.28 ± 0.94	5.17 ± 1.16	7.55 ± 0.91	7.01 ± 1.28	5.41 ± 2.01	4.96 ± 0.83
Pems-bay	1.03 ± 0.19	0.91 ± 0.21	0.96 ± 0.03	0.77 ± 0.06	0.86 ± 0.14	0.83 ± 0.03	1.41 ± 0.05	2.86 ± 1.11	2.85 ± 0.84	1.44 ± 0.16	1.58 ± 0.44	1.40 ± 0.05
Seattle-loop	2.20 ± 0.08	2.23 ± 0.24	2.34 ± 0.15	3.20 ± 0.07	3.17 ± 0.05	2.16 ± 0.05	5.94 ± 0.14	3.92 ± 0.37	5.80 ± 0.60	6.16 ± 0.17	6.04 ± 0.19	3.44 ± 0.18
FULL	STGCN	MTGNN	GTS	STGNCDE	MTGODE	RDGCN	STGCN	MTGNN	GTS	STGNCDE	MTGODE	RDGCN
Metr-la	2.57 ± 0.68	3.11 ± 0.48	3.44 ± 0.47	2.77 ± 0.35	2.31 ± 0.43	2.38 ± 0.13	5.31 ± 0.92	4.02 ± 0.31	7.04 ± 1.20	6.43 ± 1.24	4.70 ± 1.38	3.90 ± 0.10
Pems-bay	1.38 ± 0.06	1.85 ± 0.38	2.08 ± 0.51	0.83 ± 0.09	0.79 ± 0.02	0.74 ± 0.02	1.37 ± 0.06	1.85 ± 0.38	2.08 ± 0.53	1.38 ± 0.04	1.36 ± 0.04	1.38 ± 0.04
Seattle-loop	2.90 ± 0.10	2.81 ± 0.65	3.11 ± 0.11	3.32 ± 0.07	3.21 ± 0.05	2.18 ± 0.06	3.91 ± 0.45	3.81 ± 0.65	5.33 ± 0.74	6.25 ± 0.17	6.22 ± 0.17	3.58 ± 0.05

Table 4: Evaluation models under mismatched data.

	MAE		RMSE	
	SIRGCN-1	SIRGCN-n	SIRGCN-1	SIRGCN-n
Japan-Prefectures	344 ± 22	342 ± 22	871 ± 43	863 ± 44
US-States	42 ± 4	41 ± 4	123 ± 10	121 ± 10

Table 5: The computation time on the Metr-la dataset.

		# Parameters	Training (s/epoch)	Inference (s)
Metr-la	STGCN	458865	0.5649	0.0232
	MTGNN	405452	0.5621	0.0607
	GTS	38377299	1.0632	0.1641
	STGNCDE	374904	1.7114	0.3729
	MTGODE	138636	1.6158	0.3491
	RDGCN	872	0.0308	0.0037
Japan-prefectures	ColaGNN	4272	0.0297	0.0065
	EpiGNN	16875	0.0311	0.0073
	SIRGCN	181	0.0289	0.0063

We acknowledge that the computational time of SIRGCN is similar to that of the baseline models, as the baselines are not as deep or dense as traffic prediction models and do not require a large amount of data for training.

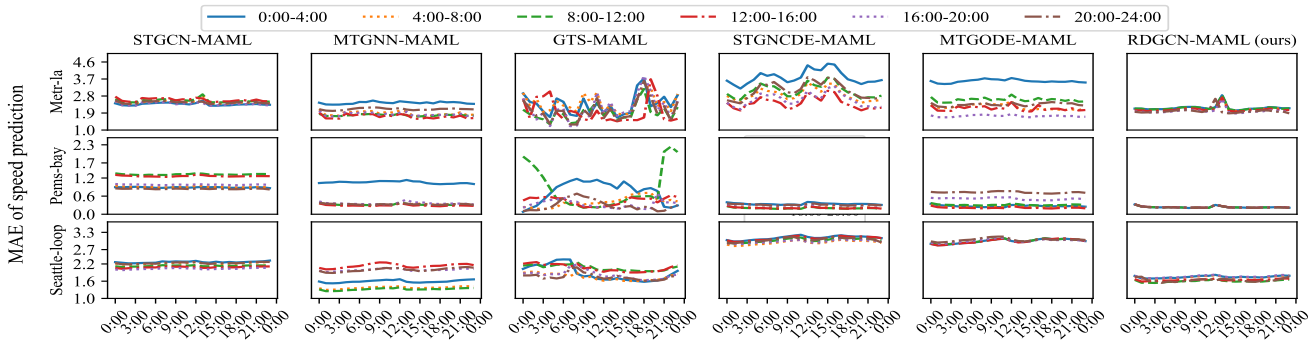
Table 6: The computation time on the Japan-Prefectures dataset.

Experimental Settings

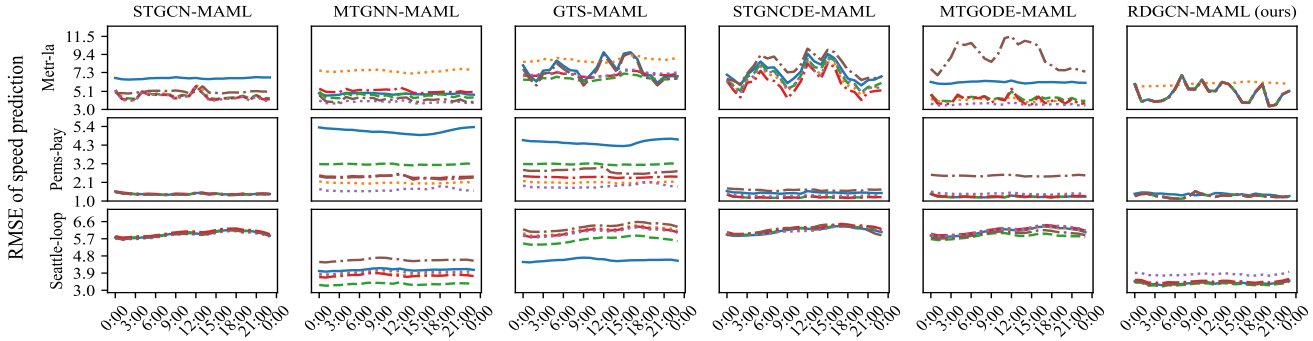
Evaluation. We assume that all zeros in the datasets are missing values, and we remove the predicted speed when the ground truth is 0, or when the last speed recorded is 0.

Hyperparameter Settings. RDGCN and SIRGCN are optimized via Adam. The batch size is set as 64. The learning rate is set as 0.001, and the early stopping strategy is used with a patience of 30 epochs. In traffic speed prediction, 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 use identical hyperparameters as released in their works. In ILI prediction, the training and validation set are split by a ratio of 5:2 from the Winter-Summer subset, and the test data is sampled from the Spring-Fall subset with different patterns. The Susceptible population at the beginning of each ILI period is 10% of the total population in each Prefectures or States. As for baselines, we also use identical hyperparameters as released in their works. We approximate the total number of populations by the average of the annual sum of infectious cases, multiplied by 10.

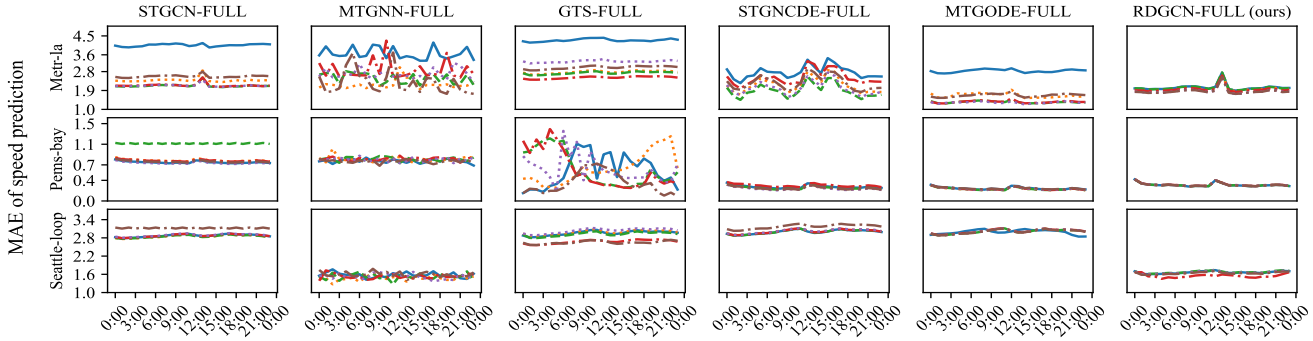
MAML Settings. Our experiment involves the following steps: (1)We randomly select sequences of 12 consecutive weekdays (same as the Limited and Mismatched Data experiment.), and sample four-hour data as the training set. We evaluate the model with hourly data on weekends. (2) We divide the training set into two equal parts: the support set and



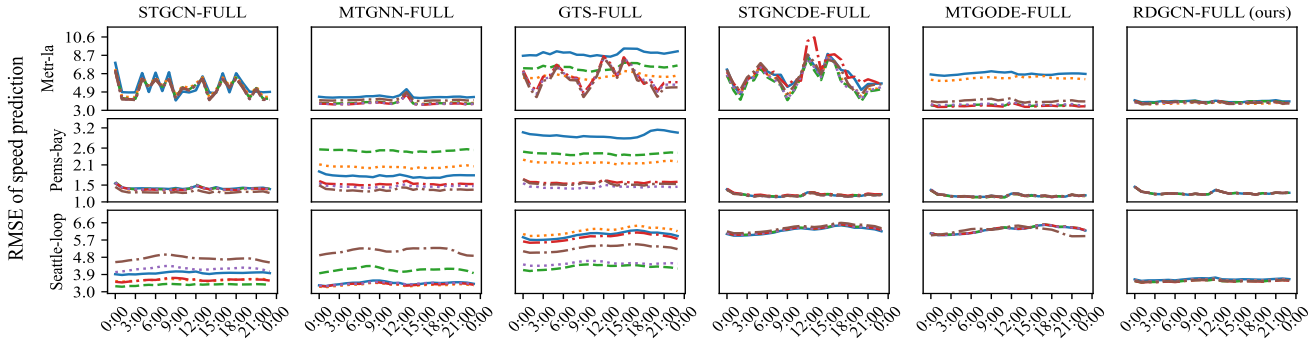
(a) Baseline models and RDGCN trained on 12 consecutive weekdays and augmented by MAML.



(b) Baseline models and RDGCN trained on 12 consecutive weekdays and augmented by MAML.



(c) Baseline models and RDGCN trained on more than half a year of weekdays.



(d) Baseline models and RDGCN trained on more than half a year of weekdays.

Figure 8: (a)(b) The results of RDGCN are very close regardless of the period of the training set. (c)(d) Even though all the models are trained using all available weekdays, the results of RDGCN are still closer, regardless of the period, compared to baseline models.

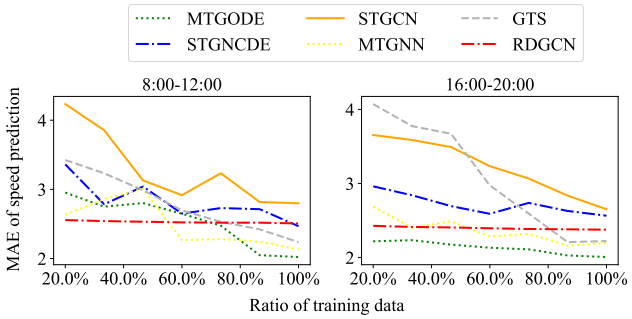


Figure 9: Feeding more training data does not lead to a significant change in the MAE of RDGCN's prediction.

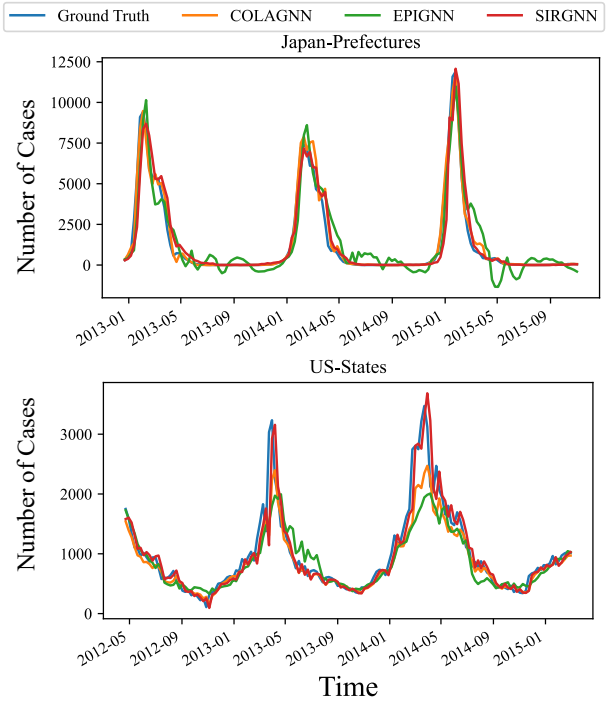


Figure 10: SIRGCN can make accurate predictions in the decreasing phase, while EpiGNN makes bad predictions in the corresponding phase.

the query set. (3) We use the support set to compute adapted parameters. (4) We use the adapted parameters to update the MAML parameters on the query set. (5) We repeat this process 200 times to obtain initial parameters for the baseline model. (6) We train baselines using the obtained initial parameters. The learning rate for the inner loop is 0.00005, and for the outer loop is 0.0005, and MAML is trained for 200 epochs.