2022 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-5099-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDM54844.2022.00084

Mest-GAN: Cross-City Urban Traffic Estimation with Meta Spatial-Temporal Generative Adversarial Networks

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

***Abstract*—The conditional urban traffic estimation problem aims to accurately estimate the future traffic status based on the changing local travel demands, which has long been an important issue in urban planning. However, most existing methods require the target city to provide a large amount of traffic data. Once traffic estimation is performed in a "new" city where many urban services and transportation infrastructures are not built and thus no prior data is available, those works would fail due to the lack of data. In this paper, we aim to solve the conditional urban traffic estimation problem in case of data scarcity (*i.e.*, the target city cannot provide any prior data) and tackle the main challenges including (1) knowledge learning from the source and (2) knowledge adaptation without prior traffic data. We propose a novel generative adversarial network — Meta Spatial-Temporal Generative Adversarial Network (Mest-GAN), which can successfully estimate traffic in the target city based on local travel demands without the access to any prior traffic data. To address the first challenge, we learn the latent distribution of travel demands with the inference network, the latent distribution also indicates the diverse spatial-temporal traffic patterns. To solve the second challenge, we use the travel demand data in the target city for adaptation, where the inference network infers a latent code guiding the generator to produce accurate traffic estimations. Extensive experiments on real-world multiple-city datasets demonstrate that our Mest-GAN produces high-quality traffic estimations and outperforms state-of-the-art baseline methods.**

***Index Terms*—generative adversarial networks; urban traffic estimation.**

I. Introduction

The conditional urban traffic estimation problem aims to accurately estimate the future traffic status based on the changing local travel demands, which has long been an important issue in urban planning, especially in land use planning, subway routes planning, *etc*. In recent years, many works [9], [32], [33] have tried to solve the conditional urban traffic estimation problem to reduce potential traffic congestion and improve transportation efficiency using both classical machine learning models and deep learning models. However, these methods require the target city to provide a large amount of traffic data for training. If traffic estimation is performed in a "new" city where many urban services and transportation infrastructures are not built and thus no prior data is available, those previous works would fail due to the lack of data. *Thus, in this paper, we aim to solve the conditional urban traffic estimation problem in case of data scarcity (i.e., the target city cannot provide any prior data).*

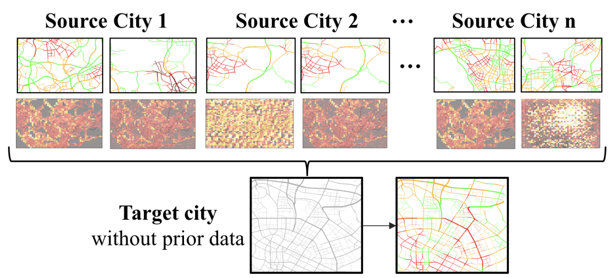
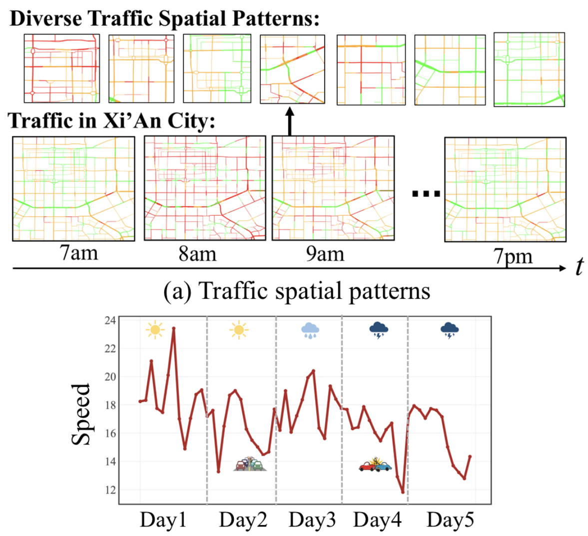


Fig. 1: Traffic before & after building subway stations.

To solve this problem, many works proposed to borrow the transfer learning or meta-learning paradigm where the knowledge learned from other source cities with abundant data is adapted to the target city to facilitate traffic estimation. For example, RegionTrans [27] proposed to predict the traffic by matching different locations in the target city to those in the source city based on traffic similarities. TL-DCRNN [15] forecast highway traffic by transferring the knowledge of the source to the target using graph neural networks. He et al. proposed a new transfer learning framework [8] for human mobility generation. STrans-GAN [34] utilized a spatially transferable generative adversarial network to estimate traffic in the target city by distinguishing and transferring diverse traffic patterns in the source cities. MetaST [28] focused on time series traffic prediction, which viewed each source city as a meta training task, and meta-learned a well-generalized model for future adaptation. However, all these methods still require prior data from the target city. Once the target city cannot provide any historical traffic data for model training or fine-tuning, all the existing methods cannot guarantee decent estimation performance.

Challenges. In this paper, we propose a novel generative adversarial meta-learning framework to solve the problem of conditional urban traffic estimation in a target city without any

978-1-6654-5099-7/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDM54844.2022.00084



(a) Traffic spatial patterns
(b) Traffic temporal patterns
Fig. 2: Illustration of traffic spatial-temporal patterns.

prior data, but we are hindered by some major challenges:
(1) *Knowledge learning from the sources.* The knowledge learned from source cities should include diverse spatial-temporal traffic patterns, which are very complicated and hard to detect. As illustrated in Figure 2, the traffic usually presents various spatial patterns across different regions due to the complex road networks, as well as complex temporal patterns from time to time. However, such spatial-temporal traffic patterns are hard to learn, some previous methods [28], [34] only consider spatial patterns without taking temporal patterns into consideration, and they manually define the number of traffic patterns, which usually lead to poor generalization and model instability.
(2) *knowledge adaptation without prior traffic data.* The learned knowledge should be quickly adapted to the target city without the access to any prior traffic data. However, the typical transfer learning methods [27], [34] require traffic data from the target city in the fine-tuning process. The supervised meta-learning methods [5], [28] also need traffic data from the target during the model adaptation process.

Contributions. In this paper, we aim to solve the conditional urban traffic estimation problem in case of data scarcity (*i.e.*, the target city cannot provide any prior data) and tackle both of the aforementioned challenges from the generative adversarial meta-learning perspective. We propose a novel generative adversarial network — Meta Spatial-Temporal Generative Adversarial Network (Mest-GAN), which can successfully estimate traffic in the target city based on local travel demands without the access to any prior traffic data. Figure 3 is our solution framework. Our Mest-GAN features a generator, a discriminator and an inference network, the well-generalized inference network and generator can be used for future traffic estimation in the target city. Moreover, to solve the first challenge, we learn the latent distribution of travel demands with the inference network, the latent distribution also indicates the diverse spatial-temporal traffic patterns. To solve the second challenge, we use the travel demand data in the target city for adaptation, where the inference network infers a latent code guiding the generator to produce accurate traffic estimations.

TABLE I: Notations

Notations	Descriptions
$\mathcal{S} = \{s_{ij}\}$	Grid cells within a city
$\mathcal{R} = \{R_{ij}\}$	Set of regions within a city
$c_s \in \mathbb{N}$	Travel demand of a grid cell s
$\boldsymbol{C}_t^R \in \mathbb{N}^{n \times n}$	Travel demand of region R at time t
$\boldsymbol{\mathcal{C}}^R = \{\boldsymbol{C}_1^R, \ldots, \boldsymbol{C}_T^R\}$	Travel demand in consecutive time slots
$\boldsymbol{x}_s \in \mathbb{R}$	Traffic status of a grid cell s
$\boldsymbol{X}_t^R \in \mathbb{R}^{n \times = n}$	Traffic distribution of region R at time t
$\boldsymbol{\mathcal{X}}^R = \{\boldsymbol{X}_1^R, \ldots, \boldsymbol{X}_T^R\}$	Traffic distribution in consecutive time slots
$\boldsymbol{c}$	latent variable
$\mathcal{V}_{\text{sr}} = \{v_{\text{sr}}^i\}$	Source cities
$\mathcal{V}_{\text{tg}} = \{v_{\text{tg}}^i\}$	Target cities
$\boldsymbol{\mathcal{C}}_{\text{sr}} = \{\boldsymbol{C}^R\}$	Travel demands from source city regions
$\boldsymbol{\mathcal{X}}_{\text{sr}} = \{\boldsymbol{X}^R\}$	traffic distributions from source city regions
$\boldsymbol{\mathcal{C}}_{\text{tg}} = \{\boldsymbol{C}^R\}$	Travel demands from target city regions
$\tilde{\boldsymbol{\mathcal{X}}}_{\text{tg}} = \{\tilde{\boldsymbol{X}}^R\}$	Estimated traffic for target city regions

Our main contributions can be summarized as follows:

- We formulate the cross-city conditional urban traffic estimation problem as an unsupervised meta-learning problem and we solve this problem from generative adversarial meta-learning perspective. A novel model — Meta Spatial-Temporal Generative Adversarial Network (Mest-GAN) is proposed, which can successfully estimate the traffic in target city in consecutive time slots. (See Section III-A.)
- We insert a latent variable to both generator and discriminator, which indicates the complex spatial-temporal traffic patterns of traffic. The inference network is designed to learn the latent distribution with travel demand data in the meta-training process. In the meta testing process, we use the travel demand data in the target city for adaptation, where the inference network infers a latent code guiding the generator to produce accurate traffic estimations, and thus the prior traffic data in the target city is not required. (See Section III-B and III-C.)
- We perform extensive experiments on multiple-city real-world datasets to evaluate our Mest-GAN. The experimental results prove that our Mest-GAN can significantly improve the urban traffic estimation performance in the data scarcity scenario and outperform state-of-the-art baseline methods. (See Section IV.) *We also made our code and dataset available to the research community [1].*

II. Preliminaries

The notations used in this paper are listed in Table I. Next, we introduce the definitions and our problem statement in detail.

Definition 1 (Grid cells). For a specific city, we partition the whole city area into $a_1 \times a_2$ grid cells with equal side-length in latitude and longitude, we denote the set of grid cells as $\mathcal{S} = \{s_{ij}\}$, where $1 \le i \le a_1, 1 \le j \le a_2$.

In our study, grid cells are the minimum units where traffic status and travel demand are measured. Alternatively, traffic estimation will be performed at a target region.

Definition 2 (Target region). A target region R formed by $n \times n$ grid cells is a square geographic region in the city. The

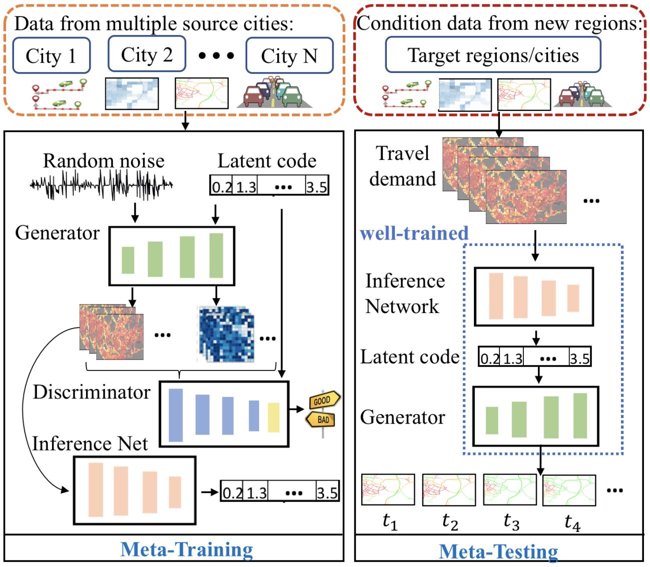


Fig. 3: Solution framework.

whole city can be split into many overlapping regions denoted as $\mathcal{R} = \{R_{ij}\}$, where $R_{ij} = \langle s_{ij}, n\rangle$ is uniquely defined by the grid cell s_{ij} in the top-left corner of the region and a number n indicating the side-length of the region.

Definition 3 (Travel demand). The travel demand of an area captures the total number of departures and arrivals during a specific time slot. In this paper, we denote the travel demand of a grid cell s in time slot t as $c_t^s \in \mathbb{N}$. Given a target region R, $\boldsymbol{C}_t^R \in \mathbb{N}^{n\times n}$ is an $n \times n$ matrix representing the travel demand of all grid cells within R during time slot t. Moreover, the travel demand of a target region R from time $t = \{1, \cdots, T\}$ is denoted as a sequence $\boldsymbol{\mathcal{C}}^R = \{\boldsymbol{C}_1^R, \dots, \boldsymbol{C}_T^R\} \in \mathbb{N}^{T\times n\times n}$.

Definition 4 (Traffic status and traffic distribution). Traffic status indicates the quality of traffic, which can be measured by traffic speed, traffic inflow/outflow, traffic volume, *etc.* We denote $x_t^s \in \mathbb{R}$ as the average traffic status of grid cell s in time slot t. Similarly, given a target region R with $n \times n$ grid cells, we denote the matrix $\boldsymbol{X}_t^R \in \mathbb{R}^{n\times n}$ as the traffic distribution of R during time slot t. Moreover, the traffic distributions of a target region R from time $t = \{1, \cdots, T\}$ is denoted as a sequence $\boldsymbol{\mathcal{X}}^R = \{\boldsymbol{X}_1^R, \dots, \boldsymbol{X}_T^R\} \in \mathbb{R}^{T\times n\times n}$.

Problem Statement: Given multiple source cities $\mathcal{V}_{\text{sr}} = \{v_{\text{sr}}^i\}$ and target cities $\mathcal{V}_{\text{tg}} = \{v_{\text{tg}}^i\}$ partitioned into grid cells, historical samples of travel demands $\boldsymbol{\mathcal{C}}_{\text{sr}} = \{\boldsymbol{C}^R\}$ and traffic distributions $\boldsymbol{\mathcal{X}}_{\text{sr}} = \{\boldsymbol{X}^R\}$ from source cities, and the travel demands $\boldsymbol{\mathcal{C}}_{\text{tg}} = \{\boldsymbol{C}^R\}$ from target cities without any prior traffic data, we aim to estimate the traffic distributions $\tilde{\boldsymbol{\mathcal{X}}}_{\text{tg}} = \{\tilde{\boldsymbol{X}}^R\}$ based on $\boldsymbol{\mathcal{C}}_{\text{tg}}$.

III. Methodologies

To estimate the urban traffic based on the local travel demands, we can simply view the travel demands as conditions, and apply the state-of-the-art generative adversarial networks including cGAN [16], TrafficGAN [32], Curb-GAN [33], C^3-GAN [35]. However, these model require large amounts of training data, and they definitely cannot work when the target city cannot provide data. Thus, in this paper, to solve the conditional urban traffic estimation problem in case of data scarcity (*i.e.*, the target city cannot provide any prior data), we propose a novel generative adversarial meta-learning framework — Meta Spatial-Temporal Generative Adversarial Network (Mest-GAN), which borrows the traffic knowledge learned from multiple source cities, adapts the knowledge to the target city without access to any prior data, and finally generates realistic traffic estimations in the target city. Mest-GAN also addresses the two challenges we mentioned in Section I with its unique designs:

(1) *Knowledge learning:* Mest-GAN has a uniquely designed architecture, which includes a generator G, a discriminator D, and an inference network Q. To enable the model automatically learn diverse spatial-temporal traffic patterns, we insert a latent variable to both generator and discriminator. The inference network is designed to learn the latent distribution with travel demand data. Novel meta-training algorithm is designed (See Section III-A and Section III-B).

(2) *Knowledge adaptation:* In the adaptation, we use the travel demand data in the target city for adaptation, where the inference network infers a latent code guiding the generator to produce accurate traffic estimations, and thus the prior traffic data in the target city is not required (See Section III-C).

A. Objective

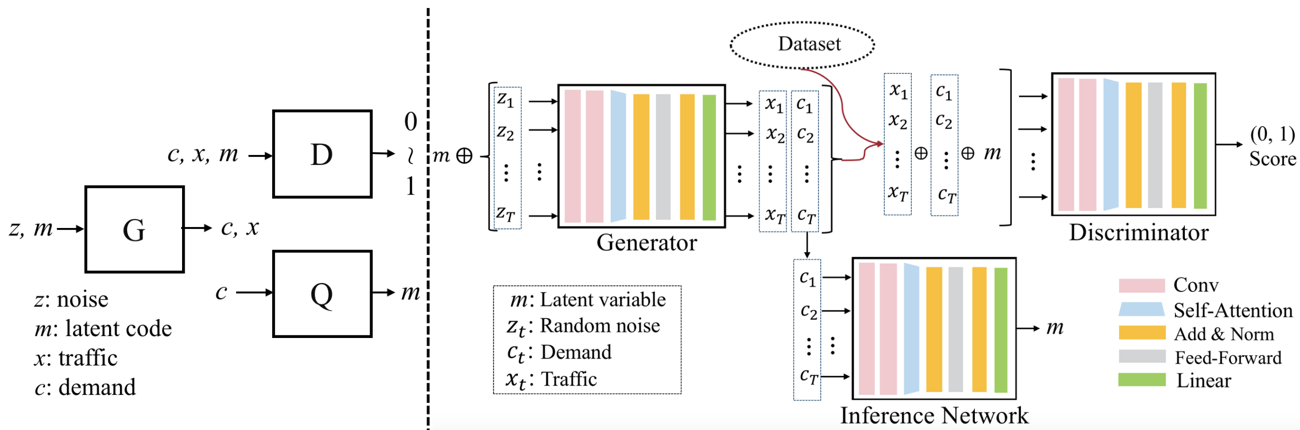
In many previous works [32]–[35], the conditional urban traffic estimation problem can be formulated as a traffic generation problem as below:

$$\min_G \max_D \mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{\boldsymbol{X}\sim p_{\text{data}}}[\log D(\boldsymbol{X}, \boldsymbol{C})] + \mathbb{E}_{\boldsymbol{Z}\sim p_Z}[\log(1 - D(G(\boldsymbol{Z}, \boldsymbol{C})))], \tag{1}$$

where $\boldsymbol{Z}$ is random noise sampled from Gaussian distribution $N(0, 1)$, $G(\boldsymbol{Z}, \boldsymbol{C})$ is a generator aiming to generate traffic distributions similar to the real traffic p_{data}, $D(\boldsymbol{X}, \boldsymbol{C})$ is a discriminator trying to distinguish the real data sampled from the dataset and the generated data sampled from the generator. The whole generation process is governed by conditions $\boldsymbol{C}$, which have a strong relation to the traffic data $\boldsymbol{X}$. Both generator and discriminator are deep neural networks, and the well-trained generator can successfully produce realistic traffic distributions matching the input travel demand C.

However, Eq 1 can only deal with the traffic estimation problem when the target city can provides large amounts of historical traffic data. Once we cannot collect enough data from the target city, this model would definitely fail. Moreover, this model can only estimate the traffic in a single city, once the data is collected from multiple source cities, the model will simply assume all the data is from the same city, and thus fail to distinguish diverse spatial-temporal traffic patters across cities and regions.

Therefore, to enable the model to automatically learn diverse traffic patterns from multiple source cities, we insert a latent variable $\boldsymbol{m}$ to the generator and discriminator. $\boldsymbol{m} \in p(\boldsymbol{m})$ is a latent code indicating a specific spatial-temporal



(a) Model Overview (b) Detailed Architecture

Fig. 4: Mest-GAN structure.

traffic pattern. Build upon the previous conditional generative adversarial networks (cGAN) based models, we transform the Eq 1 to Eq 2 as below:

$$\begin{aligned}\min_G \max_D V(G,D) = \mathbb{E}_{\boldsymbol{X},\boldsymbol{C}\sim p_{\text{data}}}[\log D(\boldsymbol{X},\boldsymbol{C},\boldsymbol{m})] \\ + \mathbb{E}_{\boldsymbol{Z}\sim p_{\boldsymbol{Z}}}[\log(1-D(G(\boldsymbol{Z},\boldsymbol{m}),\boldsymbol{m}))],\end{aligned} \tag{2}$$

where the we use the latent code $\boldsymbol{m} \in p(\boldsymbol{m})$ to guide the generation process, and the discriminator needs to tell the real traffic data from the generated one, and the real travel demand from the generated travel demand.

Moreover, since the condition (*i.e.*, travel demands) is highly related to the local traffic, which has been validated by many previous works [32], [33], we propose to add a mutual information term to Eq 2 to strength the connection between the travel demands and the latent variable $\boldsymbol{m}$. The mutual information between the latent variable $\boldsymbol{m}$ and the travel demands is denoted as $I(\boldsymbol{m},\boldsymbol{C})$, the objective with the mutual information regularizer is as follows:

$$\begin{aligned}&\min_G \max_D V_I(G,D) = V(G,D) - \lambda I(\boldsymbol{m};\boldsymbol{C}),\\ &\text{where } \boldsymbol{C} = G(\boldsymbol{Z},\boldsymbol{m}).\end{aligned} \tag{3}$$

In Eq 3, the latent variable $\boldsymbol{m}$ helps to identify different traffic patterns from the traffic data collected from multiple source cities, and it also enables the fast adaptation to a new region in the target city with only travel demand data available. However, in practice, it is hard to directly maximize the mutual information $I(\boldsymbol{m};\boldsymbol{C})$ without the access to the posterior distribution $P(\boldsymbol{m}|\boldsymbol{C})$. Instead, we calculate the variational lower bound [2], [21] of $I(\boldsymbol{m};\boldsymbol{C})$ and use an auxiliary distribution $Q(\boldsymbol{m}|\boldsymbol{C})$ to approximate the true posterior $P(\boldsymbol{m}|\boldsymbol{C})$:

$$\begin{aligned}&I(\boldsymbol{m};G(\boldsymbol{Z},\boldsymbol{m})) = H(\boldsymbol{m}) - H(\boldsymbol{m}\mid G(\boldsymbol{Z},\boldsymbol{m}))\\
&= \mathbb{E}_{\boldsymbol{C}\sim G(\boldsymbol{Z},\boldsymbol{m})}\left[\mathbb{E}_{\boldsymbol{m}'\sim P(\boldsymbol{m}|\hat{\boldsymbol{C}})}\left[\log P\left(\boldsymbol{m}'\mid\hat{\boldsymbol{C}}\right)\right]\right] + H(\boldsymbol{m})\\
&= \mathbb{E}_{\boldsymbol{C}\sim G(\boldsymbol{Z},\boldsymbol{m})}[\underbrace{D_{\mathrm{KL}}(P(\cdot\mid\hat{\boldsymbol{C}})\|Q(\cdot\mid\hat{\boldsymbol{C}}))}_{\geq 0}\\
&+ \mathbb{E}_{\boldsymbol{m}'\sim P(\boldsymbol{m}|\hat{\boldsymbol{C}})}\left[\log Q\left(\boldsymbol{m}'\mid\hat{\boldsymbol{C}}\right)\right]] + H(\boldsymbol{m})\\
&\geq \mathbb{E}_{\boldsymbol{C}\sim G(\boldsymbol{Z},\boldsymbol{m})}\left[\mathbb{E}_{\boldsymbol{m}'\sim P(\boldsymbol{m}|\hat{\boldsymbol{C}})}\left[\log Q\left(\boldsymbol{m}'\mid\hat{\boldsymbol{C}}\right)\right]\right] + H(\boldsymbol{m})\\
&= L_I(G,Q),\end{aligned} \tag{4}$$

where $p(\boldsymbol{m})$ is a prior distribution, Q is the auxiliary distribution, and we can treat Q as an inference neural network, which uses $\boldsymbol{C}$ to infer $\boldsymbol{m}$. Moreover, we can simply omit $H(\boldsymbol{m})$ in $L_I(G,Q)$ since it is a constant when $\boldsymbol{m}$ is sampled from a fixed distribution. As a result, the final objective for our Meta Spatial-Temporal Generative Adversarial Network (Mest-GAN) is as Eq 5:

$$\min_{G,Q}\max_D V(D,G,Q) = V(G,D) - \lambda L_I(G,Q). \tag{5}$$

B. Mest-GAN Architecture

Mest-GAN can successfully solve the cross-city conditional urban traffic estimation problem and address the limitations of the state-of-the-arts. The overview of the model is shown in Figure 4(a). Mest-GAN contains a generator G which generates traffic distributions of a region, a discriminator which tries to distinguish the real data and the generated data, and an inference network which infers the latent code based on the travel demand. Besides, Mest-GAN captures complex spatial-temporal dependencies with convolutional layers and self-attention mechanism within each model component (*i.e.*, the generator, discriminator and the inference network).

The generator G aims to generate like-real traffic distributions with respect to a latent code $\boldsymbol{m}$ and the random noise $\boldsymbol{Z}$. The input of the generator G includes i) a noise vector $\boldsymbol{Z}$, which is randomly sampled from Gaussian distribution, *i.e.*, $\boldsymbol{Z} \sim p_{\boldsymbol{Z}}$, and ii) a latent code $\boldsymbol{m}$. The latent vector indicates a specific spatial-temporal traffic pattern which is learned from the corresponding travel demand. G outputs the generated traffic distribution $\boldsymbol{X} \sim G(\boldsymbol{Z}, \boldsymbol{m})$ and the recovered travel demand $\boldsymbol{C} \sim G(\boldsymbol{Z}, \boldsymbol{m})$. Inside the generator G, $\boldsymbol{m}$ and $\boldsymbol{Z}$ are concatenated together and go through several convolutional layers, a multi-head self-attention layer [25] and a feed-forward network aiming to capture the complex spatial-temporal dependencies of traffic. The multi-head self-attention layer and a feed-forward network including two fully-connected layers are activated by ReLU activation function [19], and they are followed by an addition operation and a layer normalization. The final results are activated by hyperbolic tangent function.

The discriminator D aims to tell the real data sampled the dataset from the "fake" data generated by the generator. D will give a high score if the input traffic sequence $\boldsymbol{X}$ is from the training dataset and matches the input travel demand sequence $\boldsymbol{C}$, besides, D will also produce a high score if the input travel demand sequence $\boldsymbol{C}$ is real. By contrast, D will yield a low score if the input traffic $\boldsymbol{X}$ or travel demand $\boldsymbol{C}$ is generated by the generator, or they do not match. The input of D includes i) traffic sequence $\boldsymbol{X}$, which can be real data sampled from the dataset or fake data generated by the generator, *i.e.*, $\boldsymbol{X} \sim p_{\text{data}}$ or $\boldsymbol{X} \sim G(\boldsymbol{Z}, \boldsymbol{m})$, ii) the travel demand sequence $\boldsymbol{C}$ and iii) the latent code $\boldsymbol{m}$. Inside the discriminator D, $\boldsymbol{m}$, $\boldsymbol{X}$ and $\boldsymbol{C}$ are concatenated together and go through several convolutional layers and the self-attention mechanism [25] aiming to capture the complex spatial-temporal dependencies of traffic, the self-attention mechanism is composed of a multi-head self-attention layer and a feed-forward network including two fully-connected layers activated by ReLU activation function [19]. Both the self-attention layer and the feed-forward network are followed by an addition operation and a layer normalization. The final result is activated by Sigmoid function.

The inference network Q aims to infer the latent code $\boldsymbol{m}$ based on the travel demand sequence $\boldsymbol{C}$. Since the travel demand and local traffic is highly related, we can safely assume the travel demand can successfully provide the current spatial-temporal traffic pattern which is incorporated into the latent code $\boldsymbol{m}$. The input of Q is a sequence of travel demand, and the output is a latent code $\boldsymbol{m}$ which indicates a specific spatial-temporal traffic pattern. The architecture of Q is very close to that of generator.

C. Meta-Training and Meta-Testing

To optimize our final objective function Eq 5, we propose novel meta-training and meta-testing algorithms.

Meta-Training Process. Our objective Eq 5 requires a prior distribution $p(\boldsymbol{m})$, however, in most cases, the prior distribution $p(\boldsymbol{m})$ is hard to acquire (*e.g.*, we do not know the distribution of spatial-temporal traffic patterns of a city), but we can use the following generation process to synthesize latent variable $\boldsymbol{m}$, which approximates the prior distribution $p(\boldsymbol{m})$ when the generator G and the inference network Q are trained to optimality, the effectiveness has been validated by Yu et al. [30]:

$$\boldsymbol{C} \sim p_{\text{data}}, c \sim Q(m \mid \boldsymbol{C}) \tag{6}$$

Algorithm 1 Meta-Training Process

Input: Traffic data and travel demand data collected from multiple source cities, *i.e.*, $\mathcal{D} = \{(\boldsymbol{X}, \boldsymbol{C})^R\}$, initial parameters of generator, discriminator and inference network $\boldsymbol{\psi}_0$, $\boldsymbol{\theta}_0$, $\boldsymbol{\omega}_0$, repectively.

Output: Learned generator $G_{\boldsymbol{\psi}}$, discriminator $D_{\boldsymbol{\theta}}$ and inference network $Q_{\boldsymbol{\omega}}$.

1: **repeat**
2: Sample two batches of traffic and travel demand pairs b_{real} and b'_{real}: $b_{\text{real}}, b'_{\text{real}} \sim \mathcal{D}$
3: Infer a batch of latent codes $\boldsymbol{m}$ from b_{real}: $\boldsymbol{m} \sim Q_{\boldsymbol{\omega}}(\boldsymbol{m} \mid b_{\text{real}})$.
4: Sample a batch of generated traffic and travel demand pairs b_{fake} using the generator $G_{\boldsymbol{\psi}}$, *i.e.* $b_{\text{fake}} \sim G_{\boldsymbol{\psi}}(b_{\text{fake}} \mid \boldsymbol{m})$.
5: Update $\boldsymbol{\theta}$ with Adam [13] to maximize Eq. 7 using b'_{real} and b_{fake}.
6: Update $\boldsymbol{\omega}$ with Adam [13] to minimize Eq. 9 using b_{real} and b_{fake}.
7: Update $\boldsymbol{\psi}$ with with Adam [13] to minimize Eq. 8.
8: **until** Convergence

The detailed training process is in Algorithm 1. In Algorithm 1, the generator G, the discriminator D and the inference network Q are updated using Eq. 8, Eq. 7, Eq. 9, respectively. Denote $\boldsymbol{\theta}$ as the parameters of D, η as the learning rate, we update the discriminator D with Eq. 7:

$$\begin{aligned}
&\mathcal{L}_D(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{X}, \boldsymbol{C} \sim p_{\text{data}}}[\log D(\boldsymbol{X}, \boldsymbol{C}, \boldsymbol{m})] \\
&+ \mathbb{E}_{\boldsymbol{Z} \sim p_{\boldsymbol{Z}}}[\log(1 - D(G(\boldsymbol{Z}, \boldsymbol{m}), \boldsymbol{m}))], \\
&\boldsymbol{\theta} = \boldsymbol{\theta} + \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}_D(\boldsymbol{\theta}).
\end{aligned} \tag{7}$$

For the generator G, we denote $\boldsymbol{\psi}$ as the parameters of G, the loss function and updating rule for G is as follows:

$$\begin{aligned}
&\mathcal{L}_G(\boldsymbol{\psi}) = \mathbb{E}_{\boldsymbol{Z} \sim p_{\boldsymbol{Z}}}[\log(1 - D(G(\boldsymbol{Z}, \boldsymbol{m}), \boldsymbol{m}))] \\
&- \lambda \mathbb{E}_{\boldsymbol{C} \sim G(\boldsymbol{Z}, \boldsymbol{m})}\left[\mathbb{E}_{\boldsymbol{m}' \sim P(\boldsymbol{m} \mid \hat{\boldsymbol{C}})}\left[\log Q\left(\boldsymbol{m}' \mid \hat{\boldsymbol{C}}\right)\right]\right], \\
&\boldsymbol{\psi} = \boldsymbol{\psi} - \eta \nabla_{\boldsymbol{\psi}} \mathcal{L}_G(\boldsymbol{\psi}).
\end{aligned} \tag{8}$$

For the inference network Q, we denote $\boldsymbol{\omega}$ as the parameters of Q, we update Q with the following rules:

$$\begin{aligned}
&\mathcal{L}_Q(\boldsymbol{\omega}) = -\lambda \mathbb{E}_{\boldsymbol{C} \sim G(\boldsymbol{Z}, \boldsymbol{m})}\left[\mathbb{E}_{\boldsymbol{m}' \sim P(\boldsymbol{m} \mid \hat{\boldsymbol{C}})}\left[\log Q\left(\boldsymbol{m}' \mid \hat{\boldsymbol{C}}\right)\right]\right], \\
&\boldsymbol{\omega} = \boldsymbol{\omega} - \eta \nabla_{\boldsymbol{\omega}} \mathcal{L}_Q(\boldsymbol{\omega}).
\end{aligned} \tag{9}$$

The detailed training process is shown in Algorithm 1, where D is updated in line 6, G and Q are updated in line 7 and line 8, respectively. After convergence, we get the distribution of spatial-temporal traffic patterns, *i.e.*, $p(\boldsymbol{m})$.

Meta-Testing Process. During meta-testing process, we can directly infer the latent code using the local travel demand in the target city, and the latent code will guide the generator to produce the corresponding traffic estimations. (See Alg 2).

Algorithm 2 Meta-Testing Process

Input: New travel demand data in the target city $\{\tilde{\boldsymbol{C}}\}$, well-trained G and Q.

Output: Traffic estimations $\{\tilde{\boldsymbol{X}}\}$.

1: Infer latent codes $\{\tilde{\boldsymbol{m}}\}$ based on $\{\tilde{\boldsymbol{C}}\}$ with Q.

2: Sample noise vectors $\{\boldsymbol{Z}\}$ from Gaussian distribution.

3: Output traffic estimations $\tilde{\boldsymbol{X}} = G(\boldsymbol{Z}, \tilde{\boldsymbol{m}})$ with generator G.

TABLE II: Dataset descriptions.

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

IV. Evaluation

In this section, we provide extensive experiments on datasets collected from multiple cities to validate the effectiveness of our Mest-GAN.

A. Datasets and Experiment Descriptions

Dataset Descriptions. In our experiments, we collect taxi GPS records from multiple cities including Chengdu, Xi'An, Shenzhen and HB which is a northern city in China and referred to as HB. A GPS record includes five attributes including the taxi plate ID, longitude, latitude, time stamp and passenger indicator which is a binary value indicating whether a passenger is on board (*e.g.*, 0 indicates no passenger on board and 1 otherwise). Different traffic data can be extracted from the taxi GPS records including traffic speed, travel demand and taxi inflow The detailed information of dataset is shown in Table II.

In each city, we partition the whole city area into equal-sized grid cells. The traffic status in each grid cell is measured by average traffic speed and taxi inflow. The average traffic speed in each time slot is calculated by dividing the travel distance by the time period; taxi inflow is the total number of arrivals at each grid cell within a specific time slot, and the travel demand captures the number of taxi pickup and drop-off events within a grid cell during a time slot. Since it is hard to collect the total travel demand from all transport modes, in this study, we use the demand for taxis instead, its effectiveness has been validated in many previous works [6], [18], [32], [33]. More details about each city are as follows:

- **Shenzhen.** The taxi GPS records collected in Shenzhen, China is from Jul 1st to Dec 31st, 2016. Shenzhen city is divided 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 20×20 grid cells. There are 155,520 region-wise traffic sequences in total, each traffic sequence includes 12 traffic distributions from 7am to 7pm (*i.e.*, we view one hour as a time slot).
- **HB.** The taxi GPS records 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 20×20 grid cells. There are 155,520 region-wise traffic sequences in total, each traffic sequence indicates the traffic changes within one day from 7am to 7pm.
- **Chengdu.** In Chengdu City, we collect taxi GPS records from one region which contains 20×20 grid cells, 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, there are 31 region-wise traffic sequences, each sequence includes 12 traffic distributions from 7am to 7pm per day.
- **Xi'An.** Similar to Chengdu, in Xi'An City, we collect taxi GPS records from only one region which contains 20×20 grid cells, each grid cell has a side-length $l_1 = 0.0048°$ in latitude and $l_2 = 0.0041°$ in longitude. The data time span is from Oct 1st to Oct 31st, 2016, there are 31 region-wise traffic sequences, each sequence includes 12 traffic distributions from 7am to 7pm per day.

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

- **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 Chengdu, we aim to estimate both traffic speed and taxi inflow. We use the data from Shenzhen, HB and Xi'An for meta-training, and we only provide the travel demand in Xi'An during meta-testing process.
- **Experiment 2: 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 Xi'An, we aim to estimate the corresponding traffic speed and taxi inflow. In the experiment, we use the data from Shenzhen, HB and Chengdu for meta-training, in the meta-testing process, we only provide the travel demand in Xi'An.

B. Baselines

To evaluate the effectiveness of our model, we compare our Mest-GAN with other state-of-the-art methods. We first use the following two baselines including cGAN+LSTM and Curb-GAN to validate that classical traffic estimation methods cannot deal with the data scarcity problem and cannot provide realistic traffic estimations in the target city:

- **cGAN+LSTM [10], [16]** The standard conditional GAN combined with LSTM is used to capture the spatial-temporal dependencies of traffic, which includes a generator and a discriminator with convolutional layers and LSTM inside. The model is trained on the data collected from source cities and tested on the target city.

- **Curb-GAN [33]** The Curb-GAN contains a generator and a discriminator, and dynamic convolutional layers [32] and self-attention layers are embedded into both model components. This model is trained on the data collected from source cities and tested on the target city.

Next, we compare our Mest-GAN with other state-of-the-art methods which apply transfer learning and meta-learning framework to address the traffic estimation problem in the data scarcity scenario:

- **RegionTrans [27]** This model targets time-series traffic prediction problem and only allows knowledge transfer from one source city to the target. When applying to our traffic estimation problem in case of data scarcity in the target city, we treat travel demand as external context which is the input of ConvLSTM. Besides, we only use one city as the source for model training, during testing process, we use the travel demand data in the target city for future traffic estimation.
- **MetaST [28]** MetaST is designed for traffic prediction across multiple cities, which is composed of convolutional layers and LSTM. This model supports knowledge transfer from multiple cities, which simply applies meta learning framework (*i.e.*, MAML [5]) and views the traffic data collected from one source city as a meta-training task.
- **STrans-GAN [34]** STrans-GAN combines the transfer learning with generative adversarial networks for cross-city traffic estimation, which learns traffic patterns from multiple source cities through the traffic clustering and pre-training process, and transfers the knowledge to the target city in fine-tuning process. We apply convolutional layers to both generator and discriminator. The data from source cities is for pre-training, and only one data sample from the target is used for fine-tuning.

C. Evaluation Metrics

In our experiments, we use mean absolute percentage error (MAPE) and rooted mean square error (RMSE) to evaluate our model:

$$\begin{aligned} \text{MAPE} &= \frac{1}{n_s}\sum_{i=1}^{n_s} |y_i - \hat{y}_i| / y_i, \\ \text{RMSE} &= \sqrt{\frac{1}{n_s}\sum_{i=1}^{n_s} (y_i - \hat{y}_i)^2}, \end{aligned} \tag{10}$$

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

D. Experimental Settings

In the experiment, since we parametrize the auxiliary distribution $Q(\boldsymbol{m}|G(\boldsymbol{Z}, \boldsymbol{m}))$ as a neural network, its form depends on the true posterior $P(\boldsymbol{m}|G(\boldsymbol{Z}, \boldsymbol{m}))$. We found that simply treating $Q(\boldsymbol{m}|G(\boldsymbol{Z}, \boldsymbol{m}))$ as a factored Gaussian distribution is sufficient.

For all experiments, we use Adam [13] for online optimization and apply batch normalization [11] after convolutional layers. The learning rate is set to 2×10^{-5} and the batch size is 32. The detailed structure of Mest-GAN in our experiments is as follows: the generator G contains 2 convolutional layers with kernel sizes {3, 3} and output channels {256, 1}, and a multi-head self-attention layer combined with feed-forward network with layer normalization; the discriminator D also includes two convolutional layers with kernel sizes {3, 3} and output channels {256, 256}, and a multi-head self-attention layer combined with feed-forward network. The inference network Q has similar internal layers as the generator.

E. Results

1) Overall performance results: In this part, we have the average traffic estimation performance over all the time slots within a day (*i.e.*, from 7am to 6pm). When we perform the traffic speed estimation and taxi inflow estimation in Xi'An City, we produce 31 days traffic estimation sequences and compare with the ground-truth traffic from the testing set. For each specific time slot (*i.e.*, one hour), both RMSE and MAPE are calculated, and we average the 12 statistics for 12 time slots and get the average RMSE and MAPE. For each deep model, we trained and test the model for three times and picked the best trained model.

The detailed results are shown in Table III and Table IV. When we use Chengdu City as the target city and estimate the traffic speed and taxi inflow based on the local travel demands (as shown in Table III), we found the our Mest-GAN have the best average performance compared with other baseline methods. The standard traffic estimation methods including both Curb-GAN and cGAN+LSTM have the highest error, which means when these models are trained on the data provided by other source cities, they cannot successfully estimate the traffic in the target city, since the target city has never been seen by the model. Besides, the baseline models which apply the transfer learning and supervised meta-learning frameworks (including RegionTrans, MetaST and STrans-GAN) also have high error in traffic estimation in the target city, which indicates these model cannot work well when they are not fine-tuned with the data from the target city. By contrast, our Mest-GAN can automatically learn the distribution of spatial-temporal patterns using the travel demand data, once the target city cannot provide prior traffic data, we can successfully learn its traffic pattern with the local travel demand and thus accurately estimate the corresponding traffic. The traffic estimation performance in Xi'An City (shown in Table IV) have similar results.

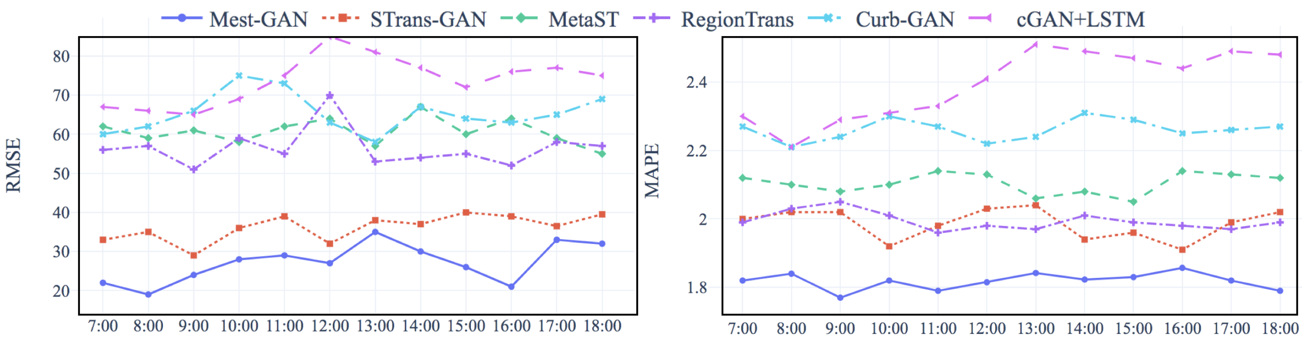
2) Traffic estimation performance in consecutive time slots: Beside the average performance, we also care about the traffic estimation performance in consecutive time slots within a day. When we use Chengdu City as the target city, the detailed traffic speed estimation performance is shown in Figure 6. We find in each specific time slot (*i.e.*, one hour), the standard traffic estimation model including the Curb-GAN and cGAN+LSTM cannot successfully capture the spatial and

TABLE III: Performance on task 1: traffic speed and taxi inflow estimation in Chengdu City.

Methods		cGAN+LSTM	Curb-GAN	RegionTrans	MetaST	STrans-GAN	**Mest-GAN**
Traffic speed	RMSE	72.22	75.21	59.99	63.96	61.23	**54.56**
	MAPE	3.64	3.41	2.68	2.96	2.78	**2.51**
Taxi inflow	RMSE	421.99	459.25	405.72	392.98	452.21	**382.65**
	MAPE	10.22	10.41	11.42	9.01	10.32	**8.67**

TABLE IV: Performance on task 2: traffic speed and taxi inflow estimation in Xi'An City.

Methods		cGAN+LSTM	Curb-GAN	RegionTrans	MetaST	STrans-GAN	**Mest-GAN**
Traffic speed	RMSE	74.21	57.72	55.34	62.75	36.62	**28.78**
	MAPE	2.44	2.27	1.99	2.12	1.98	**1.82**
Taxi inflow	RMSE	492.81	440.72	474.82	531.98	350.81	**312.51**
	MAPE	12.87	10.54	9.67	12.63	8.52	**7.33**



(a) RMSE of traffic speed estimation in consecutive time slots

(b) MAPE of traffic speed estimation in consecutive time slots

Fig. 5: Traffic speed estimation in Xi'An City in consecutive time slots.

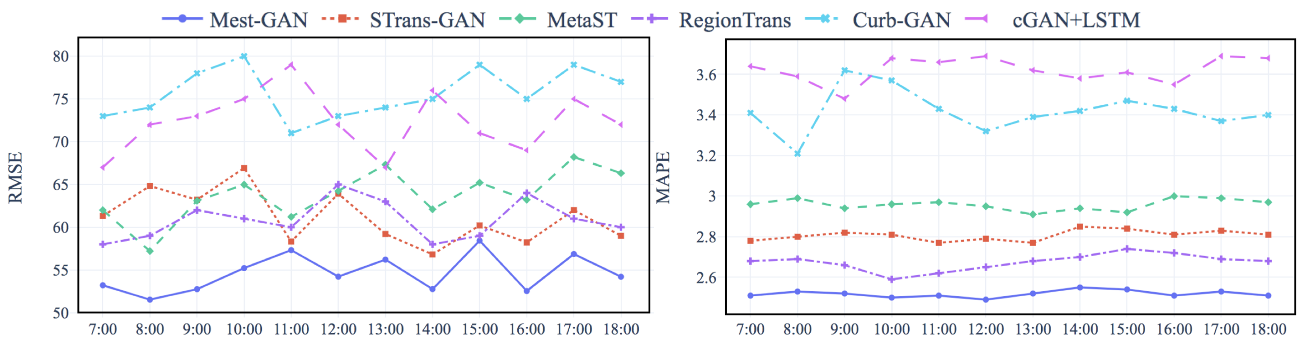
temporal traffic patterns in the target city without the access to the prior data, and they present high estimation errors. Moreover, the MetaST viewed each city as a specific task and apply MAML during meta-training, in the meta-testing process, the target city cannot provide any prior traffic data, and we directly feed the travel demand to the learned model which cannot accurately identify the traffic patterns for the target. The RegionTrans and STrans-GAN require some prior data from the target city, if we omit the fine-tuning process, both of the model would fail due to the lack of data. **Thus, Strans-GAN should be applied in the case where prior traffic data is provided for the fine-tuning process, Mest-GAN is able to estimate the traffic for the target city when no prior data is available.**

When we use Xi'An City as the target city, the detailed traffic speed estimation performance is shown in Figure 5. We find the Curb-GAN and cGAN+LSTM cannot be used to estimate the traffic in a city when they are trained on other cities. Besides, the existing meta-learning or transfer learning frameworks for traffic estimation including MetaST, RegionTrans and STrans-GAN are cannot learn the traffic patterns independently, they need pre-defined number of tasks (*e.g.*, traffic patterns), which usually lead to unstable performance. Our Mest-GAN, however, does not need any pre-defined number of patterns, it can learn the traffic patterns automatically and also capture the spatial and temporal dependencies of traffic, and thus provide stable estimation performance.

3) Evaluations on hyper-parameters: Since our Mest-GAN has different hyper-parameters including the dimension of random noise, the dimension of the latent code, the discount factor of the mutual information loss (*i.e.*, λ), batch size, *etc*, which would greatly influence the model performance. it is important to evaluate how these hyper-parameters would affect the traffic estimation results. For each the experiment in this part, we only adjust one hyper-parameter and keep the others the same.

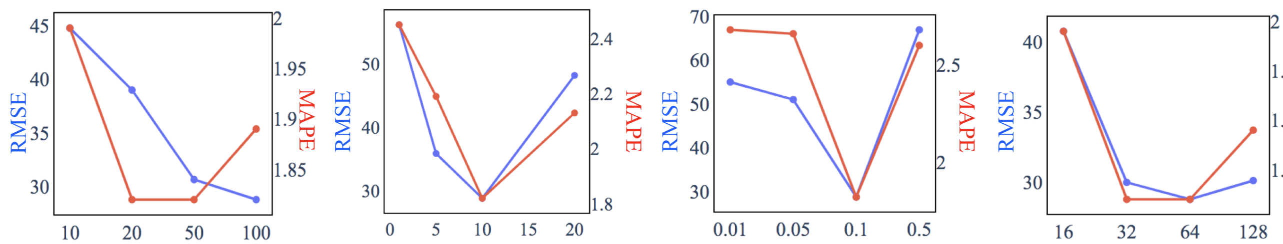
As shown in Figure 7(a), we fixed all other parameters of Mest-GAN and adjust the dimension of random noise, we find that the estimation results of our Mest-GAN is sensitive to the random noise dimension. If the noise dimension is very low, the estimation errors will be high, and similarly, if the noise dimension is very high, the estimation performance would also be affected. With low noise dimension, the weights within the generative adversarial network are not enough to learn the complex spatial and temporal dependencies of traffic. But if the noise dimension is too high, the model will contain too much randomness which make it hard to converge and learn meaningful patterns from the data.

Figure 7(b) shows the estimation performance with different dimensions of latent code λ, and we get high errors when the dimension of latent code is too high or too low. For example, if the dimension of latent code is one, which means all the information extracted from the time-series travel demands from multiple source cities should be included in only one dimensional latent code, apparently, the complex information cannot be learned in a good way, and some important information of the traffic patterns would be lost, which finally leads to poor performance. Moreover, if the dimension of the latent code is very high, the information of spatial-temporal traffic patterns learned from the travel demand sequence should be kept very well, however, it usually need huge amount of training data to finish the mapping from the demand sequences to the latent space. Thus, the dimension of the latent code cannot be too low or high. A proper dimension of the latent



(a) RMSE of traffic speed estimation in consecutive time slots (b) MAPE of traffic speed estimation in consecutive time slots

Fig. 6: Traffic speed estimation in Chengdu City in consecutive time slots.



(a) Random noise dimension (b) Latent code dimension (c) λ (d) Batch size

Fig. 7: Impact of parameters on traffic speed estimation in Xi'An City.

code is the key to the success of the traffic estimation in the target city.

Figure 7(c) is the estimation performance based on different λ in our final objective function 5. λ should be chosen based on the GAN loss scale, improper λ would affect the final generation performance, and in our experiments, the best choice of λ is 0.1. A large value of λ will lead to the situation where mutual information loss is dominant, and the discriminator cannot learn to distinguish the real data from the generated data, and the generator cannot learn to generate like-real traffic distribution. Furthermore, a small value of λ will make the mutual information loss too trivial to be considered during training, so the inference network cannot work very well. Thus, a proper λ is vital to the good performance.

Last but not the least, as shown in Figure 7(d), we evaluate the impact of the batch size. Large batch size results in bad estimation performance in our experiments, which also matches the conclusions in the work [12], [33] stating that there is a significant degradation in the generalization ability of the model is a large batch size is used in the model training.

V. Related Work

Urban traffic estimation. Traffic estimation is a vital problem in urban computing area, which helps reduce traffic congestion and provide insights for urban planning. In recent years, many works have focused on urban traffic estimation problem. For example, built upon the classic machine learning techniques, some works [14], [24] proposed novel frameworks to predict traffic volume, other works such as [4], [23] focused on predicting human mobility. However, these works ignore the complex urban conditions and cannot estimate the traffic or mobility in an geographical region. Besides, many other works tried to borrow deep learning techniques to deal with complex spatial-temporal dependencies in traffic estimation. For example, convolutional neural networks are proved to be successful in capturing traffic spatial dependencies [29], [31]. Recurrent neural networks [3] and LSTM [10] significantly improved the accuracy in traffic prediction by capturing traffic temporal dependencies very well [26]. Moreover, the convolutional neural networks and LSTM were perfectly combined in spatial-temporal prediction problems [22], [36]. However, all these works require a large amount of data to guarantee model convergence and decent performance, once facing the data scarcity scenario, all the works would fail.

Meta-Learning. Meta learning tries to learn a generalized model from training tasks which can be fast adapted into new related tasks with a few samples. The state-of-the-art meta-learning methods including MAML [5], Reptile [20], SNAIL [17], MOCA [7], *etc*. Meta learning has been applied to many areas including supervised/unsupervised learning, imitation learning and urban computing. For example, when solving the traffic estimation problem in the data scarcity scenario, MetaST [28] is proposed which is based on the standard MAMAL algorithm, and views each city as a specific task. STrans-GAN combines Reptile, transfer learning and GAN together to solve the cross-city traffic estimation problem. However, these works pre-defined the number of tasks or traffic patterns in meta-training, which usually leads to a lot of bias. And they still require some traffic data in the target city, which make them fail when the target city cannot provide any prior data.

VI. Conclusion

In this paper, we aim to solve the conditional urban traffic estimation problem in case of data scarcity (*i.e.*, the target city cannot provide any prior data) and tackle the main challenges including (1) Knowledge learning from the source and (2) knowledge adaptation without prior traffic data. We formulate the cross-city conditional urban traffic estimation problem as an unsupervised meta-learning problem and we solve this problem from generative adversarial meta-learning perspective. A novel model — Meta Spatial-Temporal Generative Adversarial Network (Mest-GAN) is proposed, which can successfully estimate the traffic in target city in consecutive time slots. In the meta testing process, we use the travel demand data in the target city for adaptation, where the inference network infers a latent code guiding the generator to produce accurate traffic estimations, and thus the prior traffic data in the target city is not required. Extensive experiments on real-world multiple-city datasets demonstrate that our Mest-GAN produces promising performance in both traffic speed and taxi inflow estimations.

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] Mest-GAN. https://www.dropbox.com/sh/suswe3x16arpemq/AADLWsvg7BFNVQEqph8kcCXXa?dl=0, 2022. [Online].

[2] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, 2016.

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

[4] 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.

[5] 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.

[6] 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.

[7] J. Harrison, A. Sharma, C. Finn, and M. Pavone. Continuous meta-learning without tasks, 2019.

[8] 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.

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

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

[11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[12] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, 2016.

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

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

[15] 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.

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

[17] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

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

[19] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

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

[21] B. Poole, S. Ozair, A. van den Oord, A. A. Alemi, and G. Tucker. On variational bounds of mutual information, 2019.

[22] 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.

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

[24] 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.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, 2017.

[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, 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.

[28] 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.

[29] 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.

[30] L. Yu, T. Yu, C. Finn, and S. Ermon. Meta-inverse reinforcement learning with probabilistic context variables. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

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

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

[33] 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.

[34] Y. Zhang, Y. Li, X. Zhou, X. Kong, and J. Luo. Strans-gan: Spatially-transferable generative adversarial networks for urban traffic estimation. 2022.

[35] 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.

[36] 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.