Spatio-Temporal Meta Learning for Urban Traffic Prediction

Zheyi Pan, Wentao Zhang, Yuxuan Liang,
Weinan Zhang, Yong Yu, Junbo Zhang, and Yu Zheng

Abstract—Predicting urban traffic is of great importance to intelligent transportation systems and public safety, yet is very challenging in three aspects: 1) complex spatio-temporal correlations of urban traffic, including spatial correlations between locations along with temporal correlations among timestamps; 2) spatial diversity of such spatio-temporal correlations, which varies from location to location and depends on the surrounding geographical information, *e.g.*, points of interests and road networks; and 3) temporal diversity of such spatio-temporal correlations, which is highly influenced by dynamic traffic states. To tackle these challenges, we proposed a deep meta learning based model, entitled ST-MetaNet$^+$, to *collectively* predict traffic in all locations at the same time. ST-MetaNet$^+$ employs a sequence-to-sequence architecture, consisting of an encoder to learn historical information and a decoder to make predictions step by step. Specifically, the encoder and decoder have the same network structure, consisting of meta graph attention networks and meta recurrent neural networks, to capture diverse spatial and temporal correlations, respectively. Furthermore, the weights (parameters) of meta graph attention networks and meta recurrent neural networks are generated from the embeddings of geo-graph attributes and the traffic context learned from dynamic traffic states. Extensive experiments were conducted based on three real-world datasets to illustrate the effectiveness of ST-MetaNet$^+$ beyond several state-of-the-art methods.

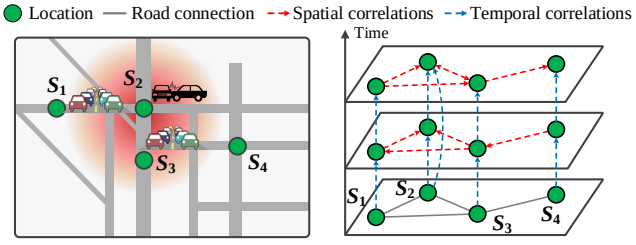
Index Terms—Urban traffic, spatio-temporal data, neural network, meta learning

1 Introduction

Recent advances in data acquisition technologies and mobile computing lead to a large collection of traffic data (*e.g.*, vehicle trajectories), enabling us to conduct urban analysis and works on downstream applications [2]. Urban traffic prediction, such as traffic speed prediction [3] and citywide flow prediction [4], has become a mission-critical work for intelligence city efforts, as it can provide insights for urban planning and traffic management to improve the efficiency of public transportation, as well as to raise early warnings for public safety emergency management [5].

However, forecasting urban traffic is very challenging due to the complex spatio-temporal (ST) correlations. Specifically, the complexity of ST correlations lies in the following two aspects:

• Complex composition of ST correlations
Urban traffic is highly dynamic-varying in both temporal and spatial domains. In the temporal domain, the current traffic readings of a certain location, such as traffic speed reported by loop detectors, are strongly correlated with its precedents. Figure 1(a) illustrates an example to support this fact. Suppose there is a car accident at S_2 at 9:00 am, it will result in a heavy traffic jam at S_2, and decrease the nearby traffic speed for a long time. On the other hand, in the spatial domain, the traffic of some locations is mutually correlated. As shown in Figure 1(a), since traffic has strong spatial dependencies [3] on road networks, the traffic congestion at S_2 will quickly diffuse to its neighbors, *i.e.*, S_1 and S_3, and impact their traffic condition.



(a) Example of ST correlations (b) ST correlations on geo-graph

Fig. 1: Complex composition of ST correlations.

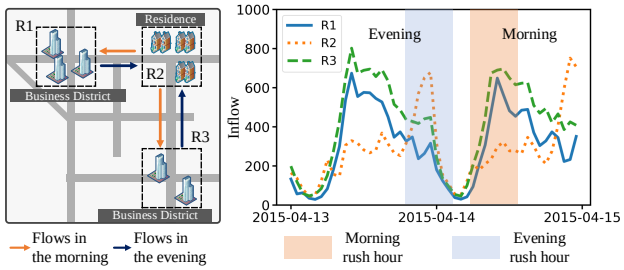
As urban traffic broadcasts in the spatial domain (*e.g.*, along road networks), and changes over time, we employ a geo-graph to describe the spatial structure, containing nodes and edges to represent locations and the relationships between pairs of locations, respectively. In general, a node can be a sensor on a road or a big region within a city, which is determined by the prediction target, such as road speed detected by sensors or citywide flows of urban regions. In the meantime, a widely used scheme to build edges is directly based on road connection between pairs of nodes, because traffic moves along road networks. As shown in Figure 1(b), the red and blue arrows represent the spatial correlations between locations and temporal correlations within each location, respectively. As both types of correlations interact with each other and affect urban traffic,

- *This paper is an extended version of [1] which has been accepted for the presentation at 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*
- *Zheyi Pan, Wentao Zhang, Weinan Zhang, and Yong Yu are with Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. E-mail: zheyi.pan@outlook.com, zwt1999@sjtu.edu.cn, {wnzhang, yyu}@apex.sjtu.edu.cn,*
- *Yuxuan Liang is with School of Computing, National University of Singapore. Email: yuxliang@outlook.com.*
- *Junbo Zhang and Yu Zheng are with JD Intelligent Cities Research and JD Intelligent Cities Business Unit, JD Digits, Beijing, China. Email: {msjunbozhang, msyuzheng}@outlook.com.*
- *Corresponding author: Junbo Zhang, Weinan Zhang, Yong Yu, Yu Zheng.*

it is necessary to *simultaneously* capture such spatial and temporal correlations.

• Diversity of ST correlations

1. Spatial diversity of ST correlations. ST correlations of urban traffic are different, across nodes and edges in the spatial domain, *i.e.*, geo-graph. In urban areas, characteristics of nodes and their mutual relationships are diverse, depending on their own geo-graph attributes: 1) *node attributes*: the condition of a node, such as the distribution of nearby points of interests (POIs) and density of road networks (RNs); 2) *edge attributes*: the relationships between two nodes, such as the features of road segments (*e.g.*, the number of lanes, speed limit, etc.) and the geospatial distance between them. As the example shown in Figure 2(a), R1 and R3 are business districts, consisting of numerous office buildings, while R2 is a residence district, which has many apartments. These districts are distinguished by their distributions of POIs and road network structures, resulting in the different characteristics of them. Besides, it can be easily seen in Figure 2(b) that the trends of their inflows are diverse, revealing that districts with different characteristics always have different types of ST correlations.



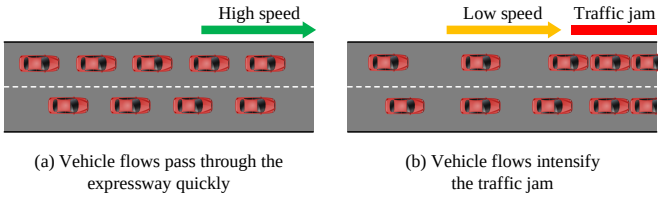
(a) Distribution of geo-attributes (b) Diversity of ST correlations

Fig. 2: Spatial diversity of ST correlations.

Nonetheless, an important fact to adhere to is that nodes with similar combinations of geo-graph attributes can lead to similar characteristics and analogous types of ST correlations. As the example shown in Figure 2(a), in general citizens usually commute from home to their workplaces in the morning, while opposite at night. Thus, business districts R1 and R3 witness similar upward trends of inflows in the morning, while the residential district R2 meets a completely different rush hour in the evening, as shown in Figure 2(b). Therefore, it is essential to model such diversity by considering the inherent relationships between geo-graph attributes and the types of ST correlations.

2. Temporal diversity of ST correlations. The types of ST correlations change over time, depending on the context included in dynamic traffic states. As the example shown in Figure 3(a), on an expressway with large amounts of vehicle flows, traffic usually quickly passes through, as vehicles can keep high speed. In this case, large amounts of vehicle flows would not impact future traffic conditions. However, sometimes there is a traffic jam blocking the expressway, as shown in Figure 3(b). Consequently, the vehicle flows would intensify the traffic jam, which shows a different type of ST correlations. Therefore, effectively modeling the relationships between dynamic traffic states and the types of ST correlations is extremely important in traffic prediction.

Recently, although there has been significant growth of works in ST prediction (including urban traffic prediction), the aforementioned challenges are still not completely solved. For instance, several studies [3], [4], [6], [7], [8], [9], [10] focus on modeling ST correlations by a single model for all locations. However, these methods cannot explicitly model the inherent relationships between geo-graph attributes and various types of ST correlations, as a result, such relationships are hard to be learned without any prior knowledge. Another group of works [11], [12], [13] adopt multi-task learning techniques, which build multiple sub-models for each location and all of them are trained together under the presence of similarity constraints between locations. These methods depend heavily on the prior knowledge (*i.e.*, the location similarity) or over-strong assumption of the specific tasks. Naturally, the main drawback of this kind of methods lies in the lack of ground truth to reflect such similarity and assumption. Therefore, such side information can only provide relatively weak supervision, producing unstable & tricked, even ruinous results in complex real-world applications.

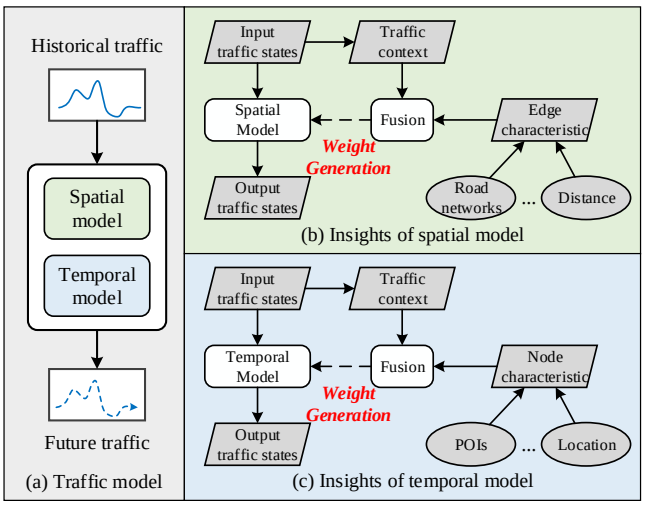


(a) Vehicle flows pass through the expressway quickly (b) Vehicle flows intensify the traffic jam

Fig. 3: Temporal diversity of ST correlations.

To tackle all the above challenges, we propose a deep meta learning based framework, entitled ST-MetaNet$^+$, for urban traffic prediction. The key insight is to regard geo-graph attributes and dynamic traffic states as the metadata of ST neural networks for capturing ST correlations. Therefore, to capture the relationships between ST correlations and these metadata, a direct solution is to employ the weight generation-based meta learning method.

More concretely, as shown in Figure 4(a), we first combine spatial and temporal models to capture these two types of correlations, simultaneously. Then, since ST correlations are implicitly affected by the characteristics of nodes (locations) and edges (inter-location relationships), as well as dynamic traffic states, we have to further capture such relationships, as presented in Figure 4(b) and Figure 4(c).



(a) Traffic model (b) Insights of spatial model (c) Insights of temporal model

Fig. 4: Insights of our framework.

Intuitively, the characteristic of an edge relies on its attributes, *e.g.,* the road connectivity and the distance between nodes. Likewise, the characteristic of a node is influenced by its attributes, like the GPS location and the distribution of nearby POIs. Besides, traffic states contain implicit information about traffic context, which impacts ST correlations. Based on these insights, ST-MetaNet$^+$ first extracts the *meta knowledge* (*i.e.,* characteristics) of nodes and edges from their attributes respectively, as well as the dynamic traffic context from traffic states. After that, the extracted information (*i.e.,* the meta knowledge and the traffic context) are simply aggregated by a data fusion module and then used to model the ST correlations, namely, generating the weights of the spatial and temporal models. The main contributions of our study are four folds:

- We design a novel framework based on deep meta learning, entitled ST-MetaNet$^+$, to forecast urban traffic. ST-MetaNet$^+$ leverages the meta knowledge extracted from geo-graph attributes and dynamic traffic context learned from traffic states to generate the parameter weights of graph attention networks and recurrent neural networks within a sequence-to-sequence architecture. As a result, it can capture the inherent relationships between diverse types of ST correlations and geo-graph attributes along with dynamic traffic states.
- An improved meta graph attention network (Meta-GAT$^+$) is proposed to model the spatial correlations. The attention mechanism can capture the dynamic mutual relationships between locations, with regard to their current states. In addition, the weights of the graph attention networks are generated by the meta knowledge of nodes and edges extracted from geo-graph attributes, as well as dynamic traffic context of nodes extracted from traffic states, such that it can model diverse spatial correlations.
- We propose an improved meta gated recurrent neural network, entitled Meta-GRU$^+$, which generates all weights of a normal gated recurrent unit from the meta knowledge and traffic context of each node. Thus each location has a unique model for its own type of temporal correlation under different traffic states.
- We evaluate our framework on three typical traffic prediction tasks in the real world. The experiment results verify that ST-MetaNet$^+$ can significantly improve the predictive performance, and learn better traffic-related knowledge from the given geo-graph attributes.

2 PRELIMINARIES

In this section, we introduce the definitions and problem statement. All frequently used notation is shown in Table 1.

Suppose there are N_l locations, which report D_t types of traffic information (*e.g.,* traffic flows and speed) on N_t timestamps respectively.

Definition 1. ***Urban traffic*** *is denoted as a tensor* $\mathcal{X} = [X_1, \cdots, X_{N_t}] \in \mathbb{R}^{N_t \times N_l \times D_t}$*, where* $X_t = \left[x_t^{(1)}, \cdots, x_t^{(N_l)}\right]$ *denotes all locations' traffic information at timestamp* t*.*

Definition 2. ***Geo-graph*** *is a directed graph that represents locations and their mutual relationships, denoted as* $\mathcal{G} = \{V, E, \mathcal{V}, \mathcal{E}\}$*. Specifically,* $V = \{1, \cdots, N_l\}$ *represents node indices, while* $E = \{(i,j) \mid 1 \le i, j \le N_l\}$ *represents directed edges, where each pair* (i,j) *denotes node* j *impacts node* i*. In addition, nodes and edges are associated with attribute vectors, denoted as* $\mathcal{V} = \left[v^{(1)}, \cdots, v^{(N_l)}\right]$ *and* $\mathcal{E} = \left\{e^{(ij)} \mid (i,j) \in E\right\}$*, to represent the geographical features of nodes and relationships between nodes, respectively. Moreover, we use* $\mathcal{N}_i = \{j \mid (i,j) \in E\}$ *to denote the neighbors of node* i*.*

TABLE 1: Notation table.

Notation	Description
N_l, N_t	Number of locations/timestamps.
$\tau_{\text{in}}, \tau_{\text{out}}$	Timestamps for historical/future traffic
X_t	The traffic readings at all timestamps.
$v^{(i)}$	The node attributes of location i.
$e^{(ij)}$	The edge attributes between node i and j.
$\mathcal{N}_i$	Neighborhoods of location i.
NMK $(\cdot)$	The function to learn node meta knowledge.
EMK $(\cdot)$	The function to learn edge meta knowledge.
CL $(\cdot)$	The context learner to learn traffic context.
$g_\theta(\cdot)$	The function to learn parameter weights θ.

With above two definitions, here we present the formal definition of the research problem in this work.

Problem 1. *Given previous* τ_{in} *traffic information* $[X_1, \cdots, X_{\tau_{\text{in}}}]$ *and geo-graph* $\mathcal{G}$*, predict the traffic for all locations in the next* τ_{out} *timestamps, denoted as* $\left[\hat{Y}_1, \cdots, \hat{Y}_{\tau_{\text{out}}}\right]$*.*

3 METHODOLOGIES

In this section, we describe the architecture of ST-MetaNet$^+$, as shown in Figure 5(a). Leveraging the sequence-to-sequence architecture [14], ST-MetaNet$^+$ is composed of two separate modules: the encoder (blue part) and the decoder (green part). The former one is used to encode the sequence of the input, *i.e.,* the historical information of urban traffic $[X_1, \cdots, X_{\tau_{\text{in}}}]$, producing the output hidden states, which are used as the initial states of the decoder that further predicts the output sequence $\left[\hat{Y}_1, \cdots, \hat{Y}_{\tau_{\text{out}}}\right]$. More specifically, the encoder and the decoder have the same network structure, consisting of three types of components:

1) ***Meta-knowledge learner***. As shown in Figure 5(b), we use two fully connected networks (FCNs), named node-meta-knowledge learner (NMK-Learner) and edge-meta-knowledge learner (EMK-Learner), to respectively learn the *meta-knowledge* of nodes (NMK) and edges (EMK) from node attributes (*e.g.,* POIs and GPS locations) and edge attributes (*e.g.,* road connectivity and the distance between nodes). Then the learned meta knowledge is further used to learn the weights of another two types of networks, *i.e.,* graph attention network (GAT) and recurrent neural network (RNN). Taking a certain node as an example, the attributes of the node are fed into the NMK-Learner, and it outputs a vector, representing the meta knowledge of that node.
2) ***Meta-GAT$^+$ (meta graph attention network$^+$)***, the improved version of Meta-GAT proposed in [1], is comprised of a context learner, a fusion gate, a meta learner, and a GAT, as shown in Figure 5(c). In this component, the traffic context is learned from the input states by an FCN, namely, the context learner. Then the fusion gate combines the meta knowledge of nodes and edges, and the learned traffic context. After that, we propose

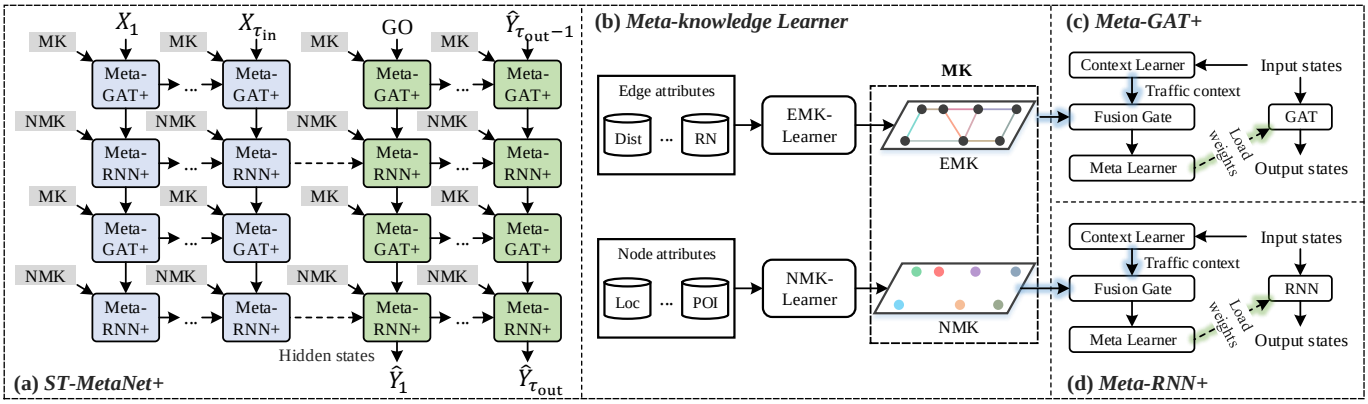


Fig. 5: Overview of ST-MetaNet$^+$.

to employ an FCN as the meta learner, which takes the output of the fusion gate as the input, and calculates the parameter weights of GAT. Meta-GAT$^+$ can capture diverse spatial correlations by individually broadcasting nodes' hidden states along edges.

3) ***Meta-RNN$^+$*** ***(meta recurrent neural network$^+$)***, the improved version of Meta-RNN proposed in [1], is comprised of a context learner, a fusion gate, a meta learner, and an RNN, as shown in Figure 5(d). Similar to Meta-GAT$^+$, the traffic context is learned from the input traffic states by the context learner, and is fused with the node meta knowledge by the fusion gate. Then we use the meta learner to generate the weights of RNN for each node from the output of the fusion gate. Meta-RNN$^+$ can capture diverse temporal correlations associated with nodes' geo-information and dynamic traffic states.

In the following subsections, we will respectively illustrate each component of ST-MetaNet$^+$ in details.

3.1 Meta-Knowledge Learner

In urban areas, characteristics of locations and their mutual relationships are diverse, depending on geographical information, *e.g.*, POIs and RNs. Such diverse characteristics bring about various types of ST correlations within urban traffic. Hence, we propose two meta-knowledge learners, *i.e.*, NMK-Learner and EMK-Learner, to learn traffic-related node and edge embeddings (meta knowledge) from geographical information, respectively. As shown in Figure 5(b), two meta-knowledge learners respectively employ different FCNs, in which the input is the attributes of a node or an edge, and the corresponding output is the embedding (vector representation) of that node or edge. Since such embeddings are used for generating weights of GAT and RNN to capture ST correlations of urban traffic, the learned embeddings can reflect traffic-related characteristics of nodes and edges. For simplicity, we use NMK $\left(v^{(i)}\right)$ and EMK $\left(e^{(ij)}\right)$ to denote the learned meta knowledge (embedding) of a node and an edge, respectively.

3.2 Meta Graph Attention Network$^+$

Urban traffic has spatial correlations that some locations are mutually affected. In addition, such correlations are diverse across nodes and edges, and related to geographical information and dynamic traffic states. Inspired by graph attention network [15], we propose to employ attention mechanisms into the framework to capture diverse spatial correlations between nodes. However, it is inappropriate to directly apply GAT because all nodes and edges would use the same attention mechanism, ignoring the relationships between spatial correlations and geographical information along with dynamic traffic states.

To capture such diverse spatial correlations, we propose an improved meta graph attention network (Meta-GAT$^+$) as shown in Figure 6, which employs an attention network whose weights are generated from the meta knowledge (the embeddings of geographical information) and the traffic context (the embeddings of input traffic states) by the meta learner. Consequently, the attention mechanisms for spatial correlation modeling are different across nodes and edges, and depending on geographical information and dynamic traffic states.

Formally, suppose the inputs of Meta-GAT$^+$ are $H = \left[h^{(1)}, \cdots, h^{(N_l)}\right] \in \mathbb{R}^{N_l \times D_h}$ (*i.e.*, the inputs of traffic states at a single timestamp) and geo-graph $\mathcal{G}$, while the output is $\bar{H} = \left[\bar{h}^{(1)}, \cdots, \bar{h}^{(N_l)}\right] \in \mathbb{R}^{N_l \times D'_h}$, where D_h and D'_h are the dimension of nodes' hidden states. The meta graph attention mechanism for each node contains two main steps: 1) attention score calculation for each edge; and 2) hidden state aggregation. As shown in Figure 6, we give an example to show the structure of Meta-GAT$^+$, that calculates the impact on the red node from its neighborhoods (the purple, orange, and green node) along edges. The details of Meta-GAT$^+$ are as follows.

Attention score calculation

First, the input H is projected to a new space by a single fully connected layer, denoted as $H' = \left[h'^{(1)}, \cdots, h'^{(N_l)}\right] \in \mathbb{R}^{N_l \times D'_h}$. Then the attention scores are calculated based on H' and the meta knowledge of geo-graph.

As we discussed, the attention score of edge (i, j) is related to the hidden states of node i and node j, the node and edge meta knowledge learned from geographical information, and the dynamic traffic context of these nodes. As shown in Figure 6, for edge (i, j), we fetch the hidden states of nodes by index, *i.e.*, $h'^{(i)}$ and $h'^{(j)}$, and the meta knowledge $\text{MK}^{(ij)}$, which is a composition of meta knowledge of nodes and edge:

$$\text{MK}^{(ij)} = \text{NMK}\left(v^{(i)}\right) \parallel \text{NMK}\left(v^{(j)}\right) \parallel \text{EMK}\left(e^{(ij)}\right), \quad (1)$$

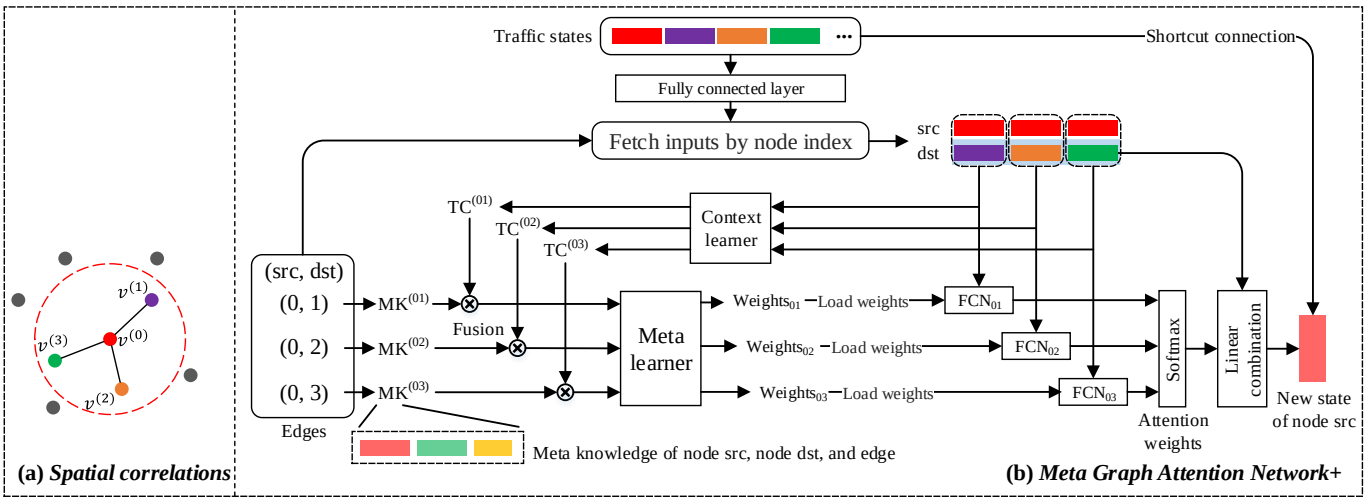


Fig. 6: Structure of Meta-GAT$^+$.

where NMK $\left(v^{(i)}\right)$, NMK $\left(v^{(j)}\right)$, and EMK $\left(e^{(ij)}\right)$ are one dimensional vectors, while $\|$ is vector concatenation operator. After that, using the traffic hidden states as the inputs, the dynamic traffic context (TC) of each edge can be calculated by:

$$\mathrm{TC}^{(ij)} = \mathrm{CL}_{\mathrm{GAT}}\left(h'^{(i)}\right) \| \mathrm{CL}_{\mathrm{GAT}}\left(h'^{(j)}\right), \tag{2}$$

where $\mathrm{CL}_{\mathrm{GAT}}(\cdot)$ is the context learner, a learnable FCN sharing parameters across all nodes. We set the output dimension of $\mathrm{CL}_{\mathrm{GAT}}(\cdot)$ such that $\mathrm{TC}^{(ij)}$ has the same dimension as $\mathrm{MK}^{(ij)}$. Then we can apply a function to calculate the attention score based on these vectors, denoted as:

$$w^{(ij)} = a(h'^{(i)}, h'^{(j)}, \mathrm{MK}^{(ij)}, \mathrm{TC}^{(ij)}) \in \mathbb{R}^{D'_h}, \tag{3}$$

where $w^{(ij)}$ is a D'_h dimension vector, denoting the importance of how $h'^{(j)}$ impacts $h'^{(i)}$ at each channel. Like GAT shown in Figure 6(b), we employ a single fully connected layer to calculate function $a(\cdot)$. However, different pairs of nodes have different meta knowledge and dynamic traffic context, resulting in different attention mechanisms. To model such diversity, we employ an edge-specific fully connected layer, followed by activation of LeakyReLU [16]:

$$\begin{aligned} &a\left(h'^{(i)}, h'^{(j)}, \mathrm{MK}^{(ij)}, \mathrm{TC}^{(ij)}\right) \\ &= \mathrm{LeakyReLU}\left(W^{(ij)}\left[h'^{(i)} \| h'^{(j)}\right] + b^{(ij)}\right), \end{aligned} \tag{4}$$

where $W^{(ij)} \in \mathbb{R}^{D'_h \times 2D'_h}$, $b^{(ij)} \in \mathbb{R}$ are edge-specific parameters of the fully connected layer. In particular, these parameters are generated from the fusion information (FI) of the meta knowledge $\mathrm{MK}^{(ij)}$ and the traffic context $\mathrm{TC}^{(ij)}$, as shown in Figure 6(b). The insight is that $\mathrm{MK}^{(ij)}$ can show static properties of this edge, while $\mathrm{TC}^{(ij)}$ can indicate how each static property takes effect under the specific traffic state. So inspired by the gating function used in long-short term memory [17], here we apply a fusion gate to calculate the fusion information, which can be formulated as:

$$\mathrm{FI}^{(ij)} = \phi\left(\mathrm{MK}^{(ij)}\right) \otimes \sigma\left(\mathrm{TC}^{(ij)}\right), \tag{5}$$

where $\otimes$ is Hadamard product, $\sigma(\cdot)$ is sigmoid function, and $\phi(\cdot)$ is tanh function. In this formula, $\sigma(\mathrm{TC}^{(ij)})$ shows the importance of each dimension in $\mathrm{MK}^{(ij)}$, making the fusion information reflect the dynamic attention mechanism of edge (i, j).

After getting the fusion information $\mathrm{FI}^{(ij)}$, we employ a meta learner, consisting of two FCNs g_W and g_b, which share parameters across all edges, to generate $W^{(ij)}$ and $b^{(ij)}$, respectively. Then, for any edge (i, j):

$$\begin{aligned} W^{(ij)} &= g_W\left(\mathrm{FI}^{(ij)}\right) \in \mathbb{R}^{D'_h \times 2D'_h}, \\ b^{(ij)} &= g_b\left(\mathrm{FI}^{(ij)}\right) \in \mathbb{R}. \end{aligned} \tag{6}$$

Note that the output of an FCN is a vector, so we need to reshape the output to the corresponding parameter shape. And finally, we can use the resulting $W^{(ij)}$ and $b^{(ij)}$ to calculate attention function $a(\cdot)$.

Hidden state aggregation

Like GAT, we firstly normalize the attention scores for a node across all its neighborhoods by softmax:

$$\alpha^{(ij)} = \frac{\exp\left(w^{(ij)}\right)}{\sum_{j \in \mathcal{N}_i} \exp\left(w^{(ij)}\right)}. \tag{7}$$

Then for each node, we calculate the overall impact of the neighborhoods by linear combinations of the hidden states corresponding to the normalized weights, and then apply a nonlinearity function ReLU, which is expressed as $\mathrm{ReLU}\left(\sum_{j \in \mathcal{N}_i} \alpha^{(ij)} h'^{(j)}\right)$. In addition, we add a shortcut connection to make network easily train. And finally, the hidden state for node i with consideration of spatial correlations can be expressed as:

$$\bar{h}^{(i)} = Uh^{(i)} + \mathrm{ReLU}\left(\sum\nolimits_{j \in \mathcal{N}_i} \alpha^{(ij)} h'^{(j)}\right), \tag{8}$$

where $Uh^{(i)}$ denotes the shortcut path, and $U \in \mathbb{R}^{D'_h \times D_h}$ is a trainable matrix projecting $h^{(i)}$ to $\mathbb{R}^{D'_h}$.

Since we extract the meta knowledge from the features of nodes and edges, as well as the dynamic traffic context from the input traffic states, and then use both information to generate the weights of graph attention network, Meta-GAT$^+$ can model the inherent relationships between diverse spatial correlations and geo-graph attributes along with dynamic traffic states.

3.3 Meta Recurrent Neural Network$^+$

Conventionally, RNN layers are employed to model the temporal correlations of urban traffic. However, as temporal correlations of urban traffic vary from node to node and from time to time, a simple shared RNN is not sufficient to simultaneously capture diverse temporal correlations for all nodes and all timestamps at once. To model such diversity, we adopt the similar idea of Meta-GAT$^+$, which generates the weights of RNN from the node embeddings learned from node attributes (*e.g.*, POIs and RNs), and the dynamic traffic context learned from traffic states.

There are various types of RNN implementation for time series analysis. Among them, as gated recurrent unit (GRU) [18] is a simple but effective structure, we introduce GRU as a running example to illustrate Meta-RNN+. Formally, a GRU is defined as:

$$h_t = \text{GRU}\left(z_t, h_{t-1} \mid W_\Omega, U_\Omega, b_\Omega\right), \tag{9}$$

where $z_t \in \mathbb{R}^D$ and $h_t \in \mathbb{R}^{D'}$ are the input vector and the encoding state at timestamp t, respectively. $W_\Omega \in \mathbb{R}^{D' \times D}$ and $U_\Omega \in \mathbb{R}^{D' \times D'}$ are weight matrices. $b_\Omega \in \mathbb{R}$ are biases ($\Omega \in \{u, r, h\}$). GRU derives the vector representation of a hidden state, which is expressed as:

$$\begin{aligned} u =& \sigma\left(W_u z_t + U_u h_{t-1} + b_u\right), \\ r =& \sigma\left(W_r z_t + U_r h_{t-1} + b_r\right), \\ h' =& \phi\left(W_h z_t + U_h\left(r \otimes h_{t-1}\right) + b_h\right) \\ h_t =& u \otimes h_{t-1} + (1-u) \otimes h', \end{aligned} \tag{10}$$

where $\otimes$ is Hadamard product, $\sigma(\cdot)$ is sigmoid function, and $\phi(\cdot)$ is tanh function.

In urban traffic prediction, we collectively encode all nodes' traffic. As the temporal correlations are diverse from node to node and related to dynamic traffic states, we adopt the parameter generation technique within Meta-GRU$^+$ like Meta-GAT$^+$. Formally, we define Meta-GRU$^+$ as:

$$H_t = \text{Meta-GRU}^+\left(Z_t, H_{t-1}, \mathcal{V}\right), \tag{11}$$

where $Z_t = \left[z_t^{(1)}, \cdots, z_t^{(N_l)}\right]$ and $H_t = \left[h_t^{(1)}, \cdots, h_t^{(N_l)}\right]$ are the inputs and the hidden states at timestamp t, respectively, and $\mathcal{V} = \left[v^{(1)}, \cdots, v^{(N_l)}\right]$ is the node attributes.

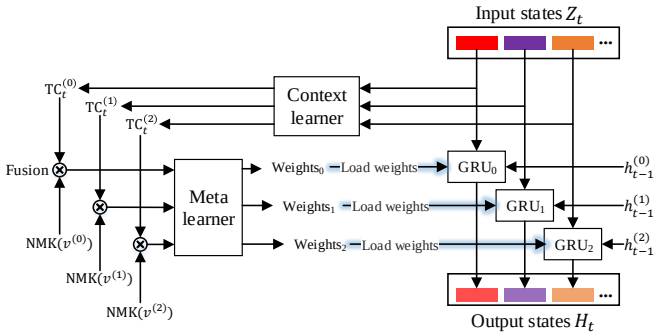


Fig. 7: Structure of Meta-GRU$^+$.

The structure of Meta-GRU$^+$ is shown in Figure 7. For any node i, we first obtain the dynamic traffic context by:

$$\text{TC}_t^{(i)} = \text{CL}_{\text{GRU}}\left(z_t^{(i)}\right), \tag{12}$$

where $\text{CL}_{\text{GRU}}(\cdot)$ is the context learner, a learnable FCN sharing parameters across all nodes. Next, we use the fusion gate to calculate the fusion information by the following equation:

$$\text{FI}_t^{(i)} = \phi\left(\text{NMK}\left(v^{(i)}\right)\right) \otimes \sigma\left(\text{TC}_t^{(i)}\right). \tag{13}$$

Finally, the output hidden states can be calculated by:

$$\begin{aligned} W_{t,\Omega}^{(i)} =& g_{W_\Omega}\left(\text{FI}_t^{(i)}\right), \\ U_{t,\Omega}^{(i)} =& g_{U_\Omega}\left(\text{FI}_t^{(i)}\right), \\ b_{t,\Omega}^{(i)} =& g_{b_\Omega}\left(\text{FI}_t^{(i)}\right), \\ h_t^{(i)} =& \text{GRU}\left(z_t^{(i)}, h_{t-1}^{(i)} \mid W_{t,\Omega}^{(i)}, U_{t,\Omega}^{(i)}, b_{t,\Omega}^{(i)}\right), \end{aligned} \tag{14}$$

where $W_{t,\Omega}^{(i)}$,$U_{t,\Omega}^{(i)}$ and $b_{t,\Omega}^{(i)}$ are the node-specific dynamic parameters generated from the fusion information by the meta learner, which consists of several FCNs g_{W_Ω}, g_{U_Ω}, g_{b_Ω} ($\Omega \in \{u, r, h\}$). As a result, all nodes have their individual and dynamic RNNs respectively, and the models represent the diverse temporal correlations related to node attributes and dynamic traffic states.

3.4 Optimization Algorithm

Suppose we employ a differentiable loss function $\mathcal{L}_{\text{train}}$ to measures the difference between the prediction values and the ground truth. Then, we can train ST-MetaNet$^+$ end-to-end by backpropagation like common neural networks. Specifically, there are two types of trainable parameters.

- The trainable parameters ω_1 in common neural networks, *e.g.*, the first fully connected layer and shortcut connection in Meta-GAT$^+$. The gradient of ω_1, denoted as $\nabla_{\omega_1}\mathcal{L}_{\text{train}}$, can be directly calculated by chain rule like a normal neural network.
- The trainable parameter ω_2 in the meta-knowledge learners, context learners, and meta learners, which generates parameters θ in the normal GATs and RNNs. The gradient of ω_2 can be calculated by chain rule, because all meta-knowledge learners, context learners, and meta learners are differentiable neural networks:

$$\nabla_{\omega_2}\mathcal{L}_{\text{train}} = \nabla_\theta \mathcal{L}_{\text{train}} \nabla_{\omega_2}\theta. \tag{15}$$

We employ a general sequence-to-sequence training process, which is similar to [1].

4 Experiments

In this section, we conduct experiments based on three real-world traffic prediction tasks to evaluate ST-MetaNet$^+$. In particular, we answer the following questions:

Q1. Can ST-MetaNet$^+$ outperform the state-of-the-art models in the traffic prediction tasks?

Q2. Do the meta learning components take effect? How large is the improvement of the meta learning method?

Q3. How do the settings of ST-MetaNet$^+$, such as the number of hidden units of each meta learned layer, impact the prediction results?

Q4. How about the stability and the convergence of ST-MetaNet$^+$ in the training phase?

Q5. Do the embeddings learned from geo-graph attributes by the meta knowledge learners reflect the properties of nodes (locations)?

4.1 Experimental Settings

4.1.1 Task Descriptions

We first introduce three traffic prediction tasks, and then illustrate the details of the datasets shown in Table 2.

TABLE 2: Details of the datasets.

Tasks	Flow prediction	Speed prediction	
Name	TAXI-BJ	METR-LA	PEMS-BAY
Region	Beijing	Los Angles	Bay Area
Prediction target	Flows	Speed	Speed
Start time	2/1/2015	3/1/2012	1/1/2017
End time	6/2/2015	6/30/2012	6/30/2017
Time interval	1 hour	5 minutes	5 minutes
# timestamps	3600	34272	52116
# nodes	1024	207	325
# edges	4114	3312	5200
# node features	989	18	18
# edge features	32	2	2

Taxi flow prediction

We partition Beijing city (lower-left GPS coordinates: 39.83°, 116.25°; upper-right GPS coordinates: 40.12°, 116.64°) into 32×32 grids, and adopt grid-based flow prediction task to evaluate our framework, where grids are regarded as nodes. The details of the dataset are as follows:

- *Taxi flow*. We obtain taxi flows from TDrive dataset [19], which contains a large number of taxicab trajectories from Feb. 1st 2015 to Jun. 2nd 2015. For each grid, we extract the hourly inflows and outflows from these trajectories by counting the number of taxis entering or exiting the grid.
- *Geo-graph attributes*. We obtain geo-graph attributes from POIs and RNs in Beijing city. Specifically, we have 982,829 POIs that belong to 668 categories, and 690,242 roads with 8 attributes, including length, width, the number of lanes, etc. The node attributes of a grid consist of many features of POIs and RNs within it, including the number of POIs in each category, the number of roads and lanes, etc. The edge attributes are the features of roads connecting pairs of grids, such as the number of roads and lanes.

In this task, we use the previous 12-hour flows to predict the next 3-hour flows. We split the traffic data along the time axis into non-overlapping training, evaluation, and test data, by the ratio of 8:1:1.

Traffic Speed prediction

The second and third tasks are traffic speed predictions. In these two tasks, we predict the traffic speed on road networks. The details of the datasets are as follows:

- *Traffic speed*. We adopt two real-world datasets to evaluate traffic speed prediction: 1) **METR-LA** [20], which contains traffic speed readings of 207 sensors in the highway of Los Angeles County; and 2) **PEMS-BAY**, which contains traffic speed readings of 325 sensors collected by California Transportation Agencies Performance Measurement System (PeMS). The readings of both datasets are aggregated into 5-minute sliding windows, and then released by [3].
- *Geo-graph attributes*. In traffic speed prediction tasks, we do not have any POI information. Thus, we only make use of GPS locations and road networks as the features of geo-graph. The node attributes consist of GPS points of nodes, and road structure information for each node, *i.e.,* a vector reflecting the road distance between the node and its k-nearest neighbors. The edge attribute is simply defined as the road distance between nodes. For the efficiency of model training and testing, we only keep the edges between each node and its k-nearest neighbors. Since the traffic correlations are directional on road networks [3], we collect node attributes and edges on both directions.

In these two tasks, we set $k = 8$, and use the historical 60-minute traffic speed to predict the traffic speed over the next 60 minutes. We partition the traffic speed dataset along the time axis into non-overlapping training, evaluation, and test data, by the ratio of 7:1:2. Our settings are exactly the same as the experiment in [3].

4.1.2 Metrics

We use Mean Absolute Error (MAE) and Rooted Mean Square Error (RMSE) to evaluate the models involved:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \qquad \text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

where n is the number of instances, $\hat{y}_i$ is the prediction result and y_i represents the ground truth.

4.1.3 Baselines

We compare ST-MetaNet$^+$ with the following baselines:

- **HA**. Historical Average. Urban traffic is modeled as the seasonal process, whose period is one day. We take the average of the previous seasons as the prediction result.
- **ARIMA**. Autoregressive Integrated Moving Average is a widely-used model for time series prediction, which combines moving average and autoregression. In the experiments, we train an individual ARIMA model for each node, and predict the future readings separately.
- **GBRT**. Gradient Boosting Regression Tree is a non-parametric statistical learning method for regression problem. For each future step (*e.g.,* next 1 hour or next 2 hour), we train a single GBRT, and predict the urban traffic, where the input consists of previous traffic information and node attributes.
- **Seq2Seq** [14]. We implement a sequence-to-sequence network with two stacking GRU layers for urban traffic prediction. The features of nodes, *i.e.,* node attributes, are firstly embedded by an FCN, and then fused with the outputs of the decoder. Finally, the fused vectors are linearly projected into the prediction results. All nodes share a model with the same parameter values.
- **GAT-Seq2Seq**. We combine graph attention networks and sequence-to-sequence architecture to model spatial and temporal correlations, respectively. It applies a similar structure as ST-MetaNet$^+$, which consists of two GAT layers and two GRU layers. Similar to Seq2Seq, the node attributes are firstly embedded by an FCN and then fused with the outputs of the decoder. Finally, the output vectors are linearly projected into the prediction results.
- **ST-ResNet** [4]. The model is widely used in grid-based flow prediction task. It models the spatio-temporal correlations by stacked residual units. To make the comparison fair, we only use the same input timestamps in our model, *i.e.,* keeping only the closeness part in ST-ResNet. Finally, we fuse the outputs with node attributes, *i.e.,* features of grids, and then make predictions by linear projection.

- **STDN** [21]. The model is used in grid-based flow prediction. It employs CNNs to capture spatial correlations, LSTMs to capture temporal correlations, and a periodically shifted attention mechanism to model long-term periodic temporal shifting. We also fuse the outputs with node attributes, and make predictions by linear projection.
- **DCRNN** [3]. It employs diffusion convolution operators within sequence-to-sequence architecture to capture spatial and temporal correlations, respectively. Besides, we embed the node attributes, and add them into the input traffic vectors as the additional input features.
- **Graph WaveNet** [9]. It employs WaveNet and graph convolution to capture temporal and spatial correlations, respectively. The authors also proposed a self-adaptive adjacency matrix to automatically uncover unseen graph structures from data without the guidance of any prior knowledge. Like DCRNN, we also add embeddings of node attributes into the input traffic vectors as the additional features.
- **ST-MetaNet** [1]. It is the prior version of this work. The main difference between them is that ST-MetaNet does not model the relationships between ST correlations and dynamic traffic states within Meta-GATs and Meta-GRUs.

For all neural network baselines, we conduct grid search on the number of hidden units in each layer, and select the best models according to the validation results. Besides, all neural networks, including our ST-MetaNet$^+$, use the same network structure to embed geo-graph attributes, *i.e.,* a two-layer FCN with $[32, 32]$ hidden units.

4.1.4 Framework Settings and Training Details

The settings of ST-MetaNet$^+$ contain three parts:

- The structures of NMK-Learner and EMK-Learner. We simply employ two FCNs (2 layers with the same number of hidden units) as NMK-Learner and EMK-Learner respectively, to learn the meta knowledge of nodes and edges. We conduct grid search on the number of hidden units over $\{8, 16, 32, 64\}$.
- The dimension of hidden states in sequence-to-sequence architecture. For simplicity, we use the same number of hidden units in all components (Meta-GAT$^+$, Meta-GRU$^+$) within the encoder and decoder, and conduct grid search on this number over $\{16, 32, 64\}$.
- Weight generation of meta learners. For each generated parameters in Meta-GAT$^+$ and Meta-GRU$^+$, *i.e.,* $W^{(ij)}$, $b^{(ij)}$, $W^{(i)}_{t,\Omega}$, $U^{(i)}_{t,\Omega}$, and $b^{(i)}_{t,\Omega}$, we simply build an FCN with hidden units $[d_g, n]$ to generate parameter weights from the meta knowledge, where n is the number of parameters in the target. We search on d_g over $\{1, 2, 4, 8\}$.

ST-MetaNet$^+$ is trained by Adam optimizer. The batch size is set as 32. We train the framework for 1000 iterations by random sampling in every epoch. The initial learning rate is 1e-2, and it is divided by 10 every 10 epochs. We also apply gradient clipping where the maximum gradient norm is set as 5. To tackle the discrepancy between training and inference in sequence-to-sequence architecture, we employ inverse sigmoid decay for scheduled sampling [22]:

$$\epsilon_i = \frac{r}{r + \exp(i/r)},$$

where r is a constant and set as 2000.

4.2 Performance Comparison (Q1)

The performance of the competitive baselines and ST-MetaNet$^+$ are shown in Table 3, Table 4, and Table 5. In addition, we also list the trainable parameters involved in deep models to show the model complexity. All models are trained and tested for five times, and the results are presented in the format: "mean $\pm$ standard deviation".

In all three tasks, the statistical models, *i.e.,* HA and ARIMA, turn out to be the worst models as they only consider the statistical features of the input data. While GBRT, which is a popular non-parametric model, has a relatively better performance. However, it does not learn any high-level temporal or spatial features. Accordingly, it still has large predicting errors.

For deep learning models, Seq2Seq is an encoder-decoder based model, capable of effectively capturing temporal correlations. However, spatial correlations are ignored in this case, resulting in low accuracy in these tasks. GAT-Seq2Seq further employs graph attention to handle spatial correlations, which upgrades the predictive performance. However, GAT-Seq2Seq still has a large margin for improvement because it does not consider the relationships between ST correlations and geographical information along with dynamic traffic states. In TAXI-BJ dataset, we also have two additional baselines, *i.e.,* ST-ResNet and STDN, to make grid-based taxi flow predictions. As shown in Table 3, ST-ResNet and STDN have unstable and very low prediction accuracy. The reason is that TAXI-BJ dataset has 1024 grids and the functions of grids are very different, but both models use the same parameters to make predictions for all grids (*e.g.,* the same convolutional kernel and the same LSTM model). Thus, they cannot effectively capture such discrepancy of ST correlations among grids.

DCRNN and Graph WaveNet are two powerful baselines using road distance to generate the adjacent matrix of graph convolution, such that the spatial correlations can be modeled. However, the graph convolution based method needs prior knowledge, *e.g.,* the function of road distance, to construct the graph. Such an assumption is relatively weak, for example, we cannot fully make use of some other important features like POIs and road network structures. As a result, in TAXI-BJ dataset, ST-MetaNet$^+$ outperforms DCRNN and Graph WaveNet by at least 3.5% MAE and 5.4% RMSE respectively. In METR-LA and PEMS-BAY datasets, our ST-MetaNet$^+$ still shows competitive results. However, as these two datasets do not have many geographical information (only GPS locations and road distance are provided), the improvement over baselines is less.

Next, we compare ST-MetaNet$^+$ with its prior version ST-MetaNet. ST-MetaNet$^+$ significantly outperforms its basic version in all three tasks, as it further tackles the inherent relationships between ST correlations and dynamic traffic states. Thus, by making the meta learners generate different parameter values at different timestamps, ST-MetaNet$^+$ can boost the expressiveness of the spatial and temporal networks, namely GAT and GRU, in advance.

Finally, we discuss the model complexity. In Meta-GAT$^+$ and Meta-GRU$^+$, we generate the parameter weights defined in GAT and GRU, which would intuitively introduce much more trainable parameters. However, as the number

TABLE 3: Predictive performance on TAXI-BJ dataset.

Models [# params]		Overall		1 hour		2 hour		3 hour	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA		26.2±0.00	56.5±0.00	26.2±0.00	56.5±0.00	26.2±0.00	56.5±0.00	26.2±0.00	56.5±0.00
ARIMA		40.0±0.00	86.8±0.00	27.1±0.00	58.3±0.00	41.2±0.00	77.0±0.00	51.8±0.00	108.0±0.00
GBRT		28.8±0.04	60.9±0.15	22.3±0.01	47.7±0.01	29.9±0.06	62.7± 0.14	34.3±0.11	70.5± 0.23
Seq2Seq	[333k]	21.3±0.06	42.6±0.14	17.8±0.05	35.1±0.07	22.0±0.06	43.6±0.16	24.2±0.09	48.1±0.20
GAT-Seq2Seq	[407k]	18.3±0.13	35.6±0.23	16.3±0.12	31.9±0.21	18.7±0.12	36.3±0.20	19.9±0.14	38.4±0.30
ST-ResNet	[445k]	18.7±0.53	36.1±0.59	16.8±0.50	31.9±0.69	18.9±0.57	36.4±0.71	20.3±0.52	39.5±0.46
STDN	[198k]	23.4±2.49	43.2±2.95	21.5±3.24	37.4±3.34	24.7±3.43	44.4±3.71	24.1±1.27	46.9±3.11
DCRNN	[405k]	17.8±0.13	36.1±0.15	15.8±0.05	32.3±0.08	18.2±0.15	36.9±0.17	19.4±0.19	38.9±0.24
Graph WaveNet	[996k]	17.1±0.06	35.0±0.14	15.2±0.08	31.1±0.27	17.4±0.09	35.7±0.30	18.6±0.11	37.8±0.25
ST-MetaNet	[129k]	16.7±0.13	33.6±0.15	14.8±0.05	29.6±0.08	17.1±0.15	34.3±0.17	18.2±0.19	36.5±0.24
ST-MetaNet$^+$	[166k]	16.5±0.16	33.2±0.35	14.7±0.18	29.7±0.40	16.9±0.16	33.9±0.35	17.8±0.17	35.8±0.53

TABLE 4: Predictive performance on METR-LA dataset.

Models [# params]		Overall		15 min		30 min		60 min	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA		4.79±0.00	8.72±0.00	4.79±0.00	8.72±0.00	4.79±0.00	8.72±0.00	4.79±0.00	8.72±0.00
ARIMA		4.03±0.00	7.94±0.00	3.27±0.00	6.14±0.00	3.99±0.00	7.78±0.00	5.18±0.00	10.10±0.00
GBRT		3.86±0.01	7.49±0.01	3.16±0.00	6.05±0.00	3.85±0.00	7.50±0.00	4.86±0.01	9.10±0.02
Seq2Seq	[81k]	3.55±0.01	7.27±0.01	2.98±0.01	5.88±0.01	3.57±0.01	7.26±0.01	4.38±0.01	8.88±0.02
GAT-Seq2Seq	[113k]	3.28±0.00	6.66±0.01	2.83±0.01	5.47±0.01	3.31±0.00	6.68±0.00	3.93±0.01	8.03±0.02
DCRNN	[372k]	3.04±0.01	6.27±0.03	2.67±0.00	5.18±0.01	3.08±0.01	6.31±0.03	3.56±0.01	7.53±0.04
Graph WaveNet	[297k]	3.05±0.01	6.16±0.03	2.70±0.01	5.16±0.01	3.08±0.01	6.20±0.03	3.55±0.12	7.35±0.05
ST-MetaNet	[124k]	3.05±0.04	6.22±0.06	2.68±0.02	5.15±0.05	3.03±0.10	6.25±0.05	3.49±0.12	7.47±0.08
ST-MetaNet$^+$	[162k]	3.00±0.01	6.16±0.02	2.65±0.01	5.11±0.01	3.04±0.01	6.16±0.02	3.48±0.02	7.37±0.04

TABLE 5: Predictive performance on PEMS-BAY dataset.

Models [# params]		Overall		15 min		30 min		60 min	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA		3.84±0.00	7.16±0.00	3.84±0.00	7.16±0.00	3.84±0.00	7.16±0.00	3.84±0.00	7.16±0.00
ARIMA		2.19±0.00	5.05±0.00	1.62±0.00	3.30±0.00	2.19±0.00	4.86±0.00	3.05±0.00	6.90±0.00
GBRT		1.96±0.02	4.48±0.00	1.49±0.01	3.21±0.00	1.99± 0.02	4.50±0.01	2.61±0.04	5.76±0.02
Seq2Seq	[81k]	1.77±0.00	4.18±0.01	1.38±0.00	2.99±0.01	1.81±0.01	4.2±0.01	2.31±0.01	5.36±0.01
GAT-Seq2Seq	[113k]	1.74±0.00	4.08±0.01	1.38±0.01	2.94±0.01	1.79±0.00	4.1±0.01	2.26±0.01	5.22±0.04
DCRNN	[372k]	1.59±0.00	3.70±0.02	1.31±0.00	2.76±0.01	1.65±0.01	3.78±0.02	1.97±0.00	4.60±0.02
Graph WaveNet	[297k]	1.59±0.01	3.66±0.04	1.31±0.01	2.75±0.01	1.65±0.01	3.73±0.04	1.98±0.03	4.56±0.06
ST-MetaNet	[124k]	1.71±0.04	3.96±0.11	1.36±0.01	2.88±0.04	1.77±0.06	4.00±0.08	2.19±0.08	5.03±0.20
ST-MetaNet$^+$	[162k]	1.60±0.01	3.72±0.02	1.31±0.00	2.78±0.01	1.66±0.01	3.81±0.01	1.99±0.01	4.62±0.04

of parameters shown in Table 3, Table 4, and Table 5, the parameters of ST-MetaNet$^+$ is acceptable, compared with state-of-the-art models, *i.e.*, ST-ResNet, STDN, DCRNN, and Graph WaveNet. This fact is related to the good expressiveness of ST-MetaNet$^+$ that small dimensional hidden states in Meta-GAT$^+$ and Meta-GRU$^+$ can already have good representation of urban traffic states, which verifies the advantage of meta learning in modeling the relationships between ST correlations and geographical information along with dynamic traffic states.

4.3 Ablation Studies on Meta Learning (Q2)

To illustrate the effectiveness of meta learning, we conduct ablation studies of ST-MetaNet$^+$ on these three datasets. In each dataset, we compare the prediction accuracy based on two settings: 1) we set RNN components as Meta-GRU$^+$, and test the prediction results under different choices of graph neural network (GNN), *i.e.*, normal GAT, Meta-GAT (no dynamic traffic context), and Meta-GAT$^+$; and 2) we set GNN components as Meta-GAT$^+$, and test the prediction results under various choices of RNN, *i.e.*, normal GRU, Meta-GRU (no dynamic traffic context), and Meta-GRU$^+$.

The comparison results of TAXI-BJ dataset are shown in Figure 8. Notice that the basic meta learning method used in Meta-GAT and Meta-GRU significantly improve the prediction accuracy of GAT and GRU respectively. The reason is that TAXI-BJ dataset has large amounts of geograph attributes, enabling the meta knowledge learners to learn meaningful embeddings and successfully build the relationships between such embeddings and diverse ST correlations. In addition, Meta-GAT$^+$ and Meta-GRU$^+$ take dynamic traffic states into account, and further improve the performance. This fact verifies that modeling the relationships between dynamic traffic states and diverse ST correlations is necessary.

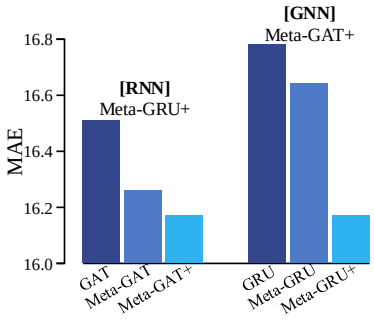
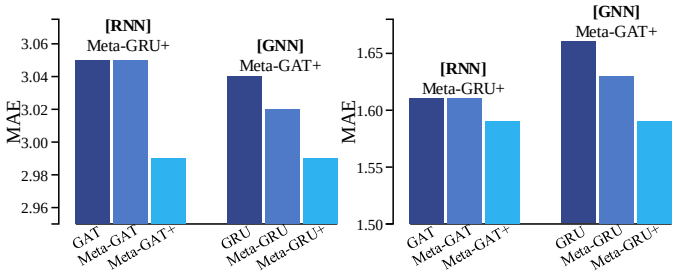


Fig. 8: Ablation studies on taxi flow prediction.

The prediction results of METR-LA and PEMS-BAY datasets are shown in Figure 9. Different from the results of TAXI-BJ dataset, the improvement of Meta-GAT over GAT is much less. The reason is that these two datasets do not have plenty of edge attributes (only road distance is provided), which does not reveal all kinds of characteristics related to diverse spatial correlations. Instead, Meta-GRU still significantly outperforms GRU, as the road structures around each node are considered, leveraging different temporal models for nodes to capture diverse temporal correlations. Notice that Meta-GAT$^+$ and Meta-GRU$^+$ have a large improvement over Meta-GAT and Meta-GRU, respectively, which demonstrates the effectiveness of modeling traffic context in meta learning.



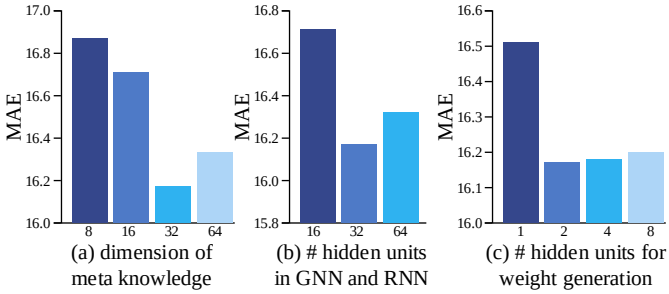
(a) METR-LA dataset (b) PEMS-BAY dataset

Fig. 9: Ablation studies on traffic speed prediction.

4.4 Evaluation on Framework Settings (Q3)

ST-MetaNet$^+$ has many settings, including the dimension of meta knowledge (outputs of NMK-Learner and EMK-Learner), the number of hidden units within RNN and GAT, and the number of hidden units for weight generation. To investigate the robustness of ST-MetaNet$^+$, for each setting, we fix other parameters and we present the results under different parameter choices of that setting.

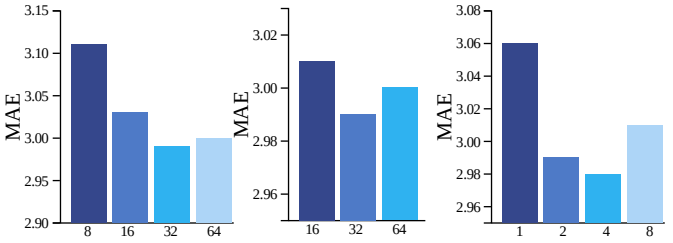
First, as shown in Figure 10(a), Figure 11(a), and Figure 12(a), increasing the value of meta knowledge dimension enhances the performance significantly in three datasets. As the dimension of meta knowledge does not impact the number of parameters in the generated RNNs and GATs, this fact illustrates that the meta knowledge learned from geo-graph attributes essentially takes effect.



(a) dimension of meta knowledge (b) # hidden units in GNN and RNN (c) # hidden units for weight generation

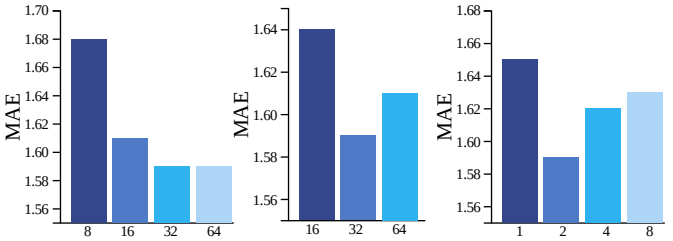
Fig. 10: Evaluation of parameter settings on TAXI-BJ.

Next, Figure 10(b), Figure 11(b), and Figure 12(b) show that increasing the number of hidden units in the generated GATs and RNNs can lower the MAE before overfitting. Note that ST-MetaNet$^+$ uses only 32 hidden units in each layer to achieve better performance than GAT-Seq2Seq (it uses the same network structure but no meta learning), that uses much more hidden units. Thus, ST-MetaNet$^+$ have a more compacted hidden representation for traffic states.



(a) dimension of meta knowledge (b) # hidden units in GNN and RNN (c) # hidden units for weight generation

Fig. 11: Evaluation of parameter settings on METR-LA.



(a) dimension of meta knowledge (b) # hidden units in GNN and RNN (c) # hidden units for weight generation

Fig. 12: Evaluation of parameter settings on PEMS-BAY.

Finally, we discuss the prediction results under different numbers of hidden units for weight generation in GAT and GRU. This parameter represents the rank of the parameters for all nodes or edges, and is very related to the number of trainable parameters. As shown in Figure 10(c), Figure 11(c), and Figure 12(c), with only 2 hidden units for weight generation, ST-MetaNet$^+$ can achieve good accuracy, and the performance is not very sensitive to the number of hidden units for weight generation when this number is further increased. This fact illustrates that though different nodes or edges have different sets of parameters, they can have a low-rank representation by several hidden units, showing similarity among nodes or edges.

4.5 Convergence Discussion (Q4)

In this subsection, we present the framework convergence during the training process, and compare ST-MetaNet$^+$ with two variants, *i.e.*, GAT-Seq2Seq and ST-MetaNet, which use the same sequence-to-sequence and graph-attention based architecture, as well as the same training strategies, including the learning rate settings, the scheduled sampling, etc.

As shown in Figure 13, there are three curves in each chart, presenting the validation losses of GAT-Seq2Seq, ST-MetaNet, and ST-MetaNet$^+$ in the training procedure. In the beginning 10,000 iterations, all models' loss values fluctuate due to the scheduled sampling strategy, that the decoder has a large probability to use ground truth as the inputs, instead of the previous decoding values. Consequently, it causes a discrepancy between the training and testing procedure. After that, with the decreasing of such probability, the validation curves tend to be stable and converged.

Comparing with GAT-Seq2Seq, ST-MetaNet$^+$ converges to a much lower MAE with faster speed in all datasets, showing the effectiveness of meta learning method, that can quickly learn genetic information of ST models. Moreover, ST-MetaNet$^+$ has much lower convergence loss values than

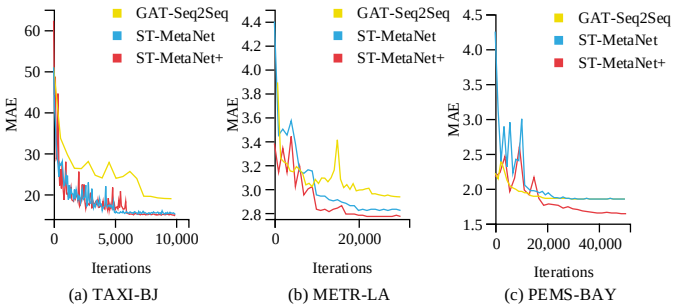


Fig. 13: Framework convergence on validation datasets.

ST-MetaNet, showing the advantage of modeling the relationships between ST correlations and dynamic traffic states.

4.6 Evaluation on Meta Knowledge (Q5)

A good meta knowledge learner should obtain the representation of geographical information that can reflect traffic similarity for nodes. To validate the effectiveness of such representation, for each node we firstly find its k-nearest neighbors in the node embedding space of geographical information, and then evaluate the similarity of traffic sequences between the node and its neighbors. We employ Pearson correlations and the first order temporal correlations [23], denoted as CORR and CORT respectively, to measure the similarity between two traffic sequences. The similarity functions can be expressed as:

$$\mathrm{CORR}(\mathbf{x},\mathbf{y}) = \frac{\sum_i (x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_i (x_i-\bar{x})^2}\sqrt{\sum_i (y_i-\bar{y})^2}},$$

$$\mathrm{CORT}(\mathbf{x},\mathbf{y}) = \frac{\sum_i (x_i-x_{i-1})(y_i-y_{i-1})}{\sqrt{\sum_i (x_i-x_{i-1})^2}\sqrt{\sum_i (y_i-y_{i-1})^2}},$$

where $\mathbf{x},\mathbf{y}$ are two temporal sequences, and $\bar{x},\bar{y}$ are their mean values. Note that the larger the criteria are, the more similar the two sequences are, and both criteria lie in $[-1,1]$.

TABLE 6: Evaluation on traffic similarity between each node and its k-nearest neighbors in the embedding space ($k=8$).

Metric	Model	TAXI-BJ	METR-LA	PEMS-BAY
CORR	GAT-Seq2Seq	0.62	0.7	0.38
	ST-MetaNet	**0.71**	**0.73**	0.38
	ST-MetaNet$^+$	0.7	0.72	**0.4**
CORT	GAT-Seq2Seq	0.39	0.48	0.14
	ST-MetaNet	**0.48**	**0.51**	0.16
	ST-MetaNet$^+$	0.46	0.5	**0.18**

We choose $k=8$ nearest neighbors for each node and calculate traffic similarity on the test dataset between each node and its neighbors. We compare the meta learning based frameworks with GAT-Seq2Seq, which uses the same sequence-to-sequence and graph attention architecture but adopts the data fusion strategy to incorporate geographical and traffic information. As shown in Table 6, the node embeddings of ST-MetaNet$^+$ and ST-MetaNet in the taxi flow prediction task shows significant improvement over embeddings of GAT-Seq2Seq, which implies that the meta learning method learns a better geographical representation that reflects traffic-related characteristics of nodes. While in the traffic prediction tasks, the improvement is less. The reason is that we have much less geographical information in METR-LA and PEMS-BAY datasets. Nonetheless, the result still shows that the meta learning based frameworks can effectively learn better traffic-related representations.

4.7 Case Study

Similarity of Geographical Information

We further show the property of embeddings learned from node attributes by ST-MetaNet$^+$ through a case study. Intuitively, a good embedding space should have the characteristic that nearby grids have similar traffic sequences. Thus, we compare ST-MetaNet$^+$ with GAT-Seq2Seq by the taxi inflows of three representative grids in Beijing: Zhongguancun (business district), Huilongyuan (residential district), and Sihui Bridge (viaduct). We present these grids on Bing Maps[1], and show their inflow trends, as shown in Figure 14.

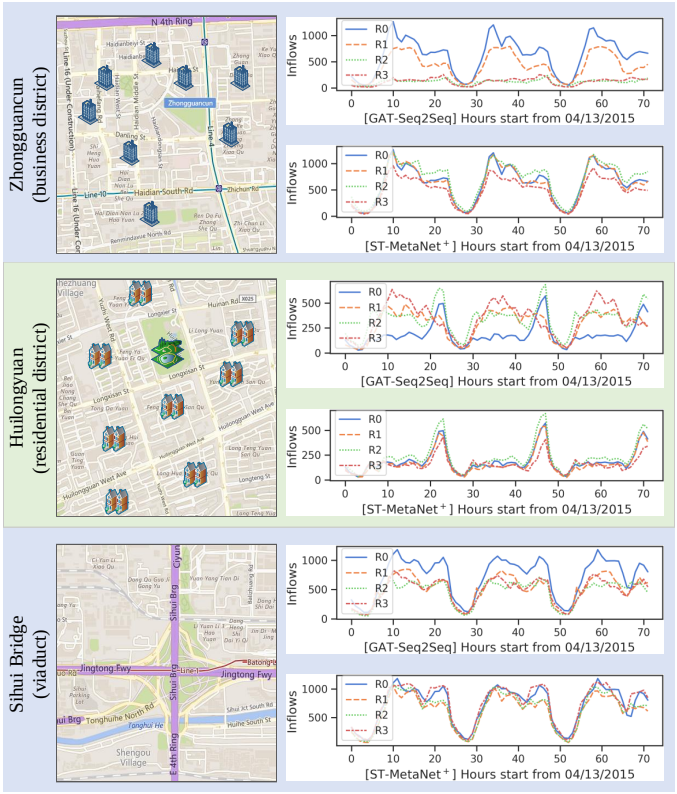
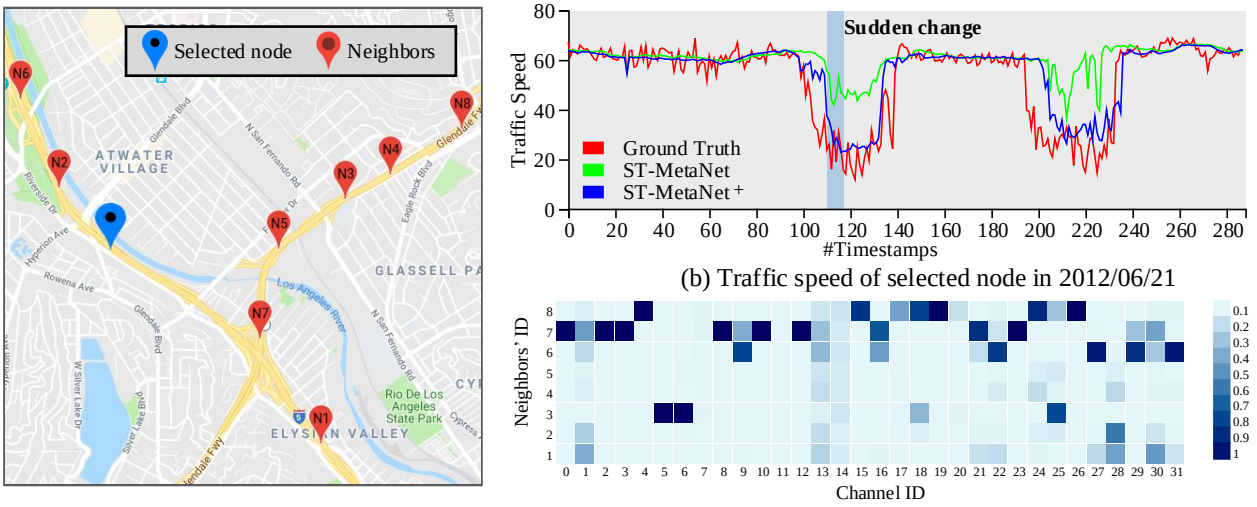


Fig. 14: The inflows of the representative grids. The left maps show the selected grids with special functions, *i.e.*, business district, residential district, and viaduct. The right charts show the inflow trends of the selected grids $R0$ and its k-nearest neighbors Rk ($k>0$) in the embedding space produced by GAT-Seq2Seq and ST-MetaNet$^+$.

The selected grids' inflows of GAT-Seq2Seq are distinct from the flows of their neighborhoods in the embedding space. While ST-MetaNet$^+$ obtains an embedding space that nearby grids have very similar flows. Specifically, in this embedding space, the neighbor grids of Zhongguancun (business district) have inflow rush hours in the morning; the neighbor grids of Huilongyuan (residential district) have inflow rush hours in the evening; while the neighbor grids

1. Bing Maps: https://cn.bing.com/maps



(a) Map of selected node and its neighbors

(b) Traffic speed of selected node in 2012/06/21

(c) Heatmap of graph attention scores when sudden change occurs

Fig. 15: Case study for a sudden change of traffic speed on METR-LA dataset.

of Sihui Bridge (viaduct) have two inflow rush hours in the morning and evening. This case demonstrates that ST-MetaNet$^+$ learns a reasonable representation of nodes, and captures the inherent relationships between geographical information and ST correlations of urban traffic.

Effectiveness of Traffic Context

Intuitively, traffic context can give the model more information about the dynamic impact on ST correlations than geographical information. Thus, ST-MetaNet$^+$ should have better predictions than ST-MetaNet when traffic meets a sudden change, such as the traffic condition at peak hour.

To show how traffic context impacts traffic prediction, we select the traffic of a node in METR-LA dataset as an example, and discuss the improvement of ST-MetaNet$^+$ over ST-MetaNet. We plot the node (blue pin) and its neighbors (red pins) on Google Maps[2] in Figure 15(a), and the predicting traffic speed (next 30 minutes) of the selected node in Figure 15(b). In this case, when traffic meets a sudden change, *i.e.*, at a peak hour, ST-MetaNet$^+$ successfully predicts the severe deceleration of traffic (traffic jam), while ST-MetaNet fails, as shown in Figure 15(b). The reason is that in off-peak hours, the traffic has high speed and has analogous types of ST correlations, however, when traffic starts changing at peak hour, the type of ST correlations is also changed. Since ST-MetaNet does not consider the relationships between ST correlations and such dynamic traffic states, it cannot give a good prediction in predicting such sudden change.

Next, we want to further illustrate how dynamic traffic state impacts spatial correlations. As traffic jam usually starts from ramp road (exits or bridges), the traffic speed near ramp road should have a large impact than the traffic speed on common roads when traffic jam occurs. To verify it, we also present the heatmap of graph attention scores for each hidden state's channel at peak hour in Figure 15(c), where the x-axis stands for the channel IDs of the hidden state, and the y-axis denotes the neighbors' IDs of the selected node. Notice that N6, N7, and N8 are near exits or bridges in Figure 15(a), and they have the largest attention scores in Figure 15(c), showing biggest impacts to the selected node. This fact verifies our assumptions, and demonstrates the effectiveness of modeling the relationships between ST correlations and dynamic traffic states.

2. Google Maps: https://www.google.com/maps

5 Related Work

Urban Traffic Prediction

There are some previously published works on predicting an individual's movement based on their location history [24], [25]. They mainly forecast millions of individuals' mobility traces rather than the aggregated traffic flows in a region. Some other researchers aim to predict travel speed or traffic volumes on single or multiple road segments [26], [27], rather than citywide ones. Recently, researchers have started to focus on city-scale traffic prediction. In the beginning, some researchers proposed non-deep models [28], [29] to predict traffic. With the development of deep learning, [4], [21], [30], [31], [32], [33] proposed to predict traffic on regular urban grids by convolution neural network (CNN) and recurrent neural network (RNN) based models, such that the high-level ST correlations can be captured effectively. In addition, [3], [6], [9], [10] employed graph convolution components in neural networks to predict urban traffic on non-grid spatial structure, *e.g.*, road networks.

Being different from all above works, we aim to model the diverse ST correlations in urban traffic. To the best of our knowledge, we are the first to study the inherent relationships between ST correlations and geographical information along with dynamic traffic states.

Deep Learning for Spatio-Temporal Modeling

Deep learning has powered many applications in spatio-temporal areas. In specific, the architectures of CNNs were widely used in modeling grid data, *e.g.*, taxi demand inference [32] and precipitation nowcasting [34]. Besides, RNNs [18] became popular due to their success in modeling temporal sequence, however, separately modeling sequences by RNNs discards the unique characteristics of spatio-temporal data, *i.e.*, spatial correlations. To tackle this issue, several studies were proposed, such as video prediction [35] and travel time estimation [36]. Very recent studies [8], [37] have

indicated that attention mechanism can enable RNNs to capture dynamic ST correlations in geo-sensory data.

By contrast, our work introduces a new perspective, that the ST models can be generated by related meta knowledge. Particularly, our meta learning framework can be also applied on the above deep ST models, *e.g.*, by generating weights of CNNs, RNNs, and attention networks.

Deep Meta Learning

The most related deep meta learning method is network weight generation. [38] firstly proposed to predict network parameters for modeling temporal data. [39] used a network called *learnet* to predict the parameters of a pupil network for few-shot learning. [40] employed hypernetworks to generate network weights, which can be regarded as weight sharing across layers. [41] proposed meta multi-task learning for NLP tasks, which also employ weight generation method to learn task-specific semantic functions by a meta network. [42] proposed to embed neural architecture and adopt hypernetworks to generate its weights, to amortize the cost of neural architecture search.

There are also other types of meta learning methods as well as related studies on graph structure. [43] proposed to use parametric functions to update network parameters. [44] proposed to maintain meta-gradients for fast model adaption. [45], [46], [47] proposed to build graphs for describing relationships between tasks or data samples, and then propagate information on graphs for few shot learning.

Our work is distinct from all above methods, as it aims to tackle a different category of problem, *i.e.*, modeling correlations on ST graphs which depends on static attributes and dynamic states.

6 Conclusion and Future Work

We propose a novel deep meta learning framework, entitled ST-MetaNet$^+$, for spatio-temporal data with applications to urban traffic prediction, capable of learning traffic-related embeddings of nodes and edges from geo-graph attributes and traffic context from dynamic traffic states, so as to model diverse spatial and temporal correlations, respectively. We evaluate ST-MetaNet$^+$ over three real-world tasks. Compared with state-of-the-art baselines, our model demonstrates very competitive performance. We visualize the similarity of meta-knowledge learned from geographical information, and the impact of dynamic traffic states, to show the interpretation of ST-MetaNet$^+$. In the future, we will extend our framework to a much broader set of urban ST prediction tasks and explore the usage of such representation learned from geographical attributes in other traffic-related tasks.

Acknowledgments

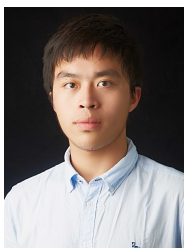
The work is supported by the National Key R&D Program of China (2019YFB2101805), APEX-MSRA Joint Research Program and the National Natural Science Foundation of China Grant (61672399, U1609217, 61773324, 61702327, 61772333, and 61632017).

References

[1] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proc. SIGKDD*, 2019.
[2] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proc. UBICOMP*, 2011.
[3] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. ICLR*, 2018.
[4] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction." in *Proc. AAAI*, 2017.
[5] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *ACM TIST*, 2014.
[6] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. IJCAI*, 2018.
[7] Z. Wang, K. Fu, and J. Ye, "Learning to estimate the travel time," in *Proc. SIGKDD*, 2018.
[8] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: Multi-level attention networks for geo-sensory time series prediction." in *Proc. IJCAI*, 2018.
[9] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proc. AAAI*, 2019.
[10] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI*, 2019.
[11] L. Zhao, Q. Sun, J. Ye, F. Chen, C.-T. Lu, and N. Ramakrishnan, "Multi-task learning for spatio-temporal event forecasting," in *Proc. SIGKDD*, 2015.
[12] Y. Liu, Y. Zheng, Y. Liang, S. Liu, and D. S. Rosenblum, "Urban water quality prediction based on multi-task multi-view learning," in *Proc. IJCAI*, 2016.
[13] J. Xu, P.-N. Tan, L. Luo, and J. Zhou, "Gspartan: a geospatio-temporal multi-task learning framework for multi-location prediction," in *Proc. SDM*, 2016.
[14] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014.
[15] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *Proc. ICLR*, 2018.
[16] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," in *ICML Deep Learning Workshop*, 2015.
[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.
[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.
[19] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: driving directions based on taxi trajectories," in *Proc. SIGSPATIAL*, 2010.
[20] H. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, "Big data and its technical challenges," *Communications of the ACM*, 2014.
[21] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proc. AAAI*, 2019.
[22] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. NIPS*, 2015.
[23] A. D. Chouakria and P. N. Nagabhushan, "Adaptive dissimilarity index for measuring time series proximity," *ADAC*, 2007.
[24] Z. Fan, X. Song, R. Shibasaki, and R. Adachi, "Citymomentum: an online approach for crowd behavior prediction at a citywide level," in *Proc. UBICOMP*, 2015.
[25] X. Song, Q. Zhang, Y. Sekimoto, and R. Shibasaki, "Prediction of human emergency behavior and their mobility following large-scale disaster," in *Proc. SIGKDD*, 2014.
[26] P.-T. Chen, F. Chen, and Z. Qian, "Road traffic congestion monitoring in social media with hinge-loss markov random fields," in *Proc. ICDM*, 2014.
[27] A. Abadi, T. Rajabioun, P. A. Ioannou *et al.*, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE TITS*, 2015.
[28] M. X. Hoang, Y. Zheng, and A. K. Singh, "Fccf: forecasting citywide crowd flows based on big data," in *Proc. SIGSPATIAL*, 2016.

[29] Y. Li, Y. Zheng, H. Zhang, and L. Chen, "Traffic prediction in a bike-sharing system," in *Proc. SIGSPATIAL*, 2015.
[30] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "Dnn-based prediction model for spatio-temporal data," in *Proc. SIGSPATIAL*, 2016.
[31] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting city-wide crowd flows using deep spatio-temporal residual networks," *AI Journal*, 2018.
[32] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, and J. Ye, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. AAAI*, 2018.
[33] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE TKDE*, 2019.
[34] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Proc. NIPS*, 2015.
[35] Y. Wang, M. Long, J. Wang, Z. Gao, and S. Y. Philip, "Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms," in *Proc. NIPS*, 2017.
[36] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, "When will you arrive? estimating travel time based on deep neural networks," in *Proc. AAAI*, 2018.
[37] W. Cheng, Y. Shen, Y. Zhu, and L. Huang, "A neural attention model for urban air quality inference: Learning the weights of monitoring stations." in *Proc. AAAI*, 2018.
[38] J. Schmidhuber, "Learning to control fast-weight memories: An alternative to dynamic recurrent networks," *Neural Computation*, vol. 4, no. 1, pp. 131–139, 1992.
[39] L. Bertinetto, J. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," in *Proc. NIPS*, 2016.
[40] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," in *Proc. ICLR*, 2017.
[41] J. Chen, X. Qiu, P. Liu, and X. Huang, "Meta multi-task learning for sequence modeling," in *Proc. AAAI*, 2018.
[42] C. Zhang, M. Ren, and R. Urtasun, "Graph hypernetworks for neural architecture search," in *Proc. ICLR*, 2019.
[43] Y. Bengio, S. Bengio, and J. Cloutier, "Learning a synaptic learning rule," in *Proc. IJCNN*, 1991.
[44] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML*, 2017.
[45] L. Liu, T. Zhou, G. Long, J. Jiang, and C. Zhang, "Learning to propagate for graph meta-learning," in *Proc. NeuralPS*, 2019.
[46] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *Proc. ICLR*, 2018.
[47] L. Liu, T. Zhou, G. Long, J. Jiang, L. Yao, and C. Zhang, "Prototype propagation networks (ppn) for weakly-supervised few-shot learning on category graph," in *Proc. IJCAI*, 2019.

Zheyi Pan is a computer science Ph.D. candidate in Apex Data & Knowledge Management Lab, Department of Computer Science, Shanghai Jiaotong University, supervised by Prof. Yong Yu. He received his B.E. degree from Zhiyuan College, Shanghai Jiao Tong University in 2015. His research interests include deep learning and data mining with a special focus on urban computing and spatio-temporal data.

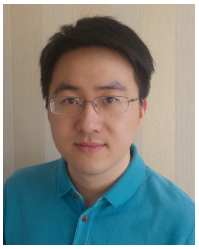


Wentao Zhang is an undergraduate research intern in Apex Data & Knowledge Management Lab, supervised by Prof. Yong Yu. He is pursing a bachelor degree in Zhiyuan College, Shanghai Jiao Tong University.

Yuxuan Liang is currently pursuing his Ph. D. degree at School of Computing, National University of Singapore. He has published several papers in refereed conferences, such as KDD, IJCAI, AAAI and GIS. His research interests mainly lie in machine learning, deep learning and their applications in urban areas.

Weinan Zhang is an assistant professor in Department of Computer Science, Shanghai Jiao Tong University. His research interests include machine learning and data mining, particularly, deep learning and reinforcement learning techniques for real-world data mining scenarios. He has published over 70 research papers and been serving as PC/SPC for conferences & journals including *KDD*, *SIGIR*, *ICML*, *ICLR*, *AAAI*, *WWW*, *WSDM*, *ICDM*, *JMLR* and *IPM* etc.

Yong Yu is a professor in Department of Computer Science in Shanghai Jiao Tong University. His research interests include information systems, web search, data mining, and machine learning. He has published over 200 papers and served as PC member of several conferences including *WWW*, *RecSys* and a dozen of other related conferences (*e.g., NIPS*, *ICML*, *SIGIR*, *ISWC* etc.) in these fields.



Junbo Zhang is a Senior Researcher of JD Intelligent Cities Research. He is leading AI Platform Department of Intelligent Cities Business Unit, JD Digits. His research interests include deep learning, data mining, artificial intelligence, big data analytics, and urban computing. He has published over 40 research papers (*e.g., AI Journal*, *IEEE TKDE*, *KDD*, *AAAI*, *IJCAI*, and *ACL*) in refereed journals and conferences. He currently serves as the Associate Editor of ACM Transactions on Intelligent Systems and Technology. He is a member of IEEE, ACM, CAAI (Chinese Association for Artificial Intelligence) and CCF (China Computer Federation), and a committee member of CCF-AI.

Yu Zheng is a Vice President and Chief Data Scientist at JD Digits, passionate about using big data and AI technology to tackle urban challenges. His research interests include big data analytic, spatio-temporal data mining, machine learning, and artificial intelligence. He also leads the JD Urban Computing Business Unit as the president and serves as the director of the JD Intelligent City Research. Before joining JD, he was a senior research manager at Microsoft Research. Zheng is also a Chair Professor at Shanghai Jiao Tong University, an Adjunct Professor at Hong Kong University of Science and Technology.