2022 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-5099-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDM54844.2022.00085

STrans-GAN: Spatially-Transferable Generative Adversarial Networks for Urban Traffic Estimation

Yingxue Zhang*, Yanhua Li†, Xun Zhou‡, Xiangnan Kong†, Jun Luo§
*Binghamton University
†Worcester Polytechnic Institute
‡The University of Iowa
§Lenovo Group Limited
yzhang42@binghamton.edu; yli15@wpi.edu; xun-zhou@uiowa.edu; xkong@wpi.edu; jluo1@lenovo.com

***Abstract*—Conditional traffic estimation is a vital problem in urban plan deployment, which can help evaluate urban construction plans and improve transportation efficiency. Conventional methods for conditional traffic estimation usually focus on supervised settings, which require a large amount of labeled training data. However, in many urban planning applications, the large amount of traffic data in a new city can be hard or impossible to acquire. To tackle the conditional traffic estimation problem in data scarcity situations, we formulate the problem as a spatial transfer generative learning problem. Compared to prior spatial transfer learning frameworks with only single source city, we propose to extracts knowledge from multiple source cities to improve the estimation accuracy and transfer stability, which is a technically more challenging task. As a solution, we propose a new cross-city conditional traffic estimation method — Spatially-Transferable Generative Adversarial Networks (STrans-GAN) with novel pre-training and fine-tuning algorithms. STrans-GAN preserves diverse traffic patterns from multiple source cities through traffic clustering, and incorporates meta-learning idea into the pre-training process to learn a well-generalized model. During fine-tuning, we propose to add a cluster matching regularizer to realize the flexible adaptation in different scenarios. Through extensive experiments on multiple-city datasets, the effectiveness of STrans-GAN is proved.**

***Index Terms*—Generative adversarial networks; meta learning; urban traffic estimation.**

I. Introduction

Conditional traffic estimation is a critical problem in urban development, especially in land use planning, subway routes planning, *etc*. Given the road network of a city, its historical traffic status, and an urban development plan which usually leads to new local travel demands, the *conditional urban traffic estimation problem* aims to accurately estimate the future traffic status based on the changing travel demands. Solving such problem not only provides insights to evaluate the feasibility of the urban deployment plan, but also helps to reduce potential traffic congestion and improves local transportation efficiency.

A number of data-driven methods have achieved success on single-city conditional traffic estimation, such as classical machine learning models [1], [3], [12] or GAN-based models including CurbGAN [39] and TrafficGAN [38]. However, these methods typically require large amount of training data of the target city and are unable to perform well in case of data scarcity, *e.g.*, when estimating the traffic in a previously unseen city or a newly-built region. A simple idea to address the data scarcity issue would be to "borrow" enough training data from other cities if they are available. However, due to spatial heterogeneity and local traffic pattern differences, such tricks usually result in poor estimation quality.

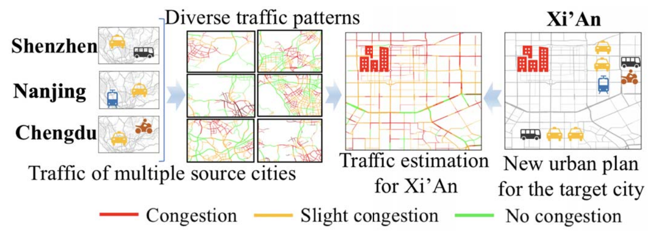


Fig. 1: An example of conditional traffic estimation by transferring knowledge from multiple source cities (Shenzhen, Chengdu, etc) to the target city (Xi'An).

An effective solution to the data scarcity problem is to employ a transfer learning paradigm, which avoids training models directly using the limited data of a "new" city but transfers urban knowledge from other source cities (with abundant data) to the target city (with insufficient data) to enable good performance [29]. *In this paper, we aim at developing a spatial transfer generative learning framework for cross-city conditional urban traffic estimation in case of data scarcity.*

Prior art. Unfortunately, existing spatial transfer learning techniques may not directly solve our cross-city conditional traffic estimation problem. For example, RegionTrans [28], TL-DCRNN [19] and STCNet [36] borrow the transfer leaning framework to forecast future traffic using historical time-series traffic data. However, in most of the urban transfer leaning methods, only one source city is used to extract transferred knowledge. To enable good transfer performance, these methods have to guarantee that the source city and the target city have a lot in common, or it would turn out to be a failure if the two cities are too different [31].

Some works try to transfer knowledge from multiple cities, which potentially increases the diversity of the source data and thus avoids the high-similarity constraint between source cities and target cities. For example, a new transfer framework [11] tries to generate mobility data for a target city by transferring knowledge from multiple source cities. MetaST [32] focuses on time series traffic prediction using multiple cities as source cities. However, both of them cannot solve the conditional

978-1-6654-5099-7/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDM54844.2022.00085

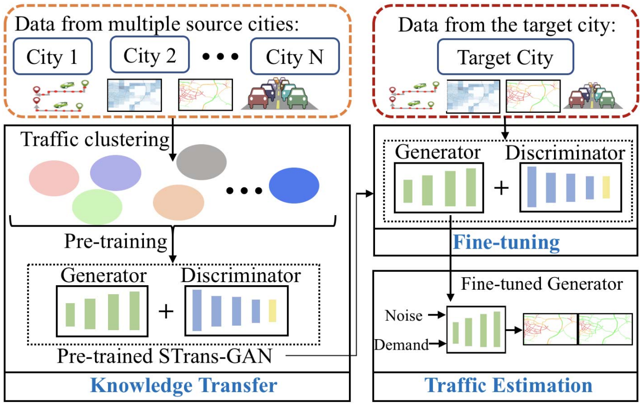


Fig. 2: Solution framework.

traffic estimation problem, since they did not consider the impact of travel demands and the diverse traffic patterns in different cities.

Our insight. When solving the cross-city conditional traffic estimation problem in the transfer learning paradigm, compared to one single source city, the shared knowledge extracted from multiple source cities would contain more comprehensive traffic patterns, and provide more information about how traffic status changes along the complex road networks in different regions. Therefore, transferring from multiple source cities may greatly improve the estimation accuracy and transfer stability. For example, as illustrated in Figure 1, with a new urban construction plan to be evaluated in Xi'An City which only has little traffic data available, we can borrow the traffic knowledge extracted from multiple source cities (*e.g.*, Shenzhen, Chengdu, etc.) to provide more accurate traffic estimations for Xi'An.

Challenges. In this paper, we study the problem of conditional traffic estimation in case of data scarcity by transferring knowledge from multiple source cities. However, such cross-city conditional traffic estimation problem is hard to solve due to two key challenges:

(1) *Knowledge extraction and transfer.* The traffic patterns between travel demands and local traffic status highly depend on the complex road networks and vary from region to region and time to time, which lead to the difficulties in knowledge extraction and transfer. When multiple cities are considered as source cities, the traffic patterns become even more complicated and thus harder to learn and transfer.

(2) *Knowledge adaptation.* The extracted knowledge should be well-adapted to the target city. Since different regions of the target city in different time slots could have different traffic patterns, the extracted knowledge should be adapted to all these scenarios in different ways. Thus, how to perform the knowledge adaption in a flexible manner is very important but challenging.

Contributions. To solve the cross-city conditional urban traffic estimation problem in case of data scarcity, we propose to estimate traffic in a data generation perspective under the multiple-city transfer learning setup. Hence, a new spatial transfer generative learning framework — Spatially-Transferable Generative Adversarial Networks (STrans-GAN) is proposed, which successfully tackles both of the aforementioned challenges and provides accurate traffic estimations based on various travel demands. Figure 2 is our solution framework. To solve the first challenge, traffic clustering is performed to aggregate all historical traffic data from multiple source cities into different clusters based on their traffic patterns, and then in the pre-training process, meta-learning idea is incorporated to learn a good global-initialized model from all clusters. To address the second challenge, the pre-trained STrans-GAN can be adapted to any scenario of the target city with only few samples by adding an extra cluster matching regularizer. Our main contributions can be summarized as follows:

- To the best of our knowledge, we are the first to solve the cross-city conditional traffic estimation problem in case of data scarcity from a spatial transfer generative learning perspective, and propose a novel method — Spatially-Transferable Generative Adversarial Networks (STrans-GAN).
- STrans-GAN preserves various traffic patterns from multiple source cities through traffic clustering, and incorporates meta-learning idea into the pre-training process to learn a global generalized model, which targets the first challenge. During fine-tuning, a new cluster matching regularizer is added to realize the flexible adaptation in different scenarios and thus addresses the second challenge.
- Extensive experiments on multiple-city datasets are performed to evaluate the effectiveness of our STrans-GAN. The experimental results prove that STrans-GAN significantly improves the urban traffic estimation performance and outperforms state-of-the-art baselines.

TABLE I: Notations

Notations	**Descriptions**
$\mathcal{S} = \{s_{ij}\}$	Grid cells within a city
$d_s \in \mathbb{N}$	Travel demand of a grid cell s
$\boldsymbol{d}_R \in \mathbb{N}^{r\times r}$	Travel demand of a region
$\mathcal{T} = \{\tau\}$	Trajectory set
$\boldsymbol{x}_s \in \mathbb{R}$	Traffic status of a grid cell s
$\boldsymbol{x}_R \in \mathbb{R}^{r\times r}$	Traffic distribution of the region R
$\mathcal{U}_{\text{source}} = \{u^i_{\text{source}}\}$	Source cities
u_{target}	Target city
$\mathcal{D}_{\text{source}} = \{\boldsymbol{d}_R\}$	Travel demands from source city regions
$\mathcal{X}_{\text{source}} = \{\boldsymbol{x}_R\}$	traffic distributions from source city regions
$\mathcal{D}_{\text{target}} = \{\boldsymbol{d}_R\}$	Travel demands from target city regions
$\mathcal{X}_{\text{target}} = \{\boldsymbol{x}_R\}$	traffic distributions from target city regions

II. Preliminaries

In this section, we first introduce the definitions and then formally define our problem.

Definition 1 (Grid cells). A city is partitioned into $m_1 \times m_2$ grid cells, each grid cell has equal side-length in latitude and longitude. The set of grid cells of one city is defined as $\mathcal{S} = \{s_{ij}\}$, where $\langle i, j\rangle$ indicates the coordinates of the grid cell s_{ij}, $1 \le i \le m_1$ and $1 \le j \le m_2$.

Definition 2 (Target region). A target region R is a square geographic region in a city, formed with $r \times r$ grid cells. A whole city can be split into multiple regions $\mathcal{R} = \{R_{ij}\}$, where $\langle i, j \rangle$ indicates the coordinates of the top-left grid cell in the region R_{ij}.

Definition 3 (Travel Demand). The travel demand of an area captures the total number of departures in a period of time. Thus, during one time period, the travel demand of a grid cell s is denoted as $d_s \in \mathbb{N}_0$, and the travel demand of a target region is denoted with a matrix $\boldsymbol{d}_R \in \mathbb{N}_0^{r \times r}$, where each entry d_s within the matrix $\boldsymbol{d}_R$ indicates the corresponding travel demand of the grid cell s within the region R. In this study, we use the demand for taxis to represent travel demand just as many literature works [9], [38], [39].

Definition 4 (Traffic status and traffic distribution). Traffic status is the basic knowledge of the road network traffic at a grid cell, which can be measured by traffic speed, traffic inflow/outflow, *etc*. We denote x_s as the average traffic status of a grid cell s within a period of time, and denote $\boldsymbol{x}_R \in \mathbb{R}^{r \times r}$ as the traffic distribution matrix of the region R, which is an $r \times r$ matrix composed of traffic status of all grid cells within the region.

Problem Statement: Given multiple source cities $\mathcal{U}_{\text{source}} = \{u^i_{\text{source}}\}$ and one target city u_{target} partitioned into grid cells, historical samples of travel demands $\mathcal{D}_{\text{source}} = \{\boldsymbol{d}_R\}$ and traffic distributions $\mathcal{X}_{\text{source}} = \{\boldsymbol{x}_R\}$ from all source cities, and only a small amount of historical samples of travel demands $\mathcal{D}_{\text{target}} = \{\boldsymbol{d}_R\}$ and traffic distributions $\mathcal{X}_{\text{target}} = \{\boldsymbol{x}_R\}$ from the target city, we aim to estimate the future traffic distributions $\{\tilde{\boldsymbol{x}}_R\}$ for a set of new travel demands $\{\tilde{\boldsymbol{d}}_R\}$ from the target city u_{target}.

III. Methodologies

Inspired by generative adversarial networks (GAN) [10], we are trying to solve the cross-city conditional traffic estimation problem in a traffic data generation perspective. The state-of-the-art conditional GAN (cGAN) model [20] seems to be a promising method, and we can view travel demands as conditions and traffic distributions as "images". However, training a good cGAN model requires a large amount of training data to ensure the convergence. If we perform traffic estimation in a city which faces data scarcity, cGAN would definitely fail due to the lack of training samples.

Thus, to better solve the cross-city conditional traffic estimation problem in case of data scarcity, we propose a spatial transfer generative learning framework — STrans-GAN which combines the generative model with multiple-city transfer learning paradigm. STrans-GAN also addresses the aforementioned challenges with novel designs:

(1) *Knowledge extraction and transfer.* To tackle the first challenge, we propose a unique architecture and a pre-training algorithm. STrans-GAN preserves various traffic patterns through traffic clustering, and incorporates meta-learning idea into the pre-training process to learn a well-initialized model.

(2) *Knowledge adaptation.* During fine-tuning, we propose to add an extra cluster matching regularizer to realize the flexible adaptation in different scenarios of the target city. Besides, a novel fine-tuning algorithm is proposed.

In this section, we first introduce our STrans-GAN architecture, and then detail the novel knowledge transfer and fine-tuning processes.

A. Model Architecture

STrans-GAN is a framework designed to solve the cross-city conditional urban traffic estimation problem in the multiple-city transfer learning setup. Since the effectiveness of GANs in traffic estimation has been proved in recent studies [38], [39], in this work, we design our STrans-GAN on top of cGAN. STrans-GAN has convolutional layers inside to better capture the complex spatial traffic dependencies, and combines the adversarial loss with L1 regularizer to improve the estimation performance during pre-training process. Figure 3(a) is the detailed architecture of our STrans-GAN, which is composed of a generator G and a discriminator D.

The generator G aims to learn a distribution $\boldsymbol{x} \sim G(\boldsymbol{z}, \boldsymbol{d})$ which matches the real data distribution p_{data}. The input of the generator G includes i) a noise vector $\boldsymbol{z}$ randomly sampled from Gaussian distribution, *i.e.*, $\boldsymbol{z} \sim p_{\boldsymbol{z}}$, and ii) the travel demand $\boldsymbol{d}$. G outputs the generated traffic distribution $\boldsymbol{x} \sim G(\boldsymbol{z}, \boldsymbol{d})$. Inside the generator, $\boldsymbol{d}$ and $\boldsymbol{z}$ are first concatenated together and then pass the stacked transposed convolutional layers, all the layers are batch normalized and activated by ReLU or hyperbolic tangent function.

The discriminator D aims to distinguish the real traffic distributions from the generated ones by giving a high score if the input $\boldsymbol{x}$ is sampled from the training set and matches the travel demand $\boldsymbol{d}$, or producing a low score if the input $\boldsymbol{x}$ is a generated one or does not match the input $\boldsymbol{d}$. The discriminator has two components including D_{Body} and D_{Head}, the input of D_{Body} includes i) a traffic distribution $\boldsymbol{x}$, which could be sampled from the dataset or generated by the generator, *i.e.*, $\boldsymbol{x} \sim p_{\text{data}}$ or $\boldsymbol{x} \sim G(\boldsymbol{z}, \boldsymbol{d})$, and ii) the travel demand $\boldsymbol{d}$. Inside D_{Body}, $\boldsymbol{d}$ and $\boldsymbol{x}$ are concatenated together and pass all stacked convolutional layers, all the layers are batch normalized and activated by leaky ReLU or hyperbolic tangent function. The output of D_{Body} is a vector which can be viewed as the embedding of the input pair (*i.e.*, travel demand and traffic distribution), the embedding vector passes D_{Head} which contains a single fully-connected layer to get the final score.

The basic objective function is as Eq. 1:

$$\begin{aligned}\mathcal{L}_{cGAN}(G, D) =& \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D(\boldsymbol{x}, \boldsymbol{d})] \\ &+ \mathbb{E}_{\boldsymbol{z} \sim p_z}[\log(1 - D(G(\boldsymbol{z}, \boldsymbol{d}), \boldsymbol{d}))].\end{aligned} \tag{1}$$

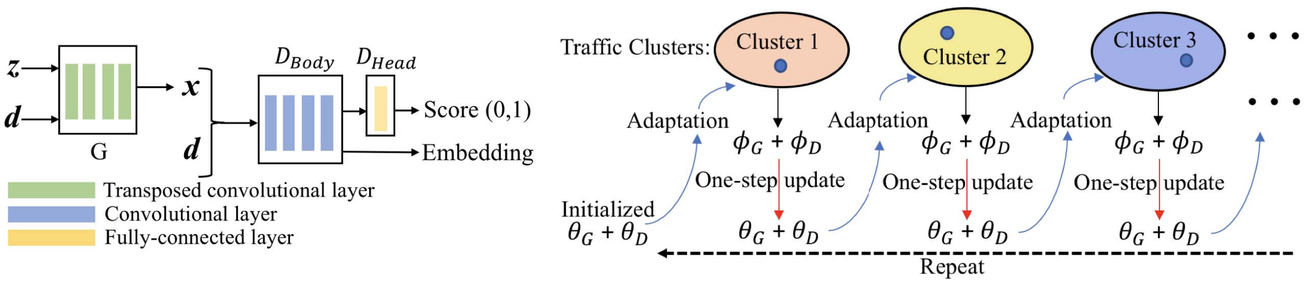
To avoid overfitting and improve the estimation performance, we combine the Eq. 1 with L1 regularizer [14]:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x \sim p_{\text{data}}, \boldsymbol{z} \sim p_z}\left[\|\boldsymbol{x} - G(\boldsymbol{z}, \boldsymbol{d})\|_1\right], \tag{2}$$

thus, the final objective is as Eq. 3:

$$\mathcal{L}(G, D) = \mathcal{L}_{cGAN}(G, D) + \alpha \mathcal{L}_{L1}(G), \tag{3}$$

and the optimization process is a min-max game, *i.e.*, $\min_G \max_D \mathcal{L}(G, D)$, where the generator tries to minimize the loss while the discriminator tries to maximize it.



(a) STrans-GAN Architecture

(b) Pre-training process

Fig. 3: STrans-GAN Overview.

B. *Knowledge Transfer*

The STrans-GAN introduced above only works when enough training data available, once data scarcity appears, it is hard to get a well-trained model. To better solve the traffic estimation problem especially in a city facing data scarcity problem, we propose to enable the STrans-GAN to learn from multiple source cities, which means the knowledge learned from source cities can be subsequently transferred to new target cities.

However, knowledge extraction and transfer from multiple source cities to the target city is very challenging. Given the traffic data (including traffic status and travel demands) and the road networks from multiple source cities, the key steps of knowledge transfer includes (i) disentangling and preserving different traffic patterns from multiple sources, and (ii) learning an initialized STrans-GAN by which the learned knowledge can be transferred. Thus, we first perform traffic clustering to detect different traffic patterns, and then design a novel pre-training algorithm which incorporates meta-learning idea to learn a well-initialized STrans-GAN.

1) Traffic Clustering: Each region in each source city may present various traffic patterns in different time slots, if the traffic distributions from all source cities are simply mixed together, the diversities of traffic patterns are ignored, which usually leads to very limited improvements than leaning from scratch (proved in the experiments). Thus, it is important to recognize different traffic patterns from multiple source cities.

Clustering is an effective way to disentangle patterns from data. However, given different source cities, we usually do not know what the proper number of clusters is, and whether the traffic data is clustered in a good way. To study how the number of clusters affect the performance and what number of clusters would be better for the given source cities, we apply k-means [18] to cluster the traffic data (*i.e.*, travel demand and traffic distribution pairs from all source cities). K-means is a flexible clustering method, it would produce different clusters based on different initial centroids, which helps to explore the best clustering way for different source cities, more empirical proofs are shown in Section IV-E3.

For a region R in a specific time slot, we have a travel demand $\boldsymbol{d}$ and a traffic distribution $\boldsymbol{x}$, and the average travel demand $\bar{d}$ and average traffic status $\bar{x}$ can be viewed as features to perform k-means clustering, where $\bar{d} = (\sum \boldsymbol{d})/r^2$, $\bar{x} = (\sum \boldsymbol{x})/r^2$.

In the traffic clustering, we execute the following four steps alternatively: (1) choose the number of clusters and randomly initialize the centroid for each traffic cluster; (2) assign all the traffic data pairs to their closest cluster centroid; (3) calculate the average feature of all traffic data within the cluster, and assign the new centroid for each newly formed cluster to the data point which is the closest to the average feature, (4) repeat previous two steps. Besides, the number of clusters is a hyper-parameter which will be tuned in the experiments.

Algorithm 1 STrans-GAN Pre-training Process

Input: Total training iterations K_1, n traffic clusters (*i.e.*, tasks), the innerloop k, initialized $\boldsymbol{\theta}_G$, $\boldsymbol{\theta}_D$, $\boldsymbol{\phi}_G$, and $\boldsymbol{\phi}_D$.
Output: Pre-trained $\boldsymbol{\theta}_G$, $\boldsymbol{\theta}_D$.

```
for iteration ← 1 to K_1 do
    for cluster ← 1 to n do
        φ_G ← θ_G.
        φ_D ← θ_D.
        Sample a batch of data {(x, d)}.
        for innerloop ← 1 to k do
            Sample a batch of noise vectors {z}.
            Update φ_D with Eq. 5.
            Update φ_G with Eq. 6.
        end for
        Update θ_D one step with Eq. 7.
        Update θ_G one step with Eq. 8.
    end for
end for
```

2) Pre-training: After clustering, the traffic data with similar patterns is clustered together, and the traffic patterns detection is finished, the next step is to find a way to make the traffic knowledge preserved in each cluster transferable. Thus, we propose a novel pre-training algorithm for our STrans-GAN on top of meta-leaning method Reptile [22], this new algorithm trains a global initialized STrans-GAN using the data from all clusters, and the traffic knowledge is preserved in the parameters of the pre-trained STrans-GAN.

Since the data in each cluster has similar patterns, we can treat each cluster as a traffic estimation task τ and get the estimation loss L_τ using the traffic data (*i.e.*, traffic demand and traffic distribution pairs) within cluster τ. During the pre-training process, our goal is to find the well-initialized generator and discriminator parameters $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_D$ using all training tasks (*i.e.*, clusters), which can quickly converge on

Algorithm 2 STrans-GAN Fine-tuning Process

Input: Total fine-tuning iterations K_2, n traffic clusters (*i.e.*, tasks), dataset for fine-tuning, pre-trained $\boldsymbol{\theta}_G$, $\boldsymbol{\theta}_D$.

Output: Fine-tuned $\boldsymbol{\theta}_G$, $\boldsymbol{\theta}_D$.

1: Get the centroid embeddings for each cluster $\boldsymbol{v}_c = D_{\text{body}}(\boldsymbol{x}_c, \boldsymbol{d}_c)$.
2: **for** iteration $\leftarrow 1$ to K_2 **do**
3: Sample a batch of data $\{(\boldsymbol{x}, \boldsymbol{d})\}$ from fine-tuning set
4: Sample a batch of noise vectors $\{\boldsymbol{z}\}$.
5: Update $\boldsymbol{\theta}_D$ with Eq. 12.
6: Update $\boldsymbol{\theta}_G$ with Eq. 13.
7: **end for**

a new task τ' with little data and few adaptation steps by minimizing the loss $L_{\tau'}$.

The objective of the pre-training process is as follows:

$$\min_{\boldsymbol{\theta}_G, \boldsymbol{\theta}_D} \mathbb{E}_{\tau} \left[L_{\tau} \left(U_{\tau}^{k}(\boldsymbol{\theta}_G, \boldsymbol{\theta}_D) \right) \right], \tag{4}$$

where U corresponds to one step of stochastic gradient descent [2] on D and G with respect to the loss in Eq 3, and we denote $\boldsymbol{\phi}_G, \boldsymbol{\phi}_D = U_{\tau}^{k}(\boldsymbol{\theta}_G, \boldsymbol{\theta}_D)$ as the adapted parameters of G and D after k steps of gradient descent in task τ.

The detailed pre-training algorithm is as Alg 1. The adapted parameters $\boldsymbol{\phi}_D$ and $\boldsymbol{\phi}_G$ of STrans-GAN for each cluster is updated using Eq 5 and Eq 6. After adapting $\boldsymbol{\phi}_G$ and $\boldsymbol{\phi}_D$ k steps in the innerloop, we update one step of $\boldsymbol{\theta}_D$ and $\boldsymbol{\theta}_G$ using Eq 7 and Eq 8.

$$\begin{aligned} &\mathcal{L}(\boldsymbol{\phi}_D) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} [\log D(\boldsymbol{x}, \boldsymbol{d})] \\ &+ \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}} [\log(1 - D(G(\boldsymbol{z}, \boldsymbol{d}), \boldsymbol{d})], \\ &\boldsymbol{\phi}_D = \boldsymbol{\phi}_D + \eta \nabla_{\boldsymbol{\phi}_D} \mathcal{L}(\boldsymbol{\phi}_D). \end{aligned} \tag{5}$$

$$\begin{aligned} &\mathcal{L}(\boldsymbol{\phi}_G) = \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}} [\log(1 - D(G(\boldsymbol{z}, \boldsymbol{d}), \boldsymbol{d}))] + \alpha \mathcal{L}_{L1}(G), \\ &\boldsymbol{\phi}_G = \boldsymbol{\phi}_G - \eta \nabla_{\boldsymbol{\phi}_G} \mathcal{L}(\boldsymbol{\phi}_G). \end{aligned} \tag{6}$$

$$\boldsymbol{\theta}_D = \boldsymbol{\theta}_D + \lambda(\boldsymbol{\theta}_D - \boldsymbol{\phi}_D). \tag{7}$$

$$\boldsymbol{\theta}_G = \boldsymbol{\theta}_G + \lambda(\boldsymbol{\theta}_G - \boldsymbol{\phi}_G). \tag{8}$$

We can get a good global initialized STrans-GAN through this pre-training process since its final outputs $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_D$ can reach the point which has minimum distance to each task just as illustrated in Eq 9, and this has been proved by many works [5], [22].

$$\min_{\boldsymbol{\theta}_G, \boldsymbol{\theta}_D} \sum_{\tau} |\boldsymbol{\theta}_D - \boldsymbol{\phi}_D^{\tau}| + |\boldsymbol{\theta}_G - \boldsymbol{\phi}_G^{\tau}|. \tag{9}$$

In Eq 9, $\boldsymbol{\phi}_G^{\tau}$ and $\boldsymbol{\phi}_D^{\tau}$ are the optimal parameters of discriminator and generator for task τ. Hence, if we have a new task τ' which is similar to one of the training tasks, $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_D$ can adapt to $\boldsymbol{\phi}_G^{\tau'}$, $\boldsymbol{\phi}_D^{\tau'}$ very fast, and thus the rapid and easy model generalization is realized.

C. Model Fine-tuning

Once we get a well-initialized STrans-GAN, we need to adapt it to the target city. However, a target city would have many different regions, each region would present different traffic patterns during different time slots, and it is common that the traffic distributions for a specific region belong to different traffic clusters. Thus, fine-tuning the pre-trained STrans-GAN with Eq 3 is not good enough, since it cannot guarantee that the pre-trained STrans-GAN can be correctly adapted to the corresponding traffic clusters for different traffic distributions.

To fine-tune the STrans-GAN parameters on the target city with few fine-tuning samples, we need to ensure the STrans-GAN can detect the traffic patterns for each sample (*i.e.*, travel demand and traffic distribution pair) and also generate reasonable traffic distributions, in other words, we need to make the fine-tuning process flexible for different traffic patterns. To realize this goal, firstly, we should guarantee the STrans-GAN can produce like-real traffic distributions based on different travel demands, which has been realized by adversarial loss and the L1 loss shown in Eq 3. Besides, we add a cluster matching regularizer, which enables the pre-trained STrans-GAN to be automatically adapted to different clusters based on different traffic patterns presented by the data. During fine-tuning, for each data sample of the target city, the cluster matching regularizer tries to minimize the distance between the data sample embedding and its corresponding cluster embedding, which helps the STrans-GAN to produce better generation results and present more clear traffic patterns. The cluster matching regularizer is defined as follows:

$$\begin{aligned} &\mathcal{L}_{\text{c}}(D) = \mathbb{E}\left[\|\boldsymbol{v}_c - \boldsymbol{v}\|_1\right], \\ &\boldsymbol{v}_c = D_{\text{body}}(\boldsymbol{x}_c, \boldsymbol{d}_c), \boldsymbol{v} = D'_{\text{body}}(\boldsymbol{x}, \boldsymbol{d}). \end{aligned} \tag{10}$$

In Eq 10, for each data sample $(\boldsymbol{x}, \boldsymbol{d})$ from the target city, we first figure out the exact cluster c that the data sample $(\boldsymbol{x}, \boldsymbol{d})$ belongs to by calculating the distance between the data sample and each cluster centroid and choosing the closest cluster, and then we get the centroid embedding $\boldsymbol{v}_c$ for cluster c by passing the centroid to D_{body}, *i.e.*, $\boldsymbol{v}_c = D_{\text{body}}(\boldsymbol{x}_c, \boldsymbol{d}_c)$, where $(\boldsymbol{x}_c, \boldsymbol{d}_c)$ is the centroid of cluster c and D_{body} is from the pre-trained discriminator. For the data sample $(\boldsymbol{x}, \boldsymbol{d})$ in the target city, we also get its current embedding $\boldsymbol{v}$ using D'_{body}, which is the discriminator being fine-tuned. We minimize the distance between the two embeddings, and thus realize the flexible fine-tuning.

The final objective during the fine-tuning process is as follows:

$$V(G, D) = \mathcal{L}_{cGAN}(G, D) + \alpha \mathcal{L}_{L1}(G) - \beta \mathcal{L}_{\text{c}}(D), \tag{11}$$

and the optimization process is $\min_G \max_D V(G, D)$.

The detailed fine-tuning algorithm is as Alg 2. The parameters of STrans-GAN are fine-tuned using Eq 12 and Eq 13.

After fine-tuning, the generator G can be used for traffic estimation in any region of the target city.

$$V(\boldsymbol{\theta}_D) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D(\boldsymbol{x}, \boldsymbol{d})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D(G(\boldsymbol{z}, \boldsymbol{d}), \boldsymbol{d})] - \beta\mathcal{L}_{\text{c}}(D),$$
$$\boldsymbol{\theta}_D = \boldsymbol{\theta}_D + \eta\nabla_{\boldsymbol{\theta}_D} V(\boldsymbol{\theta}_D). \tag{12}$$

$$V(\boldsymbol{\theta}_G) = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D(G(\boldsymbol{z}, \boldsymbol{d}), \boldsymbol{d}))] + \alpha\mathcal{L}_{L1}(G),$$
$$\boldsymbol{\theta}_G = \boldsymbol{\theta}_G - \eta\nabla_{\boldsymbol{\theta}_G} V(\boldsymbol{\theta}_G). \tag{13}$$

TABLE II: Dataset descriptions.

City	Data	Timespan	City size	Data size
Shenzhen	Speed	07/01/16-12/31/16	40×50	(155520,5,5)
	Inflow			
	Demand			
HB	Speed	07/01/15-12/31/15	40×50	(151440,5,5)
	Inflow			
	Demand			
Chengdu	Speed	10/01/16-10/31/16	20×20	(4784,5,5)
	Inflow			
	Demand			
Xi'An	Speed	10/01/16-10/31/16	20×20	(4368,5,5)
	Inflow			
	Demand			

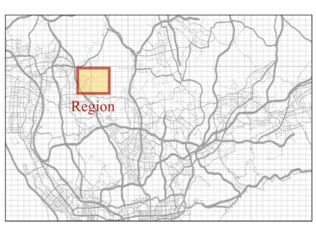
IV. Experiments

A. Dataset and Experiment Descriptions

Data Preprocessing In our experiments, we collect traffic data including travel demand, taxi inflow and traffic speed from Chengdu, Xi'An, Shenzhen and HB which is a northern city in China and referred to as HB. All the traffic data is extracted from (1) taxi GPS data; (2) road map data. Here we take the Shenzhen city as an example:

- Taxi GPS data. We collect the GPS records from 17,877 taxis in Shenzhen, China from Jul 1st to Dec 31st, 2016. The GPS sensors equipped in the taxis generate a GPS record every 40 seconds on average, and more than 51,000,000 GPS records can be collected each day. Each record contains five features including taxi ID, time stamp, passenger indicator, latitude and longitude. The passenger indicator is a binary value indicating whether a passenger is on board.
- Road map. In our study, we use the Google GeoCoding[1] to retrieve the bounding box of Shenzhen. The bounding box is defined between 22.534° to 22.87° in latitude and 113.77° to 114.40° in longitude. Shenzhen road map is shown in Fig. 4(a).

(a) Shenzhen road map



(b) Map gridding

Fig. 4: Shenzhen road map and map gridding.

[1]https://developers.google.com/maps/documentation/geocoding/

As shown in Figure 4(b), we apply map gridding method to the Shenzhen road map. The whole city is partitioned into 40×50 grid cells, and regions are formed by 5×5 grid cells. With the gridded road map and the taxi GPS records, we extract the travel demand, taxi inflow and traffic speed. In each time slot, with the passenger indicator feature in the GPS records of a taxi, we can easily monitor the passenger pickup and drop-off information, and we count the total number of pickup and drop-off events within each grid cell as the travel demand. And we count all taxis which stay or arrive at each grid cell as the taxi inflow. Traffic speed is calculated by the travel distance of a taxi and its corresponding time period.

Dataset Descriptions. After data preprocessing, we got our datasets for model training. The detailed information of datasets is shown in Table II.

In each city, with map gridding method, the whole city is partitioned into equal-sized grid cells, and we collect three different traffic datasets, *i.e.*, traffic speed, taxi inflow and travel demand. During each time slot (*i.e.*, one hour), traffic speed is the average speed extracted from taxis GPS records; taxi inflow indicates the total number of taxis that get into a grid cell; and travel demand calculates the total number of taxi pickup events within a grid cell. In this study, we use the demand for taxis to represent travel demand, and many studies have shown the effectiveness of using taxi demand to represent travel demand [9], [21], [38], [39]. More details about the dataset in each city are as follows:

- **Shenzhen.** The traffic data collected in Shenzhen, China is from Jul 1st to Dec 31st, 2016. The whole city is partitioned into 40×50 grid cells with a side-length $l_1 = 0.0084°$ in latitude and $l_2 = 0.0126°$ in longitude, and each region is formed by 5×5 grid cells. There are 155,520 region-wise traffic distributions in total, and each traffic distribution is a 5×5 matrix.
- **HB.** The traffic data collected in HB, China is from Jul 1st to Dec 31st, 2015. HB City is partitioned into 40×50 grid cells with a side-length $l_1 = 0.0084°$ in latitude and $l_2 = 0.0126°$ in longitude, and each region is formed by 5×5 grid cells. There are 151,440 region-wise traffic distributions in total.
- **Chengdu** The traffic data is collected from only one district of Chengdu, China, which is partitioned into 20×20 grid cells with multiple 5×5 regions, each grid cell has a side-length $l_1 = 0.0038°$ in latitude and $l_2 = 0.0045°$ in longitude. The data time span is from Oct 1st to Oct 31st, 2016. The total number of traffic distributions is 4,784.
- **Xi'An.** Similar to Chengdu, the traffic data in Xi'An is also collected from one district, which is also partitioned into 20×20 grid cells with a side-length $l_1 = 0.0041°$ in latitude and $l_2 = 0.0048°$ in longitude. The data time span is from Oct 1st to Oct 31st, 2016. The total number of traffic distributions is 4,784.

We have released our code and data[2] to support the reproducibility.

[2]https://www.dropbox.com/sh/nfjzwi3z6v2rw93/AAAYVFyO8Q7K57ufu2hwcLeSa?dl=0

TABLE III: Performance on experiment 1: traffic speed and taxi inflow estimation in Xi'An.

Methods		Smoothing	cGAN	Curb-GAN	TrafficGAN	Single-TL	Multi-TL	RegionTrans	MetaST	**STrans-GAN**
Speed	RMSE	14.062±0.471	19.255±0.726	15.748±0.562	14.783±0.578	13.783±0.242	11.702±0.685	14.697±0.281	14.771±0.572	**5.881±0.133**
	MAPE	0.524±0.019	0.818±0.028	0.590±0.023	0.596±0.017	0.381±0.012	0.348±0.017	0.557±0.021	0.566±0.015	**0.241±0.007**
Inflow	RMSE	144.335±4.258	102.938±4.213	188.368±6.244	101.168±3.321	112.379±4.321	105.986±5.299	137.457±5.928	118.219±4.576	**79.362±2.784**
	MAPE	2.796±0.021	0.912±0.068	1.589±0.158	1.837±0.110	0.787±0.062	0.878±0.044	3.258±0.352	0.860±0.025	**0.445±0.016**

TABLE IV: Performance on experiment 2: traffic speed and taxi inflow estimation in Chengdu.

Methods		Smoothing	cGAN	Curb-GAN	TrafficGAN	Single-TL	Multi-TL	RegionTrans	MetaST	**STrans-GAN**
Speed	RMSE	15.879±0.262	32.650±0.821	23.098±0.679	20.738±0.592	22.229±0.671	23.322±0.542	18.991±0.312	25.393±0.524	**11.763±0.231**
	MAPE	0.758±0.037	1.655±0.048	0.890±0.028	0.911±0.016	0.395±0.022	0.274±0.032	0.679±0.032	0.947±0.034	**0.176±0.011**
Inflow	RMSE	241.975±6.634	449.551±8.946	223.745±7.267	378.909±8.918	503.079±12.982	278.867±6.213	588.33±10.192	240.394±7.214	**95.597±4.423**
	MAPE	8.198±0.625	12.833±0.827	5.653±0.261	12.740±0.672	5.181±0.337	4.681±0.224	12.754±0.482	5.729±0.121	**1.181±0.033**

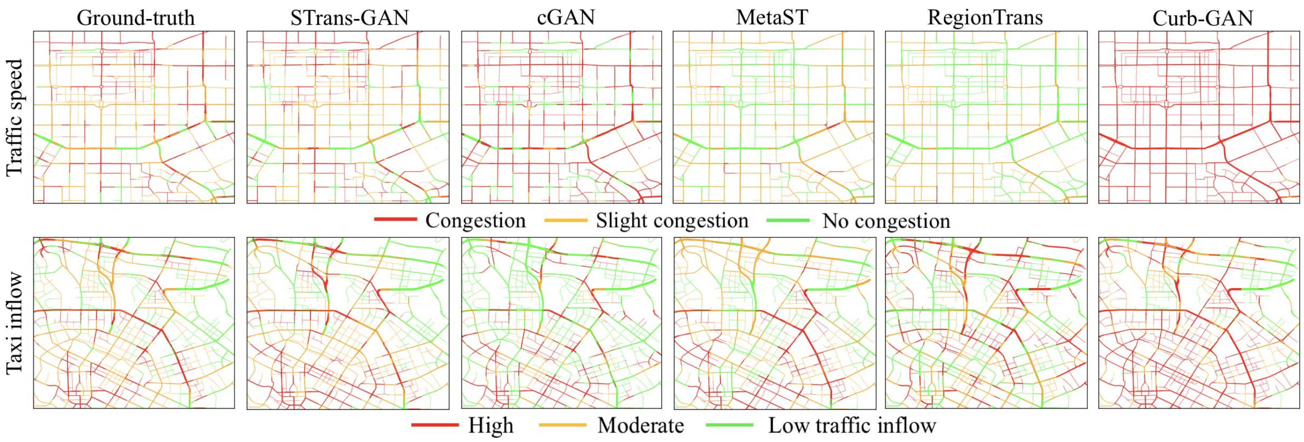


Fig. 5: Visualizations of (1st row) traffic speed estimation in Xi'An & (2nd row) taxi inflow estimation in Chengdu.

Experiment Descriptions. In our evaluation section, we conduct 2 different experiments to prove the effectiveness of our STrans-GAN:

- **Experiment 1: traffic speed and taxi inflow estimation in Xi'An.** In this experiment, we treat Shenzhen, HB and Chengdu as source cities, and view Xi'An as the target city. Given new travel demands of different regions in Xi'An, we aim to estimate the corresponding traffic speed and taxi inflow. Besides, 70% of data from Xi'An is used for fine-tuning, the remaining 30% is used for testing.
- **Experiment 2: traffic speed and taxi inflow estimation in Chengdu.** In this experiment, we treat Shenzhen, HB and Xi'An as source cities, and view Chengdu as the target city. Given new travel demands of different regions in Chengdu, we aim to estimate both traffic speed and taxi inflow. Similarly, we also use 70% of data from Chengdu for fine-tuning, and the remaining for testing.

B. Baselines

To verify the existing traffic estimation methods cannot produce good estimations when the target city has data scarcity problem, we compare our model with the state-of-the-art traffic estimation models:

- **Spatial Smoothing [8].** Given one target region and its travel demand, this method selects the traffic distributions from nearby regions which have similar travel demands and then computes an average traffic distribution as the final estimation.
- **cGAN [20].** The conditional GAN applies convolutional layers inside both generator and discriminator. We use travel demands as conditions to estimate the corresponding traffic distributions.
- **Curb-GAN [16], [39].** Curb-GAN applies self-attention and convolutional layers to deal with sequential traffic data generation problem. The generator is trained with the adversarial loss and L1 loss together.
- **TrafficGAN [14], [38]** TrafficGAN applies dynamic convolutional layers inside both generator and discriminator, and the generator is trained using both adversarial loss and L2 loss.

Besides, we compare our STrans-GAN with state-of-the-art transfer leaning methods including both single-source and multi-source transfer learning methods:

- **Multi-TL** Multi-source transfer learning uses the same architecture as STrans-GAN, but it is trained with simply mixed data from all source cities without clustering.
- **Single-TL** Single-source transfer learning uses the same architecture as STrans-GAN, but only has one source city for training.
- **RegionTrans [28]** This model targets time-series traffic prediction problem and only allows knowledge transfer from one source city to the target.
- **MetaST [32]** This model supports knowledge transfer from multiple cities, which treats each city as a task without considering different traffic patterns and directly applies MAML [7].

C. Evaluation Metrics

In our experiments, mean absolute percentage error (MAPE) and rooted mean square error (RMSE) are used to evaluate our


(a) Traffic cluster (b) λ (c) α (d) Embedding dimension (e) Innerloop

Fig. 6: Hyper-parameters studies on traffic speed estimation (Xi'An) and taxi inflow estimation (Chengdu).

model:

$$\mathrm{MAPE} = \frac{1}{n_s}\sum_{i=1}^{n_s} |y_i - \hat{y}_i| / y_i, \mathrm{RMSE} = \sqrt{\frac{1}{n_s}\sum_{i=1}^{n_s} (y_i - \hat{y}_i)^2} \quad , \tag{14}$$

where n_s is the total number of grid cells in the target region, y_i is the ground-truth traffic status observed in one grid cell s_i, and $\hat{y}_i$ is the corresponding estimated value.

D. Experimental Settings

Our experiments are running on Red Hat Enterprise Linux 7.2 with a GPU of K80 and CPU of E5-2680. The code released is in Python 3.7.3. The implementation of neural networks is based on PyTorch 1.0.14. We also employ Numpy 1.16.4 and Scipy 1.3.0 in the implementation.

For both experiments, we use data from three different source cities for pre-training, and select 70% of the data from the target city for fine-tuning and use the remaining 30% for testing. All models are updated using Adam optimizer [15]. The learning rate is set to 2×10^{-5}. The batch size is 32. Besides, we set $\alpha = 0.1$, $\beta = 0.1$, $\lambda = 0.1$. The detailed structure of STrans-GAN in our experiments is as follows: the generator G contains 4 transposed convolutional layers with kernel sizes $\{5, 5, 5, 5\}$ and output feature dimensions $\{1024, 128, 64, 1\}$; the discriminator D_{body} includes four convolutional layers with kernel sizes $\{5, 5, 5, 5\}$ and output channels $\{64, 128, 1024, 1024\}$. And the D_{head} only has one single fully-connect layer.

E. Experimental Results

1) Estimation Performance: In this part, we provide the evaluation statistics for our STrans-GAN and all baselines in both experiment. All deep models are trained and fine-tuned three times, and during testing time, we estimate the traffic for 16 different regions in the target city, and finally calculate the statistics using Eq 14. Based on the performance shown in Table III and Table IV, we have the following observations.

In both experiments, the traditional smoothing method has very high MAPE, which means the estimation at each grid cell is not accurate due to not considering local traffic patterns; other state-of-the-art traffic estimation models (including cGAN, TrafficGAN and Curb-GAN) do not provide good estimation results due to the lack of training samples. Besides, the single-source transfer learning methods (*i.e.*, RegionTrans and Single-TL) have low estimation quality, since these methods only take one city as source city, once the source city doesn't present similar traffic patterns to the target city, these methods would fail. For the multi-source transfer learning methods (*i.e.*, MetaST and Multi-TL), they cannot detect different traffic patterns in each source city, and thus they are unable to guarantee the model performance. Our STrans-GAN outperforms all baseline models in both experiments since it takes diverse traffic patterns into consideration and learns a good initialization from multiple source cities, and it is also successfully adapted to different scenarios in the target city.

2) Estimation Visualization: To further validate the effectiveness of our STrans-GAN, for both target cities including Xi'An and Chengdu, we present the visualizations of the ground-truth and the estimated traffic distributions over the road networks.

As shown in Figure 5, we visualize the traffic distributions including the ground-truth and the estimation results generated by our STrans-GAN and some competitive baselines (*i.e.*, cGAN, Curb-GAN, metaST and RegionTrans) in Xi'An City (the 1st row) and Chengdu City (the 2nd row). Obviously, in both experiments, cGAN and Curb-GAN cannot capture the traffic patterns very well due to the lack of data. MetaST and RegionTrans also present low quality estimations, they cannot identify different traffic patterns from source cities and fail to transfer useful knowledge to different target regions in the target city. In contrast, our STrans-GAN generates reasonable traffic distributions which are close to the ground-truth, and our STrans-GAN is able to successfully extract traffic knowledge from multiple source cities and adapt to the target city in a flexible way.

3) Hyper-parameter Studies: In our STrans-GAN, there are many hyper-parameters which may influence the model performance. In this section, we study how hyper-parameters influence the estimation performance of our STrans-GAN. The evaluated hyper-parameters includes the number of traffic clusters, embedding dimension, innerloop, λ and α.

From the results shown in Fig 6(a), we find in different target cities, the ideal number of traffic clusters varies since different cities would present different traffic patterns. In Xi'An City, the ideal number of traffic clusters is 5, in Chengdu City, the best number of traffic clusters is 10, which indicates Chengdu City has more complex traffic patterns compared to the traffic in Xi'An City.

In Fig 6(b), when updating the global-initialized STrans-GAN parameters $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_D$ during pre-training process, the step size λ matters. Too small or large λ hinders convergence. For both experiments, the ideal step size λ should be 0.1.

TABLE V: Ablation Study: traffic speed and taxi inflow estimation in Xi'An.

Methods		STrans-GAN$_c$	STrans-GAN$_p$	STrans-GAN$_f$	STrans-GAN$_r$	**STrans-GAN**
Speed	RMSE	11.702± 0.685	19.88±1.295	18.624±0.899	15.354±0.825	**5.881±0.133**
	MAPE	0.348±0.017	0.899±0.055	0.754±0.067	0.684±0.043	**0.241±0.007**
Inflow	RMSE	105.986±5.299	107.266±5.837	113.265±7.399	94.682±4.829	**79.362±2.784**
	MAPE	0.878±0.044	0.897±0.079	0.925±0.081	0.681±0.032	**0.445±0.016**

In Fig 6(c), we can find the estimation performance is sensitive to the value of α, α should be chosen based on the adversarial loss scale to ensure the whole loss scale keeps the same, in our experiments, the best choice of α is 0.1.

In Fig 6(d), we investigate how the embedding dimension influences the estimation performance. We find in both experiments, larger embedding dimension provides better estimation results, which indicates that larger embedding dimension can better preserve traffic patterns and potentially improve the estimation performance.

In Fig 6(e), the value of innerloop in Alg 1 is the number of steps when we adapt to each traffic cluster, and we study how the number of innerloop affects performance. It is obvious that too small or too large innerloop value usually leads to longer time to converge. In our experiments, the ideal number of innerloop should be 10 or 20.

4) Ablation Studies: Our STrans-GAN is composed of multiple components, including traffic clustering, pre-training, fine-tuning, and cluster matching regularizer, in this section, to verify the contribution of each component in our model, we present ablation studies.

As shown in Table V, we estimate traffic speed and taxi inflow with Xi'An City as the target city, and compare our STrans-GAN with STrans-GAN$_p$, STrans-GAN$_c$, STrans-GAN$_r$, STrans-GAN$_f$. STrans-GAN$_c$ removes the traffic clustering module, STrans-GAN$_p$ removes the pre-training part. In STrans-GAN$_f$, fine-tuning process is removed. And in STrans-GAN$_r$, the clustering matching regularizer is removed. Apparently, if pre-training module is removed in STrans-GAN$_p$, the cluster matching regularizer cannot work without a pre-trained model, and the limited data in the target city cannot support GAN training; if fine-tuning module is removed, the model cannot directly estimate the traffic in the target city without access to any data from the target; if the cluster matching regularizer is removed, the traffic clustering does not contribute to the fine-tuning process at all. Thus, each component in our Strans-GAN is important and contributes to the final estimation performance.

V. Related Work

In this section, we summarize the literature works from two related areas: 1) urban traffic estimation, and 2) urban transfer learning.

Urban traffic estimation. In recent years, more and more studies have focused on urban traffic estimation problem. Some works [1], [3], [12], [17], [25], [35] tried to apply classic machine learning methods to solve this problem. For example, the work [35] proposes to predict traffic volume by combining machine learning techniques and well-established traffic flow theory, and the works [6], [24], [25] propose novel frameworks to predict crowd flows and individual's movement.

Other works borrowed deep learning frameworks to solve various spatial-temporal prediction problems. For example, some works [33], [37] focus on citywide flow prediction and traffic demand prediction using CNN to better capture the spatial dependencies of urban data. Other works such as [26], [30], [34] try to solve travel time prediction and traffic speed prediction problems using recurrent neural networks [4] and LSTM [13] aiming to better capture the temporal dependencies within the data. Moreover, the work [41] tries to predict crowd density with ConvLSTM [23] to capture both spatial and temporal dependencies simultaneously. In addition, many previous works propose to solve the traffic estimation problem using generative adversarial networks [10]. For example, TrafficGAN [38], Curb-GAN [39] are proposed to estimate future traffic in an geographical region, C^3-GAN [40] tries to estimate traffic based on complex traffic related features. However, all these works would fail once lacking training samples.

Urban transfer learning. Transfer learning is a subarea in machine learning, which focuses on extracting and learning knowledge in one problem and applying it to a different but related problem. Transfer learning has been applied to many different urban scenarios, *e.g.*, many urban computing applications including traffic prediction, events detection, and urban deployment borrow transfer learning framework to solve urban data scarcity problem [29].

In previous studies, some works including [19], [28], [31], [36] propose novel traffic prediction frameworks on top of transfer leaning. For example, Wang et al. [27] propose to use transfer learning to solve ride-sourcing car detection problem. Among all these works, only one source domain is used to extract and learn knowledge. Some other works try to transfer knowledge from multiple source domains to target domains. For example, He et al. propose a novel mobility prediction framework [11] which transfers knowledge learned from multiple source cities to target cities aiming to generate mobility data for the target. Yao et al. [32] try to solve the traffic prediction problem using multiple cities as source cities. However, all these works cannot be generalized to solving conditional traffic estimation problem, and they did not consider the impact of travel demands and diverse traffic patterns.

VI. Conclusion

In this paper, we tackle the cross-city conditional traffic estimation problem in case of data scarcity, and we propose to perform traffic estimation with a novel spatial transfer generative learning framework — STrans-GAN, which combines generative models with transfer learning in multiple source cities setup. STrans-GAN preserves various traffic patterns through clustering, and incorporate meta-learning idea into the pre-training process to learn a good global generalized model. During fine-tuning, we propose to add a cluster matching regularizer aiming to realize the flexible adaptation in different scenarios. Besides, novel pre-training and fine-tuning algorithms are proposed. Through extensive experiments on multiple-city datasets, the effectiveness of STrans-GAN is proved, which significantly improves the estimation performance and outperforms all state-of-the-art baseline methods.

ACKNOWLEDGMENT

Yanhua Li and Yingxue Zhang were supported in part by NSF grants IIS-1942680 (CAREER), CNS-1952085, CMMI-1831140, and DGE-2021871. Jun Luo was partially supported by ARC Discovery Project (Grant DB210100743). Xun Zhou is funded partially by Safety Research using Simulation University Transportation Center (SAFER-SIM). SAFER-SIM is funded by a grant from the U.S. Department of Transportation's University Transportation Centers Program (69A3551747131). However, the U.S. Government assumes no liability for the contents or use thereof.

REFERENCES

[1] R. A. Anand, L. Vanajakshi, and S. C. Subramanian. Traffic density estimation under heterogeneous traffic conditions using data fusion. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 31–36, 2011.

[2] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Y. Lechevallier and G. Saporta, editors, *Proceedings of COMPSTAT'2010*, pages 177–186. Physica-Verlag HD, 2010.

[3] P. S. Castro, D. Zhang, and S. Li. Urban traffic modelling and prediction using large scale taxi GPS traces. In *Pervasive*, 2012.

[4] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, 2014.

[5] L. Clouâtre and M. Demers. FIGR: few-shot image generation with reptile. *CoRR*, 2019.

[6] Z. Fan, X. Song, R. Shibasaki, and R. Adachi. Citymomentum: An online approach for crowd behavior prediction at a citywide level. In *ACM UbiComp*, 2015.

[7] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 1126–1135, 2017.

[8] A. Getis. A history of the concept of spatial autocorrelation: A geographer's perspective. *Geographical Analysis*, 40, 2008.

[9] E. J. Gonzales, C. J. Yang, E. F. Morgul, and K. Ozbay. Modeling taxi demand with gps data from taxis and transit. Technical report, Mineta National Transit Research Consortium, 2014.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*. 2014.

[11] T. He, J. Bao, R. Li, S. Ruan, Y. Li, L. Song, H. He, and Y. Zheng. What is the human mobility in a new city: Transfer mobility knowledge across cities. In *Proceedings of The Web Conference 2020*, WWW '20, page 1355–1365, New York, NY, USA, 2020. Association for Computing Machinery.

[12] R. Herring, A. Hofleitner, P. Abbeel, and A. M. Bayen. Estimating arterial traffic conditions using sparse probe data. In *ITSC*, 2010.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 1997.

[14] P. Isola, J.-Y. Zhu, T. Zhou, and A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[16] P. Krahenbuhl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[17] X. Liu, X. Kong, and Y. Li. Collective traffic prediction with partially observed traffic history using location-base social media. In *CIKM*, 2016.

[18] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[19] T. Mallick, P. Balaprakash, E. Rask, and J. Macfarlane. Transfer learning with graph neural networks for short-term highway traffic forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10367–10374, 2021.

[20] M. Mirza and S. Osindero. Conditional generative adversarial nets. In *CoRR abs/1411.1784*, 2014.

[21] N. Mukai and N. Yoden. Taxi demand forecasting based on taxi probe data by neural network. In *IIMSS*, 2012.

[22] A. Nichol and J. Schulman. Reptile: a scalable metalearning algorithm. 03 2018.

[23] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W. chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 2015.

[24] X. Song, Q. Zhang, Y. Sekimoto, and R. Shibasaki. Prediction of human emergency behavior and their mobility following large-scale disaster. In *SIGKDD*, 2014.

[25] E. Toto, E. A. Rundensteiner, Y. Li, R. Jordan, M. Ishutkina, K. Claypool, J. Luo, and F. Zhang. Pulse: A real time system for crowd flow prediction at metropolitan subway stations. In *ECMLPKDD*, 2016.

[26] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng. When will you arrive? estimating travel time based on deep neural networks. In *AAAI*, 2018.

[27] L. Wang, X. Geng, J. Ke, C. Peng, X. Ma, D. Zhang, and Q. Yang. Ridesourcing car detection by transfer learning. *CoRR*, 2017.

[28] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang. Cross-city transfer learning for deep spatio-temporal prediction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 1893–1899. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[29] L. Wang, B. Guo, and Q. Yang. Smart city development with urban transfer learning. *Computer*, 51(12):32–41, 2018.

[30] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *NIPS*, 2017.

[31] Y. Wei, Y. Zheng, and Q. Yang. Transfer knowledge between cities. KDD '16. Association for Computing Machinery, 2016.

[32] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *The World Wide Web Conference*, WWW '19, page 2181–2191, New York, NY, USA, 2019. Association for Computing Machinery.

[33] H. Yao, F. Wu, J. ke, X. Tang, Y. Jia, S. Lu, P. Gong, and J. Ye. Deep multi-view spatial-temporal network for taxi demand prediction. 2018.

[34] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors*, 2017.

[35] X. Zhan, Y. Zheng, X. Yi, and S. Ukkusuri. Citywide traffic volume estimation using trajectory data. *TKDE*, 2017.

[36] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang. Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data. *IEEE Journal on Selected Areas in Communications*, 37(6):1389–1401, 2019.

[37] J. Zhang, Y. Zheng, and D. Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. *CoRR*, 2016.

[38] Y. Zhang, Y. Li, X. Zhou, X. Kong, and J. Luo. TrafficGAN: Off-deployment traffic estimation with traffic generative adversarial networks. In *ICDM*, 2019.

[39] Y. Zhang, Y. Li, X. Zhou, X. Kong, and J. Luo. 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*, KDD '20, 2020.

[40] Y. Zhang, Y. Li, X. Zhou, Z. Liu, and J. Luo. C^3-GAN: Complex-condition-controlled urban traffic estimation through generative adversarial networks. In *2021 IEEE International Conference on Data Mining (ICDM)*, 2021.

[41] A. Zonoozi, J. jae Kim, X.-L. Li, and G. Cong. Convolutional recurrent model for crowd density prediction with recurring periodic patterns. In *IJCAI*, 2018.