Domain Adversarial Spatial-Temporal Network:
A Transferable Framework for Short-term Traffic Forecasting across Cities

Yihong Tang*
Department of Urban Planning and Design
University of Hong Kong
Hong Kong SAR, China
yihongt@connect.hku.hk

Ao Qu*
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Boston, USA
qua@mit.edu

Andy H.F. Chow
Department of Advanced Design and Systems Engineering
City University of Hong Kong
Hong Kong SAR, China
andychow@cityu.edu.hk

William H.K. Lam
Department of Civil and Environmental Engineering
The Hong Kong Polytechnic University
Hong Kong SAR, China
william.lam@polyu.edu.hk

S.C. Wong
Department of Civil Engineering
University of Hong Kong
Hong Kong SAR, China
hhecwsc@hku.hk

Wei Ma*†
Department of Civil and Environmental Engineering
Research Institute for Sustainable Urban Development
The Hong Kong Polytechnic University
Hong Kong SAR, China
wei.w.ma@polyu.edu.hk

ABSTRACT

Accurate real-time traffic forecast is critical for intelligent transportation systems (ITS) and it serves as the cornerstone of various smart mobility applications. Though this research area is dominated by deep learning, recent studies indicate that the accuracy improvement by developing new model structures is becoming marginal. Instead, we envision that the improvement can be achieved by transferring the "forecasting-related knowledge" across cities with different data distributions and network topologies. To this end, this paper aims to propose a novel transferable traffic forecasting framework: Domain Adversarial Spatial-Temporal Network (DastNet). DastNet is pre-trained on multiple source networks and fine-tuned with the target network's traffic data. Specifically, we leverage the graph representation learning and adversarial domain adaptation techniques to learn the domain-invariant node embeddings, which are further incorporated to model the temporal traffic data. To the best of our knowledge, we are the first to employ adversarial multi-domain adaptation for network-wide traffic forecasting problems. DastNet consistently outperforms all state-of-the-art baseline methods on three benchmark datasets. The trained DastNet is applied to Hong Kong's new traffic detectors, and accurate traffic predictions can be delivered immediately (within one day) when the detector is available. Overall, this study suggests an alternative to enhance the traffic forecasting methods and provides practical implications for cities lacking historical traffic data. Source codes of DastNet are available at https://github.com/YihongT/DASTNet.

KEYWORDS

Traffic Forecasting; Transfer Learning; Domain Adaptation; Adversarial Learning; Intelligent Transportation Systems

---

*These authors contributed equally to this work.
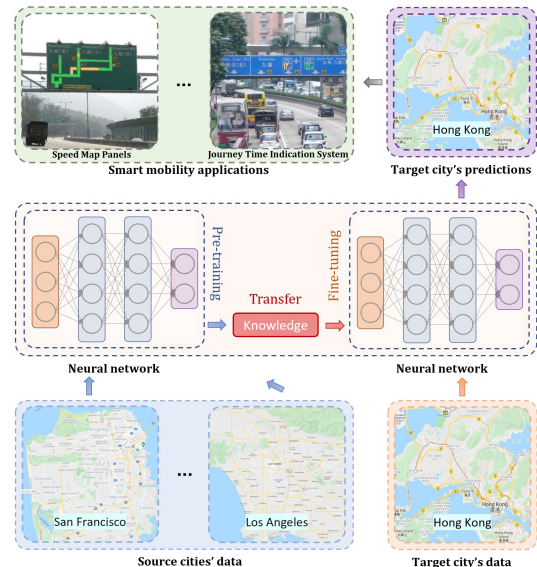†Corresponding author.

Figure 1: An overview of the transferable traffic forecasting problem and its applications.

1 INTRODUCTION

Short-term traffic forecasting [4, 24] has always been a challenging task due to the complex and dynamic spatial-temporal dependencies of the network-wide traffic states. When the spatial attributes and temporal patterns of traffic states are convoluted, their intrinsic interactions could make the traffic forecasting problem intractable. Many classical methods [11, 55] take temporal information into consideration and cannot effectively utilize spatial information. With the rise of deep learning and its application in intelligent

transportation systems (ITS) [2, 10, 65], a number of deep learning components, such as convolutional neural networks (Cnns) [38], graph neural networks (Gnns) [25], and recurrent neural networks (Rnns) [15], are employed to model the spatial-temporal characteristics of the traffic data [6, 12, 19, 27, 46]. These deep learning based spatial-temporal models achieve impressive performances on traffic forecasting tasks.

However, recent studies indicate that the improvement of the forecasting accuracy induced by modifying neural network structures has become marginal [24], and hence it is in great need to seek alternative approaches to further boost up the performance of the deep learning-based traffic forecasting models. One key observation for current traffic forecasting models is that: most existing models are designed for a single city or network. Therefore, a natural idea is to train and apply the traffic forecasting models across multiple cities, with the hope that the "knowledge related to traffic forecasting" can be transferred among cities, as illustrated in Figure 1. The idea of transfer learning has achieved huge success in the area of computer vision, language processing, and so on [30, 39, 43], while the related studies for traffic forecasting are premature [61].

There are few traffic forecasting methods aiming to adopt transfer learning to improve model performances across cities [40, 49, 52, 53, 60]. These methods partition a city into a grid map based on the longitude and latitude, and then rely on the transferability of Cnn filters for the grids. However, the city-partitioning approaches overlook the topological relationship of the road network while modeling the actual traffic states on road networks has more practical value and significance. The complexity and variety of road networks' topological structures could result in untransferable models for most deep learning-based forecasting models [35]. Specifically, we consider the road networks as graphs, and the challenge is to effectively map different road network structures to the same embedding space and reduce the discrepancies among the distribution of node embedding with representation learning on graphs.

As a practical example, Hong Kong is determined to transform into a smart city. The Smart City Blueprint for Hong Kong 2.0 was released in December 2020, which outlines the future smart city applications in Hong Kong [21]. Building an open-sourced *traffic data analytic platform* is one essential smart mobility application among those applications. Consequently, Hong Kong's Transport Department is gradually releasing the traffic data starting from the middle of 2021 [22]. As the number of detectors is still increasing now (as shown in Figure 2), the duration of historical traffic data from the new detectors can be less than one month, making it impractical to train an existing traffic forecasting model. This situation also happens in many other cities such as Paris, Shenzhen, and Liverpool [26], as the concept of smart cities just steps into the deployment phase globally. One can see that a successful transferable traffic forecasting framework could enable the smooth transition and early deployment of smart mobility applications.

To summarize, it is both theoretically and practically essential to develop a network-wide deep transferable framework for traffic forecasting across cities. In view of this, we propose a novel framework called Domain Adversarial Spatial-Temporal Network (DastNet), which is designed for the transferable traffic forecasting problem. This framework maps the raw node features to node embeddings through a spatial encoder. The embedding is induced
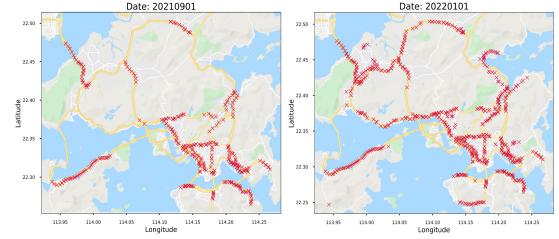

Figure 2: Available detectors in Hong Kong in September 2021 (left) and January 2022 (right).

to be domain-invariant by a domain classifier and is fused with traffic data in the temporal forecaster for traffic forecasting across cities. Overall, the main contributions of our work are as follows:

- We rigorously formulate a novel transferable traffic forecasting problem for general road networks across cities.
- We develop the domain adversarial spatial-temporal network (DastNet), a transferable spatial-temporal traffic forecasting framework based on multi-domains adversarial adaptation. To the best of our knowledge, this is the first time that the adversarial domain adaption is used in traffic forecasting to effectively learn the transferable knowledge in multiple cities.
- We conduct extensive experiments on three real-world datasets, and the experimental results show that our framework consistently outperforms state-of-the-art models.
- The trained DastNet is applied to Hong Kong's newly collected traffic flow data, and the results are encouraging and could provide implications for the actual deployment of Hong Kong's traffic surveillance and control systems such as Speed Map Panels (SMP) and Journey Time Indication System (JTIS) [48].

The remainder of this paper is organized as follows. Section 2 reviews the related work on spatial-temporal traffic forecasting and transfer learning with deep domain adaptation. Section 3 formulates the transferable traffic forecasting problem. Section 4 introduces details of DastNet. In section 5, we evaluate the performance of the proposed framework on three real-world datasets as well as the new traffic data in Hong Kong. We conclude the paper in Section 6.

2 RELATED WORKS

2.1 Spatial-Temporal Traffic Forecasting

The spatial-temporal traffic forecasting problem is an important research topic in spatial-temporal data mining and has been widely studied in recent years. Recently, researchers utilized Gnns [25, 41, 51, 57, 64] to model the spatial-temporal networked data since Gnns are powerful for extracting spatial features from road networks. Most existing works use Gnns and Rnns to learn spatial and temporal features, respectively [66]. Stgcn [63] uses Cnn to model temporal dependencies. Astgcn [19] utilizes attention mechanism to capture the dynamics of spatial-temporal dependencies. Dcrnn [28] introduces diffusion graph convolution to describe the information diffusion process in spatial networks. Dmstgcn [20] is based on Stgcn and learns the posterior graph for one day through back-propagation. [33] exploits both spatial and semantic neighbors of of each node by constructing a dynamic weighted

graph, and the multi-head attention module is leveraged to capture the temporal dependencies among nodes. Gman [67] uses spatial and temporal self-attention to capture dynamic spatial-temporal dependencies. Stgode [14] makes use of the ordinary differential equation (ODE) to model the spatial interactions of traffic flow. ST-Metanet is based on meta-learning and could conduct knowledge transfer across different time series, while the knowledge across cities is not considered [40].

Although impressive results have been achieved by works mentioned above, a few of them have discussed the transferability issue and cannot effectively utilize traffic data across cities. For example, [34] presents a multi-task learning framework for city heatmap-based traffic forecasting. [35] leverage a graph-partitioning method that decomposes a large highway network into smaller networks and uses a model trained on data-rich regions to predict traffic on unseen regions of the highway network.

2.2 Transfer Learning with Deep Domain Adaptation

The main challenge of transfer learning is to effectively reduce the discrepancy in data distributions across domains. Deep neural networks have the ability to extract transferable knowledge through representation learning methods [62]. [32] and [31] employ Maximum Mean Discrepancy (MMD) to improve the feature transferability and learn domain-invariant information. The conventional domain adaptation paradigm transfers knowledge from one source domain to one target domain. In contrast, multi-domain learning refers to a domain adaptation method in which multiple domains' data are incorporated in the training process [36, 59].

In recent years, adversarial learning has been explored for generative modeling in Generative Adversarial Networks (Gans) [17]. For example, Generative Multi-Adversarial Networks (Gmans) [13] extends Gans to multiple discriminators including formidable adversary and forgiving teacher, which significantly eases model training and enhances distribution matching. In [16], adversarial training is used to ensure that learned features in the shared space are indistinguishable to the discriminator and invariant to the shift between domains. [44] extends existing domain adversarial domain adaptation methods to multi-domain learning scenarios and proposed a multi-adversarial domain adaptation (Mada) approach to capture multi-mode structures to enable fine-grained alignment of different data distributions based on multiple domain discriminators.

3 PRELIMINARIES

In this section, we first present definitions relevant to our work then rigorously formulate the transferable traffic forecasting problem.

Definition 1 (Road Network $\mathcal{G}$). A road network is represented as an undirected graph $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}}, A_{\mathcal{G}})$ to describe its topological structure. $V_{\mathcal{G}}$ is a set of nodes with $|V_{\mathcal{G}}| = N_{\mathcal{G}}$, $E_{\mathcal{G}}$ is a set of edges, and $A_{\mathcal{G}} \in \mathbb{R}^{N_{\mathcal{G}} \times N_{\mathcal{G}}}$ is the corresponding adjacency matrix of the road network. Particularly, we consider multiple road networks consisting of $|\mathcal{I}|$ source networks and one target network. $\mathcal{G}_{S_i} = (V_{\mathcal{G}_{S_i}}, E_{\mathcal{G}_{S_i}}, A_{\mathcal{G}_{S_i}})$ denotes the ith source road network ($i \in \mathcal{I}$), $\mathcal{G}_T = (V_{\mathcal{G}_T}, E_{\mathcal{G}_T}, A_{\mathcal{G}_T})$ denotes the target road network, and we have $|V_{\mathcal{G}_{S_i}}| = N_{\mathcal{G}_{S_i}}$, $|V_{\mathcal{G}_T}| = N_{\mathcal{G}_T}$.

Definition 2 (Graph Signals X). Let $X_{\mathcal{G}} \in \mathbb{R}^{N_{\mathcal{G}} \times N_f}$ denote the traffic state observed on $\mathcal{G}$ as a graph signal with node signals $X_v \in \mathbb{R}^{N_f}$ for $v \in V_{\mathcal{G}}$, where N_f represents the number of features of each node (*e.g.*, flow, occupancy, speed). Specifically, we use $X_{\mathcal{G}}^{(t)} \in \mathbb{R}^{N_{\mathcal{G}} \times N_f}$ to denote the observation on road network $\mathcal{G}$ at time t, and $X_v^{(t)} \in \mathbb{R}^{N_f}$ denotes the observation of node v at time t, $\forall t \in \gamma$, where γ is the study time period and $v \in V_{\mathcal{G}}$.

We now define the transferable traffic forecasting problem.

Definition 3 (Transferable traffic forecasting). Given historical graph signals observed on both source and target domains as input, we can divide the transferable traffic forecasting problem into the pre-training and fine-tuning stages.

In the pre-training stage, the forecasting task $\mathcal{T}_{S_i}$ maps H' historical node (graph) signals to future H node (graph) signals on a source road network $\mathcal{G}_{S_i}$, for $v \in V_{\mathcal{G}_{S_i}}$:

$$\left[X_v^{(t-H'+1)}, \cdots, X_v^{(t)}; \mathcal{G}_{S_i} \right] \xrightarrow{\mathcal{T}_{S_i}(\cdot;\theta)} \left[\widehat{X}_v^{(t+1)}, \cdots, \widehat{X}_v^{(t+H)} \right], \quad (1)$$

where θ denotes the learned function parameters.

In the fine-tuning stage, to solve the forecasting task $\mathcal{T}_T$, the same function initialized with parameters θ shared from the pre-trained function is fine-tuned to predict graph signals on the target road network, for $v \in V_{\mathcal{G}_T}$:

$$\left[X_v^{(t-H'+1)}, \cdots, X_v^{(t)}; \mathcal{G}_T \right] \xrightarrow{\mathcal{T}_T(\cdot;\theta_*(\theta))} \left[\widehat{X}_v^{(t+1)}, \cdots, \widehat{X}_v^{(t+H)} \right], \quad (2)$$

where $\theta_*(\theta)$ denotes the function parameters adjusted from θ to fit the target domain.

Note that the topology of $\mathcal{G}_{S_i}$ can be different from that of $\mathcal{G}_T$, and $\theta_*(\theta)$ represents the process of transferring the learned knowledge θ from $\mathcal{G}_{S_i}$ to the target domain $\mathcal{G}_T$. How to construct $\theta_*(\theta)$ to make it independent of network topology is the key research question in this study. To this end, the parameter sharing mechanism in the spatial Gnns is utilized to construct $\theta_*(\theta)$ [68]. For the following sections, we consider the study time period: $\gamma = \{(t - H' + 1), \cdots, (t + H)\}$.

4 PROPOSED METHODOLOGY

In this section, we propose the domain adversarial spatial-temporal network (Dastnet) to solve the transferable traffic forecasting problem. As shown in Figure 3, Dastnet is trained with two stages, and we use two source domains in the figure for illustration. We first perform pre-training through all the source domains in turn without revealing labels from the target domain. Then, the model is fine-tuned on the target domain. We will explain the pre-training stage and fine-tuning stage in detail, respectively.

4.1 Stage 1: Pre-training on Source Domains

In the pre-training stage, Dastnet aims to learn domain-invariant knowledge that is helpful for forecasting tasks from multiple source domains. The learned knowledge can be transferred to improve the traffic forecasting tasks on the target domain. To this end, we design three major modules for Dastnet: spatial encoder, temporal forecaster, and domain classifier.
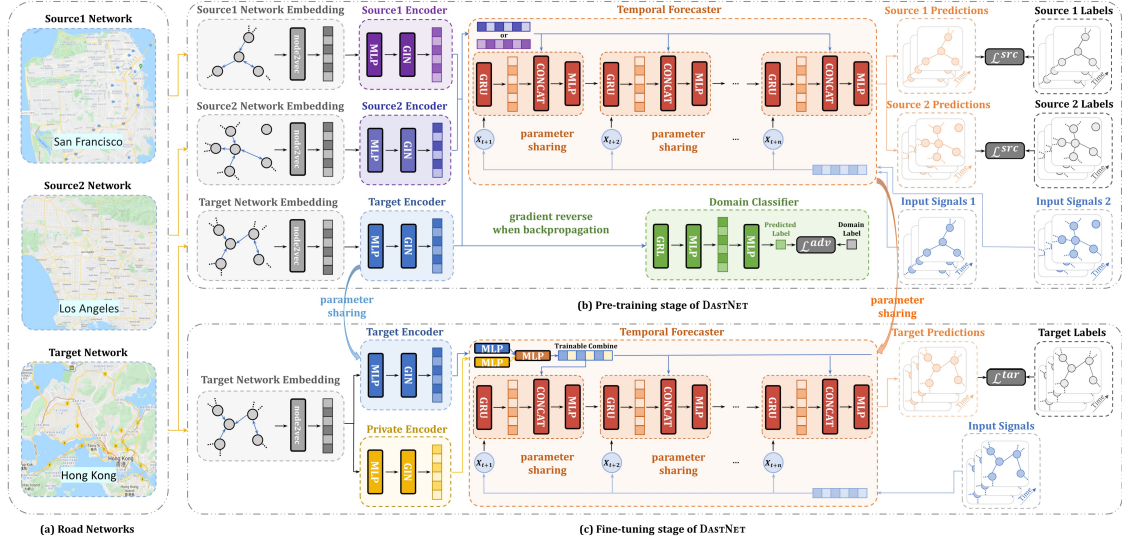
Figure 3: The proposed DASTNET architecture.

The spatial encoder aims to consistently embed the spatial information of each node in different road networks. Mathematically, given a node v's raw feature $e_v \in \mathbb{R}^{D_e}$, in which D_e is the dimension of raw features for each node, the spatial encoder maps it to a D_f-dimensional node embedding $f_v \in \mathbb{R}^{D_f}$, i.e., $f_v = M_e(e_v; \theta_e)$, where the parameters in this mapping M_e are denoted as θ_e. Note that the raw feature of a node can be obtained by a variety of methods (*e.g.*, POI information, GPS trajectories, geo-location information, and topological node representations).

Given a learned node embedding f_v for network $\mathcal{G}_i$, the temporal forecaster fulfils the forecasting task $\mathcal{T}_{S_i}$ presented in Equation 1 by mapping historical node (graph) signals to the future node (graph) signals, which can be summarized by a mapping $(\widehat{X}_v^{(t+1)}, \cdots, \widehat{X}_v^{(t+H)}) = M_y\left((X_v^{(t-H'+1)}, \cdots, X_v^{(t)}), f_v; \theta_y\right), \forall v \in V_{\mathcal{G}_{S_i}}$, and we denote the parameters of this mapping M_y as θ_y.

Domain classifier takes node embedding f_v as input and maps it to the probability distribution vector $\widehat{d}_v$ for domain labels, and we use notation d_v to denote the one-hot encoding of the actual domain label of f_v. Note that the domain labels include all the source domains and the target domain. This mapping is represented as $\widehat{d}_v = M_d(f_v; \theta_d)$. We also want to make the node embedding f_v domain-invariant. That means, under the guidance of the domain classifier, we expect the learned node embedding f_v is independent of the domain label d_v.

At the pre-training stage, we seek the parameters (θ_e, θ_y) of mappings (M_e, M_y) that minimize the loss of the temporal forecaster, while simultaneously seeking the parameters θ_d of mapping M_d that maximize the loss of the domain classifier so that it cannot identify original domains of node embeddings learned from spatial encoders. Note that the target domain's node embedding is involved in the pre-training process to guide the target spatial encoder to learn domain-invariant node embeddings. Then we can define the

loss function of the pre-training process as:

$$
\begin{aligned}
\mathcal{L}(\cdot; \theta_e, \theta_y, \theta_d) &= \mathcal{L}^{src}(\cdot; \theta_e, \theta_y) + \lambda \mathcal{L}^{adv}(\cdot; \theta_e, \theta_d) \\
&= \mathcal{L}^{src}\left(M_y\left(\cdot, M_e(e_v; \theta_e); \theta_y\right), \cdot\right) + \\
&\quad \lambda \mathcal{L}^{adv}\left(M_d\left(M_e(e_v; \theta_e); \theta_d\right), d_v\right),
\end{aligned}
\tag{3}
$$

where λ trades off the two losses. $\mathcal{L}^{src}(\cdot, \cdot)$ represents the prediction error on source domains and $\mathcal{L}^{adv}(\cdot, \cdot)$ is the adversarial loss for domain classification. Based on our objectives, we are seeking the parameters $\{\widehat{\theta}_e, \widehat{\theta}_y, \widehat{\theta}_d\}$ that reach a saddle point of $\mathcal{L}$:

$$
\begin{aligned}
\left(\widehat{\theta}_e, \widehat{\theta}_y\right) &= \arg\min_{\theta_e, \theta_y} \mathcal{L}\left(\cdot; \theta_e, \theta_y, \widehat{\theta}_d\right), \\
\widehat{\theta}_d &= \arg\max_{\theta_d} \mathcal{L}\left(\cdot; \widehat{\theta}_e, \widehat{\theta}_y, \theta_d\right).
\end{aligned}
\tag{4}
$$

Equation 4 essentially represents the min-max loss for GANs, and the following sections will discuss the details of each component in the loss function.

4.1.1 Spatial Encoder. In traffic forecasting tasks, a successful transfer of trained GNN models requires the adaptability of graph topology change between different road networks. To solve this issue, it is important to devise a graph embedding mechanism that can capture generalizable spatial information regardless of domains. To this end, we generate the raw feature e_v for a node v by node2vec [18] as the input of the spatial encoder. Raw node features learned from the node2vec can reconstruct the "similarity" extracted from random walks since nodes are considered similar to each other if they tend to co-occur in these random walks. In addition to modeling the similarity between nodes, we also want to learn localized node features to identify the uniqueness of the local topology around nodes. [58] proves that graph isomorphic network (GIN) layer is as powerful as the Weisfeiler-Lehman (WL) test [54] for distinguishing different graph structures. Thus, we adopt GIN layers with mean aggregators proposed in [58] as our

spatial encoders. Mapping $f_v = M_e(e_v; \theta_e)$ can be specified by a K-layer GIN as follows:

$$f_v^{(k)} = \text{MLP}_{\text{gin}}{}^{(k)} \left(\left(1 + \epsilon^{(k)}\right) \cdot f_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} \frac{f_u^{(k-1)}}{|\mathcal{N}(v)|} \right), \quad (5)$$

where $f_v^{(0)} = e_v$, $\mathcal{N}(v)$ denotes the neighborhoods of node v and ϵ^k is a trainable parameter, $k = 1, \cdots, K$, and K is the total number of layers in GIN. The node v's embedding can be obtained by $f_v = f_v^{(K)}$. We note that previous studies mainly use GPS trajectories to learn the location embedding [7, 56], while this study utilizes graph topology and aggregate traffic data.

4.1.2 Temporal Forecaster. The learned node embedding f_v will be involved in the mapping M_y to predict future node signals. Now we will introduce our temporal forecaster, which aims to model the temporal dependencies of traffic data. Thus, we adapted the Gated Recurrent Units (GRU), which is a powerful RNN variant [9, 15]. In particular, we extend GRU by incorporating the learned node embedding f_v into its updating process. To realize this, the learned node embedding f_v is concatenated with the hidden state of GRU (we denote the hidden state for node v at time τ as $h_v^{(\tau)}$). Details of the mapping M_y is shown below:

$$u_v^{(\tau)} = \sigma\left(\Theta_u\left[X_v^{(\tau)}; h_v^{(\tau-1)}\right] + b_u\right), \quad (6)$$

$$r_v^{(\tau)} = \sigma\left(\Theta_r\left[X_v^{(\tau)}; h_v^{(\tau-1)}\right] + b_r\right), \quad (7)$$

$$c_v^{(\tau)} = \tanh\left(\Theta_c\left[X_v^{(\tau)}; r_v^{(\tau)} \odot h_v^{(\tau-1)}\right] + b_c\right), \quad (8)$$

$$h_v^{(\tau)} = \text{MLP}_{\text{gru}}(\underset{\underset{\text{learned from spatial encoder}}{\downarrow}}{f_v}; (u_v^{(\tau)} \odot h_v^{(\tau-1)} + (1 - u_v^{(\tau)}) \odot c_v^{(\tau)})), \quad (9)$$

where $u_v^{(\tau)}$, $r_v^{(\tau)}$, $c_v^{(\tau)}$ are update gate, reset gate and current memory content respectively. Θ_u, Θ_r, and Θ_c are parameter matrices, and b_u, b_r, and b_c are bias terms.

The pre-training stage aims to minimize the error between the actual value and the predicted value. A single-layer perceptrons is designed as the output layer to map the temporal forecaster's output $h_v^{(\tau)}$ to the final prediction $\widehat{X}_v^{(\tau)}$. The source loss is represented by:

$$\mathcal{L}^{src} = \frac{1}{H} \sum_{\tau=t+1}^{t+H} \frac{1}{N_{\mathcal{G}_{S_i}}} \sum_{v \in V_{\mathcal{G}_{S_i}}} \left\| \widehat{X}_v^{(\tau)} - X_v^{(\tau)} \right\|_1. \quad (10)$$

4.1.3 Domain Classifier. The difference between domains is the main obstacle in transfer learning. In the traffic forecasting problem, the primary domain difference that leads to the model's inability to conduct transfer learning between different domains is the spatial discrepancy. Thus, spatial encoders are involved in learning domain-invariant node embeddings for both source networks and the target network in the pre-training process.

To achieve this goal, we involve a Gradient Reversal Layer (GRL) [16] and a domain classifier trained to distinguish the original domain of node embedding. The GRL has no parameters and acts as an identity transform during forward propagation. During the backpropagation, GRL takes the subsequent level's gradient, and passes its negative value to the preceding layer. In the domain classifier, given an input node embedding f_v, θ_d is optimized to

predict the correct domain label, and θ_e is trained to maximize the domain classification loss. Based on the mapping $\widehat{d}_v = M_d(f_v; \theta_d) = \text{Softmax}(\text{MLP}_d(f_v))$, $\mathcal{L}^{adv}$ is defined as:

$$\mathcal{L}^{adv} = \sum_{V \in V_{\text{all}}} -\frac{1}{|V|} \sum_{v \in V} \langle d_v, \log\left(\text{Softmax}\left(\text{MLP}_d(f_v)\right)\right)\rangle, \quad (11)$$

where $V_{all} = V_{\mathcal{G}_{S_1}} \cup V_{\mathcal{G}_{S_2}} \cup V_{\mathcal{G}_T}$, and the output of $\text{MLP}_d(f_v)$ is fed into the softmax, which computes the possibility vector of node v belonging to each domain.

By using the domain adversarial learning, we expect to learn the "forecasting-related knowledge" that is independent of time, traffic conditions, and traffic conditions. The idea of spatial encoder is also inspired by the concept of land use regression (LUR) [47], which is originated from geographical science. The key idea is that the location itself contains massive information for estimating traffic, pollution, human activities, and so on. If we can properly extract such information, the performance of location-related tasks can be improved.

4.2 Stage 2: Fine-tuning on the Target Domain

The objective of the fine-tuning stage is to utilize the knowledge gained from the pre-training stage to further improve forecasting performance on the target domain. Specifically, we adopt the parameter sharing mechanism in [39]: the parameters of the target spatial encoder and the temporal forecaster in the fine-tuning stage are initialized with the parameters trained in the pre-training stage.

Moreover, we involve a private spatial encoder combined with the pre-trained target spatial encoder to explore both domain-invariant and domain-specific node embeddings. Mathematically, given a raw node feature e_v, the private spatial encoder maps it to a domain-specific node embedding $\tilde{f}_v$, this process is represented as $\tilde{f}_v = \tilde{M}_e(e_v; \tilde{\theta}_e)$, where $\tilde{M}_e$ has the same structure as M_e and the parameter $\tilde{\theta}_e$ is randomly initialized. The pre-trained target spatial encoder maps the raw node feature e_v to a domain-invariant node embedding f_v, i.e., $f_v = M_e(e_v; \theta_{e*}(\theta_e))$, where $\theta_{e*}(\theta_e)$ means that θ_{e*} is initialized with the trained parameter θ_e in the pre-training stage. Note that $\tilde{M}_e$ and M_e are of the same structure, and the process to generate $\tilde{f}_v$ and f_v is the same as in Equation 5.

Before being incorporated into the pre-trained temporal forecaster, $\tilde{f}_v$ and f_v are combined by MLP layers to learn the combined node embedding f_v^{tar} of the target domain:

$$f_v^{tar} = \text{MLP}_{\text{cmb}}\left(\text{MLP}_{\text{pre}}(f_v) + \text{MLP}_{\text{pri}}(\tilde{f}_v)\right), \quad (12)$$

then given node signal $X_v^{(\tau)}$ ($v \in V_{\mathcal{G}_T}$) at time τ and f_v^{tar} as input, $\widehat{X}_v^{(\tau)}$ is computed based on Equation 6, 7, 8, and 9. We denote the target loss at the fine-tuning stage as:

$$\mathcal{L}^{tar} = \frac{1}{H} \sum_{\tau=t+1}^{t+H} \frac{1}{N_{\mathcal{G}_T}} \sum_{v \in V_{\mathcal{G}_T}} \left\| \widehat{X}_v^{(\tau)} - X_v^{(\tau)} \right\|_1. \quad (13)$$

5 EXPERIMENTS

We first validate the performance of DASTNET using benchmark datasets, and then DASTNET is experimentally deployed with the newly collected data in Hong Kong.

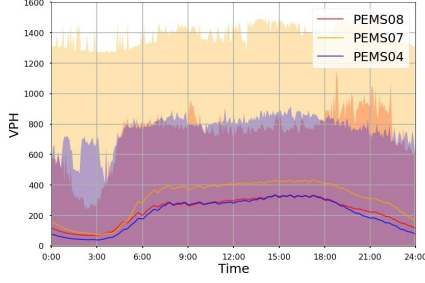## 5.1 Offline Validation with Benchmark Datasets

Figure 4: Within-day traffic flow distributions.

We evaluate the performance of DastNet on three real-world datasets, PEMS04, PEMS07, PEMS08, which are collected from the Caltrans Performance Measurement System (PEMS) [37] every 30 seconds. There are three kinds of traffic measurements in the raw data: speed, flow, and occupancy. In this study, we forecast the traffic flow for evaluation purposes and it is aggregated to 5-minute intervals, which means there are 12 time intervals for each hour and 288 time intervals for each day. The unit of traffic flow is veh/hour (vph). The within-day traffic flow distributions are shown in Figure 4. One can see that flow distributions vary significantly over the day for different datasets, and hence domain adaption is necessary.

The road network for each dataset are constructed according to actual road networks, and we defined the adjacency matrix based on connectivity. Mathematically, $A_{i,j} = \begin{cases} 1, & \text{if } v_i \text{ connects to } v_j \\ 0, & \text{otherwise} \end{cases}$, where v_i denotes node i in the road network. Moreover, we normalize the graph signals by the following formula: $X = \frac{X - \text{mean}(X)}{\text{std}(X)}$, where function mean and function std calculate the mean value and the standard deviation of historical traffic data respectively.

5.1.1 Baseline Methods.
- HA [29]: Historical Average method uses average value of historical traffic flow data as the prediction of future traffic flow.
- SVR [45]: Support Vector Regression adopts support vector machines to solve regression tasks.
- GRU [8]: Gated Recurrent Unit (GRU) is a well-known variant of RNN which is powerful at capturing temporal dependencies.
- GCN [25]: Graph Convolutional Network can handle arbitrary graph-structured data and has been proved to be powerful at capturing spatial dependencies.
- TGCN [66]: Temporal Graph Convolutional Network performs stably well for short-term traffic forecasting tasks.
- STGCN [63]: Spatial-Temporal Graph Convolutional Network uses ChebNet and 2D convolutions for traffic prediction.
- DCRNN [28]: Diffusion Convolutional Recurrent Neural Network combines GNN and RNN with diffusion convolutions.
- AGCRN [1]: Adaptive Graph Convolutional Recurrent Network learns node-specific patterns through node adaptive parameter learning and data-adaptive graph generation.
- STGODE [14]: Spatial-Temporal Graph Ordinary Differential Equation Network capture spatial-temporal dynamics through a tensor-based ODE.

To demonstrate the effectiveness of each key module in DastNet, we compare with some variants of DastNet as follows:

- TEMPORAL FORECASTER: DastNet with only the TEMPORAL FORECASTER component. This method only uses graph signals as input for pre-training and fine-tuning.
- TARGET ONLY: DastNet without training at the pre-training stage. The comparison with this baseline method demonstrate the merits of training on other data sources.
- DASTNET W/O DA: DastNet without the adversarial domain adaptation (domain classifier). The comparison with this baseline method demonstrate the merits of domain-invariant features.
- DASTNET W/O PRI: DastNet without the private encoder at the fine-tuning stage.

Above variants' other settings are the same as DastNet.

5.1.2 Experimental Settings.
To simulate the lack of data, for each dataset, we randomly select ten consecutive days' traffic flow data from the original training set as our training set, and the validation/testing sets are the same as [27]. We use one-hour historical traffic flow data for training and forecasting traffic flow in the next 15, 30, and 60 minutes (horizon=3, 6, and 12, respectively). For one dataset $\mathcal{D}$, DastNet-related methods and Trans Gru are pre-trained on the other two datasets, and fine-tuned on $\mathcal{D}$. Other methods are trained on $\mathcal{D}$. All experiments are repeated 5 times. Other hyper-parameters are determined based on the validation set. We implement our framework based on Pytorch [42] on a virtual workstation with two 11G memory Nvidia GeForce RTX 2080Ti GPUs. To suppress noise from the domain classifier at the early stages of the pre-training procedure, instead of fixing the adversarial domain adaptation loss factor $\mathcal{F}$. We gradually change it from 0 to 1: $\mathcal{F} = \frac{2}{1+\exp(-\eta \cdot \mathcal{P})} - 1$, where $\mathcal{P} = \frac{\text{current step}}{\text{total steps}}$, η was set to 10 in all experiments. We select the SGDM optimizer for stability and set the maximum epochs for fine-tuning stage to 2000 and set K of GIN encoders as 1 and 64 as the dimension of node embedding. For all model we set 64 as the batch size. For node2vec settings, we set $p = q = 1$, and each source node conduct 200 walks with 8 as the walk length and 64 as the embedding dimension.

Table 1 shows the performance comparison of different methods for traffic flow forecasting. Let $x_i \in X$ denote the ground truth and $\hat{x}_i$ represent the predicted values, and Ω denotes the set of training samples' indices. The performance of all methods are evaluated based on (1) Mean Absolute Error (MAE$(x, \hat{x}) = \frac{1}{|\Omega|} \sum_{i \in \Omega} |x_i - \hat{x}_i|$), which is a fundamental metric to reflect the actual situation of the prediction accuracy. (2) Root Mean Squared Error (RMSE$(x, \hat{x}) = \sqrt{\frac{1}{|\Omega|} \sum_{i \in \Omega} (x_i - \hat{x}_i)^2}$), which is more sensitive to abnormal values. (3) Mean Absolute Percentage Error (MAPE$(x, \hat{x}) = \frac{1}{|\Omega|} \sum_{i \in \Omega} \left| \frac{x_i - \hat{x}_i}{x_i} \right|$). It can be seen that DastNet achieves the state-of-the-art forecasting performance on the three datasets for all evaluation metrics and all prediction horizons. Traditional statistical methods like HA and SVR are less powerful compared to deep learning methods such as GRU. The performance of GCN is low, as it overlooks the temporal patterns of the data. DastNet outperforms existing spatial-temporal models like TGCN, STGCN, DCRNN, AGCRN and the state-of-the-art method STGODE.

Table 1: Performance comparison of different methods. (mean ± std)

PEMS04	15min			30min			60min		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
HA	28.36±0.00	40.55±0.00	20.14±0.00	31.75±0.00	45.14±0.00	22.84±0.00	38.52±0.00	54.45±0.00	28.48±0.00
SVR	21.21±0.05	29.68±0.07	16.05±0.14	23.90±0.04	33.51±0.02	18.74±0.40	29.24±0.14	41.14±0.10	23.46±0.73
GRU	20.96±0.29	31.08±0.20	14.78±1.86	22.71±0.21	33.77±0.19	16.54±1.73	26.25±0.28	38.87±0.32	18.66±1.95
GCN	48.65±0.04	68.89±0.06	40.53±0.88	49.49±0.05	69.97±0.06	41.42±0.78	51.63±0.06	72.65±0.07	44.03±0.49
TGCN	24.09±1.35	34.31±1.59	18.26±1.38	25.22±0.96	36.09±1.22	19.34±1.07	27.16±0.65	38.76±0.94	20.84±0.37
STGCN	27.03±1.30	38.26±1.35	25.16±4.33	27.91±0.88	39.65±0.78	25.33±5.06	35.55±2.43	49.12±4.01	37.74±5.15
DCRNN	23.73±0.62	34.27±0.71	18.84±0.75	26.68±0.94	37.63±1.00	21.39±1.90	33.79±1.77	46.70±1.91	29.68±1.76
AGCRN	24.58±0.35	42.30±0.30	14.93±0.13	26.53±0.20	48.05±0.52	15.30±0.36	30.06±0.29	52.19±0.55	16.67±0.07
STGODE	20.73±0.04	31.97±0.06	15.79±0.22	23.14±0.08	35.55±0.23	17.66±0.16	27.24±0.08	41.05±0.10	23.86±0.38
TEMPORAL FORECASTER	20.70±0.60	30.80±0.46	14.72±1.91	22.22±0.15	33.19±0.13	15.53±0.76	25.88±0.09	38.33±0.12	17.84±1.09
TARGET ONLY	19.81±0.06	29.77±0.03	13.95±0.47	21.55±0.09	32.26±0.13	14.83±0.21	24.59±0.13	36.31±0.15	17.45±0.39
DASTNET w/o DA	19.65±0.11	29.52±0.14	13.53±0.35	21.57±0.41	32.26±0.76	15.09±0.54	23.84±0.10	35.21±0.14	17.03±0.44
DASTNET w/o PRI	19.35±0.09	29.05±0.15	13.54±0.24	21.00±0.54	31.40±0.87	14.61±0.31	22.96±0.38	34.02±0.54	16.51±0.58
DASTNET	**19.25±0.03**	**28.91±0.05**	**13.30±0.22**	**20.67±0.07**	**30.78±0.04**	**14.56±0.31**	**22.82±0.08**	**33.77±0.13**	**16.10±0.18**

PEMS07	15min			30min			60min		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
HA	32.85±0.00	46.56±0.00	15.10±0.00	37.09±0.00	52.38±0.00	17.26±0.00	45.43±0.00	63.93±0.00	21.66±0.00
SVR	23.36±0.38	32.30±0.28	14.97±1.41	27.33±0.30	37.60±0.22	19.23±0.89	36.90±0.98	49.13±0.77	33.50±2.83
GRU	23.77±0.49	34.49±0.52	11.21±0.66	25.31±0.37	37.85±0.38	12.87±2.08	29.39±0.25	43.89±0.35	13.26±0.37
GCN	50.81±0.56	71.67±0.50	36.47±1.57	51.94±0.24	73.18±0.30	39.10±1.26	55.09±0.07	77.15±0.10	41.46±0.42
TGCN	30.18±0.41	42.11±0.56	15.74±0.99	30.84±2.77	43.58±3.37	15.19±1.59	33.25±1.45	47.24±1.82	16.58±1.04
STGCN	34.14±6.13	48.58±7.32	19.67±6.38	39.50±2.76	43.58±3.37	15.09±1.59	43.45±2.50	60.67±3.23	27.57±1.36
DCRNN	26.66±1.23	37.66±1.39	16.68±1.31	31.06±1.39	43.38±1.75	19.94±2.48	51.09±6.82	66.26±7.42	48.29±17.74
AGCRN	35.16±0.23	64.08±0.45	11.88±0.12	35.10±0.25	63.78±0.44	11.98±0.14	39.00±1.74	68.44±0.41	13.98±0.04
STGODE	22.30±0.13	33.89±0.14	10.92±0.20	26.02±0.18	38.52±0.14	14.23±0.57	30.87±0.43	45.27±0.25	17.21±1.57
TEMPORAL FORECASTER	23.11±0.54	34.07±0.38	10.97±1.25	24.70±0.20	37.13±0.22	10.98±0.58	28.55±0.18	42.72±0.22	12.67±0.17
TARGET ONLY	21.71±0.13	32.93±0.22	9.41±0.11	24.61±1.00	37.15±1.46	10.80±0.69	28.88±0.65	43.13±0.98	13.18±0.77
DASTNET w/o DA	21.80±0.26	33.09±0.44	9.45±0.18	24.52±0.55	37.05±0.94	10.77±0.42	28.61±0.56	42.88±0.91	12.74±0.42
DASTNET w/o PRI	21.23±0.14	32.28±0.24	9.20± 0.15	23.85±0.47	36.10±0.71	10.51±0.22	28.37±1.06	42.51±1.64	12.74±0.50
DASTNET	**20.91±0.03**	**31.85±0.05**	**8.95±0.13**	**22.96±0.10**	**34.80±0.11**	**9.87±0.19**	**26.88±0.28**	**40.12±0.29**	**11.75±0.33**

PEMS08	15min			30min			60min		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
HA	23.12±0.00	33.03±0.00	14.61±0.00	26.12±0.00	37.16±0.00	16.55±0.00	32.15±0.00	45.41±0.00	20.60±0.00
SVR	37.63±2.42	46.59±2.56	20.79±1.47	45.79±2.59	56.16±2.70	24.29±1.02	66.91±3.82	79.72±4.07	33.20±1.86
GRU	16.69±0.40	24.72±0.41	11.05±0.93	18.89±0.67	28.14±0.65	13.45±3.18	20.94±0.24	31.32±0.19	15.20±0.94
GCN	64.63±0.08	87.30±0.10	90.32±1.83	65.09±0.06	87.87±0.08	91.64±1.12	66.24±0.11	89.21±0.10	94.01±1.93
TGCN	20.65±0.96	28.77±1.13	15.06±1.20	21.60±1.44	30.40±1.78	15.97±2.42	24.33±2.51	34.20±3.14	17.91±4.77
STGCN	25.90±1.60	35.58±1.98	18.91±2.35	26.20±1.75	36.52±2.34	17.73±0.74	31.89±4.23	43.94±5.56	20.99±2.41
DCRNN	20.61±0.97	29.03±1.08	20.36±1.62	23.23±1.24	32.76±1.44	24.53±2.77	39.14±7.12	51.97±8.41	47.62±19.08
AGCRN	18.50±0.16	30.76±0.30	10.77±0.09	19.45±0.12	32.34±0.23	11.30±0.09	23.44±0.13	37.55±0.19	13.71±0.07
STGODE	20.42±0.69	37.92±3.06	17.82±1.08	23.41±0.48	36.41±2.89	21.00±2.18	26.86±0.28	39.85±0.57	24.43±0.02
TEMPORAL FORECASTER	15.99±0.10	23.95±0.11	9.93±0.45	17.77±0.40	26.56±0.31	12.08±1.75	20.03±0.33	29.86±0.21	14.80±2.10
TARGET ONLY	16.50±0.12	24.58±0.12	11.07±0.16	17.95±1.04	26.63±1.24	11.90±2.09	19.69±0.33	29.37±0.40	12.48±0.37
DASTNET w/o DA	16.51±0.30	24.50±0.38	10.55±0.99	17.58±0.81	26.31±1.21	11.22±0.76	19.37±0.46	28.87±0.54	11.95±0.36
DASTNET w/o PRI	15.75±0.25	23.60±0.41	10.00±0.22	16.87±0.38	25.38±0.68	10.55±0.14	18.90±0.20	28.28±0.20	12.52±0.64
DASTNET	**15.26±0.18**	**22.70±0.17**	**9.64±0.37**	**16.41±0.34**	**24.57±0.39**	**10.46±0.31**	**18.84±0.12**	**28.06±0.17**	**11.72±0.29**

DASTNET achieves approximately 9.4%, 8.6% and 10.9% improvements compared to the best baseline method in MAE, RMSE, MAPE, respectively. Table 2 summarize the improvements of our methods, where "-" denotes no improvements..

Table 2: Comparison between 1) GRU and the TARGET ONLY (Upper); 2) DASTNET and the best baseline (Lower).

Impv.	15min			30min			60min		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
04	5.4%	4.2%	6.2%	5.1%	4.4%	10%	6.3%	6.5%	6.5%
07	8.6%	4.5%	16%	2.8%	1.8%	16%	1.7%	1.7%	0.6%
08	1.1%	0.6%	-	5.0%	5.4%	11.5%	6.0%	6.2%	18.0%

Impv.	15min			30min			60min		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
04	7.1%	2.6%	10.9%	9.0%	8.2%	4.8%	13.1%	12.9%	3.4%
07	6.2%	6.5%	18.0%	9.3%	7.4%	17.6%	8.5%	8.6%	11.4%
08	8.6%	8.2%	10.5%	13.1%	12.7%	7.4%	10.0%	10.4%	14.5%

Ablation Study. From Table 1, MAE, RMSE and MAPE of the TARGET ONLY are reduced by approximately 4.7%, 7% and 10.6% compared to GRU (see Table 2), which demonstrates that the temporal forecaster outperforms GRU due to the incorporation of the learned node embedding. The accuracy of DASTNET is superior to TARGET ONLY, DASTNET w/o DA, TEMPORAL FORECASTER and DASTNET w/o PRI, which shows the effectiveness of pre-training, adversarial domain adaptation, spatial encoders and the private encoder. Interestingly, the difference between the results of the DASTNET and DASTNET w/o PRI on PEMS07 is generally larger than that on dataset PEMS04 and PEMS08. According to Figure 4, we know that the data distribution of PEMS04 and PEMS08 datasets are similar, while the data distribution of PEMS07 is more different from that of PEMS04 and PEMS08. This reflects differences between spatial domains and further implies that our private encoder can capture the specific domain information and supplement the information learned from the domain adaptation.

Effects of Domain Adaptation. To demonstrate the effectiveness of the proposed adversarial domain adaptation module, we visualize the raw feature of the node e_v (generated from node2vec) and the learned node embedding f_v (generated from spatial encoders) in Figure 5 using t-SNE [50]. As illustrated, node2vec learns graph connectivity for each specific graph, and hence the raw features are separate in Figure 5. In contrast, the adversarial training process
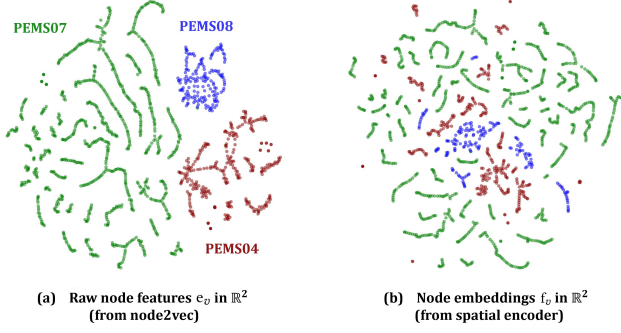
(a) Raw node features e_v in $\mathbb{R}^2$
(from node2vec)

(b) Node embeddings f_v in $\mathbb{R}^2$
(from spatial encoder)

Figure 5: Visualization of e_v and f_v by t-SNE.

successfully guides the spatial encoder to learn more uniformly distributed node embeddings on different graphs.
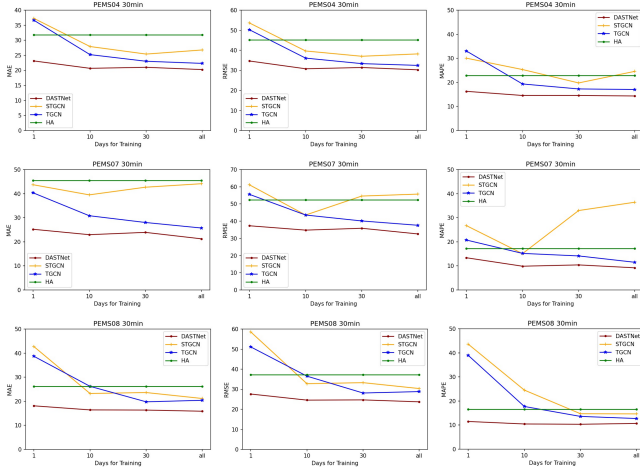

Figure 6: Sensitivity analysis, future 30-minute traffic flow forecasting results under different training set sizes.

Sensitivity Analysis. To further demonstrate the robustness of DASTNET, we conduct additional experiments with different sizes of training sets. We change the number of days of traffic flow data in the training set. To be more specific, we use four training sets with 1 day, 10 days, 30 days and all data, respectively. Then we compare DASTNET with STGCN and TGCN. The performance of DCRNN degrades drastically when the training set is small. To ensure the complete display in the Figure, we do not include it in the comparison and we do not include STGODE because of its instability. We measure the performance of DASTNET and the other two models on PEMS04, PEMS07, and PEMS08, by changing the ratio (measured in days) of the traffic flow data contained in the training set.

Experimental results of the sensitivity analysis are provided in Figure 6. In most cases, we can see that STGCN and TGCN underperform HA when the training set is small. On the contrary, DASTNET consistently outperforms other models in predicting different future time intervals of all datasets. Another observation is that the improvements over baseline methods are more significant for few-shot settings (small training sets). Specifically, the approximate gains

on MAE decrease are 42.1%/ 23.3% /14.7% /14.9% on average for 1/10/30/all days for training compared with TGCN and 46.7%/35.7% /30.7% /34% compared with STGCN.

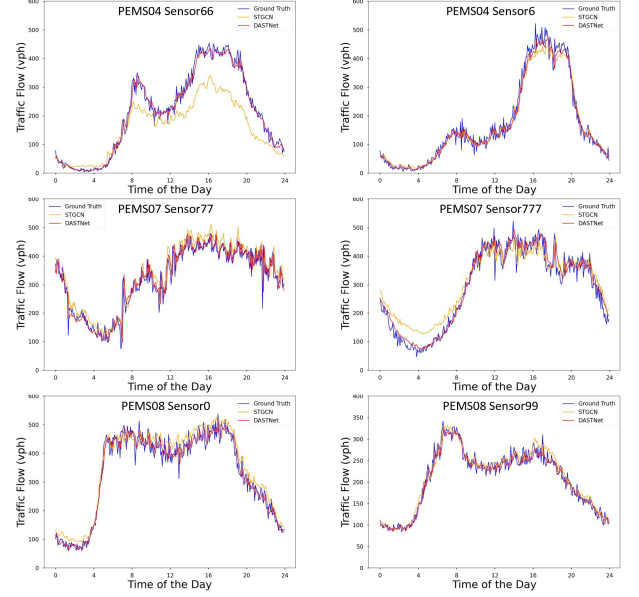**Figure 7: Visualization of the predicted flow.**

Case Study. We randomly select six detectors and visualize the predicted traffic flow sequences of DASTNET and STGCN follow the setting in [14], and the visualizations are shown in Figure 7. Ground true traffic flow sequence is also plotted for comparison. One can see that the prediction generated by DASTNET are much closer to the ground truth than that by STGCN. STGCN could accurately predict the peak traffic , which might be because DASTNET learns the traffic trends from multiple datasets and ignores the small oscillations that only exist in a specific dataset.

5.2 Experimental Deployment in Hong Kong

By the end of 2022, we aim to deploy a traffic information provision system in Hong Kong using traffic detector data on strategic routes from the Transport Department [23]. The new system could supplement the existing Speed Map Panels (SMP) and Journey Time Indication System (JTIS) by employing more reliable models and real-time traffic data. For both systems, flow data is essential and collected from traffic detectors at selected locations for the automatic incident detection purpose, as the JTIS and SMP make use of the flow data to simulate the traffic propagation, especially after car crashes [48]. Additionally, DASTNET could be further extended for speed forecasting. As we discussed in Section 1, the historical traffic data for the new detectors in Hong Kong are very limited. Figure 8 demonstrates: a) the spatial distribution of the newly deployed detectors in January 2022 and b) the corresponding traffic flow in Hong Kong. After the systematic process of the raw data as presented in c), traffic flow on the new detectors can be predicted and fed into the downstream applications once the detector is available.

We use the traffic flow data from three PEMS datasets for pretraining, and use Hong Kong's traffic flow data on January 10,
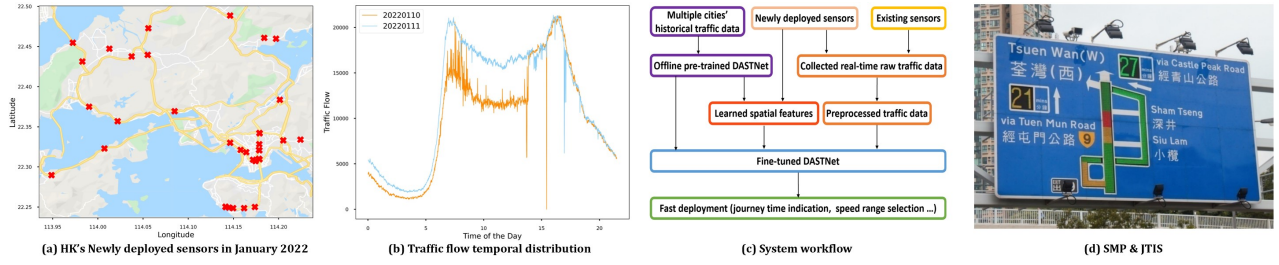
(a) HK's Newly deployed sensors in January 2022   (b) Traffic flow temporal distribution   (c) System workflow   (d) SMP & JTIS

Figure 8: Traffic data and system workflow for the experimental deployment of DastNet in Hong Kong.

2022 to fine-tune our model. All Hong Kong's traffic flow data on January 11, 2022 are used as the testing set. We use 614 traffic detectors (a combinations of video detectors and automatic licence plate recognition detectors) to collect Hong Kong's traffic flow data for the deployment of our system, and the raw traffic flow data is aggregated to 5-minute intervals. We construct Hong Kong's road network $\mathcal{G}_{HK}$ based on distances between traffic detectors and define the adjacency matrix through connectivity.. Meanwhile, Ha and spatial-temporal baseline methods Tgcn, Stgcn and Stgode are adopted for comparisons. All experiments are repeated for 5 times, and the average results are shown in Table 3. One can read from the table that, with the trained DastNet from other datasets, accurate traffic predictions can be delivered to the travelers immediately (after one day) when the detector data is available.

Table 3: Performance comparison on the newly collected data in Hong Kong.

HK	15min			30min			60min		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Ha	15.79	23.95	16.96%	17.84	27.00	18.70%	21.66	33.41	22.42%
Tgcn	22.39	30.50	27.54%	22.39	30.48	26.76%	25.95	35.61	27.98%
Stgcn	39.86	55.79	46.80%	39.34	55.34	45.62%	42.52	58.95	52.94%
Stgode	63.46	86.08	54.77%	66.19	87.36	69.23%	66.76	92.83	58.65%
DastNet	**11.71**	**17.69**	**12.89**%	**13.87**	**21.25**	**14.91**%	**17.09**	**26.47**	**18.24**%

6 CONCLUSION

In this study, we formulated the transferable traffic forecasting problem and proposed an adversarial multi-domain adaptation framework named Domain Adversarial Spatial-Temporal Network (DastNet). This is the first attempt to apply adversarial domain adaptation to network-wide traffic forecasting tasks on the general graph-based networks to the best of our knowledge. Specifically, DastNet is pre-trained on multiple source datasets and then fine-tuned on the target dataset to improve the forecasting performance. The spatial encoder learns the uniform node embedding for all graphs, the domain classifier forces the node embedding domain-invariant, and the temporal forecaster generates the prediction. DastNet obtained significant and consistent improvements over baseline methods on benchmark datasets and will be deployed in Hong Kong to enable the smooth transition and early deployment of smart mobility applications.

We will further explore the following aspects for future work: (1) Possible ways to evaluate, reduce and eliminate discrepancies of

time-series-based graph signal sequences across different domains. (2) The effectiveness of the private encoder does not conform to domain adaptation theory [3], and it is interesting to derive theoretical guarantees for the necessity of the private encoder on target domains. In the experimental deployment, we observe that the performance of existing traffic forecasting methods degrades drastically when the traffic flow rate is low. However, this situation is barely covered in the PEMS datasets, which could potentially make the current evaluation of traffic forecasting methods biased. (3) The developed framework can potentially be utilized to learn the node embedding for multi-tasks, such as forecasting air pollution, estimating population density, etc. It would be interesting to develop a model for a universal location embedding [5], which is beneficial for different types of location-related learning tasks [7, 56].

REFERENCES

[1] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in Neural Information Processing Systems* 33 (2020), 17804–17815.

[2] Richard Barnes, Senaka Buthpitiya, James Cook, Alex Fabrikant, Andrew Tomkins, and Fangzhou Xu. 2020. BusTr: Predicting Bus Travel Times from Real-Time Traffic. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3243–3251.

[3] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79, 1 (2010), 151–175.

[4] Ella Bolshinsky and Roy Friedman. 2012. *Traffic flow forecast survey*. Technical Report. Computer Science Department, Technion.

[5] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).

[6] Ling Cai, Krzysztof Janowicz, Gengchen Mai, Bo Yan, and Rui Zhu. 2020. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS* 24, 3 (2020), 736–755.

[7] Yile Chen, Xiucheng Li, Gao Cong, Zhifeng Bao, Cheng Long, Yiding Liu, Arun Kumar Chandran, and Richard Ellison. 2021. Robust Road Network Representation Learning: When Traffic Patterns Meet Traveling Semantics. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 211–220.

[8] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[9] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR* abs/1412.3555 (2014). arXiv:1412.3555 http://arxiv.org/abs/1412.3555

[10] Rui Dai, Shenkun Xu, Qian Gu, Chenguang Ji, and Kaikui Liu. 2020. Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3074–3082.

[11] Harris Drucker, Chris JC Burges, Linda Kaufman, Alex Smola, Vladimir Vapnik, et al. 1997. Support vector regression machines. *Advances in neural information processing systems* 9 (1997), 155–161.

[12] Shengdong Du, Tianrui Li, Xun Gong, Yan Yang, and Shi Jinn Horng. 2017. Traffic flow forecasting based on hybrid deep learning framework. In *2017 12th international conference on intelligent systems and knowledge engineering (ISKE)*. IEEE, ISKE, 1–6.

[13] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. 2016. Generative multiadversarial networks. *arXiv preprint arXiv:1611.01673* (2016).

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

[15] Rui Fu, Zuo Zhang, and Li Li. 2016. Using LSTM and GRU neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE, 324–328.

[16] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. PMLR, 1180–1189.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[18] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

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

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

[21] HKGov. 2022. Hong Kong Smart City Blueprint. https://www.smartcity.gov.hk/.

[22] HKGov. 2022. Hong Kong's Transport Department. https://data.gov.hk/en/.

[23] HKGov. 2022. Traffic Detectors on strategic routes. https://www.td.gov.hk/en/transport_in_hong_kong/its/intelligent_transport_systems_strategy_review_and_/traffic_detectors/index.html.

[24] Weiwei Jiang and Jiayun Luo. 2021. Graph neural network for traffic forecasting: A survey. *arXiv preprint arXiv:2101.11174* (2021).

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

[26] Vivacity Labs. 2022. Sustainable Travel Innovations by Liverpool John Moores University. https://vivacitylabs.com/sustainable-travel-innovation-liverpool/

[27] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Depeng Jin, and Yong Li. 2021. Dynamic Graph Convolutional Recurrent Network for Traffic Prediction: Benchmark and Solution. *arXiv preprint arXiv:2104.14917* (2021).

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

[29] Jing Liu and Wei Guan. 2004. A summary of traffic flow forecasting methods [J]. *Journal of highway and transportation research and development* 3 (2004), 82–85.

[30] Ruijun Liu, Yuqian Shi, Changjiang Ji, and Ming Jia. 2019. A survey of sentiment analysis based on transfer learning. *IEEE Access* 7 (2019), 85401–85412.

[31] Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Transferable representation learning with deep adaptation networks. *IEEE transactions on pattern analysis and machine intelligence* 41, 12 (2018), 3071–3085.

[32] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*. PMLR, 97–105.

[33] Bin Lu, Xiaoying Gan, Haiming Jin, Luoyi Fu, and Haisong Zhang. 2020. Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1025–1034.

[34] Yichao Lu. 2021. Learning to Transfer for Traffic Forecasting via Multi-task Learning. *arXiv preprint arXiv:2111.15542* (2021).

[35] Tanwi Mallick, Prasanna Balaprakash, Eric Rask, and Jane Macfarlane. 2021. Transfer learning with graph neural networks for short-term highway traffic forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 10367–10374.

[36] Hyeonseob Nam and Bohyung Han. 2016. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4293–4302.

[37] California Department of Transportation. 2021. Caltrans PeMS. https://pems.dot.ca.gov/.

[38] Keiron O'Shea and Ryan Nash. 2015. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458* (2015).

[39] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[40] Zheyi Pan, Wentao Zhang, Yuxuan Liang, Weinan Zhang, Yong Yu, Junbo Zhang, and Yu Zheng. 2020. Spatio-temporal meta learning for urban traffic prediction. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[41] Cheonbok Park, Chunggi Lee, Hyojin Bahng, Yunwon Tae, Seungmin Jin, Kihwan Kim, Sungahn Ko, and Jaegul Choo. 2020. ST-GRAT: Spatio-Temporal Graph Attention Network for Traffic Forecasting. In *29th ACM International Conference on Information and Knowledge Management, CIKM 2020*. Association for Computing Machinery.

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019), 8026–8037.

[43] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. 2015. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine* 32, 3 (2015), 53–69.

[44] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. 2018. Multiadversarial domain adaptation. In *Thirty-second AAAI conference on artificial intelligence*.

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

[46] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. 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. 914–921.

[47] Michael Steininger, Konstantin Kobs, Albin Zehe, Florian Lautenschlager, Martin Becker, and Andreas Hotho. 2020. Maplur: Exploring a new paradigm for estimating air pollution using deep learning on map images. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 6, 3 (2020), 1–24.

[48] Mei Lam Tam and William HK Lam. 2011. Application of automatic vehicle identification technology for real-time journey time estimation. *Information Fusion* 12, 1 (2011), 11–19.

[49] Chujie Tian, Xinning Zhu, Zheng Hu, and Jian Ma. 2021. A transfer approach with attention reptile method and long-term generation mechanism for few-shot traffic prediction. *Neurocomputing* 452 (2021), 15–27.

[50] Laurens Van Der Maaten. 2013. Barnes-hut-sne. *arXiv preprint arXiv:1301.3342* (2013).

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

[52] Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. 2018. Cross-city transfer learning for deep spatio-temporal prediction. *arXiv preprint arXiv:1802.00386* (2018).

[53] Senzhang Wang, Hao Miao, Jiyue Li, and Jiannong Cao. 2021. Spatio-Temporal Knowledge Transfer for Urban Crowd Flow Prediction via Deep Attentive Adaptation Networks. *IEEE Transactions on Intelligent Transportation Systems* (2021).

[54] Boris Weisfeiler and Andrei Leman. 1968. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series* 2, 9 (1968), 12–16.

[55] Billy M Williams and Lester A Hoel. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of transportation engineering* 129, 6 (2003), 664–672.

[56] Ning Wu, Xin Wayne Zhao, Jingyuan Wang, and Dayan Pan. 2020. Learning effective road network representation with hierarchical graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 6–14.

[57] Qinge Xie, Tiancheng Guo, Yang Chen, Yu Xiao, Xin Wang, and Ben Y Zhao. 2020. Deep graph convolutional networks for incident-driven traffic speed prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1665–1674.

[58] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[59] Yongxin Yang and Timothy M Hospedales. 2014. A unified perspective on multi-domain and multi-task learning. *arXiv preprint arXiv:1412.7489* (2014).

[60] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. 2019. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *The World Wide Web Conference*. 2181–2191.

[61] Xueyan Yin, Genze Wu, Jinze Wei, Yanming Shen, Heng Qi, and Baocai Yin. 2021. Deep learning on traffic prediction: Methods, analysis and future directions. *IEEE*

[62] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792* (2014).

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

[64] Xiyue Zhang, Chao Huang, Yong Xu, and Lianghao Xia. 2020. Spatial-temporal convolutional graph attention networks for citywide traffic flow forecasting. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1853–1862.

[65] Yingxue Zhang, Yanhua Li, Xun Zhou, Xiangnan Kong, and Jun Luo. 2020. Curb-GAN: Conditional Urban Traffic Estimation through Spatio-Temporal Generative Adversarial Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 842–852.

[66] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* 21, 9 (2019), 3848–3858.

[67] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1234–1241.

[68] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81.