REST: Reciprocal Framework for Spatiotemporal-coupled Predictions

Haozhe Lin
Department of Automation, Tsinghua University
Beijing, China
linhz16@mails.tsinghua.edu.cn

Yushun Fan
Department of Automation, Tsinghua University, China
Beijing, China
fanyus@tsinghua.edu.cn

Jia Zhang
Department of Computer Science, Southern Methodist University
Dallas, Texas, USA
jiazhang@smu.edu

Bing Bai
Cloud and Smart Industries Group, Tencent
Beijing, China
icebai@tencent.com

ABSTRACT

In recent years, Graph Convolutional Networks (GCNs) have been applied to benefit spatiotemporal predictions. The current shell for spatiotemporal predictions often relies heavily on the quality of handcraft, fixed graphical structures, however, we argue that such a paradigm could be expensive and sub-optimal in many applications. To raise the bar, this paper proposes to jointly mine the spatial dependencies and model temporal patterns in a coupled framework, i.e., to make spatiotemporal-coupled predictions. We come up with a novel Reciprocal SpatioTemporal (REST) framework, which introduces Edge Inference Networks (EINs) to couple with GCNs. From the temporal side to the spatial side, EINs infer spatial dependencies among time series vertices and generate multi-modal directed weighted graphs to serve GCNs. And from the temporal side to the spatial side, GCNs utilize these spatial dependencies to make predictions and then introduce feedback to optimize EINs. The REST framework is incrementally trained for higher performance of spatiotemporal prediction, powered by the reciprocity between its comprised two components from such an iterative joint learning process. Additionally, to maximize the power of the REST framework, we design a phased heuristic approach, which effectively stabilizes training procedure and prevents early-stop. Extensive experiments on two real-world datasets have demonstrated that the proposed REST framework significantly outperforms baselines, and can learn meaningful spatial dependencies beyond predefined graphical structures.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; • **Computing methodologies → Neural networks**; *Learning from implicit feedback.*

KEYWORDS

Spatiotemporal prediction, graph convolutional networks, Web traffic prediction, spatial dependencies inference

ACM Reference Format:
Haozhe Lin, Yushun Fan, Jia Zhang, and Bing Bai. 2021. REST: Reciprocal Framework for Spatiotemporal-coupled Predictions. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3442381.3449928

1 INTRODUCTION

Time series prediction has remained an enduring research topic for both academia and industry for decades. By understanding past observations, such technologies predict the future trend in a certain horizon, which has been benefiting many real-world applications, including weather forecasting [21], Web service invocation predictions [17], environmental analysis [10], and so on. Early research on this topic mainly concentrates on studying individual time series; while, in recent years, researchers have started to investigate how to utilize the interactions among multiple time series to promote prediction accuracy [16, 30]. As an example, the traffic flow of a certain road is determined by the inflow of its upstream intersections, whose traffic flows are in turn determined by other intersections. Therefore, taking into consideration of the temporal information of the roads spatially close by may significantly enhance the performance of transportation predictions. In such a context, Graph Convolutional Networks (GCNs) [6, 9, 14, 29] have been successfully applied to make spatiotemporal predictions [16, 28, 34], where different time series and their interrelated relationships are seen as vertices and edges, respectively.

The precondition of GCN-based spatiotemporal prediction models is a well defined graph. In many real-world applications, however, it is not always easy to obtain high-quality spatial information. Let us take Web traffic prediction[1] as an example to illustrate how to identify spatial dependencies, namely edges, among entries. One may define that an edge refers to a hyperlink from one Wikipedia entry to another. Nevertheless, it may be difficult to further quantify the weights of such edges (e.g., the transition probability contained in a hyperlink), especially when considering the contents (including hyperlinks) of Wikipedia entries are frequently edited and changed.

[1] https://www.kaggle.com/c/web-traffic-time-series-forecasting

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449928

Apart from explicit hyperlinks, implicit authorship or similarities in the content could also be exploited to construct an isomeric graph, thus yielding multi-modal spatial dependencies. In such situations, profound domain knowledge is required to construct an accurate graph; however, human involvement may become not only costly but also sub-optimal. To this end, we argue that jointly mining the spatial dependencies and modeling temporal patterns in a coupled framework may largely benefit spatiotemporal predictions, which lead to our targeted goal, i.e., spatiotemporal-coupled prediction.

Three challenges exist regarding spatiotemporal-coupled predictions. Firstly, **from the data property aspect**, there lacks existing edge labels to learn the spatial dependencies in a supervised manner. Moreover, the information of time series (i.e., historical observations with different timestamps) may be limited and noisy, making it difficult to find the distance (i.e., correlation or causation) among time series and cluster them as a graph [1]. Secondly, **from the learning aspect**, without effective inductive bias, a model is easy to overfit the noises and the learning procedure may become unstable. Since the spatial and temporal dependencies among vertices couple compactly, the changes of spatial dependencies may make learning temporal patterns more strenuous, and vice versa, especially when both sides are initialized from random states. Thirdly, **from the practicality aspect**, mining potential links between two arbitrary time series pairs also brings significant computational burden, as the possible links are in n^2 order, if we assume n time series exist. Existing research concerning spatiotemporal prediction is either based on predefined graph structure [16, 30] or can only infer potential links with strong domain knowledge [11] or in rather small graphs, e.g., with less than 400 vertices [28].

In this paper, we propose a novel Reciprocal Spatiotemporal (REST) framework to address the aforementioned three challenges synergistically. As illustrated in Figure 1 (details to be explained in Section 4), the REST framework consists of two integral parts: our introduced Edge Inference Networks (EINs) for mining spatial dependencies among time series; and integrated GCNs, e.g., DCRNNs [16], for making spatiotemporal prediction. The spatial dependencies inferred by EINs promote GCNs to make more accurate prediction, while supervisedly trained GCNs help EINs learn better distance measurement. To address the data property challenge, EINs project time series from time domain to frequency domain, and thus fertilize the original time series data and quantify the multi-modal spatial dependencies among them. To address the practicality challenge, EINs firstly sample a fixed number of possible time series neighbors for all the central time series vertices of interest before each training epoch, and then during the training procedure, EINs try to learn a more accurate distance function with the help of the GCN part. Through such manner, the REST framework can theoretically explore all possible linkages from the whole dataset, while remain the sparsity of graph as $\frac{kn}{n^2}$ for training, where k refers to predefined number of neighbor candidates and $k \ll n$. To address the learning challenge, we propose a phased heuristic approach as a warm-up to drive the REST framework. As a consequence, such an iterative learning cycles within the REST framework shall incrementally enhance the spatiotemporal-coupled prediction over time.

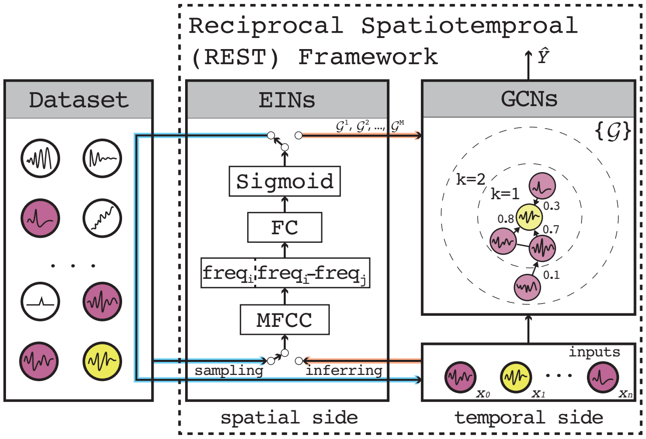


Figure 1: The overview of the proposed REST framework. Inside REST, EINs and GCNs cooperate with each other through the orange lines. From the temporal side to the spatial side, EINs first infer pairwise vertices distance through the frequency features of time series and construct multi-modal graphs $\mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^M$ for GCNs. From the spatial side to the temporal side, GCNs receive deduced spatial dependencies and regress prediction results. From inputs to outputs, EINs provide spatial information for GCNs; while from outputs to inputs, GCNs propagate feedback to EINs for spatial distance learning, and thus reciprocity is developed. Outside REST framework, through the turquoise lines, EINs explore whole time series datasets and sample potential neighbors, i.e., purple circles, for central vertex, i.e., yellow circle, before each training epoch.

Our main contributions are three-fold:

- We have proposed a REST framework for spatiotemporal predictions, when structural information is incomplete or under frequent changes. In particular, our introduced EINs component is able to infer multi-modal, directed and weighted spatial dependencies.
- We have developed a phased heuristic strategy and spatiotemporal coupled learning paradigm, which help to stabilize the training procedure of REST framework, while making it possible to explore unsuspected linkages in the full domain without introducing unaffordable computational training burden.
- We have designed and conducted a collection of experiments on two real-world datasets. Our empirical studies show that REST outperforms state-of-the-art spatiotemporal prediction algorithms in terms of prediction accuracy, and demonstrate its capability of finding meaningful linkages between time series vertices.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 gives notations and mathematically restates the spatiotemporal-coupled prediction problem. Section 4 introduces the proposed model, and Section 5 describes the training details. Section 6 reports our experimental results. Finally, Section 7 draws conclusions.

2 RELATED WORK

In this section, we review important related work in the literature in two categories, time series predictions and graph convolutional networks, and compare our work with them.

2.1 Time Series Predictions

Time series prediction is a long-standing problem. At the beginning, researchers concentrated on studying the pattern of an individual time series and making predictions. For instance, Auto-Regressive Integrated Moving Average model (ARIMA) [4] and Support Vector Regression (SVR) [23] study the linearity and non-linearity of time series. There are also some attempts to investigate the interactions among time series, where the representatives are Vector Auto-Regressive model (VAR) [18] and multiple-output SVR [26]. In recent years, with the advancement of deep learning, the accuracy of time series prediction has achieved noticeable progress. On the one hand, in Recurrent Neural Networks (RNNs), Long-Short Term Memories (LSTMs) [13] and Gated Recurrent Units (GRUs) [7] show significant advantages in capturing long-term dependencies of time series. Specially, RNNs based on LSTMs with fully connected layers (FC-LSTM) are one of the most representative models for sequence prediction [25]. On the other hand, in Convolutional Neural Networks (CNNs), dilated causal convolution [31] together with gated mechanism [8] is also able to accumulate sequential information within a large receptive field, with WaveNet as a successful application [27].

The above earlier approaches do not (or poorly) consider the spatial structure of time series. However, the interactions among time series could provide helpful information for models to enhance prediction accuracy [15]. To this end, researchers have introduced Graph Convolutional Networks (GCNs) to aggregate irregular spatial dependencies among time series. Two recent representative works are Diffusion Convolutional Recurrent Neural Networks (DCRNNs) [16] and Graph WaveNet [28]. DCRNNs substitute multiplication in GRUs with diffusion convolution operators to utilize the spatial dependencies. Graph WaveNet [28] replace 1D convolution in the residual block of WaveNet with diffusion convolution, enabling it to make spatiotemporal predictions.

Although these models have shown attractive abilities to exploit spatial dependencies among time series, their performance mainly relies on the quality of the graph underneath. To date, few existing work studies to voluntarily learn spatial dependencies for temporal prediction. We argue that such a manner is important to further increase the degree of freedom of spatiotemporal models, without depending on many handcraft features, and enhance prediction accuracy. Related to this idea, an adaptive adjacent matrix proposed in [28] provides a partial solution. Such a method, however, is difficult to be applied to graphs with large sizes, because it requires to compare all pairwise similarities during training procedure, which makes the graph dense and introduces enormous computational burden. By comparison, in our REST framework, input time series (i.e., central vertices and several adjacent vertices) are sampled before each training epoch, and thus the sparsity is restricted by $\frac{kn}{n^2}$, remaining computational complexity acceptable.

2.2 Graph Convolutional Networks

Most spatiotemporal prediction models are based on spectral graph convolution, which has shown big potential in aggregating non-euclidean spatial dependencies among time series vertices. Spectral-based Graph Convolutional Networks (GCNs) were first introduced by Bruna et al. [6], which incorporate spectral graph theory into deep learning models. To efficiently apply spectral graph theory, Defferrard et al. [9] proposed ChebNet, in which spectral graph convolution is approximated through Chebyshev polynomial and gain localized features. Kipf and Welling [14] further enhanced the efficiency of GCNs by restricting graph convolution within one-step adjacency and stacking linear GCNs to expand receptive fields. To model undirected graphs, ChebNet and linear GCNs [14] have shown potentials. For directed graphs, Atwood and Towsley [2] proposed diffusion convolution. These graph convolution filters have been successfully applied in spatiotemporal models, e.g., DCRNNs and Graph WaveNet, to improve prediction accuracy.

There is an important branch of GCNs studying link prediction in graphs, which seems similar to our spatial distance inference task. For example, Weisfeiler-Lehman Neural Machine (WLNM) [32], SEAL [33], and Graph Agreement Models (GAM) [24] have been used to predict links in citation networks. However, these models are based on vertex features and categorical labels to learn the similarities among vertices. In our spatiotemporal prediction task, the features of time series vertex are limited and noisy, meanwhile there is no obvious labels to represent the similarities between vertices. As a result, existing models cannot solve our problem.

3 PRELIMINARIES

To begin with, we define some important notations, and then mathematically restate the spatiotemporal-coupled prediction problem.

Definition 1 (Observation records). *Given one time series, we denote* ***observation records*** *by $\boldsymbol{x}_i$, where $\boldsymbol{x}_i$ can be broken down to $\boldsymbol{x}_i = \{x_i^0, x_i^1, \ldots, x_i^p\}$ for time series i in the past p time steps. In the problem that we consider, we further write $\boldsymbol{X} = \{\boldsymbol{x}_1; \boldsymbol{x}_2; \ldots; \boldsymbol{x}_n\}$ to denote the whole N time series.*

Definition 2 (Prediction trend). *The* ***prediction trend*** *of one time series is denoted by $\hat{\boldsymbol{y}}_i = \{\hat{y}_i^{p+1}, \hat{y}_i^{p+2}, \ldots, \hat{y}_i^{p+q}\}$, where $\hat{y}_i^t (t \in (p, p+q]))$ refers to the prediction value of time series i in the next t-th time step. Likewise, $\hat{Y} = \{\hat{\boldsymbol{y}}_1; \hat{\boldsymbol{y}}_2; \ldots; \hat{\boldsymbol{y}}_n\}$ denotes the trend of all time series.*

Definition 3 (Spatial dependencies). *We construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ to represent* ***spatial dependencies*** *among time series, with $\mathcal{V}$ as vertex set, $\mathcal{E}$ as edge set, and $\mathcal{W}$ as corresponding weights for each edge. We treat each time series as a vertex, so $|\mathcal{V}| = N$. In particular, within a mini-batch, we call the vertices of interest as the* ***central vertices****, and the other vertices that can reach the central vertices within K steps as the* ***adjacent vertices****. In our problem, we consider multi-modal, weighted and directed spatial dependencies, i.e., $\mathcal{W} = \{\boldsymbol{w}^m, m = 0, 1, \ldots, M\}$, which can start from a handcraft graph, i.e., spatial dependencies, $\mathcal{G}_0$, or be inferred by our REST framework. In particular, weight $w_{ij}^m \in \mathcal{W}$ refers to spatial dependency from time series i to j under modality m.*

Based on these definitions, our targeted problem is formally introduced as follows.

Problem (Spatiotemporal coupled prediction). *Given N time series, our goal is to jointly learn $f(\cdot)$ and $g(\cdot)$ to predict the trend of time series $\boldsymbol{Y}$, as well as to infer their underline spatial dependencies $\mathcal{G}$, based on observation records $\boldsymbol{x}_i = \{x_i^0, x_i^1, \ldots, x_i^p\}$ for all i and their predefined or randomly initialized spatial dependencies $\mathcal{G}$, which can be described in Equation* (1).

$$[\boldsymbol{X}^1, \boldsymbol{X}^2, \ldots, \boldsymbol{X}^p, \mathcal{G}^0] \xrightarrow[g(\mathcal{G})]{f(X)} [\hat{\boldsymbol{Y}}^{p+1}, \ldots, \hat{\boldsymbol{Y}}^{p+q}, \mathcal{G}^1, \ldots, \mathcal{G}^M] \quad (1)$$

As for other frequently used notations, the readers could refer to Table 4 in Appendix A for details.

4 MODEL ARCHITECTURE

The overall structure and mechanism of action of the REST framework are depicted in Figure 1. We will sequentially discuss its two comprised building blocks (EINs and GCNs), and then show how they support each other.

4.1 Spatial Inference

In REST framework, we introduce Edge Inference Networks (EINs) to discover and quantify spatial dependencies among time series vertices, whose internal procedure is illustrated in Figure 1. As mentioned in Section 1, the information of time series observations may be limited and noisy, which makes it difficult to precisely measure the dependencies (e.g., distance) between two time series. Therefore, inside EINs, our idea is to project the observations from time domain to frequency domain. Here we adopt the Mel-Frequency Cepstrum Coefficients (MFCCs) [5, 19], as effective features. Presenting the envelope of the frequency spectrum of sound signals, MFCCs are widely used in audio compressing and speech recognition. Here we use this frequency warping to represent time series. We calculate MFCCs through Equation (2):

$$\begin{aligned} X[k] &= \text{fft}(x[n]) \\ Y[c] &= \log\left(\sum_{k=f_{c-1}}^{f_{c+1}} \mid X[k] \mid^2 B_c[k]\right) \\ c_x[n] &= \frac{1}{C}\sum_{c=1}^{C} Y[c] \cos\left(\frac{\pi n(c-\frac{1}{2})}{C}\right), \end{aligned} \quad (2)$$

where $x[n]$ refers to the time series observations; fft$(\cdot)$ refers to fast Fourier transform; $B_c[k]$ refers to filter banks; C refers to the number of MFCCs to retain; and $c_x[n]$, also denoted by $\boldsymbol{c}_x$, refers to MFCCs of time series x. As for the choice of MFCCs, we consider the observations of time series in the frequency domain are usually more meaningful in practice. Moreover, MFCCs, presenting the envelope of the frequency feature from Fourier transform, could generally be smoother features for neural networks. The readers could refer to [20] for the details of MFCCs.

Taking MFCCs as effective features, EINs then estimate the spatial dependencies between two time series through Equation (3) using a sigmoid function:

$$\boldsymbol{a}_{ij} = \sigma\left(\boldsymbol{W}^\top \text{concat}([\boldsymbol{c}_i, \boldsymbol{c}_i - \boldsymbol{c}_j]) + \boldsymbol{b}\right), \quad (3)$$

where $\boldsymbol{a}_{ij} \in \mathbb{R}^M$ refers to inferred asymmetric distance from time series i to j under M considered modalities; $\boldsymbol{c}_i \in \mathbb{R}^C$, namely $c_x[n]$, refers to MFCCs of time series i; $\boldsymbol{W} \in \mathbb{R}^{2C\times M}$ and $\boldsymbol{b} \in \mathbb{R}^M$ refer to learnable parameters for time series distance inference. Note that, in Equation (3), $\boldsymbol{c}_i$ is concatenated with $\boldsymbol{c}_i - \boldsymbol{c}_j$, which models the directed spatial dependencies, while to consider undirected (or symmetric) relationship between time series i and j, $\boldsymbol{c}_i$ should be concatenated with $\boldsymbol{c}_j$. We thus built a fully connected layer to measure the time series distance. We will conduct more in-depth study about time series metrics learning in our future work.

Based on their capability of inferring spatial dependencies among time series, EINs play two important roles in REST framework: sampling and inferring. Figure 1 illustrates the two functions initiated by turquoise line and orange line, respectively. To begin with, during data preparation phase, EINs go through the entire dataset to select possible adjacent candidates, i.e., purple vertices in Figure 1, for the central vertices of interest, i.e., yellow vertex. Afterwards, given these sampled vertices as input, EINs infer and quantify their spatial dependencies under M modalities for GCNs. In this section, we have shown how EINs generate spatial dependencies for GCNs, and we will discuss how it learns from GCNs for optimization in Section 4.3.

4.2 Temporal Prediction

Receiving inferred spatial dependencies from EINs (triggered by the orange line), REST framework can integrate a GCN-based spatiotemporal prediction model, e.g., DCRNN [16] or Graph WaveNet [28], as backends to make predictions. Generally speaking, by defining different Laplacian matrix, e.g., normalized Laplacian and random walk Laplacian, the (un)directed spatial dependencies can be aggregated through different graph convolution operators, e.g., Chebyshev convolution [9] and diffusion convolution [2]. In this paper, we will mainly discuss the directed spatial dependencies and consider diffusion convolution on random walk Laplacian. When considering only one modality, the random walk Laplacian is defined as $L^{\text{rw}} = I - D^{-1}A$, with L^{rw} relating to transition matrix, based on which the bidirectional diffusion convolution can be formulated in Equation (4):

$$\boldsymbol{Z} \star_{\mathcal{G}} g_\theta \approx \sum_{k=0}^{K-1}\left(\theta_{k,0}\left(D_I^{-1}A\right)^k + \theta_{k,1}\left(D_O^{-1}A^\top\right)^k\right)\boldsymbol{Z}, \quad (4)$$

where $\boldsymbol{Z}$ refers to inputs of graph convolution filter; g_θ refers to diffusion convolution filter with $\theta \in \mathbb{R}^{K\times 2}$ as trainable parameters; $\boldsymbol{A}$ refers to adjacent matrix and $\boldsymbol{D}_I$ and $\boldsymbol{D}_O$ refer to input and output degree matrix, respectively. A diffusion convolution can be truncated by a predefined graph convolution depth K, which is empirically not more than 3 [12]. Note that due to the sparsity of most graph, the complexity of recursively calculating Equation (4) is $O(K|\mathcal{E}|) \ll O(N^2)$.

In this paper, we consider that the spatial dependencies, namely edge features, deduced by EINs are multi-modal. Therefore, an enhanced diffusion convolution is formulated by Equation (5):

$$\boldsymbol{h}_s = \text{ReLU}\left(\sum_{m=0}^{M-1}\sum_{k=0}^{K-1} \boldsymbol{Z} \star_{\mathcal{G}^m} g_\Theta\right), \quad (5)$$

where $\boldsymbol{h}_s \in \mathbb{R}^{N \times d_o}$ refers to spatial hidden states, namely output of diffusion convolution operators; M refers to predefined number of modality, and specifically, forward adjacent matrix $\boldsymbol{A}$ and backward one $\boldsymbol{A}^\top$ in bidirectional random walk are deemed as one modality; then $\Theta \in \mathbb{R}^{M \times K \times d_i \times d_o}$, extension of θ, refer to multi-modal, high-order graph convolution parameters.

Given enhanced diffusion convolution filters to aggregate spatial information, existing approaches typically adopt either RNNs (e.g., DCRNNs [16]) or CNNs (e.g., Graph WaveNet [28]) to capture temporal dependencies of time series. Without losing generality, we adopt DCRNNs as a prototype backend to show how REST framework incorporates it to generate predictions. To capture temporal dependencies, DCGRUs replace multiplication in GRUs with diffusion convolution, as shown in Equation (6):

$$\begin{aligned}
\boldsymbol{r}^t &= \sigma(f_r \star_{\mathcal{G}^m} [\boldsymbol{X}^t, \boldsymbol{H}^{t-1}] + \boldsymbol{b}_r) \\
\boldsymbol{u}^t &= \sigma(f_u \star_{\mathcal{G}^m} [\boldsymbol{X}^t, \boldsymbol{H}^{t-1}] + \boldsymbol{b}_u) \\
\boldsymbol{C}^t &= \tanh(f_C \star_{\mathcal{G}^m} [\boldsymbol{X}^t, (\boldsymbol{r}^t \odot \boldsymbol{H}^{t-1})] + \boldsymbol{b}_C) \\
\boldsymbol{H}^t &= \boldsymbol{u}^t \odot \boldsymbol{H}^{t-1} + (1 - \boldsymbol{u}^t) \odot \boldsymbol{C}^t,
\end{aligned} \tag{6}$$

where $\boldsymbol{X}^t$ refer to the observations of all included vertices, i.e., all colored vertices in Figure 1; $\boldsymbol{H}^{t-1}$ refer to temporal hidden states generate by last DCGRUs; $\star_{\mathcal{G}^m}$ refers to diffusion convolution operator given graph $\mathcal{G}^m$, where, particularly, $\mathcal{G}^m$ is inferred by EINs; $\boldsymbol{r}^t$ and $\boldsymbol{u}^t$ refer to the output of reset and update gates at time t; f_r, f_u, and f_C refer to graph convolutional filters with different trainable parameters; finally, $\boldsymbol{H}^t$ refer to temporal hidden states. In DCRNNs, DCGRUs are stacked to construct encoders and decoders, and based on the temporal hidden states $\boldsymbol{H}^t$ from decoders, a fully connected layer then generate predictions:

$$\hat{Y}^{t+1} = \boldsymbol{W}^\top \boldsymbol{H}^t + \boldsymbol{b}. \tag{7}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are trainable parameters.

4.3 Reciprocity

REST framework enables reciprocity between its comprising EINs and GCNs, as shown in Figure 1. On the one hand, EINs could help generate better spatial structures than prior knowledge. Thus in forward propagation, EINs will promote GCNs to make more accurate predictions. On the other hand, limited by the data property, it is difficult for EINs to learn the spatial dependencies in a supervised manner. However, in the REST framework, EINs can get optimized through temporal labels from GCNs in backward propagation. In other words, GCNs help EINs to learn better distance measurement. To this end, the reciprocity of REST framework is established through such an iterative learning process. Note that warm-started EINs can further be used to explore the full dataset and sample possible unknown linkages before each training epoch. Assuming there are n time series in total, since the EINs only sample at most k adjacent vertices for the central vertices, the sparsity is restricted within $\frac{kn}{n^2}, k \ll n$.

However, if EINs and GCNs do not perform normally, REST framework may not work as expected. It might pose a problem that both sides could hamper each other, which we may encounter especially in an initialization phase. We will discuss how to solve the parameter initialization issue in Section 5.2.

5 LEARNING DETAILS

In this section, we discuss parameter tuning and optimization for REST framework.

5.1 Loss Function

We choose mean absolute error (MAE) as loss function to supervise the training process under REST framework, which is formulated by Equation (8):

$$\mathcal{L} = \frac{1}{n} \sum_{i,t} \left| y_i^t - \hat{y}_i^t \right|, \tag{8}$$

where n refers to the number of observations of all time series in a batch; $y_{i,t}$ and $\hat{y}_{i,t}$ refer to the ground truth and predictions of time series i at time t. Note that for applications where the orders of magnitude of time series significantly differ from each other, we evaluate loss under logarithmic scale.

5.2 Phased Heuristics

Well trained EINs and GCNs can benefit each other. However, at the beginning stage, since both sides are initialized with random states, EINs and GCNs may hamper each other, making it more difficult to train. To bolster training process, we have developed a phased heuristics strategy to enable a warm start. As the training process going, information closed loop is formed, meaning that REST only relies on information within the framework rather than depending on prior knowledge.

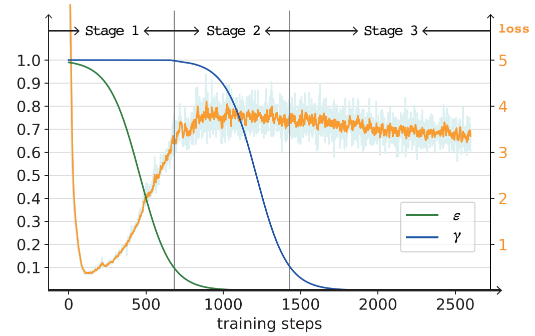


Figure 2: Heuristics. There are three stages to drive REST framework. According to inverse sigmoid decay, stage 1 begins with the whole training process, and ends when the decay factor ϵ reduces to 0.1. In stage 1, only GCNs are trained and gradually become stable. Likewise, stage 2 begins with the ending of stage 1, and last before decay factor γ reduce to 0.1. In stage 2, GCNs hinge on spatial dependencies from either EINs or prior knowledge, and steadily rely more on the former ones. In this stage, EINs get trained. Stage 3 starts when stage 2 ends. In this stage, EINs begin to explore possible links beyond prior graph structure, and REST start to count to early stop.

We develop a three-stage heuristics to train spatiotemporal models under REST framework, which is illustrated in Figure 2. In stage 1, we apply the scheduled sampling strategy [3] to learn the parameters of graph convolutions. Specifically, the inputs of DCRNNs' decoders are either from ground truth or previous predictions. We set a probability ϵ to control using ground truth or previous predictions, which will gradually decay and make the models entirely rely on their predictions. We choose inverse sigmoid decay to gradually reduce ϵ:

$$\epsilon = \frac{k}{k + \exp(\frac{i}{k})}, \tag{9}$$

where i refers to the current number of training steps, and k is a hyper-parameter to control how much steps it will take to reduce ϵ to 0. Generally, we adjust k to end the first stage within two epochs. As shown in Figure 2, in stage 1, loss rapidly drops and then rebounds, because the decoders gradually begin to conduct self-regression, rather than depending on the ground truth. In this stage, GCNs begin to learn how to capture temporal dependencies, while our EINs do not work. When ϵ decays to 0.1, stage 2 starts and EINs begin to learn how to measure the distance between time series and construct multi-modal spatial dependencies for GCNs.

In stage 2, the spatial dependencies come from either prior knowledge or inferred values from EINs. Likewise, We then set another decay factor γ, which reduces from 1 to 0 with the same speed as ϵ. In this stage, the spatial dependencies are restricted in predefined structure, with the weight changing. When γ decays to 0.1, stage 3 starts and the EINs begin to explore possible links among all pairwise time series. Though the phased heuristics, different part of REST framework can be sequentially and synergistically trained, which makes the REST framework easier to achieve local optima.

6 EXPERIMENTS

We have conducted extensive experiments to evaluate the effectiveness and efficiency of our proposed REST framework. In this section, we first introduce experimental settings, and then analyze experimental results in detail.

6.1 Experimental Settings

6.1.1 Dataset. We verified our REST framework on two widely used open datasets: a traffic dataset released by Li et al. [16] and Wu et al. [28], and a web traffic dataset from WikiState[2]. The traffic dataset, called Metr-LA, records four months (from March 1, 2012 to June 30, 2012) of statistics on traffic speed on 207 sensors on the highways of Los Angeles in five minutes period. We adopted the same distance measurement, as in [16], to construct the initial adjacent matrix. The web traffic dataset, named Wiki-EN, consists of 793 daily (from July 1, 2015 to August 31, 2017) web page views of 4,118 Wikipedia entries randomly sampled from the English Wikiproject. We crawled the hyperlinks among these Wikipedia entries to construct their initial spatial dependencies. For both datasets, we used Z-score normalization to preprocess input data, and for the Wiki-EN dataset, we further took logarithm to the inputs to eliminate the potential hazard caused by the huge difference in the order of magnitude. During the experiments, both datasets were chronologically split, with the first 70% as training set, the following 10% as validating set, and final 20% as testing set. Table 1 shows the detailed statistics of the two datasets.

Table 1: Statistics of Metr-LA and Wiki-EN

Datasets	# vertex	# edge	# observations
Metr-LA	207	1, 515	7, 094, 304
Wiki-EN	4, 118	8, 173	3, 265, 574

6.1.2 Evaluation Scheme. We evaluated REST framework and baselines by mean absolute error (MAE), root mean squared error (RMSE) and mean absolute percentage error (MAPE) for Metr-LA dataset, which is consistent with [16, 28]. However, considering huge difference in the order of magnitude of time series in Wiki-EN, we chose MAE, root mean squared logarithmic error (RMSLE) and symmetric mean absolute percentage error (SMAPE) to evaluate. Among these metrics, MAE, RMSE and RMSLE reflect absolute prediction errors, while MAPE and SMAPE reflect relative prediction errors. Note that the lower value of these metrics represent the higher prediction accuracy. Detailed formations of these metrics are summarized in Appendix B.

6.1.3 Baselines. We chose DCRNNs as backend of REST framework in this paper, and compared it with seven representative baselines: (i) Auto-Regressive Integrated Moving Average (ARIMA) [4]; (ii) Vector AutoRegressive (VAR) [18]; (iii) Support Vector Regression (SVR) [23]; (iv) RNN with fully connected LSTM hidden units (FC-LSTM) [25]; (v) WaveNet [27]; (vi) Diffusion Convolutional Recurrent Neural Network (DCRNN) [16]; and (vii) Graph WaveNet [28]. Among these approaches, ARIMA and SVR are designed for individual time series; VAR considers the interactions among multiple time series; FC-LSTM and WaveNet are RNN- and CNN-based deep learning models, which show significant capability of modeling nonlinear and long-term dependencies for individual time series; DCRNNs and Graph WaveNet introduce graph convolution filter to exploit spatial dependencies among time series, which are the state of the arts in this domain. Note that Graph WaveNet can also infer and quantify the spatial dependencies from time series.

6.1.4 Hyper-parameters and other settings. All of our experiments were conducted on an Ubuntu server [CPU: Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz, GPU: NVIDIA GTX 1080 Ti]. For Metr-LA dataset, we use the same experimental settings reported in [16, 28]. For the Wiki-EN dataset, the hyper-parameters were tuned for different models to achieve their best performance. In particular, we consider one-step adjacent vertices for VAR to predict the trend of central vertices. We set 128 temporal hidden states for each recurrent units, for FC-LSTM, DCRNNs and REST. In REST, we empirically set $C = 13$ for the number of MFCCs to retain; set 128 spatial hidden states for graph convolution filters. As for the most sensitive hyper-parameters, i.e., predefined graph convolution depth K and modality M, we conducted several experiments to carefully study them and will discuss it in detail in Section 6.2.2.

6.2 Experimental Results and Analyses

6.2.1 Main Results. To compare the overall prediction accuracy of REST framework with those of baselines, we conducted repeated experiments for ten times with different initialization. Table 2 records the average of MAE, RMSE, RMSLE, MAPE and SMAPE for Metr-LA

[2] https://dumps.wikimedia.org

Table 2: Performance comparison of REST framework and other baselines. (a) shows the metrics of Metr-LA dataset, where 15, 30 and 60 minutes refer to 3, 6, 12 steps predictions. (b) shows the metrics of Wiki-EN dataset, where 3, 7, 14 days refer to 3, 7, 14 steps predictions. The lower value reflects the higher prediction accuracy. The bold font highlight the best performance.

(a) Metr-LA

Models	15 min			30 min			60 min		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
ARIMA [4]	3.99	8.21	9.60%	5.15	10.45	12.70%	6.90	13.23	17.40%
VAR [18]	4.42	7.89	10.20%	5.41	9.13	12.70%	6.52	10.11	15.80%
SVR [23]	3.99	8.45	9.30%	5.05	10.87	12.10%	6.72	13.76	16.70%
FC-LSTM [25]	3.44	6.30	9.60%	3.77	7.23	10.90%	4.37	8.69	13.20%
WaveNet [27]	2.99	5.89	8.04%	3.59	7.28	10.25%	4.45	8.93	13.62%
DCRNN [16]	2.77	5.38	7.30%	3.15	6.45	8.80%	3.60	7.59	10.50%
Graph WaveNet [28]	2.69	5.15	6.90%	3.07	6.22	8.37%	3.53	7.37	10.01%
REST	**2.66**	**4.88**	**6.78%**	**2.94**	**5.63**	**7.83%**	**3.35**	**6.62**	**9.35%**

(b) Wiki-EN

Models	3 days			7 days			14 days		
	MAE	RMSLE	SMAPE	MAE	RMSLE	SMAPE	MAE	RMSLE	SMAPE
ARIMA [4]	1646	0.6854	33.11%	1758	0.7293	34.95%	2380	0.6820	36.11%
VAR [18]	2194	0.7636	42.52%	2711	0.8884	47.20%	3390	1.0425	53.58%
SVR [23]	1423	0.5151	30.05%	1526	0.5427	31.43%	1653	0.5773	33.20%
FC-LSTM [25]	827	0.3421	19.20%	878	0.3786	21.25%	952	0.4160	23.71%
WaveNet [27]	755	0.3435	19.11%	830	0.3797	21.03%	917	0.4175	23.41%
DCRNN [16]	765	0.3423	19.01%	827	0.3793	21.03%	903	0.4166	23.39%
Graph WaveNet [28]†	-	-	-	-	-	-	-	-	-
REST	**743**	**0.3400**	**18.63%**	**816**	**0.3776**	**20.70%**	**897**	**0.4148**	**23.07%**

† Graph WaveNet ran out of GPU memory and thus could not work on the Wiki-EN dataset.

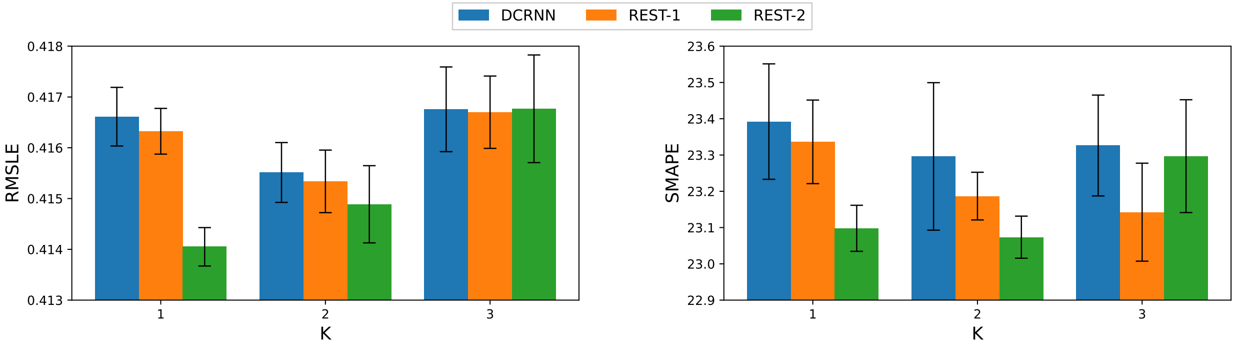
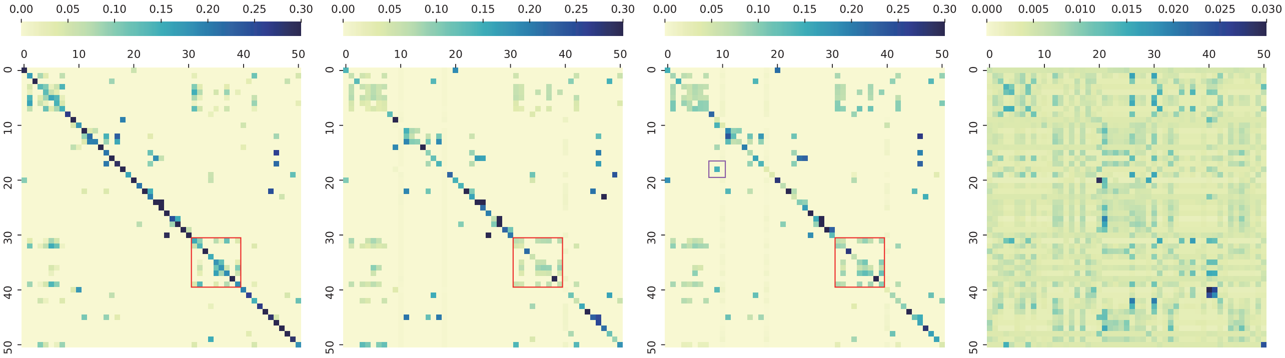


Figure 3: Performance comparison for DCRNNs and REST framework with different graph convolution depth K and modality M. In particular, REST-1 and REST-2 refer to 1 and 2 modalities that are concerned. In general, REST framework with $K = 2$, $M = 2$ beats other settings.

and Wiki-EN. The results of all baseline methods on Metr-LA are directly taken from Li et al. [16] and Wu et al. [28], and other results are based on our experiments. In particular, we chose DCRNNs as GCNs backend for REST framework in this paper. Examining the results over both datasets, we notice four consistent phenomena. First, VAR considers the spatial dependencies of time series, however its prediction error is dramatically higher than that of ARIMA, which indicates that handcraft spatial dependencies may carry many noises and thus may not always offer a good base. Second, all neural network based models significantly outperform the previous ones, especially when the prediction horizons become



(a) handcraft graph (b) EINs inferred, modality 1 (c) EINs inferred, modality 2 (d) Graph WaveNet inferred

Figure 4: Visualization of the spatial dependencies of the last 51 vertices on Metr-LA dataset, where the deeper color represents the higher correlation. In particular, (a) is defined by road network distance based on expert knowledge; (b) and (c) are two modalities inferred by EINs; (d) is inferred by Graph WaveNet. Note that the spatial dependencies inferred by Graph WaveNet are dense and minor. Best viewed in color.

longer. That is because RNN and TCN structures are with great abilities to model the non-linear and long-term temporal dependencies. Third, spectral graph convolution based approaches, i.e., DCRNNs, Graph WaveNet and REST framework, gain further improvement, comparing with other deep learning models without considering spatial features. It demonstrates that graph convolution is capable of making use of spatial dependencies. Note that Graph WaveNet, considering all possible time series pairs during training procedure, cannot work in dataset with relative large graph size, like Wiki-EN with 4,118 vertices. Fourth, REST framework made the most accurate predictions among these methods, which should attribute to the EINs inferring better spatial dependencies than the handcraft ones.

6.2.2 Important Hyper-parameters Discussion. In REST framework, the predefined number of graph convolution depth K of GCNs and the predefined number of modality M of EINs are two major hyperparameters. In this section, we carefully compare the performance of REST framework with different K and M to study how they couple with each other in our REST framework. Figure 3 reports the average and standard deviation of RMSLE and SMAPE of DCRNN and REST framework, with different hyper-parameters. As shown, within each graph convolution depth K, REST framework outperforms DCRNNs. In particular, REST framework predicts slightly more accurately than DCRNNs with $M = 1$. However, with the increment of predefined modality, REST framework significantly outperforms DCRNNs with $M = 2$. This phenomenon indicates that EINs are able to infer high-quality spatial dependencies for GCNs. Moreover, with the predefined number of modality being larger, EINs can learn more abundant spatial representations and promote GCNs to make more accurate predictions. Besides, given various predefined modality, RMSLE and SMAPE of both DCRNNs and REST framework both drop from 1 to 2, and then rebound. Since K related to the receptive field of graph convolution, we suspect $K = 2$ best fit Wiki-EN dataset, which is consistent with common conclusion in [12].

Table 3: Phased heuristics ablation.

REST framework	MAE	RMSLE	SMAPE
with heuristics	897 ± 9	0.4148 ± 0.0007	23.07% ± 0.05
w/o heuristics	904 ± 12	0.4152 ± 0.0011	23.34% ± 0.13

6.2.3 Heuristics Ablations. To evaluate the effect of the phased heuristics, we conducted an ablation experiment with ten times for each configuration. Table 3 reports the RMSLE and SMAPE of REST framework adopted the phased heuristics or not. Examining the average and standard deviation of both metrics in Table 3, REST framework driven by the phased heuristics performs significantly better than REST without it. Through observing the experimental results, we find REST framework without the phased heuristics could also achieve as lower as the prediction errors of REST framework with it. However, without heuristics, REST framework is more likely to get early stopped, which thus leads to higher prediction error and standard deviation.

6.2.4 Visualization. Both REST framework and Graph WaveNet can infer spatial dependencies from time series. Figure 4 visualizes the spatial dependencies of the last 51 vertices on the Metr-LA dataset. Based on handcraft spatial dependencies [16, 22] (i.e., Figure 4 (a)), EINs generated 2 modalities of spatial dependencies (i.e., Figure 4 (b) and (c)). Comparing Figure 4 (b) and (c) with (a), we found EINs can quantify the spatial dependencies from different points (see the red region). Besides, it can also identify new linkages (see the magenta region). Examining Figure 4 (d), we found the spatial dependencies inferred by Graph WaveNet are dense and similar, which is not practical when the size of graph becomes larger.

7 CONCLUSIONS

In this paper, we have presented Reciprocal Spatiotemporal (REST) framework for spatiotemporal-coupled prediction. In the REST framework, on the one hand, our introduced Edge Inference Networks (EINs) infer the spatial dependencies between time series vertices and serve GCNs. On the other hand, GCNs integrate state-of-the-art spatiotemporal prediction models, e.g., DCRNNs, to utilize EINs inferred spatial dependencies and make more precise time series prediction, while introduce feedback to optimize EINs. To maximize the power of the REST framework, we also designed a phased heuristics to stabilize training procedure and help it quickly converge to their local optima. Through iterative joint learning process, the performance of EINs and GCNs mutually benefit each other, and eventually lead to accurate spatiotemporal predictions. Extensive experimental results over real-world datasets have demonstrated the effectiveness and efficiency of our REST framework in terms of prediction accuracy and spatial dependencies inference, without introducing unaffordable computational and storage burden.

In the future, we plan to focus on the following three aspects: (1) to conduct in-depth study of the features of general time series and upgrade EINs to model the temporal patterns of time series and exclude their noise; (2) to study how to introduce more feedback to EINs to facilitate its exploration of possible links; (3) to apply REST to other domains to verify its generality, such as software invocation prediction and pandemic prediction.

ACKNOWLEDGMENTS

This research has been partially supported by the National Key Research and Development Program of China (No. 2018YFB1402500), the National Nature Science Foundation of China (No.61673230). Yushun Fan is the corresponding author.

REFERENCES

[1] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. 2015. Time-series clustering–a decade review. *Information Systems* 53 (2015), 16–38.

[2] James Atwood and Don Towsley. 2016. Diffusion-convolutional neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems.* 2001–2009.

[3] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems.* 1171–1179.

[4] George E. P. Box and Gwilym M. Jenkins. 1970. *Time series analysis forecasting and control.* HOlden-Day.

[5] John S Bridle and Michael D Brown. 1974. An experimental automatic word recognition system. *JSRU report* 1003, 5 (1974), 33.

[6] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs. In *Proceedings of International Conference on Learning Representations.*

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[8] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of International Conference on Machine Learning.* 933–941.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems.* 3844–3852.

[10] Junxiang Fan, Qi Li, Junxiong Hou, Xiao Feng, Hamed Karimian, and Shaofu Lin. 2017. A spatiotemporal prediction framework for air pollution based on deep RNN. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4 (2017), 15.

[11] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3656–3663.

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st Conference on Neural Information Processing Systems.* 1025–1035.

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[14] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[15] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval.* 95–104.

[16] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In *Proceedings of International Conference on Learning Representations.*

[17] Haozhe Lin, YuShun Fan, Jia Zhang, and Bing Bai. 2020. MSP-RNN: Multi-Step Piecewise Recurrent Neural Network for Predicting the Tendency of Services Invocation. *IEEE Transactions on Services Computing* (2020).

[18] Helmut Lütkepohl. 2005. *New introduction to multiple time series analysis.* Springer Science & Business Media.

[19] Paul Mermelstein. 1976. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence* 116 (1976), 374–388.

[20] Md Sahidullah and Goutam Saha. 2012. Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition. *Speech communication* 54, 4 (2012), 543–565.

[21] Sebastian Scher and Gabriele Messori. 2019. Weather and climate forecasting with neural networks: using general circulation models (GCMs) with different complexity as a study ground. *Geoscientific Model Development* 12, 7 (2019), 2797–2809.

[22] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine* 30, 3 (2013), 83–98.

[23] Alex J Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Statistics and Computing* 14, 3 (2004), 199–222.

[24] Otilia Stretcu, Krishnamurthy Viswanathan, Dana Movshovitz-Attias, Emmanouil Platanios, Sujith Ravi, and Andrew Tomkins. 2019. Graph agreement models for semi-supervised learning. In *Proceedings of the 32th International Conference on Neural Information Processing Systems.* 8713–8723.

[25] I Sutskever, O Vinyals, and QV Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Vol. 2. 3104–3112.

[26] Souhaib Ben Taieb, Antti Sorjamaa, and Gianluca Bontempi. 2010. Multiple-output modeling for multi-step-ahead time series forecasting. *Neurocomputing* 73, 10-12 (2010), 1950–1957.

[27] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. In *9th ISCA Speech Synthesis Workshop.* 125–125.

[28] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence.* 1907–1913.

[29] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations.*

[30] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence.* 3634–3640.

[31] Fisher Yu and Vladlen Koltun. 2016. Multi-scale context aggregation by dilated convolutions. In *Proceedings of International Conference on Learning Representations.*

[32] Muhan Zhang and Yixin Chen. 2017. Weisfeiler-lehman neural machine for link prediction. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 575–583.

[33] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *Proceedings of the 31th International Conference on Neural Information Processing Systems.* 5165–5175.

[34] Qi Zhang, Jianlong Chang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. 2020. Spatio-temporal graph structure learning for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1177–1185.

A NOTATIONS

B EVALUATION METRICS

Obey our definitions in Section 3, aforementioned metrics are defined as follows, where Ω refers to the indices of observed samples. Note that missing data are not included in Ω.

Table 4: Notations

Name	Explanations
$\mathcal{G}$	Graph, or spatial dependencies.
$\mathcal{V}$, N	Vertex set of $\mathcal{G}$, $\lvert\mathcal{V}\rvert = N$
$\mathcal{E}$, $\mathcal{W}$	Edge set and corresponding wight set of $\mathcal{G}$.
$\boldsymbol{L}$	Laplacian matrix of $\mathcal{G}$, which refers to random walk definition in this work.
$\boldsymbol{D}$, $\boldsymbol{D}_I$, $\boldsymbol{D}_O$	Degree matrix, which can break down into input degree $\boldsymbol{D}_I$, and output degree $\boldsymbol{D}_O$.
$\boldsymbol{A}$, M	Adjacent matrix, which are considered with M modalities.
g_Θ, Θ, K	Graph convolutional filter with trainable parameters Θ. $\Theta \in \mathbb{R}^{M \times K \times d_i \times d_o}$, with M modalities, K predefined graph convolution depth, d_i and d_o as input and output dimensions of graph convolutional filters.
$\boldsymbol{H}$	Hidden states, output of graph convolution operators, $H \in \mathbb{R}^{N \times d_o}$.

- mean absolute error (MAE):

$$\mathrm{MAE} = \frac{1}{\Omega} \sum_{i \in \Omega} \mid y_i - \hat{y}_i \mid$$

- mean squared error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{\Omega} \sum_{i \in \Omega} (y_i - \hat{y}_i)^2}$$

- root mean squared logarithmic error (RMSLE):

$$\mathrm{RMSLE} = \sqrt{\frac{1}{\Omega} \sum_{i \in \Omega} [\log(y_i + 1) - \log(\hat{y}_i + 1)]}$$

- mean absolute percentage error (MAPE):

$$\mathrm{MAPE} = \frac{1}{\Omega} \sum_{i \in \Omega} \frac{\mid y_i - \hat{y}_i \mid}{y_i}$$

- symmetric mean absolute percentage error (SMAPE):

$$\mathrm{SMAPE} = \frac{2}{\Omega} \sum_{i \in \Omega} \frac{\mid y_i - \hat{y}_i \mid}{y_i + \hat{y}_i}$$