Personalized Federated Learning for Cross-City Traffic Prediction

Yu Zhang[1,2], **Hua Lu**[3*], **Ning Liu**[1,2], **Yonghui Xu**[1,2], **Qingzhong Li**[1,2*] and **Lizhen Cui**[1,2]

[1]School of Software, Shandong University (SDU), China

[2]Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, China

[3]Department of People and Technology, Roskilde University, Denmark

z.yu@mail.sdu.edu.cn, luhua@ruc.dk, {liun21cs, lqz, clz}@sdu.edu.cn, xu.yonghui@hotmail.com

Abstract

Traffic prediction plays an important role in urban computing. However, many cities face data scarcity due to low levels of urban development. Although many approaches transfer knowledge from data-rich cities to data-scarce cities, the centralized training paradigm cannot uphold data privacy. For the sake of inter-city data privacy, Federated Learning has been used, which follows a decentralized training paradigm to enhance traffic knowledge of data-scarce cities. However, spatio-temporal data heterogeneity causes client drift, leading to unsatisfactory traffic prediction performance. In this work, we propose a novel personalized Federated learning method for Cross-city Traffic Prediction (`pFedCTP`). It learns traffic knowledge from multiple data-rich source cities and transfers the knowledge to a data-scarce target city while preserving inter-city data privacy. In the core of `pFedCTP` lies a Spatio-Temporal Neural Network (ST-Net) for clients to learn traffic representation. We decouple the ST-Net to learn space-independent traffic patterns to overcome cross-city spatial heterogeneity. Besides, `pFedCTP` adaptively interpolates the layer-wise global and local parameters to deal with temporal heterogeneity across cities. Extensive experiments on four real-world traffic datasets demonstrate significant advantages of `pFedCTP` over representative state-of-the-art methods.

1 Introduction

Traffic prediction plays an important role in urban computing. The accuracy of traffic prediction depends on the availability of sufficient data, which is mostly collected by road sensors deployed by authorities or vehicle devices involved in spatial crowdsourcing tasks [Liu *et al.*, 2022; Zhong *et al.*, 2023]. However, due to diverse levels of urban development, some cities only have limited traffic data insufficient for accurate traffic prediction. To cope with this issue of unbalanced data availability, an intuitive solution is to share knowledge from data-rich cities with data-scarce cities, i.e., cross-city traffic knowledge transfer.

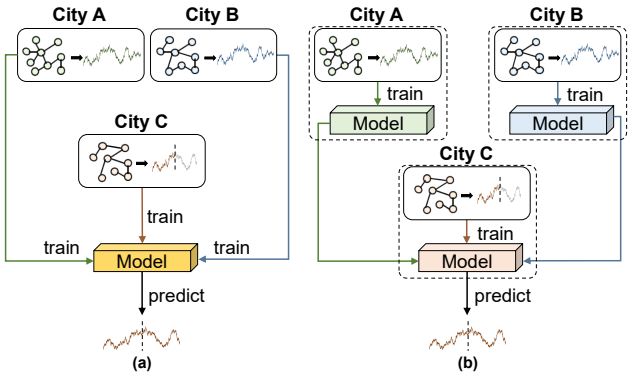

Figure 1: Illustration of cross-city traffic prediction. Cities A and B are source cities with abundant data, while city C is the target city with scarce data. **(a) Centralized Cross-City Traffic Prediction:** A centralized model is trained by using the data from all three cities to help the target city C improve prediction results. **(b) Decentralized Cross-City Traffic Prediction**: Cities A and B train models separately and transfer their traffic knowledge to the target city C.

Extensive research has been carried out on cross-city traffic knowledge transfer. Based on how urban areas are being divided, the research works fall into 1) Convolutional Neural Network (CNN)-based methods and 2) Graph Neural Network (GNN)-based methods. CNN-based methods divide a city into equal-size grid cells to learn region embeddings [Zhang *et al.*, 2022b; Jin *et al.*, 2022a; Wang *et al.*, 2018; Yao *et al.*, 2019a]. However, equal-size grid cells fail to reflect the actual urban structure. Neither can CNNs effectively model non-Euclidean road networks. In contrast, GNN-based methods can model irregular urban structures by capturing the spatial relevance of the regions [Yao *et al.*, 2023; Ouyang *et al.*, 2023b; Ouyang *et al.*, 2023a].

Both CNN- and GNN-based methods may suffer from negative transfer, i.e., the performance of a target city degrades after domain knowledge is transferred from a source city [Zhang *et al.*, 2022a]. Many graph-based cross-city transfer learning approaches have attempted to address this problem. Jin *et al.* [2022b] reweighted the source regions to transfer similar knowledge to the target city. However, external auxiliary data (e.g., POI, check-in data) is essen-

*Corresponding authors.

tial to measuring regional similarity. Domain adversarial learning is commonly used in transfer learning to solve the negative transfer problem caused by domain shift. Recent works [Ouyang *et al.*, 2023b; Yao *et al.*, 2023; Tang *et al.*, 2022; Ouyang *et al.*, 2023a] have adopted the domain classifier to align the distribution by distinguishing the source and target domain features. As the training objective of the domain classifier is to identify the source of samples, retraining is necessary when a new city joins. Lu *et al.* [2022] and Liu *et al.* [2023] focused on graph few-shot learning based on the Meta-Learning framework, using a reconstructed graph structure to constrain the shift of the original spatial structure. However, the aforementioned centralized methods do not uphold data privacy concerns by either the source or the target cities, as shown in Figure 1(a). To uphold privacy, we adopt a decentralized training strategy, as shown in Figure 1(b). Cities keep data private and transfer the traffic knowledge to a data-scarce target city to improve the performance of prediction.

Federated learning (FL) [Yang *et al.*, 2020] is a decentralized collaborative learning paradigm designed to preserve data privacy. In the FL paradigm, each city works as a client, while an FL server orchestrates the collaborative training of an FL model across participating cities without exposing local data. Chen *et al.* [2022] proposed a Cross-city Federated Transfer Learning framework, CcFTL, to transfer knowledge from multi-source urban data (e.g., POIs and population density). However, the FedAvg [McMahan *et al.*, 2017] algorithm used in CcFTL requires client data to be independent and identically distributed (IID), and thus encounters convergence issues on non-IID spatio-temporal traffic data. Such data leads to client drift, deteriorating the performance of FL.

Thus motivated, we propose a personalized Federated learning method for Cross-city Traffic Prediction (`pFedCTP`). We design a Spatio-Temporal Neural Network (ST-Net) to extract spatial structure features, spatio-temporal knowledge, and traffic patterns from raw traffic data. To overcome spatial heterogeneity caused by different city spatial structures, we decouple the components within ST-Net and selectively transfer space-independent traffic patterns to the FL server. On the other hand, each city's traffic patterns vary temporally, causing temporal heterogeneity in cross-city traffic prediction. To deal with temporal heterogeneity across cities, we propose an adaptive layer-wise model interpolation method to customize personalized local client models.

The contributions of this paper are as follows:

- We propose a novel personalized Federated learning method for Cross-city Traffic Prediction (`pFedCTP`). It learns traffic knowledge from multiple data-rich source cities and transfers the knowledge to a data-scarce target city while preserving inter-city data privacy.
- We design an ST-Net for cross-city traffic prediction under FL. To overcome spatial heterogeneity, we decouple the ST-Net and share space-independent traffic patterns with the server. Meanwhile, we propose an adaptive layer-wise model interpolation method to alleviate the effect of temporal heterogeneity.
- We conduct extensive experiments on four real-world traffic speed datasets to verify that `pFedCTP` achieves superior performance over several representative state-of-the-art methods. It is capable of reducing average MAE and RMSE by 1.9% and 0.8% respectively compared to the best-performing baseline.

2 Related Work

2.1 Traffic Prediction

Traffic prediction [Tedjopurnomo *et al.*, 2020] is an important area of research with real-world impact. Conventional approaches, like HA and ARIMA [Williams and Hoel, 2003], utilize statistical information from traffic data. Deep neural networks have further boosted this area of research. CNN-based methods use a grid to partition a city for traffic prediction [Zhang *et al.*, 2017; Yao *et al.*, 2018; Yao *et al.*, 2019b], but they fail to account for spatial correlation among road networks. In contrast, graph neural networks perform better in a non-Euclidean space. A popular paradigm of spatio-temporal prediction combines GNN and Recurrent Neural Network (RNN) to capture spatio-temporal features simultaneously [Li *et al.*, 2017a; Zhao *et al.*, 2019; Bai *et al.*, 2020].

2.2 Traffic Knowledge Transfer across Cities

Traffic knowledge transfer across cities aims to help data-scarce cities improve traffic prediction by transferring knowledge from data-rich cities. Related work can be divided into three categories: 1) Similarity-based methods, 2) DA-based methods, and 3) ML-based methods. Similarity-based methods calculate the similarity scores between the source and the target regions based on the external auxiliary information (e.g., POI, check-in, weather) to avoid interference from irrelevant knowledge [Wang *et al.*, 2018; Jin *et al.*, 2022a]. Although urban knowledge mining from multi-source data has been widely verified [Li *et al.*, 2023; Zhang *et al.*, 2021; Huang *et al.*, 2023], the data is not always available. DA-based methods refer to those that leverage domain adversarial learning to alleviate domain distribution discrepancies [Yao *et al.*, 2023; Ouyang *et al.*, 2023a; Ouyang *et al.*, 2023b]. The training objective of the domain discriminator makes it essentially difficult to adapt to new cities. ML-based methods adopt the Meta-Learning framework Model-Agnostic Meta-Learning (MAML) [Yao *et al.*, 2019a; Lu *et al.*, 2022] and Reptile [Liu *et al.*, 2023]. Such methods regard the target city as a new task and learn generalizable knowledge from multiple related tasks to adapt to new tasks rapidly. However, the aforementioned centralized methods fail to uphold data privacy concerns.

2.3 Personalized Federated Learning

As a new paradigm for distributed machine learning, FL aims to train a global model without collecting data to the server. However, FedAvg [McMahan *et al.*, 2017] based approaches cannot deal with non-IID client data effectively. Personalized Federated Learning (PFL) methods have emerged to address this limitation. Previous research on PFL can be categorized into two types [Tan *et al.*, 2022]: 1) global model personalization and 2) personalized local models. Global model personalization aims to learn a well-generalized global model

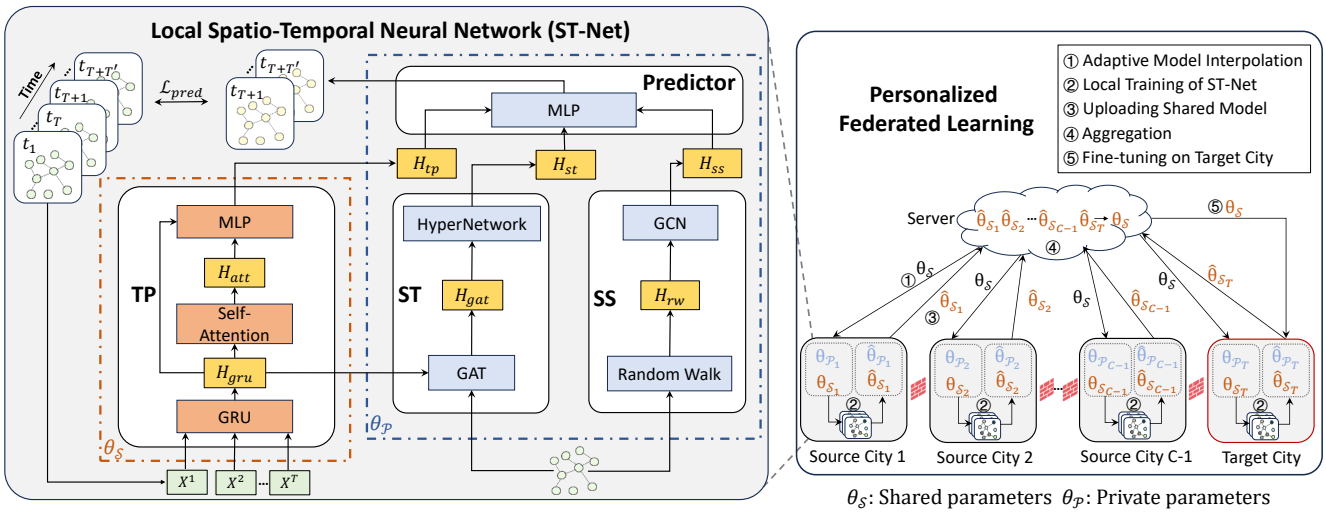

Figure 2: The framework of pFedCTP. The left side corresponds to a local ST-Net composed of four modules, in which the spatial structure (SS), spatio-temporal feature (ST), and predictor are private, and their parameters are denoted as $\theta_{\mathcal{P}}$; traffic pattern (TP) is shared with other cities and the parameters are denoted as $\theta_{\mathcal{S}}$. The right side corresponds to Personalized Federated Learning, where cities act as clients participating in FL to train a well-generalized global model that is transferred to the data-scarce target city for fine-tuning on the local data.

to adapt rapidly to local models. Personalized local models customize model architecture for each client by modifying the FL model aggregation process. Model decoupling [Arivazhagan *et al.*, 2019] and model interpolation [Hanzely and Richtárik, 2020; Zhang *et al.*, 2023] are two commonly used techniques to achieve local model personalization. Model decoupling keeps part of local parameters private to balance the performance between generalization and personalization. Model interpolation mixes the global and local models to save valuable local model parameter information. In this paper, we attempt to learn a generalized global model that absorbs traffic knowledge from multiple cities and can quickly adapt to a target city. Meanwhile, we also learn a personalized local model that customizes spatio-temporal features from the local traffic data of a city.

3 Preliminary

Traffic Spatio-Temporal Graph. Given a city, we define its traffic spatio-temporal graph as $\mathcal{G}_{ST} = (\mathcal{V}, \mathcal{E}, A, X)$. Specifically, $\mathcal{V}$ denotes the node set, $N = |\mathcal{V}|$ is the number of nodes, and $\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V})$ denotes the edge set. Moreover, $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix of the traffic spatio-temporal graph, where $a_{ij} = 1$ indicates an edge exists between node i and j; otherwise, $a_{ij} = 0$. Furthermore, $X \in \mathbb{R}^{N \times T_{total} \times d_f}$ is the traffic feature (e.g., traffic flow and traffic speed) matrix, T_{total} is the total time steps of traffic data, d_f is the node feature dimensionality, and $X^t \in \mathbb{R}^{N \times d_f}$ represents the traffic data at time t.

Traffic Spatio-Temporal Graph Prediction. Given the historical traffic data of T steps, the goal of traffic spatio-temporal graph prediction is to predict the traffic data at future T' steps by learning a function $f(\cdot)$ based on the traffic spatio-temporal graph $\mathcal{G}_{ST}$:

$$[X^{t-T+1}, \ldots, X^t; \mathcal{G}_{ST}] \xrightarrow{f(\cdot)} [X^{t+1}, \ldots, X^{t+T'}]. \quad (1)$$

Federated Cross-City Traffic Prediction. Given $C-1$ source cities with abundant traffic data $\mathcal{G}_{ST}^{S} = \{\mathcal{G}_{ST}^{1}, \ldots, \mathcal{G}_{ST}^{C-1}\}$ and a target city with scarce traffic data $\mathcal{G}_{ST}^{T}$, the goal of federated cross-city traffic prediction is to learn traffic knowledge from the source cities without sharing their local data and transfer the traffic knowledge to the target city to improve the traffic prediction performance.

4 Methodology

We propose a novel personalized federated learning framework to transfer the traffic knowledge from the data-rich source cities to the data-scarce target city. As shown in Figure 2, the framework consists of two stages. ***Stage I: Local Spatio-Temporal Neural Network (ST-Net).*** We design an ST-Net to learn global shared traffic knowledge and local personalized spatio-temporal features for the cross-city traffic prediction task. Each city is seen as a client and deploys an ST-Net to train on the local traffic data. ***Stage II: Personalized Federated Learning.*** We decouple the components within ST-Net and selectively share space-independent traffic patterns to overcome spatial heterogeneity. Meanwhile, to alleviate temporal heterogeneity, we propose adaptive layer-wise aggregation (model interpolation) to balance global generalization and local personalization.

4.1 ST-Net

In this section, we introduce the ST-Net designed for cross-city traffic prediction under FL. An ST-Net instance is deployed on each client to extract spatio-temporal features and predict traffic patterns. It has four modules: spatial structure, traffic pattern, spatio-temporal feature, and predictor.

Spatial Structure

City spatial structure is tightly connected to factors such as terrain, history and culture, and population density. Learn-

ing city spatial features helps understand spatial topological structure and improves the performance of spatio-temporal prediction tasks. Following a previous work [Tan *et al.*, 2023], we extract the random walk information to generate spatial structure embedding H_{rw}, using the raw adjacent matrix. Then H_{rw} is fed into the Graph Convolutional Neural Network (GCN) [Kipf and Welling, 2016] to generate the spatial features H_{ss}.

$$H_{ss}^{(l+1)} = \sigma(\widetilde{A} H_{ss}^{(l)} W^l), \tag{2}$$

where σ is an activation function, $H_{ss}^{(l)}$ represents the node features at layer l, $H_{ss}^{(0)} = H_{rw}$, $\widetilde{A} = \widehat{D}^{-\frac{1}{2}} \widehat{A} \widehat{D}^{-\frac{1}{2}}$, $\widehat{A} = A + I_N$, $\widehat{D}$ is the degree matrix of $\widehat{A}$, I_N is the identity matrix, and N is the number of nodes.

Traffic Pattern

The current or future traffic condition is tightly correlated with its previous observations. We use Gated Recurrent Unit (GRU) [Chung *et al.*, 2014] to generate temporal embedding H_{gru} from the historical traffic data X, for GRU has fewer parameters than RNN but a faster training speed.

Furthermore, we extract representative traffic knowledge, i.e., traffic patterns, from temporal embeddings. We feed the temporal embedding H_{gru} into a self-attention module to adaptively extract temporal traffic features.

$$Q = H_{gru} W_Q, K = H_{gru} W_K, V = H_{gru} W_V, \tag{3}$$

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d'}})V, \tag{4}$$

where W_Q, W_K, W_V are learnable parameters, d' is the dimension of the query, key, and value matrix. Then we extend the temporal attention mechanism to multi-heads to capture the temporal features H_{att}. We further concatenate the temporal embedding H_{gru} and traffic features H_{att}, and feed the concatenation result into Multi-Layer Perceptrons (MLP) to generate the temporal pattern $H_{tp} = MLP([H_{gru}||H_{att}])$, where $||$ represents the concatenation operation.

Spatio-Temporal Feature

We combine GRU and Graph Attention Network (GAT) [Veličković *et al.*, 2017] to learn spatio-temporal features from the observed historical data. We feed the adjacent matrix A and temporal embedding H_{gru} into GAT to generate spatio-temporal features H_{gat}.

$$e_{ij} = attention(W' h_{gru}^i, W' h_{gru}^j), j \in N_i, \tag{5}$$

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in N_i} exp(e_{ik})}, \tag{6}$$

where W' is the weight matrix, e_{ij} represents the importance of node j to node i, N_i is node v_i's neighbor node set, and $softmax$ normalizes node v_j's neighbor nodes. Finally, the output features H_{gat} are obtained by weighting the input features:

$$h_{gat}^i = \sigma(\sum_{j \in N_i} \alpha_{ij} h_{gru}^j). \tag{7}$$

To further personalize a local ST-Net, we include a hypernetwork [Ha *et al.*, 2016] to dynamically generate the weights for the network based on the input features. We pass the spatio-temporal features H_{gat} as input to the hypernetwork to produce personalized model weight $W \in \mathbb{R}^{d \times d}$ and bias $b \in \mathbb{R}^d$. Accordingly, we customize spatio-temporal features using the linear transformation $H_{st} = W \times H_{gat} + b$.

Predictor

We design a predictor to predict the traffic situation of the future T' steps. We combine the private spatial structure embedding H_{ss}, the shared traffic patterns H_{tp}, and the private spatio-temporal features H_{st} to obtain the final representation for traffic prediction. Specifically, we use an MLP as the predictor.

$$\widehat{Y} = MLP([H_{ss}||H_{tp}||H_{st}]), \tag{8}$$

where $\widehat{Y} \in \mathbb{R}^{N \times T' \times d_f}$ is the prediction result. Then we use Mean Squared Error (MSE) loss to measure the performance of our model in training. Given the ground truth $Y \in \mathbb{R}^{N \times T' \times d_f}$, the loss function of ST-Net for traffic prediction is defined as:

$$\mathcal{L}_{pred} = \frac{1}{T'} \sum_{t=1}^{T'} (Y^t - \widehat{Y}^t)^2. \tag{9}$$

4.2 Personalized Federated Learning

In this section, we describe the PFL designed for the cross-city traffic prediction task to handle client drift caused by spatio-temporal data heterogeneity.

Parameter Decoupling

To overcome spatial heterogeneity, we decouple ST-Net and only learn the shared module (traffic pattern), keeping specific spatial-related features private. The ST-Net parameters are divided into the shared model parameters $\theta_{\mathcal{S}}$ and the private model parameters $\theta_{\mathcal{P}}$. For the four modules in ST-Net (spatial structure, traffic pattern, spatio-temporal feature, and predictor), only traffic pattern is shared during FL, and the rest modules contain city-specific spatial knowledge, which can be seen as noise to the target city and should be kept private. Parameter decoupling not only avoids unrelated traffic knowledge but also reduces communication costs between server and clients.

Adaptive Model Interpolation

To alleviate temporal heterogeneity, we adopt model interpolation to customize the local model parameters at the start of a communication round rather than overwriting them with the global model parameters. The original local model has effective information, and thus dropping them will cause information loss. A trade-off parameter $\lambda \in [0, 1]$ is generally used to determine the mixing degree of the local shared model and global mean model. $\lambda = 1$ means the global model replaces the local model completely; $\lambda = 0$ means the opposite. The model interpolation on client i at round r is denoted as:

$$\theta_{\mathcal{S}_i}^r = \lambda \cdot \theta_{\mathcal{S}}^r + (1 - \lambda) \cdot \theta_{\mathcal{S}_i}^{r-1}. \tag{10}$$

However, it is hard to denote a constant λ to all clients because they have different contributions. Besides, it is infeasible to consider that each layer in the model has an equal

mixing degree. Therefore, we propose an adaptive layer-wise model interpolation method for local ST-Net initialization in each round. For layer $l_i^r \in \theta_{\mathcal{S}_i}^r$ in client i at round r, the model interpolation process works as follows:

$$l_i^r = l_i^{r-1} + sim(l^r, l_i^{r-1}) \cdot (l^r - l_i^{r-1}), \tag{11}$$

where $sim(\cdot)$ captures the similarity of the global and the local shared layer, l^r represents the corresponding layer in global model $\theta_{\mathcal{S}}^r$ at round r, and $l^r - l_i^{r-1}$ denotes the update of the parameters. In this paper, we use the Cosine similarity function.

`pFedCTP` Learning Process

Algorithm 1 outlines the learning process of `pFedCTP`. After the initialization (lines 1–2), it adopts a workflow of FL (lines 3–10) followed by transfer (lines 11–14). In FL, clients receive a global module from the server (line 4) and adaptively aggregate with the local traffic pattern module (lines 5–7). Then clients train ST-Net on samples of their local dataset and upload the updated shared module to the server in parallel (lines 8–9). The server aggregates all modules from clients for the next round of communication (line 10). The FL ends after R rounds of communication, and the server stores a well-generalized global traffic pattern module. Subsequently, the target city receives the global module transferred from the server (line 11) and performs layer-wise traffic pattern module updates (lines 12–13). Finally, the target city fine-tunes the ST-Net with local scarce traffic data to improve traffic prediction performance (line 14).

Notably, all clients (source and target) participate in the first stage but only the target client participates in the second. Thus, the total communication cost between the server and clients is $2 \times C \times R + 1$ times. Moreover, the local training on each client only uses sampled data instead of the entire local dataset lest unbalanced client data deteriorates the performance of the target client. To this end, we introduce a hyperparameter B to denote the batch number in local training. Its effect will be analyzed in Section 5.3. Assuming that a data-rich source city gets m samples from local data during ST-Net training, we have $m << M$[1], where M is the total number of possible data samples. The computational complexity mainly includes the local model training cost $O(m \times C \times R + p)$, where p is the number of samples in the target city, and the adaptive model interpolation cost $O(L \times C \times R + L)$, where L is the number of layers in the shared module. In contrast to the conventional FL approach that trains a model on the entire dataset and shares the entire model architecture, our model's complexity is significantly reduced.

Algorithm 1: The `pFedCTP` Framework.

Input: Communication rounds R, client number C, ST-Net shared parameters θ_S, batch number B
Output: The target client ST-Net $\widehat{\theta}_t^{R+1}$

```
// Inititialization
for client i from 1 to C in parallel do
    Initialize client model θ_i^0;
// Personalized Federated Learning
for round r = 1, 2, ..., R do
    Server sends θ_S^r to all clients;
    for client i from 1 to C in parallel do
        for layer l_i^r ∈ θ_{S_i}^r do
            update l_i^r according to Eq. (11);
        θ̂_i^r ← Train ST-Net in B batches on samples;
        update θ̂_{S_i}^r to the Server;
    Server computes θ_S^{r+1} = (Σ_{i=1}^C θ̂_{S_i}^r) / C;
// Fine-Tuning on Target City
Server sends θ_S^{R+1} to the target city;
for layer l_t^{R+1} ∈ θ_{S_t}^{R+1} do
    update l_t^{R+1} according to Eq. (11);
θ̂_t^{R+1} ← Fine-tune ST-Net on target city's data;
```

Datasets	# Sensors	Time Span	Time Interval
PEMS-BAY	325	52,116	5 min
METR-LA	207	34,272	5 min
DiDi-Chengdu	524	17,280	10 min
DiDi-Shenzhen	627	17,280	10 min

Table 1: Statistics of traffic datasets.

5 Experiments

5.1 Experiment Settings

Datasets. We evaluate the performance of `pFedCTP` on four traffic speed datasets: PEMS-BAY, METR-LA [Li *et al.*, 2017b], DiDi-Chengdu, and DiDi-Shenzhen. PEMS-BAY and METR-LA include traffic information from the San Francisco Bay Area and Los Angeles County in the USA, respectively. DiDi-Chengdu and Didi-Shenzhen are provided by the Didi GAIA Initiative [DiDi, 2020]. The detailed information of each dataset is shown in Table 1. We alternately use three datasets as source cities with abundant traffic data and the fourth as the target city with relatively less data. To simulate data scarcity in the target city, we only use 3 days of traffic data as training data.

Metrics. We use two commonly used metrics to evaluate the performance of traffic prediction: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

Baselines. We compare `pFedCTP` with relevant methods of three types: Target-only methods, Cross-city methods, and Federated Cross-city methods.

Target-only methods only train a model on the data-scarce target city, without involving any source cities or their data. (1) Spatio-Temporal Graph Convolution Network (**STGCN**) [Yu *et al.*, 2017] integrates graph convolution and gated temporal convolution through spatio-temporal convolutional blocks for traffic prediction; (2) **T-GCN** [Zhao *et al.*, 2019] uses temporal GCN to learn complex topological structures and GRU to obtain temporal correlation from dynamic

[1]Refer to the dataset Didi-Chengdu shown in Table 1. Suppose that the batch number $B = 150$ and the batch size $= 32$. We have $m = 150 \times 32 = 4,800$, and $M \approx 17,280$, i.e., the time span.

Methods	**PEMS-BAY** (Target city)						**METR-LA** (Target city)					
	MAE (↓)			**RMSE (↓)**			**MAE (↓)**			**RMSE (↓)**		
	5 min	**15 min**	**30 min**	**5 min**	**15 min**	**30 min**	**5 min**	**15 min**	**30 min**	**5 min**	**15 min**	**30 min**
Target-only baselines												
STGCN [Yu *et al.*, 2017]	2.636	2.770	3.009	4.033	4.436	5.127	4.263	4.418	4.864	6.320	6.666	7.351
T-GCN [Zhao *et al.*, 2019]	1.352	1.799	2.278	2.175	3.329	4.664	2.691	3.324	4.048	4.349	5.834	7.216
ST-Net (ours)	**1.058**	1.625	2.365	1.850	3.275	4.686	2.482	3.058	**3.780**	4.210	5.733	7.104
Cross-city baselines												
DastNet [Tang *et al.*, 2022]	1.518	2.056	2.473	3.139	4.397	5.404	2.830	3.379	4.042	5.220	6.390	7.671
ST-GFSL [Lu *et al.*, 2022]	1.158	1.637	**2.169**	1.981	3.261	4.638	2.464	3.106	3.816	4.141	**5.614**	**6.928**
Federated cross-city baselines												
FedAvg [McMahan *et al.*, 2017]	2.030	2.741	3.144	2.864	4.111	5.305	3.234	3.679	4.430	5.243	6.427	7.691
FedProx [Li *et al.*, 2020]	2.030	2.279	2.800	3.357	4.286	5.591	3.157	3.911	4.512	4.773	6.185	7.412
Per-FedAvg [Fallah *et al.*, 2020]	2.305	2.646	3.084	4.121	4.965	5.910	3.022	3.572	4.214	4.894	6.189	7.439
pFedCTP (ours)	1.076	**1.596**	2.201	**1.827**	**3.212**	**4.580**	**2.327**	**3.010**	3.813	**4.082**	5.677	7.066

Methods	**Didi-Chengdu** (Target city)						**Didi-Shenzhen** (Target city)					
	MAE (↓)			**RMSE (↓)**			**MAE (↓)**			**RMSE (↓)**		
	10 min	**30 min**	**60 min**	**10 min**	**30 min**	**60 min**	**10 min**	**30 min**	**60 min**	**10 min**	**30 min**	**60 min**
Target-only baselines												
STGCN [Yu *et al.*, 2017]	2.930	2.905	3.093	4.249	4.248	4.491	2.740	2.738	2.873	3.858	3.877	4.087
T-GCN [Zhao *et al.*, 2019]	2.340	2.850	3.350	3.369	4.136	4.845	2.109	2.589	3.010	3.177	3.920	4.572
ST-Net (ours)	2.327	2.768	3.135	3.256	3.994	4.570	1.932	2.355	**2.675**	2.804	3.551	4.124
Cross-city baselines												
DastNet [Tang *et al.*, 2022]	2.874	3.230	3.739	4.145	4.661	5.346	2.335	2.651	3.110	3.508	4.018	4.704
ST-GFSL [Lu *et al.*, 2022]	2.189	**2.639**	3.004	3.151	**3.840**	**4.339**	1.948	2.361	2.718	2.806	**3.515**	4.129
Federated cross-city baselines												
FedAvg [McMahan *et al.*, 2017]	3.181	3.56	4.005	4.323	4.852	5.428	2.621	3.131	3.571	3.623	4.380	5.009
FedProx [Li *et al.*, 2020]	3.086	3.645	4.088	3.940	4.777	5.400	2.473	2.904	3.290	3.290	4.003	4.613
Per-FedAvg [Fallah *et al.*, 2020]	2.401	2.932	3.431	3.355	4.138	4.812	2.146	2.606	2.946	2.972	3.715	4.289
pFedCTP (ours)	**2.144**	2.661	**3.003**	**3.099**	3.896	4.394	**1.889**	**2.339**	2.699	**2.746**	3.522	**4.123**

Table 2: Performance comparison of all methods. The **best** and second best results are highlighted.

traffic data; (3) We also train an **ST-Net** (cf. Section 4.1) on a target city only to validate its effectiveness.

From Cross-city methods, for fairness, we choose those that transfer knowledge from multiple source cities to the target city. (1) Domain Adversarial Spatial-Temporal Network (**DastNet**) [Tang *et al.*, 2022] adopts domain adversarial learning to learn the domain-invariant node embedding; (2) Spatio-temporal few-shot traffic prediction model (**ST-GFSL**) [Lu *et al.*, 2022] adopts MAML [Finn *et al.*, 2017] as the base Meta-Learning framework.

We also compare with several FL-based methods. (1) **FedAvg** [McMahan *et al.*, 2017]: We combine both source cities and the target city to train a global ST-Net, and we test the final global model on the target city data; (2) **FedProx** [Li *et al.*, 2020] uses a preliminary term to guide local models to pay more attention to model weights that are close to the global model weights during updates; (3) **Per-FedAvg** [Fallah *et al.*, 2020] is a combination of FL and Meta-Learning methods to optimize the global model for fast personalization. The source cities are trained in the meta-training phase and the target city is tested in the meta-testing phase.

Implement Details. For all the experiment methods, we set the history time steps $T = 12$ and the future time steps $T' = 6$. We simulate short-, medium-, and long-term prediction results using 1, 3, and 6 steps respectively. Other important hyperparameters are set as follows: the client number $C = 4$, the batch size = 32, the learning rate = 0.01, the number of GCN layers = 1, and the hidden dimensions = 32 for all methods. For the baselines, we train the target-only model for 100 epochs. The code is available at https://github.com/ZYuSdu/pFedCTP.

5.2 Results and Discussion

We show the traffic prediction results of all methods on the four datasets in Table 2. We have the following observations: (1) Target-only baselines mostly perform not well because they are trained on limited target city traffic data. Compared to T-GCN, STGCN struggles to predict short- and mid-term traffic using scarce data but has a significant advantage in long-term prediction in the Didi-Chengdu and Didi-Shenzhen datasets. (2) For Cross-city baselines, the state-of-the-art method ST-GFSL performs better on some long-term traffic prediction indicators, while DastNet is unable to handle cross-city traffic prediction well for scarce data in the target city. (3) In respect of FL-based baselines, Per-FedAvg outperforms the others. As a Meta-Learning based method, Per-FedAvg samples tasks to learn a generalized global model, which can quickly adapt to the new target city. (4) Compared to the other target-only baselines, our ST-Net shows clear advantages in the prediction task on scarce traffic data. Overall, pFedCTP achieves the best results among all, reducing the average MAE and RMSE by 1.9% and 0.8% respectively compared to the best-performing baseline, i.e., ST-GFSL.

5.3 Hyperparameter Analysis

We investigate the effect of the communication rounds and the local batches in each round on Didi-Chengdu. The results are shown in Figure 3.

Communication Rounds. We tune R according to the set of $\{30, 60, 90, 120\}$. We can see that as the number of communication rounds increases, the performance of the model is improved. However, excessive communication rounds not only increase costs but also increase the over-fitting risk of the model. Our pFedCTP achieves good results when $R = 90$.

Local Batches. We vary the local batch number B according to the set of $\{50, 100, 150, 200\}$ to find the suitable setting. Small B values require massive communication between the server and clients, causing a waste of resources, whereas large B values interfere with the prediction performance in long-term traffic prediction. When $B = 150$, pFedCTP makes better short- and long-term predictions.

5.4 Ablation Study

We also conduct ablation studies to verify the effectiveness of our framework designs.

Ablation Study of ST-Net. We compare the ST-Net (ST) with several variants: ST-w/o TP (temporal pattern), ST-w/o SS (spatial structure), and ST-w/o HN (hypernetwork). The ablation results are shown in Figure 4. We observe that the prediction performance deteriorates severely after we remove the traffic pattern module, meaning that the module is necessary for ST-Net and FL. Without the spatial structure module, the performance deteriorates clearly on the two Didi datasets, as the rich information in their complex spatial structure features is not used. Moreover, we observe that the hypernetwork might slightly enhance the prediction performance in some cases by customizing the network parameters.

Ablation Study of pFedCTP. The ablation study on pFedCTP includes pFedCTP-w/o FT (Fine-tuning on the target city), pFedCTP-All (sharing the entire ST-Net), and pFedCTP-T that only involves source cities in FL and trains each ST-Net on the entire local traffic dataset. The results are shown in Figure 5. We can see that pFedCTP-w/o FT is slightly worse than pFedCTP, verifying that the model transferred from the server can adapt to the target city through fine-tuning. Next, pFedCTP-All can result in higher errors since sharing the entire ST-Net with the server can also transfer irrelevant noise from other clients. Further, pFedCTP-T achieves comparable performance with pFedCTP. However, pFedCTP-T uses the entire dataset for local training and thus incurs much longer training time than pFedCTP that uses sampled data. For instance, on the Didi-Chengdu dataset, pFedCTP-T needs 72 minutes but pFedCTP only 9 minutes.

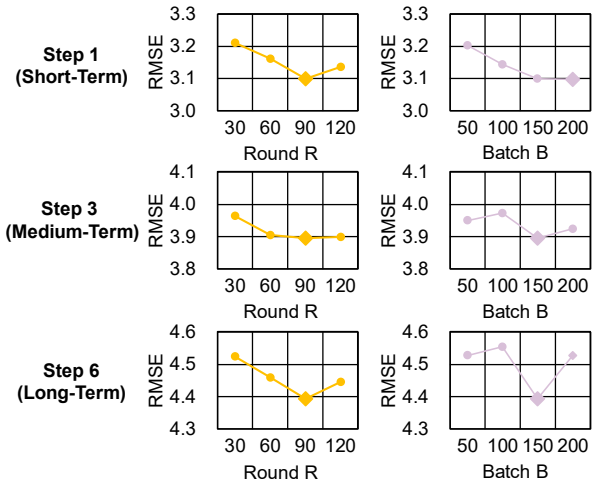

Figure 3: Hyperparameter analysis on Didi-Chengdu.

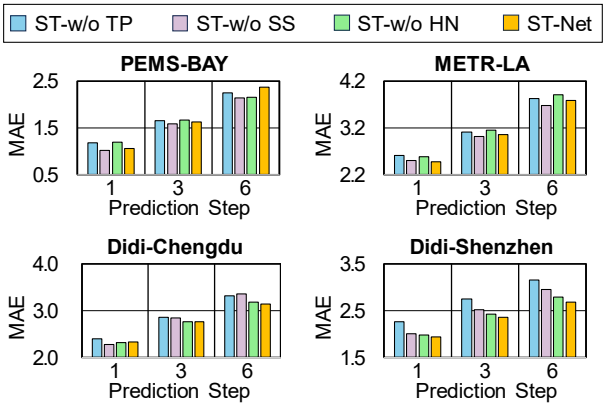

Figure 4: Ablation study of ST-Net.

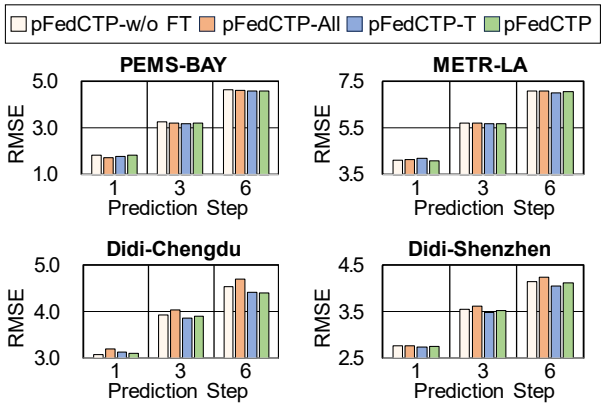

Figure 5: Ablation study of pFedCTP.

6 Conclusion

In this paper, we have proposed a personalized federated learning framework for cross-city traffic prediction, pFedCTP, that aims to help a data-scarce city improve traffic prediction performance without violating data privacy. Following the FL paradigm, pFedCTP treats cities as clients each of which trains a Spatio-Temporal Neural Network (ST-Net) on its local traffic data. To overcome spatial heterogeneity, pFedCTP decouples the components within ST-Net and shares space-independent traffic knowledge. Meanwhile, pFedCTP uses a layer-wise model interpolation method to deal with temporal heterogeneity. Experiments on four real traffic datasets demonstrate the effectiveness of pFedCTP.

Acknowledgments

This research is partially supported by the National Key R&D Program of China 2021YFF0900800 and NSFC No. 92367202.

References

[Arivazhagan *et al.*, 2019] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[Bai *et al.*, 2020] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

[Chen *et al.*, 2022] Gaode Chen, Yijun Su, Xinghua Zhang, Anmin Hu, Guochun Chen, Siyuan Feng, Ji Xiang, Junbo Zhang, and Yu Zheng. A cross-city federated transfer learning framework: A case study on urban region profiling. *arXiv preprint arXiv:2206.00007*, 2022.

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

[DiDi, 2020] DiDi. GAIA Initiative DiDi. https://outreach.didichuxing.com/app-outreach/CTIE, 2020. Accessed: 2024-05-23.

[Fallah *et al.*, 2020] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[Ha *et al.*, 2016] David Ha, Andrew Dai, and Quoc V. Le. Hypernetworks, 2016.

[Hanzely and Richtárik, 2020] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

[Huang *et al.*, 2023] Weiming Huang, Daokun Zhang, Gengchen Mai, Xu Guo, and Lizhen Cui. Learning urban region representations with pois and hierarchical graph infomax. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:134–145, 2023.

[Jin *et al.*, 2022a] Yilun Jin, Kai Chen, and Qiang Yang. Selective cross-city transfer learning for traffic prediction via source city region re-weighting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 731–741, 2022.

[Jin *et al.*, 2022b] Yilun Jin, Kai Chen, and Qiang Yang. Selective cross-city transfer learning for traffic prediction via source city region re-weighting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 731–741, 2022.

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

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

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

[Li *et al.*, 2020] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[Li *et al.*, 2023] Yi Li, Weiming Huang, Gao Cong, Hao Wang, and Zheng Wang. Urban region representation learning with openstreetmap building footprints. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1363–1373, 2023.

[Liu *et al.*, 2022] Jiaxin Liu, Liwei Deng, Hao Miao, Yan Zhao, and Kai Zheng. Task assignment with federated preference learning in spatial crowdsourcing. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1279–1288, 2022.

[Liu *et al.*, 2023] Zhanyu Liu, Guanjie Zheng, and Yanwei Yu. Cross-city few-shot traffic forecasting via traffic pattern bank. *arXiv preprint arXiv:2308.09727*, 2023.

[Lu *et al.*, 2022] Bin Lu, Xiaoying Gan, Weinan Zhang, Huaxiu Yao, Luoyi Fu, and Xinbing Wang. Spatio-temporal graph few-shot learning with cross-city knowledge transfer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1162–1172, 2022.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[Ouyang *et al.*, 2023a] Xiaocao Ouyang, Yan Yang, Yiling Zhang, Wei Zhou, Jihong Wan, and Shengdong Du. Domain adversarial graph neural network with cross-city graph structure learning for traffic prediction. *Knowledge-Based Systems*, 278:110885, 2023.

[Ouyang *et al.*, 2023b] Xiaocao Ouyang, Yan Yang, Wei Zhou, Yiling Zhang, Hao Wang, and Wei Huang. Citytrans: Domain-adversarial training with knowledge transfer for spatio-temporal prediction across cities. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[Tan *et al.*, 2022] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Tan *et al.*, 2023] Yue Tan, Yixin Liu, Guodong Long, Jing Jiang, Qinghua Lu, and Chengqi Zhang. Federated learning on non-iid graphs via structural knowledge sharing. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 9953–9961, 2023.

[Tang *et al.*, 2022] Yihong Tang, Ao Qu, Andy HF Chow, William HK Lam, SC Wong, and Wei 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*, pages 1905–1915, 2022.

[Tedjopurnomo *et al.*, 2020] David Alexander Tedjopurnomo, Zhifeng Bao, Baihua Zheng, Farhana Murtaza Choudhury, and Alex Kai Qin. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 34(4):1544–1561, 2020.

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

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

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

[Yang *et al.*, 2020] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu, editors. *Federated Learning*. Springer, Cham, 2020.

[Yao *et al.*, 2018] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[Yao *et al.*, 2019a] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *The world wide web conference*, pages 2181–2191, 2019.

[Yao *et al.*, 2019b] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5668–5675, 2019.

[Yao *et al.*, 2023] Zhixiu Yao, Shichao Xia, Yun Li, Guangfu Wu, and Linli Zuo. Transfer learning with spatial–temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

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

[Zhang *et al.*, 2017] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[Zhang *et al.*, 2021] Mingyang Zhang, Tong Li, Yong Li, and Pan Hui. Multi-view joint graph representation learning for urban region embedding. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4431–4437, 2021.

[Zhang *et al.*, 2022a] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2022.

[Zhang *et al.*, 2022b] Yingxue Zhang, Yanhua Li, Xun Zhou, and Jun Luo. Mest-gan: Cross-city urban traffic estimation with meta spatial-temporal generative a dversarial networks. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 733–742. IEEE, 2022.

[Zhang *et al.*, 2023] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11237–11244, 2023.

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

[Zhong *et al.*, 2023] Xiaolong Zhong, Hao Miao, Dazhuo Qiu, Yan Zhao, and Kai Zheng. Personalized location-preference learning for federated task assignment in spatial crowdsourcing. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3534–3543, 2023.