This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2023.3283520

CityTrans: Domain-Adversarial Training with Knowledge Transfer for Spatio-Temporal Prediction across Cities

Xiaocao Ouyang, Yan Yang, *Member, IEEE,* Wei Zhou, Yiling Zhang, Hao Wang, and Wei Huang

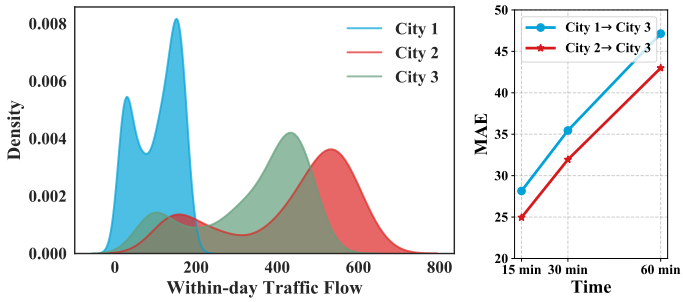
Abstract—As the spatio-temporal data of a city is not always available, insufficient data would lead to poor performance in some urban prediction tasks. Existing works utilize transfer learning to solve the data scarcity problem, but they ignore the differences in data distributions across cities, which leads to the ineffectiveness of knowledge transfer. In this paper, we propose a domain adversarial model with knowledge transfer for spatio-temporal prediction across cities, entitled *CityTrans*. Specifically, 1) the self-adaptive spatio-temporal knowledge (namely ST-Knowledge) is mined, to learn the latent spatial and temporal patterns among cities; 2) the domain-adversarial training strategy is introduced to enhance domain invariance; 3) a knowledge attention mechanism is proposed to extract the transferable information from the ST-Knowledge. Note that our CityTrans is an end-to-end domain adversarial spatio-temporal network without two-stage training (i.e., pre-training and fine-tuning). Finally, we conduct extensive experiments on two spatio-temporal prediction tasks: traffic (flow and speed) prediction, and air quality prediction. Experimental results demonstrate that CityTrans outperforms state-of-the-art models on all tasks by a significant margin.

Index Terms—Urban computing; Graph neural network; Transfer learning; Spatio-temporal prediction.

1 Introduction

The construction of smart cities has substantially improved the lives of citizens. Accurate spatio-temporal prediction plays an important role in smart cities, such as traffic and air quality prediction, which has guidance effects on traffic control and air pollution management. Because of its great practical value, many efforts have been devoted to the accurate spatio-temporal prediction.

Spatio-temporal data is a typical type of time-series data since its inherent temporal consecutiveness and spatial correlations. Classic time-series analysis methods extract the linear dependencies for forecasting, representatives of which include auto-regressive integrated moving average (ARIMA) and vector auto-regression (VAR). But these methods cannot handle the complex nonlinear correlations in spatio-temporal data, because they are based on the linear dependency assumption. Additionally, they lack the ability to extract the spatial proximity. With the rapid increase of urban data volume, deep learning methods are widely applied for spatio-temporal prediction in smart cities, such as traffic flow [1], [2], [3], taxi demand [4], [5], air quality [6], [7] and precipitation predictions [8]. Particularly, recurrent neural networks (RNNs), including gated recurrent unit (GRU) [9] and long short-term memory network (LSTM) [10], perform well in capturing temporal dependencies. Convolution neural networks (CNNs) extracts local spatial features to model spatio-temporal relationships of a city, which is divided into regular grids. However, most sensors that collect data in smart cities present an irregular distribution, it is hard for CNNs to process data with non-Euclidean structure. Graph neural networks (GNNs) [11], such as graph convolutional network (GCN) [12], significantly improves the prediction performance because of its advantage in handling spatio-temporal relationships in non-Euclidean spaces. Moreover, the combinations of CNNs, GNNs, and RNNs [1], [2], [3], [8], [13] have achieved impressive performances in spatio-temporal prediction. Despite these achievements in deep learning, challenges remain for spatio-temporal prediction in smart cities. The superior performance of these deep learning methods is based on large-scale training data, which is not always accessible in the real world. For example, cities that are newly developed or only released a few days of data face the data-insufficient problem. Therefore, existing methods still struggle in making an accurate prediction in cities with limited data. Transfer learning provides effective solutions to address the data scarcity problem, which adapts existing knowledge from data-rich cities to the data-scarce city. [14] constructed semantically related dictionaries that were transferred from a source city to a target city, defined in advance as shared knowledge. [15] computed the similarity between inter-city regions to transfer fine-grained region-level knowledge. Transfer



(a) Distribution differences across cities (b) Prediction results

Fig. 1. Transfer knowledge between different cities, *City 1→City 3* denotes that transfer knowledge from City 1 to City 3, and likewise for *City 2→City 3*.

- *Xiaocao Ouyang, Yan Yang (the corresponding author), Wei Zhou, Yiling Zhang, and Wei Huang are with School of Computing and Artificial Intelligence, Institute of Artificial Intelligence, Southwest Jiaotong University, Chengdu, 611756, China. E-mails: ouyangxiaocao@my.swjtu.edu.cn, yyang@swjtu.edu.cn, wzhou@my.swjtu.edu.cn, geniuszyl@163.com, huangweifujian@126.com.*
- *Hao Wang is with the Research Institute of Artificial Intelligence, Zhejiang Lab, Hangzhou, China. E-mail: cshaowang@gmail.com.*

learning has been demonstrated to help solve the data scarcity problem [16]. However, knowledge transfer between different cities is challenging, as the data from different cities may have different distributions. Hence, knowledge transfer across cities faces two key challenges.

(1) **How to transfer effective knowledge from source to target cities with different distributions?** The difference in data distribution between source and target cities may cause unstable performance and bring the risk of negative transfer [17]. As shown in Fig. 1(a), the data distribution of City 3 is similar to that of City 2 but significantly different from that of City 1. So we show that transferring knowledge from City 2 to City 3 may be more effective than that from City 1. The results in Fig. 1(b) also demonstrate that *City 2→City 3* has a smaller prediction error than *City 1→City 3*. Thus, a method capable of mitigating the discrepancy between data distributions of source and target cities is needed.

(2) **How to capture potential spatio-temporal patterns from cities?** Existing methods demonstrate that capture similar patterns among cities or regions can significantly improve the prediction performance [14], [15], [17]. But spatio-temporal patterns in cities always present complexity and dynamics, as the example shown in Fig. 2. From temporal perspective (i.e., Fig. 2(a)), different districts (e.g., office building and school) in two cities share similar change trends, i.e., temporal patterns. From the spatial perspective (i.e., Fig. 2(b)), different cities both encounter traffic congestion in the morning peak, it shows the consistency of traffic speed in spatial distribution. Similarly, the high traffic speed and fluent traffic conditions are happened at midnight in different cities. Pre-constructing explicit modules to capture complex spatio-temporal patterns is almost unattainable. Thus, it is challenging to capture potential spatio-temporal patterns from cities.

To address above challenges, we propose a novel domain adversarial model with knowledge transfer for spatio-temporal prediction across cities, entitled **CityTrans**. It is composed of four parts: feature extractor, feature domain discriminator, knowledge domain discriminator and predictor. Specifically, a spatio-temporal network (ST-Net) is constructed as a *feature extractor* to capture the hidden spatio-temporal representations. Then, parameterized spatio-temporal knowledge (ST-Knowledge) is adaptively learned through ST-Net, which simultaneously captures the potential spatial and temporal patterns. Then, we construct a *feature domain discriminator* and a *knowledge domain discriminator*, which introduce the domain-adversarial training strategy to assist in modeling domain-invariant feature distribution. Finally, a *predictor* composed of a knowledge attention mechanism and a fully connected layer is devised. The knowledge attention mechanism is utilized to extract the transferable information from ST-Knowledge, and the fully connected layer is employed for the final linear transformation. We summarize our contributions as follows.

- To the best of our knowledge, we are the first to propose an end-to-end graph-based domain adversarial framework for cross-city spatio-temporal prediction.
- To capture more comprehensive spatio-temporal knowledge for transferring, we learn self-adaptive ST-Knowledge that simultaneously extracts potential spatial and temporal patterns across cities. Besides, a knowledge attention mechanism is designed to explicitly exploit the ST-Knowledge.

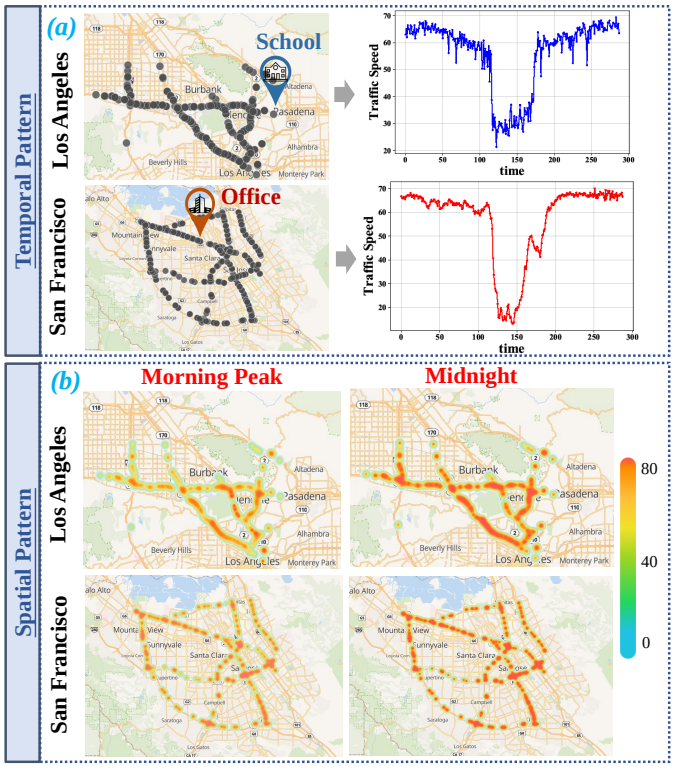


Fig. 2. The illustration of spatio-temporal patterns of Los Angeles and San Francisco.

- We build a feature domain discriminator and a knowledge domain discriminator with domain-adversarial training strategy, which jointly obtain domain-invariant features and ST-Knowledge.
- We conduct extensive experiments on two tasks: traffic (flow and speed) prediction and air quality prediction. The experimental results verify that CityTrans significantly improves prediction performance in target cities with limited data.

The rest of this paper is organized as follows. Section 2 discusses related works. Section 3 introduces definitions and problem statement of this paper. Section 4 presents the methodology proposed in this paper. Section 5 shows and discusses the experimental results on two spatio-temporal prediction tasks. Finally, Section 6 concludes the work.

2 Related Works

2.1 Spatio-Temporal Prediction

Spatio-temporal prediction is a fundamental problem in smart cities. Majority of deep learning methods for spatio-temporal prediction can be roughly divided into grid-based and graph-based models [18]. Grid-based models partition a area or city into regular grids, then adopt CNNs to model the spatial relationships [8]. [19] took one grid and the surrounding grids into a local CNN to capture localized spatial correlations, then introduced a weighted graph for the global spatial relationship. For various temporal dependencies, [20], [21] constructed data with different temporal patterns (i.e., temporal closeness, period and trend). A Siamese architecture was proposed to simultaneously predict inflow and outflow of traffic demand with data

from multiple temporal views [5]. To simultaneously predict the crowd flows and Origin-Destination of the flows, [22] designed a shared-private feature learning framework with an adversarial loss and a discriminative orthogonality loss to extract the task-specific and shared spatio-temporal features. The attention mechanism was adopted to learn a spatial attention map and dynamic spatio-temporal representations [23].

Most graph-based models treat a city as a graph, the sensors that collect data in the city as nodes of graph, and edges of graph are built by geographical proximity or road connectivity. Graph-based models use GNNs [12], [24] to model spatio-temporal correlations due to their powerful capability for extracting spatial features in non-Euclidean space. Most existing works combined GNNs and RNNs to jointly capture spatial and temporal relationships [2], [13], [25]. To adaptively extract more relevant information, [26], [27] applied attention mechanisms to graph-based models. [28] utilized external information to learn meta-knowledge of nodes and edges, then incorporated these meta-knowledge to boost prediction performance. [29] modified GCN by combining time-varying affinity to capture dynamic spatio-temporal correlations, and integrated multi-task learning to predict traffic accident risk and traffic flow. In [30], a node-wise proximity measurement and signal-wise differential operation integrated a GCN-based model for multi-granularity traffic risk prediction. Heterogeneous information was leveraged in [31] and a vehicle-trajectories based network was proposed. Unlike grid- and graph-based models, [32], [33] treated vehicle trajectories as first-class citizens, propagating information along the trajectories to capture the spatial dependency.

Although the aforementioned works have achieved remarkable performance on spatio-temporal prediction, they relied on a large set of training samples. Besides, the construction of data-based temporal patterns usually requires sufficient data, which is difficult to acquire for cities with limited data. Therefore, transfer knowledge from a data-rich source city to a data-scarce target city is essential.

2.2 Knowledge Transfer

Transfer learning leverages the previously learned knowledge from a source domain to adapt to a target domain with limited data [16]. Existing works pre-constructed shared knowledge through similarity between source and target domains to enhance prediction performance. A multi-modal transfer learning method learned semantically related dictionaries by clustering meteorology data, then transferred dictionaries to a target domain for predicting air quality categories [14]. To predict traffic flow, [15] linked each target city region to a similar source region based on short period of service data or correlated check-in records. The aforementioned methods elaborately constructed shared knowledge in advance or required external related data. However, the additional related information (e.g., meteorology information for regions of a city, check-in records in traffic system, etc.) is often hard to access, which restricts the capability to construct knowledge to transfer. [17] further introduced a meta-learning approach and learned a spatio-temporal memory based on regional functional categories from multi-source cities. For station-sparse demand prediction, an augmented memory module was combined with LSTM to get task-specific and shared knowledge [34]. These methods built parameterized knowledge or memory cell that captured short- and long-period temporal patterns from source cities and transferred them to a target city.

However, these methods focused on extracting explicit spatio-temporal patterns in source cities and ignored the complex spatio-temporal patterns which are potential. Besides, due to the significant differences in data distributions between the source and target cities, knowledge transfer may hurt the performance. Based on the spatio-temporal distribution similarities, [35] considered the different impacts of multi-source cities on the target city, and proposed a generation mechanism to learn long-term temporal features from source cities. But it may not be effective in knowledge transfer since the differences in data distributions still exist. *Different from previous works, to enhance the performance of spatio-temporal prediction in a target city, we aim to transfer the potential spatio-temporal patterns and domain-invariant knowledge from a data-rich source city to a data-scarce target city.*

2.3 Domain Adaptation

Domain adaptation is a subtopic of transfer learning, which can effectively bridge the gap in data distributions across domains. Motivated by the adversarial theory, [36], [37] utilized maximum mean discrepancy (MMD) to learn transferable features and obtain domain-invariant information from task-specific layers. Similar to generative adversarial networks (GANs) [38], domain adversarial problem is formulated as a minimax game to learn the domain-invariant representations [39]. [40] aligned multi-modal distributions by leveraging discriminative information from the label classifier. [41] leveraged an adversarial domain adaptation method with GCN to learn class discriminative node representations.

To the best of our knowledge, most domain adaptation methods are constructed for classification tasks [42], [43], [44], there are few attempts on regression tasks. To solve the temporal covariate shift problem, [45] split time series into periods with large distribution gaps, then reduced distribution divergence via domain adaptation. For spatio-temporal crowd flow prediction, [46] used a grid-based adaptation network with MMD to match the domain distributions of the embedding. To predict the parking occupancy, [47] leveraged convolutional LSTM (ConvLSTM) with heterogeneous contextual information and utilized adversarial domain adaptation. *Different from these methods that worked on time-series or grid-based spatio-temporal prediction, we focus on the graph-based spatio-temporal prediction task with limited data, which is rarely considered. Therefore, the aforementioned methods cannot be directly applied.* [48] proposed a transferable model to learn domain-invariant node embeddings between cities. *Compared with it, we not only learn the self-adaptive spatio-temporal knowledge to transfer but also learn the domain-invariant hidden representations and knowledge across cities via domain-adversarial training.*

3 Preliminary

In this section, we first present definitions used in this paper and then formulate the spatio-temporal prediction problem.

Definition 1 (Graph $\mathcal{G}$). The topological structure of a city is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$. $\mathcal{V}$ is a set of nodes and $|\mathcal{V}| = N$. $\mathcal{E}$ indicates a set of edges. In addition, $\mathcal{A}$ is a weighted matrix that represents the adjacency between nodes of the graph. Each weight in $\mathcal{A}$ represents the proximity between nodes. Particularly, we set the source city to $\mathcal{G}_{\mathcal{S}} = (\mathcal{V}_{\mathcal{S}}, \mathcal{E}_{\mathcal{S}}, \mathcal{A}_{\mathcal{S}})$ and set the target city to $\mathcal{G}_{\mathcal{T}} = (\mathcal{V}_{\mathcal{T}}, \mathcal{E}_{\mathcal{T}}, \mathcal{A}_{\mathcal{T}})$, $|\mathcal{V}_{\mathcal{S}}| = N_{\mathcal{S}}$ and $|\mathcal{V}_{\mathcal{T}}| = N_{\mathcal{T}}$.

Definition 2 (Graph Signal $\mathcal{X}$). In a city, we define the graph signal as

$$\mathcal{X} = \{X^{(t)} | t \in P\} = \{x_i^{(t)} | i \in N, t \in P\} \in \mathbb{R}^{N \times P \times D}, \quad (1)$$

where $x_i^{(t)}$ is the graph signal of node i at time step t in graph $\mathcal{G}$, and P is the given historical time step. We define graph signals in source city and target city as $\mathcal{X}_\mathcal{S}$ and $\mathcal{X}_\mathcal{T}$, respectively.

In this paper, the source city has abundant data, whereas the target city suffers from data scarcity problem. $P_\mathcal{S}$ and $P_\mathcal{T}$ are total number of time steps for source and target cities respectively, i.e., $|P_\mathcal{T}| \ll |P_\mathcal{S}|$. We formulate the spatio-temporal prediction problem in the target city as follows.

Problem Definition (Spatio-temporal Prediction Problem). Given limited data in a target city and sufficient data in a source city, we aim to learn a function f that can forecast P' future graph signals in the target city based on P historical graph signals:

$$\widetilde{X}_\mathcal{T}^{(t+1):(t+P')} = f(X_\mathcal{S}^{(t-P):(t)}, X_\mathcal{T}^{(t-P):(t)}), \quad (2)$$

$$\min_f error(\widetilde{X}_\mathcal{T}^{(t+1):(t+P')}, X_\mathcal{T}^{(t+1):(t+P')}), \quad (3)$$

where $\widetilde{X}_\mathcal{T}$ is predicted graph signals in the target city. The $error$ metric can be mean absolute error, mean squared error, etc.

4 METHODOLOGY

In this section, we propose a domain adversarial model with knowledge transfer to solve the spatio-temporal prediction problem across cities, entitled **CityTrans**. Specifically, we first introduce a spatio-temporal network (ST-Net) as the basic model for spatio-temporal modeling. Then, we feed the raw spatio-temporal data into ST-Net for obtaining the hidden representations of the raw data and randomly initialize spatio-temporal knowledge (ST-Knowledge) with learnable parameters. The parameterized ST-Knowledge is jointly trained with ST-Net in an end-to-end manner. Furthermore, we build two domain discriminators consisting of a feature domain discriminator and a knowledge domain discriminator, which are incorporated with domain-adversarial training to facilitate city-level domain adaption. Finally, we design a predictor where a knowledge attention module is proposed to extract the transferable information from ST-Knowledge, and a fully connected layer is adopted to obtain the final linear mapping. The framework of CityTrans is depicted in Fig. 3, which we explicate in detail as follows.

4.1 Spatio-Temporal Network

Recently, GCN [12] has been widely used due to its powerful capability to capture spatial relationships in non-Euclidean space. The combinations of GRU and GCN have achieved state-of-the-art performances on spatio-temporal prediction [2], [13], [49]. Therefore, we construct our spatio-temporal network (ST-Net) by adopting GRU and GCN, which are built as the basic model to obtain hidden representations with spatio-temporal correlations. Unlike previous works [26], [50], which are elaborately designed with multiple modules to handle the spatio-temporal relationships, we introduce ST-Net as the spatio-temporal feature extractor. It can not only achieve superior performance, but also reduce the number of parameters.

The historical graph signal in source city is denoted as $\mathcal{X}_\mathcal{S} \in \mathbb{R}^{N_\mathcal{S} \times P \times D}$, and the target city has $\mathcal{X}_\mathcal{T} \in \mathbb{R}^{N_\mathcal{T} \times P \times D}$. To jointly learn hidden spatio-temporal representations of the source and target cities, we feed $\mathcal{X} = [\mathcal{X}_\mathcal{S}; \mathcal{X}_\mathcal{T}]$ into ST-Net. First, the raw data is mapped into feature space through a linear layer $FC(\cdot)$. Second, a GRU is adopted to memorize long-term information, which iteratively inputs the hidden states at previous time steps and the current graph signal to handle temporal dependence. Then, a batch normalization layer (BN) [51] normalizes features to accelerate the convergence of the model. The temporal feature extractor process is formulated as:

$$\mathcal{H}^{(l)} = ReLU(BN(GRU(FC(\mathcal{X})))), \quad (4)$$

where $\mathcal{H}^{(l)}$ is the hidden representation of l-th layer.

GCN is utilized to extract spatial correlations, and an augmented adjacency matrix is required to update node features and aggregate neighboring information in the source and target cities. The augmented adjacency matrix $\mathcal{A}$ is constructed by arranging $\mathcal{A}_\mathcal{S}$ and $\mathcal{A}_\mathcal{T}$ on the diagonal of $\mathcal{A}$ respectively, the rest positions of $\mathcal{A}$ are set to 0. In this way, we have $\mathcal{A} \in \mathbb{R}^{(N_\mathcal{S}+N_\mathcal{T}) \times (N_\mathcal{S}+N_\mathcal{T})}$. We denote the graph convolution operation to $GCN(\cdot)$, and the graph convolutional representations are learned by:

$$\mathcal{H}^{(l+1)} = GCN(\mathcal{H}^{(l)}) = \sigma(\widetilde{\mathcal{A}}\mathcal{H}^{(l)}\mathcal{W}^{(l+1)}), \quad (5)$$

where $\widetilde{\mathcal{A}}$ is the normalized adjacency matrix of $\mathcal{A}$, and each node has a self-loop. $\mathcal{H}^{(l)}$ is the output of previous layer. $\mathcal{W}^{(l+1)}$ is the projection matrix with trainable parameters, $\sigma(\cdot)$ is the activation function. For better illustration, we visualize the structure of ST-Net in Fig. 3.

4.2 Spatio-Temporal Knowledge

Most existing methods [2], [3], [8], [26] for spatio-temporal prediction are based on a huge volume of training data, and long period of data plays an important role in these methods to extract potential temporal knowledge (e.g., temporal periodicity) [5], [19], [21]. However, mining long-term knowledge from cities with limited data is very challenging. [17] learned a global memory to capture long-period patterns, then a clustering loss based on region functional similarity was constructed, which enforced the consistency of representations extracted from memory with clustering results. This approach stored explicit spatio-temporal patterns (i.e., spatial functionality), but its performance heavily depended on the pre-clustering results.

Different from existing works [14], [15], [17], we learn the self-adaptive spatio-temporal knowledge, named **ST-Knowledge**. ST-Knowledge does not need external related information or to be pre-defined, which is learned in an end-to-end manner through stochastic gradient descent. Through iterative training, the model automatically extracts the potential spatio-temporal patterns exist in source and target cities by itself. To simultaneously consider the latent spatial and temporal patterns, we randomly initialize ST-Knowledge with learnable parameters. Specifically, we denote ST-Knowledge as $\mathcal{K} = [\mathcal{K}_\mathcal{S}; \mathcal{K}_\mathcal{T}] \in \mathbb{R}^{2G_k \times P_k \times d_k}$. $\mathcal{K}_\mathcal{S}$ and $\mathcal{K}_\mathcal{T}$ denote the ST-Knowledge from source and target cities, respectively. And $\mathcal{K}_\mathcal{S}, \mathcal{K}_\mathcal{T} \in \mathbb{R}^{G_k \times P_k \times d_k}$, where G_k denotes the number of categories of latent spatial patterns, P_k is the length of underlying temporal patterns, and d_k is the dimension of the knowledge. Note that ST-Knowledge does not guarantee patterns across cities with explicit spatial functionalities or temporal periods, but rather potential spatio-temporal patterns.

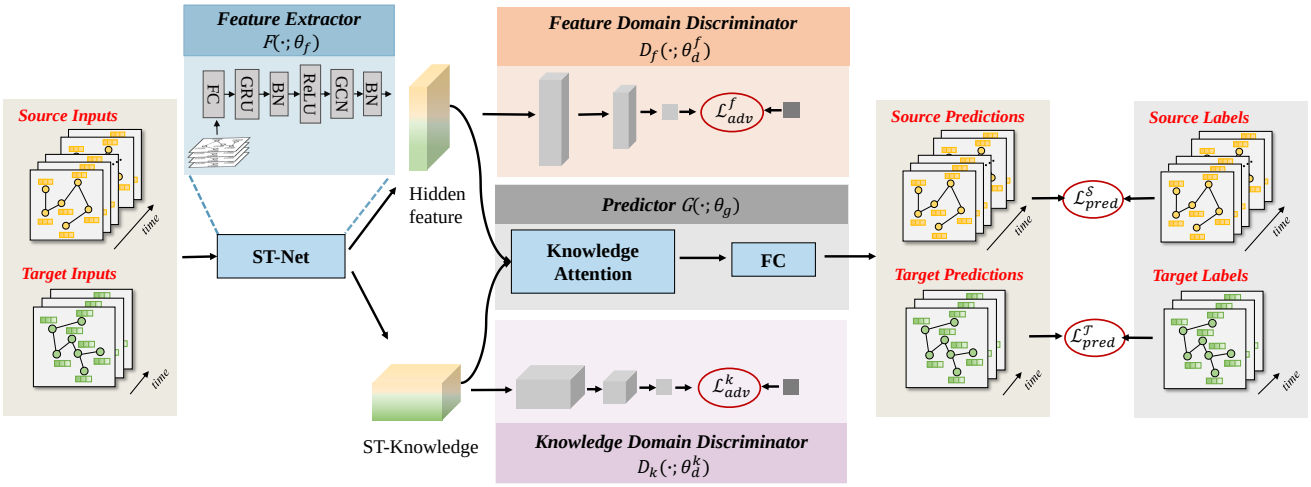


Fig. 3. Framework overview of CityTrans.

4.3 Domain-Adversarial Training

During the transferring phase, the distribution discrepancy between source and target cities may lead to negative transfer [16], i.e., the previous knowledge from the source city harms the performance of the target city. For spatio-temporal prediction across cities, we regard source city as a source domain and target city as a target domain. Therefore, we adopt a domain-adversarial training strategy to capture domain-invariant features and knowledge. The basic idea is to achieve adversarial training between domain discriminator and feature extractor. Concretely, we train domain discriminators to predict the domain label of each representation or ST-Knowledge (i.e., to predict whether the representation or ST-Knowledge comes from source or target domain). Meanwhile, to fool the domain discriminators, we train the ST-Net to generate domain-invariant representations and shared knowledge. Our proposed domain discriminators are comprised of two components: a **feature domain discriminator** and a **knowledge domain discriminator**. In the following, we present the domain discriminators in detail.

4.3.1 Feature Domain Discriminator

Specifically, we define a binary class domain discriminator for feature classification, given domain labels $d_{\mathcal{S}} = 0$ and $d_{\mathcal{T}} = 1$ for source and target domains, respectively. The feature domain discriminator $D_f(\cdot;\theta_d^f)$ follows the feature extractor $F(\cdot;\theta_f)$ (i.e., ST-Net), as illustrated in Fig. 3. Besides, to reverse the gradient during the back propagation, we incorporate a gradient reversal layer $R(\cdot)$ [39] between F and D_f. Next, we update parameters θ_d^f from the feature domain discriminator D_f:

$$\begin{aligned}\arg\min_{\theta_d^f} \mathcal{L}_{adv}^f(d_{\mathcal{S}}, D_f(R(F(\mathcal{X}_{\mathcal{S}})))+ \\ \mathcal{L}_{adv}^f(d_{\mathcal{T}}, D_f(R(F(\mathcal{X}_{\mathcal{T}})))),\end{aligned} \tag{6}$$

where $d_{\mathcal{S}}$ and $d_{\mathcal{T}}$ represent true domain labels of source and target domains, respectively.

The objective contains a categorical cross-entropy loss $\mathcal{L}_{adv}^f$ with the one-hot domain label d, the feature domain discriminator is formulated as:

$$\mathcal{L}_{adv}^f(d, q) = -\sum_{i=1}^{n} d_i \log q_i, \tag{7}$$

where d_i is the true domain label of each representation, and the softmax probability output of D_f is denoted to q_i, n is the number of samples.

4.3.2 Knowledge Domain Discriminator

As aforementioned, we propose to construct ST-Knowledge for source and target cities, denoted as $\mathcal{K}_{\mathcal{S}}$ and $\mathcal{K}_{\mathcal{T}}$ respectively. Since spatio-temporal patterns shared between source and target domains are supposed to be domain-invariant, we design a knowledge domain discriminator of ST-Knowledge $D_k(\cdot;\theta_d^k)$ with domain-adversarial training to mitigate the difference between $\mathcal{K}_{\mathcal{S}}$ and $\mathcal{K}_{\mathcal{T}}$. We also define a binary class domain discriminator for knowledge classification, given domain labels $d_{\mathcal{S}} = 0$ and $d_{\mathcal{T}} = 1$ for $\mathcal{K}_{\mathcal{S}}$ and $\mathcal{K}_{\mathcal{T}}$, respectively.

Formally, we update the parameter θ_d^k as following:

$$\begin{aligned}\arg\min_{\theta_d^k} \mathcal{L}_{adv}^k(d_{\mathcal{S}}, D_k(R(\mathcal{K}_{\mathcal{S}})))+ \\ \mathcal{L}_{adv}^k(d_{\mathcal{T}}, D_k(R(\mathcal{K}_{\mathcal{T}})))\end{aligned} \tag{8}$$

Similar to the loss function $\mathcal{L}_{adv}^f$, we define the objective function $\mathcal{L}_{adv}^k$ of knowledge domain discriminator as:

$$\mathcal{L}_{adv}^k(d, q') = -\sum_{i=1}^{n} d_i \log q_i', \tag{9}$$

where q_i' indicates the softmax probability output of D_k.

4.4 Predictor

To predict the future graph signals across cities, we devise a predictor comprising two components: a knowledge attention module and a fully connected layer.

The knowledge attention module is proposed to extract the transferable spatio-temporal information from ST-Knowledge. After learning the self-adaptive and domain-invariant ST-Knowledge from both source and target cities, we utilize each hidden node representation of data to query the ST-Knowledge through the knowledge attention mechanism. The detailed structure of knowledge attention is illustrated in Fig. 4.

Specifically, we first project the hidden representation of ST-Net to obtain the query vector, which is formulated as:

$$v_i^{(t)} = FC(h_i^{(t)}), \tag{10}$$

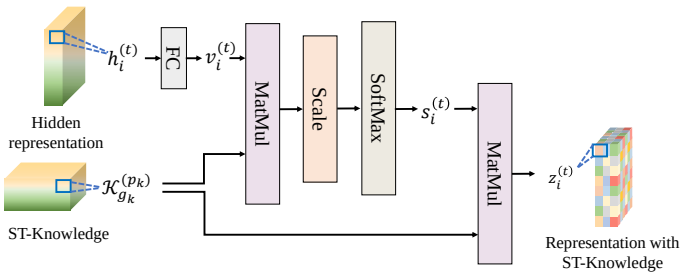


Fig. 4. Structure of knowledge attention module.

where $FC(\cdot)$ is a fully connected layer, $h_i^{(t)} \in \mathbb{R}^{d_h}$ is the representation of node i at time t, and d_h is the dimension of node representations. The query vector is $v_i^{(t)} \in \mathbb{R}^{d_k}$.

Then, we calculate the similarity score between hidden representations and ST-Knowledge. A higher similarity score indicates that the knowledge tends to be transferred for prediction. Formally, we obtain the transferable spatio-temporal information as:

$$s_i^{(t)}(p_k, g_k) = \frac{exp(< v_i^{(t)}, \mathcal{K}_{g_k}^{(p_k)} >)}{\sum_{g'_k=1}^{G_k} \sum_{p'_k=1}^{P_k} exp(< v_i^{(t)}, \mathcal{K}_{g'_k}^{(p'_k)} >)}, \tag{11}$$

$$z_i^{(t)} = \sum_{g_k=1}^{G_k} \sum_{p_k=1}^{P_k} (s_i^{(t)}(p_k, g_k) * \mathcal{K}_{g_k}^{(p_k)}), \tag{12}$$

where $\mathcal{K}_{g_k}^{(p_k)} \in \mathbb{R}^{d_k}$ is the ST-Knowledge of g_k-th spatial pattern and p_k-th temporal pattern, $s_i^{(t)}(p_k, g_k)$ is the similarity score, and $z_i^{(t)}$ indicates the representation with ST-Knowledge.

Finally, we input the concatenation of hidden representations $h_i^{(t)}$ and the representation with ST-Knowledge $z_i^{(t)}$ into a fully connected layer to obtain the prediction results, formulated as:

$$\widetilde{x}_i^{(t)} = FC([h_i^{(t)}, z_i^{(t)}]), \tag{13}$$

where $\widetilde{x}_i^{(t)}$ is the predicted graph signal of node i at time t.

We formulate the prediction loss function as:

$$\mathcal{L}_{pred} = \frac{1}{N}\frac{1}{P'} \sum_{i=1}^{N} \sum_{t=1}^{P'} (\widetilde{x}_i^{(t)} - x_i^{(t)})^2, \tag{14}$$

where $x_i^{(t)}$ is ground truth of node i at time t. $\mathcal{L}_{pred}^{\mathcal{S}}$ and $\mathcal{L}_{pred}^{\mathcal{T}}$ indicate prediction losses of source and target cities, respectively.

We denote the parameters of feature extractor F as θ_f, and the parameters in the predictor G as θ_g. Then, we update the model parameters θ_f and θ_g as follows:

$$\begin{aligned} \arg\min_{\theta_f, \theta_g} & \mathcal{L}_{pred}(\mathcal{X}_{\mathcal{S}}, G(F(\mathcal{X}_{\mathcal{S}}))) + \\ & \mathcal{L}_{pred}(\mathcal{X}_{\mathcal{T}}, G(F(\mathcal{X}_{\mathcal{T}}))) \end{aligned} \tag{15}$$

4.5 Overall Loss and Model Training

The overall loss of our proposed CityTrans is formulated as:

$$\mathcal{L} = \mathcal{L}_{pred}^{\mathcal{S}} + \mathcal{L}_{pred}^{\mathcal{T}} + \lambda(\mathcal{L}_{adv}^{f} + \mathcal{L}_{adv}^{k}), \tag{16}$$

where λ is a hyper-parameter to tune the trade-off of domain discriminators during learning process.

We summarize the training procedure for CityTrans in Algorithm 1.

Algorithm 1 Training Algorithm of CityTrans

Input: source city data $\mathcal{X}_{\mathcal{S}}$, target city data $\mathcal{X}_{\mathcal{T}}$, trade-off parameter λ
Output: parameter sets θ_f and θ_g

/* In the following, the feature domain discriminator and the knowledge domain discriminator are referred to as F-DD and K-DD respectively, while the knowledge attention module is referred to as KA for simplicity.*/

Randomly initialize $\theta_f, \theta_g, \theta_d^f, \theta_d^k, \mathcal{K}_{\mathcal{S}}$ and $\mathcal{K}_{\mathcal{T}}$
$epoch \leftarrow 0$
while $epoch \leq max\_epoch$ **do**
  for $x_i^{\mathcal{S}} \in \mathcal{X}_{\mathcal{S}}, x_i^{\mathcal{T}} \in \mathcal{X}_{\mathcal{T}}$ **do**
    $[h_i^{\mathcal{S}}, h_i^{\mathcal{T}}; \mathcal{K}_{\mathcal{S}}, \mathcal{K}_{\mathcal{T}}] \leftarrow$ ST-Net $([x_i^{\mathcal{S}}, x_i^{\mathcal{T}}])$
    $[q_i^{\mathcal{S}}, q_i^{\mathcal{T}}] \leftarrow$ F-DD $([h_i^{\mathcal{S}}, h_i^{\mathcal{T}}])$
    $[q_i'^{\mathcal{S}}, q_i'^{\mathcal{T}}] \leftarrow$ K-DD $([\mathcal{K}_{\mathcal{S}}, \mathcal{K}_{\mathcal{T}}])$
    $[z_i^{\mathcal{S}}, z_i^{\mathcal{T}}] \leftarrow$ KA $([h_i^{\mathcal{S}}, h_i^{\mathcal{T}}; \mathcal{K}_{\mathcal{S}}, \mathcal{K}_{\mathcal{T}}])$
    $[\widetilde{x}_i^{\mathcal{S}}, \widetilde{x}_i^{\mathcal{T}}] \leftarrow$ FC $([h_i^{\mathcal{S}}; z_i^{\mathcal{S}}], [h_i^{\mathcal{T}}; z_i^{\mathcal{T}}])$
  end for
  calculate $\mathcal{L}$ by Eq. (16)
  update parameters θ_f and θg by Eq. (15)
  update parameters θ_d^f and θ_d^k by Eqs. (6) and (8)
end while
return θ_f, θ_g

5 Experiments

In this section, we conduct extensive experiments for two spatio-temporal prediction tasks: traffic (flow and speed) prediction and air quality prediction across cities. Experiments are designed to answer the following research questions (RQ):

- **RQ1**: Whether our proposed CityTrans outperforms baseline methods for spatio-temporal prediction in cities with limited data?
- **RQ2**: How do our proposed sub-modules (i.e., ST-Knowledge, feature domain discriminator, and knowledge domain discriminator) contribute to the prediction performance?
- **RQ3**: What are the impacts of the training days and historical time steps of the target city on the prediction performance?
- **RQ4**: How do various hyper-parameters (i.e., the latent spatial categories G_k, the length of underlying temporal pattern P_k, the dimension of the ST-Knowledge d_k, and the trade-off λ) affect the prediction performance?

We also visualize learned node representations after domain-adversarial training and predicted results of different methods to provide an intuitive understanding of our proposed model.

5.1 Experiment Setup

5.1.1 Datasets

We evaluate the performance of CityTrans on two kinds of public real-world datasets: traffic and air quality datasets.

Traffic Datasets: We first conduct experiments on two traffic tasks: (1) flow prediction and (2) speed prediction. For flow prediction, we evaluate experiments on four public real-world traffic datasets, i.e., PEMS03, PEMS04, PEMS07, and PEMS08, which are collected from Caltrans Performance Measure System (PeMS) [52]. Note that we use PEMS03, PEMS07 as source cities, and PEMS04, PEMS08 as target cities. For speed prediction, we

TABLE 1
Summary of datasets.

Datasets		# Sensors	Time Span (m/d/y)	Time Interval
Traffic Flow	PEMS03	358	9/1/2018-11/30/2018	5 minutes
	PEMS04	307	1/1/2018-2/28/2018	5 minutes
	PEMS07	883	5/1/2017-8/31/2017	5 minutes
	PEMS08	170	7/1/2016-8/31/2016	5 minutes
Traffic Speed	METR-LA	207	3/1/2012-6/30/2012	5 minutes
	PEMS-BAY	325	1/1/2017-5/31/2017	5 minutes
Air Quality	Beijing	34	1/1/2017-1/31/2018	1 hour
	London	20	1/1/2017-3/27/2018	1 hour

verify CityTrans on two datasets, i.e., METR-LA and PEMS-BAY, which are released by [2]. The time interval for traffic datasets is 5 minutes, which means 12-time intervals of 1 hour and 288-time intervals of 1 day. Given the historical 1-hour observations, we predict traffic flow or speed for the next hour.

Air quality Datasets: We also evaluate experiments on air quality prediction, the air quality is represented by the value of PM2.5 concentrations. Two air quality datasets: Beijing and London datasets are provided by Biendata[1]. Since the time interval of air quality data is 1 hour, we predict PM2.5 concentrations for the next 12 hours using the previous 12-hour observations.

Additionally, we normalize these datasets by the standard normalization method. The detailed information of these datasets is shown in Table 1.

5.1.2 Experimental Settings

For all prediction tasks, we split datasets of source cities with a ratio of 7:1:2 into training, validation, and test sets. To simulate the condition of target cities that suffer from the data scarcity problem, (1) for traffic prediction, we use 10-day data from target cities for training, (2) for air quality prediction, we use 3-month training data since the time interval is 1 hour. The validation and test sets are the same as [1], [3]. We use the 12-time intervals of historical data to predict the future 12-time intervals data in the target city.

All deep learning models are implemented in Python with PyTorch and executed on one NVIDIA TITAN V (12GB) GPU. In the experiments, we optimize our model by AdamW optimizer for a maximum of 100 epochs, and an early stop strategy is adopted. We manually set the learning rate to 0.001 and set the batch size to 16 and 32 for traffic prediction and air quality prediction, respectively. The hyper-parameters G_k, P_k and d_k in ST-Knowledge $\mathcal{K}_{\mathcal{S}}$, $\mathcal{K}_{\mathcal{T}}$ are set to 16, 12 and 16, respectively. Moreover, we set the trade-off λ in loss function $\mathcal{L}$ to 0.5.

5.1.3 Baselines and Evaluation Metrics

We compare CityTrans with both non-transfer learning baselines and transfer learning-based prediction models. For non-transfer baselines, only the limited data from target cities are used for model training. The non-transfer baselines are as follows:

- **VAR**: Vector Auto-Regression is a linear function that captures the dependencies between multiple time series variables.
- **GBRT**: Gradient Boosting Regression Tree consists of gradient boosting and regression tree to extract non-linear patterns.
- **GRU** [9]: Gated Recurrent Unit is a variant of recurrent neural network, which is powerful for handling temporal dependencies. We realize this GRU with 2 hidden layers and 64 neurons in each hidden layer.
- **DCRNN** [2]: Diffusion Convolution Recurrent Neural Network combines RNN and GCN with a diffusion operation to simultaneously capture the spatial and temporal correlations. The diffusion step and the number of diffusion recurrent convolution layer are set to 2.
- **STGCN** [1]: Spatio-Temporal Graph Convolution Network incorporates graph convolution with gated temporal convolution. In our experiments, 2 ST-Conv blocks are stacked, and the channels of 3 layers in each block are 64, 32, and 128, respectively. The size of kernels is set to 3.
- **GWN** [3]: Graph WaveNet is a spatio-temporal model that integrates diffusion graph convolution and 1D dilated convolutions. We select different dimension of node embedding to learn the adaptive adjacency matrix. We implement this model with 4 dilated causal convolution layers and 4 temporal convolution blocks.
- **STSGCN** [50]: Spatial-Temporal Synchronous Graph Convolution Network that captures spatiotemporal correlations by stacking multiple localized GCN with a designed adjacency matrix over the time axis. We exploit 4 synchronous graph convolutional layers and each layer contains 3 convolutional operations with 64, 64, 64 filters respectively.
- **AGCRN** [13]: Adaptive Graph Convolutional Recurrent Network automatically learns a data-adaptive graph structure and a node-specific pattern to capture the spatial and temporal relationship. We deploy different dimensions of the node-specific embedding for the datasets and utilize 2 hidden layers.
- **HetETA** [31]: HetETA combines gated convolution neural networks and GNN to capture spatio-temporal correlations. Specifically, three different components are developed to model periodicity, and two Het-ChebNets are employed for the road network and vehicle trajectories. We only take its recent component and one Het-ChebNet based on road network because of the limited amount and missing type of data.

The transfer learning-based baselines including fine-tuning based models and the state-of-the-art spatio-temporal transfer learning model.

- **Fine-tuning Methods**: The model is pre-trained with sufficient data from the source city, and then fine-tuned using limited data from the target city. The following methods are conducted: GRU, DCRNN, STGCN, GWN, STSGCN, and AGCRN. The corresponding fine-tuning methods are denoted as **GRU-FT**, **DCRNN-FT**, **STGCN-FT**, **GWN-FT**, **STSGCN-FT**, and **AGCRN-FT**.
- **DASTNET** [48]: Domain Adversarial Spatial-Temporal Network is the state-of-the-art graph-based transfer learning model. It leverages the adversarial domain adaptation technique to learn domain-invariant node embeddings for source and target cities, which are then incorporated to model the spatio-temporal data. We set the dimension of node embedding to 64.

For the spatio-temporal models (i.e., DCRNN, STGCN, GWN, STSGCN, AGCRN, HetETA, and DASTNET), we conduct the experiments based on codes provided in corresponding papers. For fine-tuning baseline methods, we transfer parameters with

1. http://biendata.com/competition/kdd_2018/data/

TABLE 2
Performance comparison of different models on traffic flow datasets PEMS04 and PEMS08 with 10-day training data. (The best performance are marked in bold, and the second best are underlined.)

	Method		PEMS04											
			15 min			30 min			60 min			Average		
			MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
Non-Transfer	**VAR**		22.84	34.73	18.04	28.09	41.46	23.94	39.33	55.29	38.15	29.13	42.62	25.54
	GBRT		21.54	33.58	15.25	24.88	37.98	18.13	31.49	46.58	24.69	25.33	38.52	18.75
	GRU		29.08	41.08	33.25	38.18	54.28	38.59	50.22	70.63	49.58	37.97	53.61	38.04
	DCRNN		21.34	33.17	14.31	25.15	38.51	16.89	33.47	49.89	22.94	25.85	39.41	17.49
	STGCN		22.24	33.92	16.93	24.93	37.58	18.31	30.10	44.43	21.59	25.39	38.11	18.81
	GWN		25.76	38.61	28.20	31.35	45.66	30.41	43.57	62.32	38.06	32.38	47.22	30.50
	STSGCN		22.26	34.63	15.45	24.25	37.38	16.51	28.66	43.49	18.95	24.56	37.84	16.71
	AGCRN		21.09	34.24	14.33	21.69	35.68	14.82	23.68	38.32	15.84	21.98	35.71	14.84
	HetETA		23.05	35.04	17.68	25.46	38.69	18.48	31.55	47.41	22.03	26.36	40.02	19.26
Transfer	**GRU-FT**	*PEMS03→PEMS04*	26.15	37.40	34.72	33.45	45.82	37.96	46.14	61.89	62.25	34.12	47.03	42.84
		PEMS07→PEMS04	24.95	35.90	27.57	31.94	46.83	20.83	42.99	60.90	33.80	32.13	46.47	26.37
	DCRNN-FT	*PEMS03→PEMS04*	20.61	32.21	13.99	23.74	36.57	16.17	30.68	46.13	21.33	24.35	37.38	16.70
		PEMS07→PEMS04	20.35	31.90	13.99	23.49	36.29	16.15	30.24	45.59	21.39	24.06	37.05	16.67
	STGCN-FT	*PEMS03→PEMS04*	21.19	32.65	15.12	23.83	36.31	17.25	28.84	43.71	19.82	24.74	37.62	17.39
		PEMS07→PEMS04	20.50	31.78	15.06	22.74	34.80	17.03	27.03	40.64	20.57	23.33	35.50	17.65
	GWN-FT	*PEMS03→PEMS04*	24.59	36.99	19.72	30.77	45.32	23.98	44.34	63.62	33.48	32.23	47.30	24.93
		PEMS07→PEMS04	24.31	36.64	21.90	30.41	44.93	22.87	45.03	64.45	32.16	32.13	47.20	24.36
	STSGCN-FT	*PEMS03→PEMS04*	21.48	33.20	15.58	23.77	36.55	17.18	28.96	44.02	20.17	24.20	37.14	17.23
		PEMS07→PEMS04	21.87	33.75	15.37	24.20	37.10	16.90	29.70	44.74	20.86	24.71	37.77	17.35
	AGCRN-FT	*PEMS03→PEMS04*	19.49	30.89	13.12	20.61	32.63	13.84	23.53	37.99	**15.44**	20.78	32.89	13.98
		PEMS07→PEMS04	19.18	30.42	13.07	20.44	32.33	13.94	**22.99**	37.53	15.68	20.54	32.48	14.06
	DASTNET	*PEMS03→PEMS04*	21.24	32.82	15.54	24.10	36.83	17.29	30.63	45.42	22.79	24.67	37.00	18.00
		PEMS07→PEMS04	21.50	33.47	14.34	24.87	38.32	16.81	32.61	48.96	23.07	25.59	39.00	18.00
	CityTrans	*PEMS03→PEMS04*	17.22	26.97	13.75	19.20	29.99	14.91	24.70	38.05	19.08	19.69	30.71	15.45
		PEMS07→PEMS04	**16.71**	**26.47**	**11.98**	**18.75**	**29.52**	**13.10**	24.10	**37.37**	16.65	**19.17**	**30.13**	**13.47**

	Method		PEMS08											
			15 min			30 min			60 min			Average		
			MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
Non-Transfer	**VAR**		18.33	27.51	12.96	22.69	33.63	16.45	32.11	45.22	24.44	23.50	34.33	17.25
	GBRT		17.08	28.81	11.43	19.47	32.15	12.58	24.80	40.07	16.56	19.87	32.69	12.92
	GRU		20.72	29.71	17.89	30.18	40.11	32.03	37.22	50.99	45.78	28.23	39.18	29.05
	DCRNN		17.16	26.49	10.93	20.30	31.61	12.89	27.09	41.17	17.58	20.80	32.03	13.34
	STGCN		17.67	27.04	12.42	20.11	30.79	13.83	24.39	36.88	16.45	20.39	31.09	14.07
	GWN		19.98	29.35	18.83	24.91	36.55	21.58	35.89	51.03	30.10	26.56	37.73	23.16
	STSGCN		17.87	27.81	11.97	19.16	29.73	12.67	21.96	33.54	14.37	19.30	29.85	12.81
	AGCRN		18.46	30.56	11.03	19.25	31.90	11.35	23.39	37.46	**13.62**	19.97	32.86	11.77
	HetETA		21.11	30.74	16.52	23.28	34.45	17.21	27.98	41.90	18.64	23.90	35.25	17.38
Transfer	**GRU-FT**	*PEMS03→PEMS08*	22.36	30.98	28.01	30.02	39.55	39.85	37.10	50.72	44.66	28.25	38.65	35.39
		PEMS07→PEMS08	20.91	29.51	19.65	32.56	42.98	29.79	39.59	52.85	34.42	29.96	40.57	25.45
	DCRNN-FT	*PEMS03→PEMS08*	16.04	24.70	10.19	18.35	28.36	11.60	23.24	35.47	14.82	18.67	28.71	11.86
		PEMS07→PEMS08	15.75	24.39	10.05	17.88	27.77	11.40	22.43	34.20	14.46	18.18	28.02	11.61
	STGCN-FT	*PEMS03→PEMS08*	16.97	25.95	11.51	19.45	29.74	13.16	23.67	36.01	15.50	20.12	30.61	13.49
		PEMS07→PEMS08	16.66	25.54	10.52	18.80	28.85	11.96	22.81	34.64	14.04	19.38	29.51	12.26
	GWN-FT	*PEMS03→PEMS08*	20.33	29.86	13.59	25.58	37.19	17.12	35.66	50.39	24.16	26.26	38.01	17.71
		PEMS07→PEMS08	17.84	29.24	12.19	25.59	37.04	16.26	35.86	49.87	25.43	26.29	37.63	17.56
	STSGCN-FT	*PEMS03→PEMS08*	17.83	27.00	13.92	19.54	29.67	15.14	23.22	34.87	16.92	19.79	29.94	15.16
		PEMS07→PEMS08	17.16	26.42	11.79	18.59	28.66	12.75	22.07	33.61	15.32	18.90	29.01	13.06
	AGCRN-FT	*PEMS03→PEMS08*	17.25	27.65	13.35	18.92	29.51	13.66	22.42	34.83	16.15	19.17	29.82	13.87
		PEMS07→PEMS08	16.61	25.72	10.84	18.75	29.56	11.89	23.30	36.43	14.77	19.00	30.04	12.20
	DASTNET	*PEMS03→PEMS08*	17.37	26.46	12.77	20.10	30.88	14.81	26.52	39.59	19.79	20.66	31.00	15.00
		PEMS07→PEMS08	17.03	25.96	12.96	19.39	29.78	13.99	24.86	37.30	17.52	19.82	30.00	14.00
	CityTrans	*PEMS03→PEMS08*	**13.63**	**20.88**	**9.65**	**15.59**	**23.92**	11.54	**20.27**	**31.24**	14.13	**15.96**	**24.55**	11.64
		PEMS07→PEMS08	14.48	22.01	9.69	15.84	24.47	**11.00**	20.40	31.26	13.90	16.52	25.30	**11.50**

the same size between source and target cities, while parameters with different sizes are learned from scratch.

For traffic prediction task, we use *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE) and *Mean Absolute Percentage Error* (MAPE) to evaluate our model. For air quality prediction task, MAE and RMSE are used as evaluation metrics. Lower values indicate better performance.

5.2 Performance Comparison (RQ1)

In this section, we consider two different spatio-temporal prediction tasks across cities. The first task involves traffic prediction, which includes both traffic flow and traffic speed prediction. The second task is the prediction of air quality.

5.2.1 Traffic Prediction Performance Comparison

We implement our proposed CityTrans and compare it with the baselines on traffic flow and traffic speed datasets. We report the performance for the next 15, 30, and 60 minutes (i.e., the 3rd, 6th, and 12th time step). For non-transfer baselines, we only use the limited data of target cities for training. For transfer-based models, we pre-train them with sufficient data from the source city and then fine-tune them on the target city. Different source cities are adopted to transfer. All experiments are repeated 5 times and the average results are presented in Table 2 and Table 3. According to these tables, we make the following observations:

- For traffic flow prediction, CityTrans achieves the best and second-best performance with different source and target cities in most cases. Compared with the state-of-the-art baselines, the averaged relative improvements of

TABLE 3
Performance comparison of different models on traffic speed datasets METR-LA and PEMS-BAY with 10-day training data. (The best performance are marked in bold, and the second best are underlined.)

Method		**METR-LA** (*PEMS-BAY→METR-LA*)											
		15 min			30 min			60 min			Average		
		MAE	**RMSE**	**MAPE(%)**	**MAE**	**RMSE**	**MAPE(%)**	**MAE**	**RMSE**	**MAPE(%)**	**MAE**	**RMSE**	**MAPE(%)**
Non-Transfer	**VAR**	6.64	12.00	11.72	8.30	14.60	14.87	10.43	17.51	19.11	8.25	14.38	14.82
	GBRT	7.60	16.29	9.63	8.99	17.81	12.61	10.54	19.09	16.97	8.87	17.47	12.62
	GRU	3.42	6.71	9.43	4.12	8.11	11.97	5.28	9.87	16.65	4.17	8.02	12.18
	DCRNN	3.30	6.64	8.72	4.08	8.32	11.11	5.30	10.50	14.94	4.10	8.23	11.21
	STGCN	3.19	6.23	8.67	3.79	7.69	10.77	4.74	9.60	14.37	3.82	7.63	10.93
	GWN	3.23	6.24	9.16	3.93	7.77	11.52	5.04	9.83	15.31	3.95	7.70	11.63
	STSGCN	3.68	7.34	9.35	4.23	8.57	11.39	5.10	10.21	15.02	4.25	8.54	11.60
	AGCRN	5.56	11.52	9.19	6.58	13.43	11.54	8.15	15.39	16.00	6.60	13.52	11.66
	HetETA	3.23	6.28	9.06	3.80	7.73	11.14	4.85	10.66	14.80	4.03	7.97	11.69
Transfer	**GRU-FT**	3.14	6.15	8.54	3.81	7.69	10.86	4.93	9.67	15.08	3.86	7.60	11.07
	DCRNN-FT	3.03	5.86	8.26	3.58	7.17	10.43	4.41	8.80	13.66	3.58	7.08	10.45
	STGCN-FT	3.07	5.99	8.44	3.61	7.34	10.46	4.44	9.11	13.96	3.63	7.29	10.66
	GWN-FT	3.05	6.53	9.75	4.12	8.13	12.29	5.25	10.07	16.48	4.13	8.00	12.43
	STSGCN-FT	3.63	7.23	9.16	3.99	8.03	10.61	4.51	9.10	12.66	4.00	8.01	10.61
	AGCRN-FT	4.62	10.04	9.72	5.82	12.34	12.64	7.64	16.53	15.83	5.86	12.75	12.47
	DASTNET	3.62	7.11	9.50	4.46	8.83	12.06	5.71	10.80	16.25	4.48	9.00	12.00
	CityTrans	**2.44**	**4.40**	**6.11**	**2.69**	**5.04**	**6.96**	**3.24**	**6.38**	**8.87**	**2.72**	**5.11**	**7.09**

Method		**PEMS-BAY** (*METR-LA→PEMS-BAY*)											
		15 min			30 min			60 min			Average		
		MAE	**RMSE**	**MAPE(%)**	**MAE**	**RMSE**	**MAPE(%)**	**MAE**	**RMSE**	**MAPE(%)**	**MAE**	**RMSE**	**MAPE(%)**
Non-Transfer	**VAR**	2.12	3.84	4.85	2.68	5.07	6.35	3.52	6.63	8.67	2.69	5.02	6.41
	GBRT	1.67	3.56	3.68	2.29	4.92	5.19	3.09	6.41	7.14	2.26	4.77	5.12
	GRU	1.61	3.40	3.36	2.12	4.83	4.74	2.98	6.47	7.55	2.14	4.71	4.94
	DCRNN	1.48	3.20	3.08	2.01	4.67	4.47	2.80	6.50	6.63	2.01	4.58	4.51
	STGCN	1.62	3.40	3.68	2.09	4.69	5.02	2.67	5.99	6.53	2.08	4.57	4.96
	GWN	1.58	3.39	3.36	2.13	4.76	5.00	2.97	6.49	7.72	2.15	4.69	5.16
	STSGCN	1.69	3.52	3.61	2.13	4.70	4.76	2.66	5.89	6.18	2.09	4.56	4.69
	AGCRN	1.60	3.39	3.51	1.95	4.39	4.39	2.62	6.03	6.14	2.00	4.58	4.53
	HetETA	1.67	3.45	3.73	2.09	4.66	4.95	2.71	5.96	6.55	2.13	4.69	5.00
Transfer	**GRU-FT**	1.57	3.41	3.30	2.12	4.99	4.81	2.86	6.42	7.35	2.09	4.73	4.88
	DCRNN-FT	1.42	3.03	3.00	1.85	4.26	4.21	2.40	5.60	5.85	1.82	4.13	4.18
	STGCN-FT	1.53	3.23	3.28	1.96	4.41	4.44	2.46	5.61	5.93	1.93	4.28	4.42
	GWN-FT	1.64	3.39	3.70	2.31	4.98	6.08	3.27	7.61	9.85	2.33	5.13	6.36
	STSGCN-FT	1.61	3.31	3.39	1.95	4.32	4.30	2.33	5.23	5.24	1.91	4.15	4.17
	AGCRN-FT	1.56	3.59	3.24	2.07	4.54	4.98	2.85	6.66	6.69	2.08	5.05	4.62
	DASTNET	1.58	3.43	3.29	2.15	4.93	4.71	2.96	6.71	6.97	2.14	5.00	5.00
	CityTrans	**1.20**	**2.37**	**2.51**	**1.47**	**3.11**	**3.23**	**1.87**	**4.18**	**4.49**	**1.46**	**3.09**	**3.28**

TABLE 4
Performance comparison of different models on air quality datasets Beijing and London with 3-month training data. (The best performance are marked in bold, and the second best are underlined.)

Method		**Beijing** (*London→Beijing*)								**London** (*Beijing→London*)							
		3 h		6 h		12 h		Average		3 h		6 h		12 h		Average	
		MAE	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
Non-Transfer	**VAR**	19.91	37.00	28.88	49.41	38.61	59.96	27.90	47.11	5.43	7.92	6.57	9.74	7.42	11.27	6.33	9.44
	GBRT	19.96	42.22	28.48	51.45	37.62	59.24	27.44	49.32	4.59	6.96	5.69	8.66	7.31	11.24	5.64	8.61
	GRU	25.22	45.56	37.13	61.54	46.19	73.16	34.86	58.40	4.56	6.79	5.62	8.26	6.79	10.17	5.39	8.01
	DCRNN	16.41	33.81	24.61	45.39	35.50	59.74	24.25	44.52	**4.06**	**5.94**	5.34	8.02	6.18	**9.27**	5.02	**7.50**
	STGCN	20.19	42.49	25.70	50.22	33.12	56.28	25.81	49.10	6.07	9.72	6.92	10.61	7.86	12.12	6.88	10.73
	GWN	22.69	42.04	34.80	57.11	45.27	70.65	32.84	54.89	5.20	7.83	6.92	10.42	8.25	12.55	6.62	10.06
	STSGCN	19.88	40.89	24.52	46.28	35.48	59.51	25.83	48.05	5.78	9.44	6.87	10.71	7.88	12.38	6.76	10.78
	AGCRN	19.13	39.16	27.14	48.18	36.58	58.71	26.90	48.75	5.74	9.40	6.70	10.50	7.36	11.50	6.49	10.40
	HetETA	24.71	48.83	28.82	52.48	35.32	58.47	29.16	52.85	6.48	10.04	7.09	10.79	8.01	12.32	7.16	11.02
Transfer	**GRU-FT**	23.93	43.88	38.66	64.21	45.69	72.10	35.05	58.74	4.33	6.45	5.58	8.24	6.68	9.90	5.34	7.92
	DCRNN-FT	17.84	36.03	26.88	47.05	38.05	59.81	26.61	46.65	4.44	6.44	5.85	8.41	7.35	11.14	5.71	8.43
	STGCN-FT	19.83	41.77	25.98	49.52	33.45	56.53	26.07	48.86	5.62	8.94	6.60	10.06	7.74	11.90	6.57	10.20
	GWN-FT	22.20	41.98	34.86	57.74	44.43	69.97	32.43	54.65	4.76	6.91	6.57	9.65	8.30	12.66	6.33	9.43
	STSGCN-FT	19.04	40.98	23.48	46.50	33.01	56.42	24.59	47.34	5.55	8.75	6.67	10.32	7.57	11.74	6.48	10.14
	AGCRN-FT	19.56	40.70	24.59	46.12	34.07	56.24	25.42	47.49	5.47	8.70	6.44	9.85	7.22	11.07	6.26	9.81
	DASTNET	17.32	34.58	25.55	46.93	35.72	58.73	25.06	45.49	4.64	6.64	5.83	8.47	6.79	10.14	5.53	8.00
	CityTrans	**13.57**	**30.21**	**18.87**	**37.92**	**25.62**	**47.75**	**18.60**	**37.52**	4.24	6.33	**5.12**	**7.69**	**6.12**	9.53	**5.01**	7.63

all evaluation metrics are 6% and 9% on PEMS04 and PEMS08, respectively. For traffic speed prediction, CityTrans is significantly outperforms the baselines. Compared with the second best results on METR-LA and PEMS-BAY, the averaged relative improvements are 28% and 22%, respectively.

- Traditional methods (i.e., VAR and GBRT) achieve competitive performance compared with the deep learning method GRU. With limited traffic data, traditional methods can still be effective in capturing the periodicity in the data to assist in improving forecasting performance.
- CityTrans is superior to non-transfer spatio-temporal models (i.e., DCRNN, STGCN, GWN, STSGCN, AGCRN, and HetETA) in most cases. This result indicates that well-designed spatio-temporal models require abundant data to achieve promising performance. However, with insufficient training data, the prediction performance of these models disastrously drop.
- Compared with the corresponding non-transfer models, almost all transfer models improve prediction performance. This result implies that the knowledge transfer between cities is effective in predicting when training data is insufficient.
- CityTrans performs superior under different source cities (e.g., *PEMS03→PEMS08* and *PEMS07→PEMS08*). This shows CityTrans is effective in reducing domain differences between source and target cities. This also demonstrates the stability of CityTrans in transferring process of different source cities.

5.2.2 Air Quality Prediction Performance Comparison

We conduct experiments on an air quality prediction task and compare CityTrans with the baselines. The performance in the next 3, 6, and 12 hours are presented. The experiments are repeated for 5 times and the average of MAE, RMSE are reported in Table 4. Most of the observations we have are similar to the traffic prediction. In addition, from this table, we have the following observations:

- In most cases, CityTrans achieves the best performance. In other cases (e.g., MAE and RMSE for 3-hour prediction and RMSE for 12-hour prediction on London dataset), CityTrans gets the comparable results. This phenomenon indicates that CityTrans can effectively achieve promising performance in different spatio-temporal prediction tasks across cities.
- CityTrans outperforms the non-transfer baselines by a large margin on Beijing dataset. Moreover, it shows consistent superiority over transfer baselines. We suggest that CityTrans is able to capture potential spatio-temporal patterns and extract transferable information, which assists in improving prediction performance.

5.2.3 Parameters Comparison

We further compare the parameter number of CityTrans with the spatio-temporal baseline models on six datasets, as shown in Table 5. STSGCN has the most parameters on all datasets since it synchronously captures the localized spatio-temporal relationships at the adjacent time. AGCRN has almost 10 times more parameters than CityTrans on PEMS04 because of the learning of the node-specific pattern. Although the parameter number of DASTNET is comparable to that of CityTrans, it is far less effective than CityTrans. That means CityTrans achieves a significant improvement in performance with a minimum number of parameters.

TABLE 5
Number of parameters for different models on datasets.

Model	# Parameters					
	Traffic Flow Datasets		Traffic Speed Datasets		Air Quality Datasets	
	PEMS04	PEMS08	METR-LA	PEMS-BAY	Beijing	London
DCRNN	149,056	149,056	149,440	149,440	149,440	149,440
STGCN	383,969	296,289	319,969	395,489	209,249	200,289
GWN	143,176	140,436	141,208	143,568	137,748	137,468
STSGCN	2,024,445	1,401,232	1,536,390	2,131,501	1,116,784	1,106,396
AGCRN	748,810	150,112	150,186	300,064	298,900	300,380
HetETA	380,005	292,325	316,005	391,525	205,285	196,325
DASTNET	85,343	85,343	85,343	85,343	85,343	85,343
CityTrans	82,497	82,497	82,561	82,561	82,561	82,561

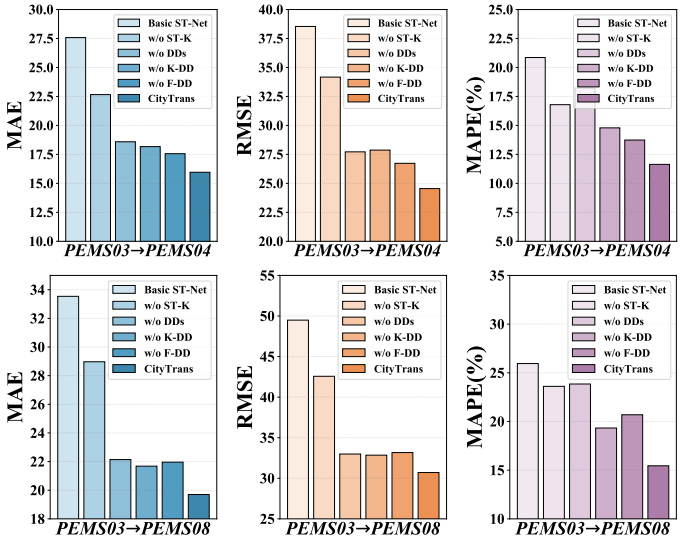


Fig. 5. Comparison the overall prediction results in variants of CityTrans on *PEMS03→PEMS04* and *PEMS03→PEMS08*.

5.3 Ablation Study (RQ2)

To demonstrate the effectiveness of the proposed ST-Knowledge, feature domain discriminator, and knowledge domain discriminator in CityTrans, we compare CityTrans with the following variants on *PEMS03→PEMS04* and *PEMS03→PEMS08*:

- Basic ST-Net: We keep ST-Net as the feature extractor and add a fully connected layer for prediction.
- w/o ST-K: We remove ST-Knowledge and the corresponding knowledge domain discriminator, as well as the knowledge attention module in the predictor.
- w/o DDs: We remove feature domain discriminator and knowledge domain discriminator.
- w/o K-DD: We remove knowledge domain discriminator.
- w/o F-DD: We remove feature domain discriminator.

Prediction performance of the variants are illustrated in Fig. 5, we make the following observations:

- Basic ST-Net shows the poor performance compared with the other variants, which manifests the effectiveness of the proposed modules in CityTrans.
- The performance of CityTrans outperforms w/o ST-K by a large margin, it indicates that ST-Knowledge is capable

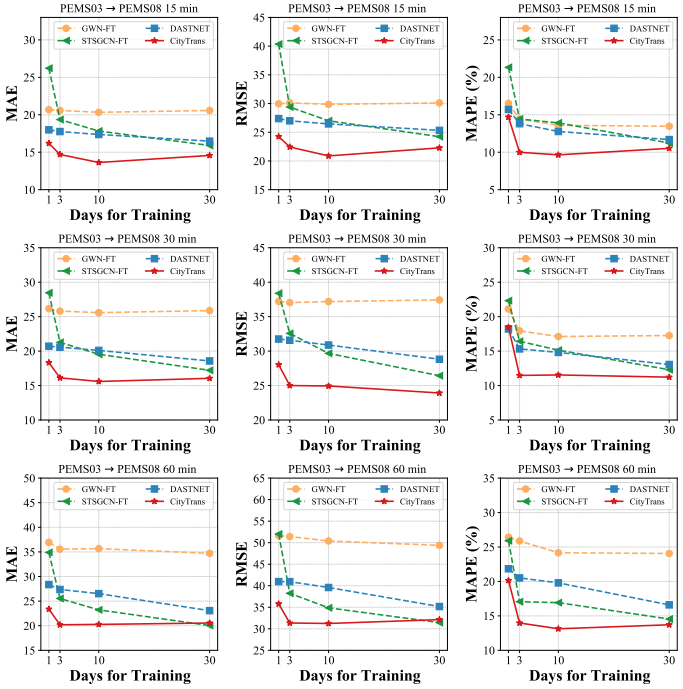


Fig. 6. Prediction performance under different size of training sets on *PEMS03→PEMS08*.

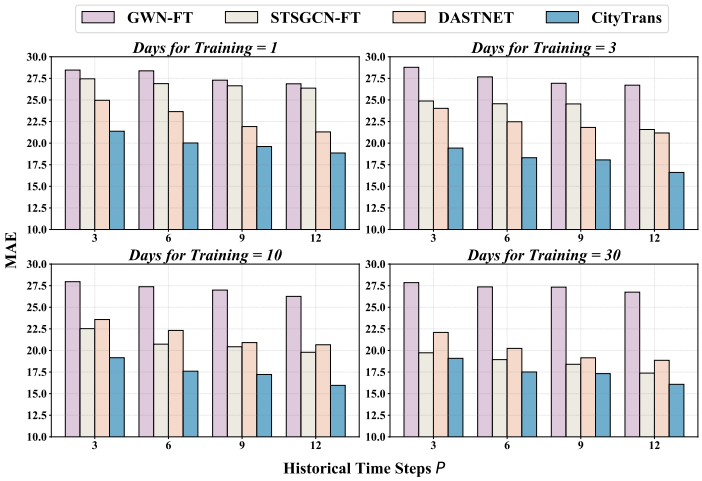


Fig. 7. Effects of different historical time steps on *PEMS03→PEMS08*.

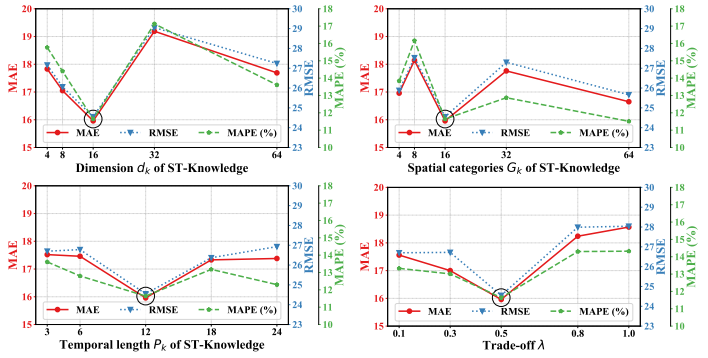


Fig. 8. Effects of different hyper-parameters on *PEMS03→PEMS08*.

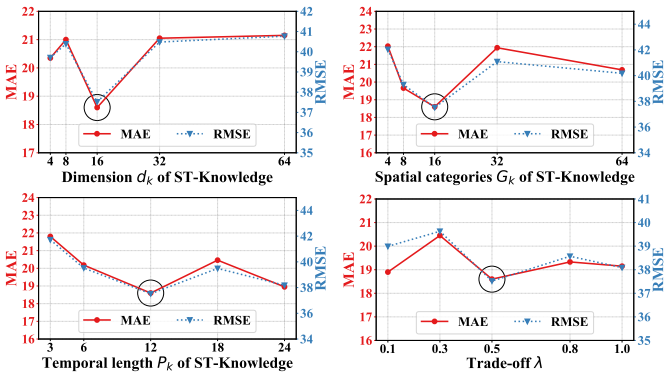
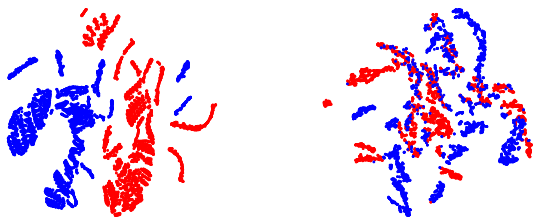


Fig. 9. Effects of different hyper-parameters on *London→Beijing*.



(a) Raw features (b) Hidden representations after domain-adversarial training

Fig. 10. Visualization of extracted features of *PEMS03→PEMS08* by t-SNE. Each point represents node representation. Blue points are from PEMS03, red points are from PEMS08.

of capturing global and useful spatio-temporal patterns for knowledge transfer.

- CityTrans shows substantial performance improvements compared with w/o DDs. Additionally, models with feature or knowledge domain discriminator (i.e., w/o K-DD and w/o F-DD) outperform the variants without either domain discriminators (i.e., w/o DDs). These phenomena suggest that domain-invariant representations obtained from domain-adversarial training contribute to prediction performance.
- Compared with w/o F-DD, CityTrans demonstrates superiority by incorporating the feature domain discriminator to mitigate the feature distribution difference between cities. Furthermore, it shows considerable improvement over both w/o K-DD and w/o F-DD.

5.4 Effects of Training Days and Historical Time Steps (RQ3)

To further explore the robustness of our proposed CityTrans, we conduct experiments on *PEMS03→PEMS08* with different sizes of training sets. We use 1-day, 3-day, 10-day, and 30-day data for training, and use the same test data as in previous experiments for testing. We compare CityTrans with the following transfer learning-based methods: GWN-FT, STSGCN-FT, and DASTNET. As Fig. 6 shows, GWN-FT performs worst on average. In the case of very small amounts of training data (i.e., 1-day and 3-day), these baseline models show very poor performance. With the increasing amount of training data, the performance of most baselines improve significantly. Despite being equipped with only a simple spatio-temporal feature extractor, CityTrans achieves comparable performance to STSGCN-FT on 30-day

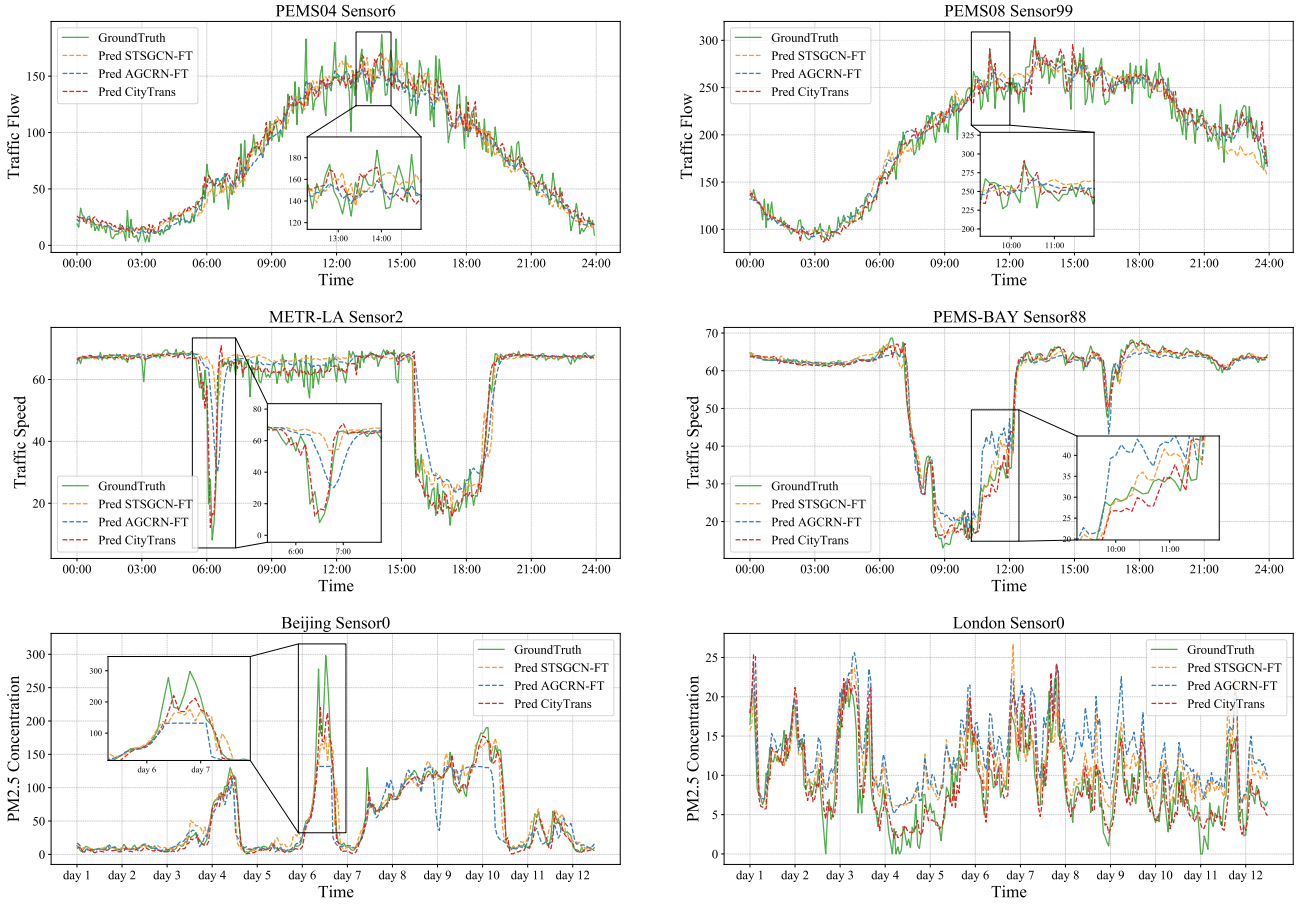


Fig. 11. Visualization predicted results by CityTrans and baselines.

training data. This result verifies our hypothesis that most existing spatio-temporal prediction methods heavily rely on a large amount of training data. Our CityTrans outperforms these models in different forecasting time steps.

Furthermore, to investigate the impact of different historical time steps on prediction performance, we conduct experiments on *PEMS03→PEMS08*, using varying lengths of historical data. We set the number of historical time steps P in Eq. (2) to 3, 6, 9, and 12, which correspond to 15, 30, 45, and 60 minutes of historical data, respectively. We vary the number of training days and utilize these steps to predict the data for the next hour. The average results are illustrated in Fig. 7. From the results, we can see that CityTrans achieves the best performance under varying historical time steps.

5.5 Parameter Sensitivity (RQ4)

We investigate how various hyper-parameters involved in CityTrans affect the prediction performance. Specifically, we analyze four key parameters: the dimension d_k, latent spatial categories G_k, underlying temporal pattern length P_k of ST-Knowledge and the trade-off λ in Eq. (16). We conduct experiments on *PEMS03→PEMS08* and *London→Beijing*. As shown in Fig. 8 and Fig. 9, we tune d_k and G_k in the grid of {4, 8, 16, 32, 64} and tune P_k in the grid of {3, 6, 12, 18, 24}. Moreover, we change λ from 0.1 to 1.0. We can observe that the performance initially improves but later decreases in most cases. One potential reason could be that ST-Knowledge provides insufficient shared knowledge for prediction at the beginning. However, with the increase of d_k, G_k and P_k, the information is redundant or irrelevant for prediction. The optimal performance of different parameters is marked with black circles in Fig. 8 and Fig. 9. Consequently, we set d_k and G_k to 16, and set P_k and λ to 12, 0.5 respectively.

5.6 Visualization of Node Representations

To verify the effectiveness of the domain-adversarial training strategy, we visualize the raw features of nodes and the hidden node representations after domain-adversarial training on *PEMS03→PEMS08* using t-SNE [53], as illustrated in Fig. 10. Blue points represent nodes from the source city PEMS03, and red points are nodes from the target city PEMS08. As shown in Fig. 10(a), the raw features of different cities are separated due to different data distributions. In contrast, we can observe that blue and red points are clustered together in Fig. 10(b). It suggests that domain-adversarial training strategy effectively mitigates distribution divergence between different cities and learns more uniformly distributed representations.

5.7 Case Study

We randomly select six sensors from six datasets (i.e., PEMS04, PEMS08, METR-LA, PEMS-BAY, Beijing and London) and visualize the predicted sequences of CityTrans and state-of-the-art baselines (i.e., STSGCN-FT and AGCRN-FT), as shown in Fig. 11. The ground truth of corresponding sequences is also plotted for comparison. We can observe that: (1) The prediction sequences generated by CityTrans are closer to the ground truth

than STSGCN-FT and AGCRN-FT; (2) CityTrans makes more accurate predictions than STSGCN-FT and AGCRN-FT when violent fluctuations occur, such as 6:00 in METR-LA. These phenomena further confirm the effectiveness and robustness of CityTrans in spatio-temporal prediction across cities.

6 Conclusion

In this paper, a novel domain adversarial model with knowledge transfer is proposed for spatio-temporal prediction in data-scarce target cities, named CityTrans. CityTrans is an end-to-end domain adversarial framework different from the models in a two-stage learning way, i.e., pre-train and fine-tune. Specifically, a self-adaptive spatio-temporal knowledge is learned to simultaneously capture the potential spatial and temporal patterns. Additionally, CityTrans is capable of jointly learning the domain-invariant features and spatio-temporal knowledge across cities via domain-adversarial training to boost the prediction performance. Experiments demonstrate that CityTrans obtains significant and consistent improvements over the baseline models on traffic flow, traffic speed, and air quality prediction tasks.

For future work, we plan to explore the following directions: (1) Investigating a transfer framework that utilizes multiple source cities to enhance prediction performance. (2) Modeling single-node representations to address the data scarcity problem resulting from newly added sensors in cities. (3) Making further attempts to model heterogeneous spatio-temporal data, such as trajectories, weather conditions, and points of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (No. 61976247).

References

[1] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.

[2] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.

[3] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of the 28th International Joint Conferences on Artificial Intelligence*, 2019.

[4] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, D. Chuxing, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, 2018.

[5] W. Zhou, Y. Yang, Y. Zhang, D. Wang, and X. Zhang, "Deep flexible structured spatial-temporal model for taxi capacity prediction," *Knowledge-Based Systems*, vol. 205, p. 106286, 2020.

[6] S. Du, T. Li, Y. Yang, and S.-J. Horng, "Deep air quality forecasting using hybrid deep learning framework," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2412–2424, 2021.

[7] X. Ouyang, Y. Yang, Y. Zhang, W. Zhou, and D. Guo, "Dual-channel spatial-temporal difference graph neural network for pm2.5 forecasting," *Neural Computing and Applications*, vol. 35, no. 10, pp. 7475–7494, 2022.

[8] 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 *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[9] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proceedings of the 28th Conference on Neural Information Processing Systems Workshop on Deep Learning*, 2014.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016.

[13] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 17 804–17 815.

[14] Y. Wei, Y. Zheng, and Q. Yang, "Transfer knowledge between cities," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1905–1914.

[15] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, "Cross-city transfer learning for deep spatio-temporal prediction," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 1893–1899.

[16] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[17] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities A meta-learning approach for spatial-temporal prediction," in *The World Wide Web Conference*, 2019, pp. 2181–2191.

[18] R. Jiang, D. Yin, Z. Wang, Y. Wang, J. Deng, H. Liu, Z. Cai, J. Deng, X. Song, and R. Shibasaki, "Dl-traff Survey and benchmark of deep learning models for urban traffic prediction," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 4515–4525.

[19] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 468–478, 2020.

[20] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the 31st AAAI conference on Artificial Intelligence*, 2017.

[21] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, "Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 1020–1027.

[22] H. Miao, J. Shen, J. Cao, J. Xia, and S. Wang, "Mba-stnet: Bayes-enhanced discriminative multi-task learning for flow prediction," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[23] L. Liu, J. Zhen, G. Li, G. Zhan, Z. He, B. Du, and L. Lin, "Dynamic spatial-temporal representation learning for traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7169–7183, 2021.

[24] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[25] D. Chai, L. Wang, and Q. Yang, "Bike flow prediction with multi-graph convolutional networks," in *Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2018, pp. 397–400.

[26] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 922–929.

[27] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

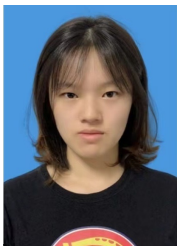
[28] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1720–1730.

[29] Z. Zhou, Y. Wang, X. Xie, L. Chen, and H. Liu, "Riskoracle: a minute-level citywide traffic accident forecasting framework," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1258–1265.

[30] Z. Zhou, Y. Wang, X. Xie, L. Chen, and C. Zhu, "Foresee urban sparse traffic accidents: A spatiotemporal multi-granularity perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3786–3799, 2022.

[31] H. Hong, Y. Lin, X. Yang, Z. Li, K. Fu, Z. Wang, X. Qie, and J. Ye, "Heteta: Heterogeneous information network embedding for estimating time of arrival," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2444–2454.

[32] B. Hui, D. Yan, H. Chen, and W.-S. Ku, "Trajnet: A trajectory-based deep learning model for traffic prediction," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 716–724.
[33] B. Hui, D. Yan, H. Chen, and W.-S. Ku, "Trajectory wavenet: A trajectory-based model for traffic forecasting," in *2021 IEEE International Conference on Data Mining*, 2021, pp. 1114–1119.
[34] C. Li, L. Bai, W. Liu, L. Yao, and S. T. Waller, "Knowledge adaption for demand prediction based on multi-task memory neural network," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 715–724.
[35] C. Tian, X. Zhu, Z. Hu, and J. Ma, "A transfer approach with attention reptile method and long-term generation mechanism for few-shot traffic prediction," *Neurocomputing*, vol. 452, pp. 15–27, 2021.
[36] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015, pp. 97–105.
[37] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 12, pp. 3071–3085, 2019.
[38] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.
[39] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.
[40] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
[41] Q. Dai, X.-M. Wu, J. Xiao, X. Shen, and D. Wang, "Graph transfer learning via adversarial domain adaptation with graph convolution," *IEEE Transactions on Knowledge and Data Engineering*, 2022.
[42] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.
[43] Y. Li, Y. Song, L. Jia, S. Gao, Q. Li, and M. Qiu, "Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2833–2841, 2021.
[44] X. Yang, C. Deng, T. Liu, and D. Tao, "Heterogeneous graph attention network for unsupervised multiple-target domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 1992–2003, 2022.
[45] Y. Du, J. Wang, W. Feng, S. Pan, T. Qin, R. Xu, and C. Wang, "Adarnn: Adaptive learning and forecasting of time series," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 402–411.
[46] S. Wang, H. Miao, J. Li, and J. Cao, "Spatio-temporal knowledge transfer for urban crowd flow prediction via deep attentive adaptation networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021.
[47] W. Shao, S. Zhao, Z. Zhang, S. Wang, M. S. Rahaman, A. Song, and F. D. Salim, "Fadacs: A few-shot adversarial domain adaptation architecture for context-aware parking availability sensing," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2021, pp. 1–10.
[48] Y. Tang, A. Qu, A. H. Chow, W. H. Lam, S. Wong, and W. Ma, "Domain adversarial spatial-temporal network: A transferable framework for short-term traffic forecasting across cities," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 1905–1915.
[49] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.
[50] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 914–921.
[51] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015, pp. 448–456.
[52] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: Mining loop detector data," *Transportation Research Record*, vol. 1748, no. 1, pp. 96–102, 2001.
[53] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

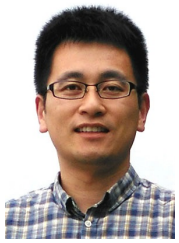


Xiaocao Ouyang received the B.S. degree from Southwest Jiaotong University, China. She is a Ph. D. candidate at School of Computing and Artificial Intelligence, Southwest Jiaotong University, China. Her research interests include deep learning, spatio-temporal data mining, and urban computing.

Yan Yang received the B.S. and M.S. degrees from Huazhong University of Science and Technology, China in1984 and 1987, respectively. She received the Ph. D. degree from Southwest Jiaotong University in 2007. In 2002/2003 and 2004/2005, she was a visiting scholar at the University of Waterloo. She is currently a professor and vice dean in School of Computing and Artificial Intelligence, Southwest Jiaotong University, China. Her research interests include artificial intelligence, big data mining, and multi-view learning.

Wei Zhou received the B.S. degree from Southwest Jiaotong University, China, in 2016. He is currently working toward the Ph. D. degree in the School of Computing and Artificial Intelligence, Southwest Jiaotong University. His research focuses on multi-view learning and spatial-temporal data mining.

Yiling Zhang received her B.S. degree from Southwest Jiaotong University, China, in 2016. In the same year, she was admitted to further study at Southwest Jiaotong University. She received the Ph.D. degree from Southwest Jiaotong University in 2022. Her research focuses on multi-task learning and multi-view learning.



Hao Wang received his PhD degree in Computer Science from Southwest Jiaotong University. He is a researcher with the Research Institute of Artificial Intelligence at Zhejiang Lab. His research interests include spatio-temporal data mining, lifelong machine learning, multi-view learning, sentiment analysis and NLP.

Wei Huang received the B.S. degree from Fujian University of Technology, Fuzhou, China in 2016. She received M.S. degree from Fujian Normal University, Fuzhou, China in 2019. She is a Ph. D. candidate at Southwest Jiaotong University, Chengdu, China. Her research interests include federated learning, deep learning, data mining, and urban computing, etc. She has published several papers in journals such as IEEE Transactions on Knowledge and Data Engineering, Information Sciences, Information Fusion etc.