This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3014511, IEEE Transactions on Big Data

Off-Deployment Traffic Estimation — A Traffic Generative Adversarial Networks Approach

Yingxue Zhang, Yanhua Li, Xun Zhou, Xiangnan Kong and Jun Luo

Abstract—The rapid progress of urbanization has expedited the process of urban planning, *e.g.*, new residential, commercial areas, which in turn boosts the local travel demand. We propose a novel "off-deployment traffic estimation problem", namely, to foresee the traffic condition changes of a region prior to the deployment of a construction plan. This problem is important to city planners to evaluate and develop urban deployment plans. However, this task is challenging. Traditional traffic estimation approaches lack the ability to solve this problem, since no data about the impact can be collected before the deployment and old data fails to capture the traffic pattern changes. In this paper, we define the off-deployment traffic estimation problem as a traffic generation problem, and develop a novel deep generative model TrafficGAN that captures the shared patterns across spatial regions of how traffic conditions evolve according to travel demand changes and underlying road network structures. In particular, TrafficGAN captures the road network structures through a dynamic filter in the dynamic convolutional layer. We evaluate our TrafficGAN using a large-scale traffic data collected from Shenzhen, China. Results show that TrafficGAN can more accurately estimate the traffic conditions compared with all baselines. We also showcase that TrafficGAN can identify potential traffic issues in some regions and suggest possible reasons.

Index Terms—Traffic estimation, TrafficGAN, generative model.

1 Introduction

Over the past a few decades, we have witnessed drastic urbanization at the global scale. It is reported that the world's urban population ratio has reached 54% in 2014, and it is projected that by 2050, two-thirds of the world population will live in urban areas [1].

With the rapid progress of urbanization, urban planning is becoming a vital problem concerning with resources allocation, urban transportation efficiency and living environment. The fast development of new residential and commercial areas always comes with population growth, which in turn increases the travel demands and the risk of worsening traffic conditions due to the overload of the transportation infrastructures. For example, the Olympic Village was built in the northern area of Beijing for the 2008 Olympic Games with many new residential and commercial areas constructed in its nearby areas as illustrated in Fig. 1. The population in that region increased drastically after 2008, which significantly worsened the local traffic conditions [2]. This could have been avoided if more thorough and accurate traffic evaluation had been done before the constructions. Therefore, it is crucial to foresee both positive and negative impacts on traffic conditions before an urban construction plan is deployed. In our work, we refer to such a problem as "*off-deployment traffic estimation*" problem.

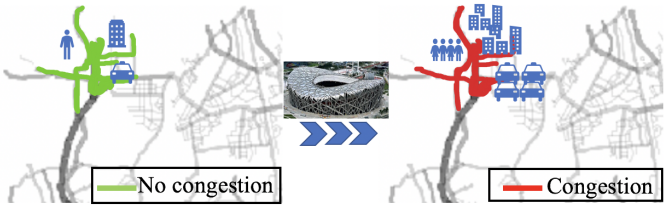


Fig. 1: Traffic condition changes around Olympic Village in Beijing, China

Solving this problem is technically challenging, since no new data can be collected before deployment in an area, while old data collected before deployment fails to capture the traffic pattern changes.

The traffic estimation problem has been extensively studied in the literature [3]–[7]. These works use the historical traffic data of regions to build machine learning models that capture the correlations among the past traffic, environmental features and the future traffic. However, when predicting the traffic impact of a newly developed construction plan, these models will fail because they cannot capture the future traffic pattern changes caused by the new deployment plan due to the lack of training samples. Traditionally in civil engineering, agent-based simulation models [8] or physical models [9] are used to estimate the projected traffic volume after constructions. However, these models rely heavily on model choice and parameter settings, which are not transferable across urban regions.

In this paper, we propose a novel traffic generative adversarial network (TrafficGAN[1]) to tackle the off-deployment traffic estimation problem. The proposed TrafficGAN can capture the traffic correlations along the underlying road networks, and estimate traffic conditions prior to

- *Yingxue Zhang and Yanhua Li are with the Department of Data Science, Worcester Polytechnic Institute, Worcester, MA, 01609. E-mail: yzhang31@wpi.edu, yli15@wpi.edu*
- *Xun Zhou is with Department of Business Analytics, University of Iowa, Iowa City, IA, 52242. E-mail: xun-zhou@uiowa.edu*
- *Xiangnan Kong is with the Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA, 01609. E-mail: xkong@wpi.edu*
- *Jun Luo is with Machine Intelligence Center, Lenovo Group Limited, Hong Kong. E-mail: jluo1@lenovo.com*

1. A preliminary version of the results in this paper appeared in [10].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3014511, IEEE Transactions on Big Data

deployment of a construction plan. Our **main contributions** are summarized as follows:

• We model the off-deployment traffic estimation problem as a traffic data generation problem, and propose a novel deep generative model – TrafficGAN, which captures the shared patterns across spatial regions of how traffic conditions evolve according to travel demand changes and underlying road network structures. (See Sec 4.)

• We evaluate TrafficGAN using a large scale traffic data collected during 7/2016-12/2016 from Shenzhen, China. The unique dataset represents a wide range of regions with diverse travel demands and traffic conditions in both rural and urban areas. Our results demonstrate that our proposed TrafficGAN can accurately estimate the traffic conditions compared with all baselines. When applying TrafficGAN to a number of representative regions with higher (than their current) travel demands, we showcase the issues of those regions (in case of new construction plan deployed) and identify potential reasons of the issues. (See Sec 6.)

• Compared with the preliminary version of this work in [10], we have (i) added a threshold to control the traffic correlation matrix and provided more detailed analysis on how the threshold of traffic correlation influenced the size and shape of the dynamic filter (in Section 4.3 and 4.4.1); (ii) provided the preliminaries of state-of-the-art generative models (in Section 4.2); (iii) detailed how the results obtained from TrafficGAN can be utilized by urban planner to make decisions (in Section 5); (iv) added more metrics and presented more evaluation results on threshold selection, statistics comparisons and traffic condition visualizations, and looked into real traffic condition evaluation cases (in Section 6.4 and 6.5); (v) added comprehensive related work (in Section 7).

This paper is organized as follows. Section 2 formally defines the problem, outlines the solution framework for off-deployment traffic estimation. Section 3 – 5 detail the three key stages of the solution. Section 6 presents our experimental results. Related works are discussed in Section 7 and we conclude our paper in Section 8.

2 Overview

In this section, we define the off-deployment traffic estimation problem, describe the datasets we use, and outline our solution framework.

2.1 Problem Definition

Urban planning, especially, governmental zoning[2], is a process of planning land use and development in a target region, in which certain land uses (*e.g.*, residential, commercial) are permitted or prohibited [11]. In this work, we focus on urban deployment and zoning plans when developing certain new residential or commercial areas in a target region, which will potentially promote the population size and influence the travel demand in the region. Denote a city under planning as R_0, *e.g.*, Shenzhen City in China bounded by $[22.534°, 22.87°]$ in latitude and $[113.77°, 114.40°]$ in longitude. As defined below, we partition R_0 into *grid cells* (as the smallest granular region to characterize the traffic status) and form *target regions* R's as collections of grid cells.

2. https://en.wikipedia.org/wiki/Zoning

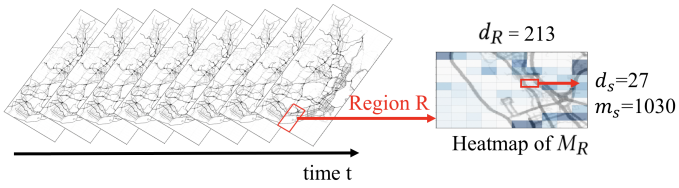


Fig. 2: Travel demand and traffic distribution of region R

Definition 1 (Grid cell s). The planning city R_0 is partitioned into N_0 grid cells with equal side-length in latitude and longitude, denoted as $\mathcal{S} = \{s_i\}$, where $1 \leq i \leq N_0, i \in \mathbb{N}$.

Definition 2 (Target region R). A target region R is a square geographic region in R_0, formed by $\ell \times \ell$ grid cells. Formally, $R = \langle s, \ell \rangle$ is uniquely defined by an anchor grid cell s on its top-left corner and a number ℓ of grid cells on the side[3].

In our study, grid cells are the minimum units where traffic status and travel demands are measured. Alternatively, urban planning and traffic estimation will be performed at a target region. As a result, a region is analogous to an "image", where grid cells are viewed as "pixels". Fig. 5a visualizes grid cells of Shenzhen, China and Fig. 5b illustrates the target region examples with $\ell = 10$.

Definition 3 (Travel demand of a grid cell and a target region). The travel demand of a geographic area captures the total number of departures in a period of time, *e.g.*, one hour interval. Thus, we denote the travel demand of a grid cell s as $d_s \in \mathbb{N}$. Given a target region R, $\boldsymbol{D}_R$ is an $\ell \times \ell$ matrix representing the travel demand distribution of all grid cells in R. Moreover, we denote the *total travel demand* of a target region R as $d_R \in \mathbb{N}$, which is the sum of travel demands in all grid cells within R, *i.e.*, $d_R = \sum_{s \in R} d_s = \sum_{1 \leq i,j \leq l} \boldsymbol{D}_R(i,j)$.

In general, it is hard to obtain the total travel demand in a region including all transport modes. In this work, we use the demand for taxis to represent the regional travel demand, where many studies have shown that taxi demands represent the total demands quite well [12], [13].

Definition 4 (Traffic status of a grid cell and traffic distribution of a target region). Traffic status includes various measures representing the quality of traffic in a geographic region, such as average driving speed, traffic inflow/outflow, traffic volume, *etc*. Taking traffic inflow as an example, we denote m_s as the traffic inflow of grid cell s in a period of time. Similar, given a target region R with $\ell \times \ell$ grid cells, we denote an $\ell \times \ell$ matrix $\boldsymbol{M}_R$ as the traffic distribution in R.

Each element of $\boldsymbol{M}_R$ represents the taxi inflow in a grid cell. As shown in Fig. 2, the whole matrix $\boldsymbol{M}_R$ can be viewed and visualized as an "image" characterizing the underlying traffic distribution of a target region R, where each pixel represents a grid (*e.g.*, gird s, the small red box in the map).

Definition 5 (Urban deployment plan). An urban deployment plan in a target region R is referred to a plan to construct new residential or commercial areas in the region R without changing the road structures. As a part of the plan,

3. Note that target regions can also be defined as rectangles rather than squares. For simplicity, we use square shape of target regions in this work.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3014511, IEEE Transactions on Big Data

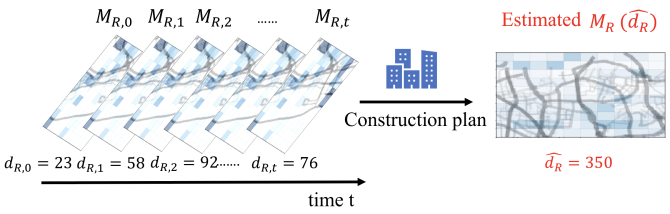


Fig. 3: Off-deployment traffic estimation problem

the expected travel demand after deployment is specified[4], denoted by $\hat{d}_R$.

Clearly, an urban deployment plan will lead to an updated regional travel demand $\hat{d}_R$, which in turn would significantly affect the regional traffic distribution $\boldsymbol{M}_R$. The goal of our work is to estimate $\boldsymbol{M}_R(\hat{d}_R)$, which reflects the potential traffic burden to be introduced by a deployment plan and can be used by the planning authorities to evaluate the pros and cons of urban deployment plans. The problem is formally defined as below.

Problem definition. Given a city area R_0 partitioned into grid cells $\mathcal{S}$, the citywide historical travel demands and traffic distributions $\boldsymbol{D}_{R_0,t}$ and $\boldsymbol{M}_{R_0,t}$ are available over a time span $1 \leq t \leq T$. For a target region $R = \langle s, \ell \rangle$ and a deployment plan in R with the expected travel demand $\hat{d}_R$, we aim to estimate the traffic distribution $\boldsymbol{M}_R(\hat{d}_R)$.

Fig. 3 illustrates an example of the problem. The series of matrices on the left are the historical traffic distributions and travel demands for each time slot. The map on the right is the estimated traffic distribution $\boldsymbol{M}_R(\hat{d}_R)$ based on an expected travel demand $\hat{d}_R$=350 of the proposed plan.

2.2 Data Description

In effect, all kinds of personal vehicle data can be used, especially the GPS trajectories and other spatial-temporal data records. In this paper, we use two real-world traffic datasets, (1) taxi GPS data; (2) road map data. For consistency, both of the datasets are collected from the same time interval, *i.e.*, from Jul 1st to Dec 31st, 2016 in Shenzhen, China.

Taxi GPS data contains GPS records collected from taxis in Shenzhen, China from Jul 1st to Dec 31st, 2016. There are 17,877 taxis equipped with GPS sensors, each GPS sensor generates a GPS record every 40 seconds on average. Overall, a total number of 51,485,760 GPS records are collected each day, each record contains five key data fields including taxi ID, time stamp, passenger indicator, latitude and longitude. The passenger indicator field is a binary value indicating whether a passenger is on board.

Road map. In our study, we use the Google GeoCoding[5] 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[6] is shown in Fig. 5.

4. The expected travel demand $\hat{d}_R$ after deploying a construction plan is assumed given in this paper, which can be done by commonly used Four-Steps demand forecasting approaches in Civil Engineering [14].

5. https://developers.google.com/maps/documentation/geocoding/

6. http://www.openstreetmap.org/

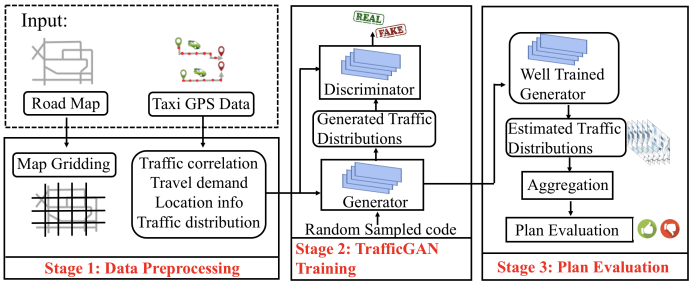


Fig. 4: Solution framework

2.3 Solution Framework

Fig. 4 outlines our off-deployment traffic estimation framework, which takes taxi GPS data and road map data as inputs, processes the data in three stages to get the output:

- **Stage 1 (Data Preprocessing):** In this stage, we partition the road map into small grid cells and calculate the travel demands and traffic status (*i.e.*, taxi inflow) of all grid cells and time intervals.
- **Stage 2 (TrafficGAN Training):** In this stage, we train TrafficGAN, a novel generative model for traffic estimation. TrafficGAN automatically captures the shared patterns across spatial regions of how traffic distributions evolve according to travel demand changes and underlying road network structures.
- **Stage 3 (Urban Plan Evaluation):** In this stage, the generator obtained from Stage 2 will be used to estimate the future traffic distribution $\boldsymbol{M}_R(\hat{d}_R)$ for a target region R, given a deployment plan with the expected travel demand $\hat{d}_R$. Depending on traffic distribution requirement defined by the urban planners/authorities, the deployment plan is accepted or rejected.

3 Stage 1: Data Preprocessing

3.1 Map Gridding

For the ease of implementation in practice, we adopt the grid based method, which simply partitions the map into equal side-length grids [15], [16]. The grid based method has the advantage of allowing us to adjust the side-length of grids, which helps to examine and understand impacts of the grid size. In this paper, we divide the map of Shenzhen City into 40×50 grid cells with a side-length $l_1 = 0.0084°$ in latitude and $l_2 = 0.0126°$ in longitude.

3.2 Training Sets Construction

Given all 40×50 grid cells in Shenzhen, we choose target region size $\ell = 10$ as an example in this study, where our TrafficGAN actually applies to any target region size. Thus, there are in total $1,271$ possible target regions with size 10×10. As shown in Fig. 5b, the upper-left red box is the first region in Shenzhen, to get new regions, we can slide it horizontally for p grid cells, $0 \leq p \leq 40$, and/or vertically for q grid cells, $0 \leq q \leq 30, p, q \in \mathbb{N}$. The location of each region is described with a tuple (i, j) which indicates the coordinates of the first grid cell (the upper-left one) in the region, *i.e.*, the row and column index $0 \leq i \leq 30, 0 \leq j \leq 40, i, j \in \mathbb{N}$. However, it is unnecessary and too costly to use data from all $1,271$ regions as training

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3014511, IEEE Transactions on Big Data

(a) Shenzhen road map gridding (b) Regions illustration

Fig. 5: Map gridding and regions illustration

data. Instead, we set $p = q = 5$ and get 63 regions covering entire Shenzhen city as target regions, extract their traffic distributions and travel demands over time.

Travel Demand. We use six months taxi GPS records of Shenzhen, China in 2016 to extract the travel demand of each grid cell and region in Shenzhen. In each time slot, *i.e.*, one hour, we count the total pickup events within each grid cell and each $\ell \times \ell$ region. Since in each GPS record, the passenger indicator value indicates whether a passenger is on board, which can be easily used to monitor the pickup information. In addition, for other personal vehicle trajectory data, if the passenger indicator is not available, alternative approaches can be employed to detect the demand (start of a trip), for example, if the car start moving (in GPS record), we can safely claim it is a starting location and time of a demand.

Traffic Distribution. Traffic distributions reflect the traffic status in a region, which can be quantified with many measures such as traffic speed, volume, inflow/outflow. Taking traffic inflow as an example, it is a crucial metric capturing the amount of arrivals in each grid cell. Since it is hard to obtain the total traffic inflow in a grid cell including all transport modes, in this paper, we use taxi inflow to represent the traffic inflow In each time slot of each day, we count all taxis which stay or arrive at each grid cell as the taxi inflow. Since taxi may not be representative to all vehicles, taxi data may introduce bias to the traffic status estimation, the limitations of taxi data can be easily tackled by [17] and [18] as complementary techniques to our work.

4 Stage 2: TrafficGAN Training

Taking an analogy, our "off-deployment traffic estimation" problem is similar as image generation problem, where the traffic distribution of a region can be viewed as a gray-scale "image", where the traffic status (*e.g.*, inflow) of each grid can be viewed as a "pixel" value. Thus, image generation approaches, such as GANs [19], sound a promising solution. However, the unique challenges of our problem prevent the state-of-the-art GAN models from solving it. In this section, we highlight the technical challenges of our problem, summarize the state-of-the-art generative models, and introduce our TrafficGAN for off-deployment traffic estimation problem.

4.1 Challenges

To solve the off-deployment traffic estimation problem, we aim to generate the traffic distributions with respect to various travel demands and the road network structures in the target region, which is a challenging task for the following reasons:

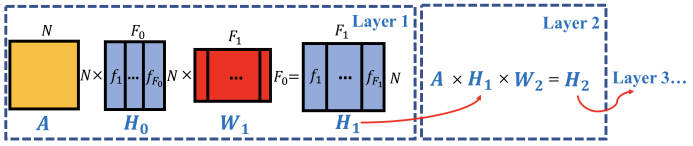


Fig. 6: Propagation rule of dynamic convolutional layer, f refers to traffic features

• **Traffic correlations along road networks.** In a target region R, the traffic of neighboring grids along the underlying road networks has strong correlations. Capturing such correlations is non-trivial since the correlation patterns are defined by the road network structures, which may have irregular shapes (rather than squares or rectangles).

• **Conditioned Traffic Distribution Generation.** The generated traffic distribution is meaningful only when conditioned on the given region R and the travel demand d_R. However, how to design a generative model that outputs the traffic distributions for a desired region and travel demand is challenging.

4.2 Preliminaries

Generative adversarial networks (a.k.a. GANs) [19] have been widely employed to many applications, including image, text generation, domain adaptation, *etc*. GAN includes a generator which generates a new data instance with input as a random code in a low dimensional space, and a discriminator which evaluates input data instances for authenticity. **Conditional GANs with deep convolutional layers (cDCGANs)** [20] are composed of multiple convolutional layers in both generator and discriminator to obtain better generation quality for primarily image data, and introduce a condition as input in both generator and discriminator to guarantee not only the generated data is close to the real, but also matches the input condition. cDCGAN seems a feasible method to solve the off-deployment traffic estimation problem since it can control the outputs by conditions and the convolutional layers can capture local patterns with filters. However, it is still hard to capture the traffic correlations along road networks accurately due to filters' fixed size and shape.

Below, we introduce a measure of traffic correlation across grid cells and develop TrafficGAN. As a generative model, TrafficGAN integrates the traffic correlations for traffic generations.

4.3 Quantifying Traffic Correlation

We introduce traffic correlation to capture the inherent traffic dependence between a grid cell pair. For each grid cell, there is time series traffic data (taxi inflow) over the entire study time period. We calculate the Pearson correlation coefficient between time series of a grid cell pair to quantify their traffic correlation. *Pearson correlation coefficient* [21] measures the linear correlation between variables X and Y, where X and Y are time series taxi inflow data of two grid cells in our case. The Pearson correlation coefficient a can be calculated by the formula below, where $\overline{X}$ and $\overline{Y}$ are the mean of X and Y:

$$a_{XY} = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2} \sqrt{\sum_{i=1}^{n} \left(Y_i - \overline{Y}\right)^2}}, \quad (1)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3014511, IEEE Transactions on Big Data

(a) Filter in standard convolutional layer (b) Dynamic filter with threshold λ_1 (c) Dynamic filter with threshold λ_2

Fig. 7: Convolutional filter comparison, $\lambda_1 > \lambda_2$

where $a \in [-1, 1]$. If $a \in [-1, 0)$, the two grid cells are negatively correlated, $a = 0$ means two grid cells don't have any linear correlation, $a \in (0, 1]$ indicates two cells are positively correlated. For an $\ell \times \ell$ region R, its corresponding traffic correlation matrix $\boldsymbol{A}$ is a symmetric $\ell^2 \times \ell^2$ matrix, where the entry a_{ij} is the traffic correlation between grid cell s_i and s_j, $s_i, s_j \in R$.

In a road network, nearby road segments (resp. nearby grid cells) often have stronger correlations in traffic according to the First Law of Geography [22]. In fact, the effective traffic correlations are generally positive, since if a road segment (resp. a grid cell) has traffic congestion, the road segments adjacent to it (resp. nearby grid cells) are likely to have high traffic volumes, the similar trend indicates positive correlations. However, the traffic correlations between distant road segments (resp. distant grid cells) are weak due to the lack of direct traffic connections. In our work, for a specific grid cell, we only consider its nearby grid cells which are directly connected with it by roads and thus (likely) to have positive traffic correlations. To remove other uncorrelated grids, we set a threshold $\lambda \in (0, 1)$ such that we set $a_{ij} = 0$, if $a_{ij} < \lambda$. After removing the uncorrelated grids by the threshold λ, we perform row normalization for the traffic correlation matrix $\boldsymbol{A}$ as Eq. 2, so it will not affect the scale of features when multiplied to the feature matrix in Eq. 3.

$$a_{ij} = \frac{a_{ij}}{\sum_{j=1}^{\ell^2} a_{ij}}. \tag{2}$$

Next, we will elaborate on how to integrate the traffic correlation matrix for traffic distributions estimation and generation.

4.4 TrafficGAN

In this paper, to solve the challenges mentioned above, we propose a novel conditional generative model – TrafficGAN which can capture the traffic correlations of road networks, control the generation results with desired region and travel demand conditions, and generate realistic traffic distributions. TrafficGAN consists of a generator G and a discriminator D, and it applies dynamic convolutional layers in G and D.

4.4.1 Dynamic Convolutional Layer

The goal of dynamic convolutional layer is to learn a function of traffic features in a region including traffic inflow, volume, speed, *etc*. The input of dynamic convolutional layer includes two parts:

• A traffic feature matrix $\boldsymbol{H}$ of size $N \times F_0$ (N: number of grid cells in a region, $N = \ell \times \ell$; F_0: initial number of traffic features).

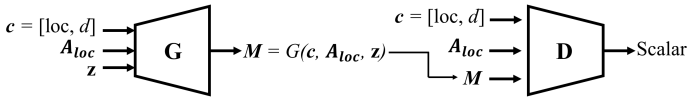


Fig. 8: TrafficGAN

• A non-negative and row-normalized traffic correlation matrix $\boldsymbol{A}$ of size $N \times N$.

The output is a new feature matrix after one-layer convolution. The layer-wise propagation rule is:

$$\boldsymbol{H}_{i+1} = f(\boldsymbol{H}_i, \boldsymbol{A}) = \sigma(\boldsymbol{A}\boldsymbol{H}_i\boldsymbol{W}_{i+1}), \tag{3}$$

where $\boldsymbol{H}_i$ is the feature matrix of a region got after ith layer and is the input of the $(i+1)$th layer, $\boldsymbol{W}_{i+1}$ is the weight matrix in $(i+1)$th layer and σ is an activation function. The rule is illustrated in Fig. 6.

Dynamic Filter. In the propagation, since the traffic correlation matrix is multiplied by the traffic feature matrix, for each grid cell, the new features after one-layer propagation is the weighted sum of all grid cells features within the corresponding region, and we can treat the correlations between the current grid cell and any other grid cells (*i.e.*, the corresponding row in correlation matrix) as a filter, whose shape and size are irregular. Hence, we say such filters are dynamic since the the filter of each grid cell would be different and changeable.

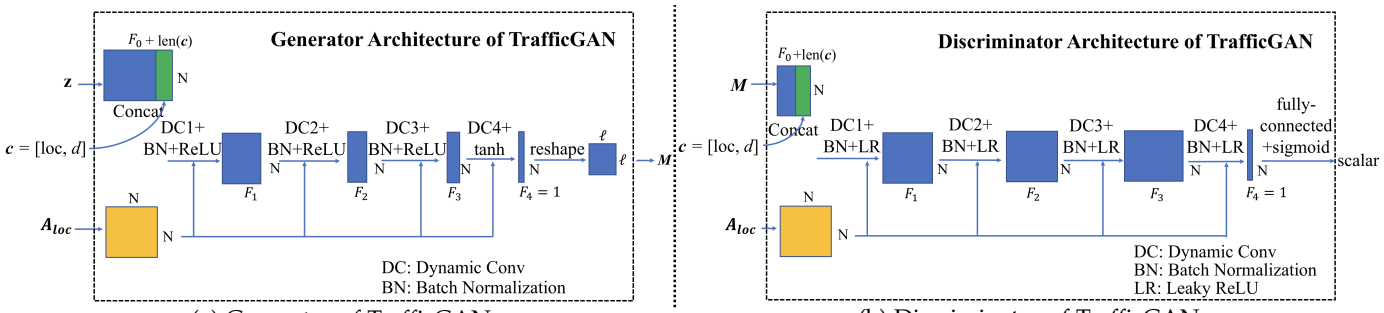
Dynamic Convolutional Layer *vs*. Standard Convolutional Layer. Fig. 7 illustrates the difference between a standard convolutional layer and a dynamic convolutional layer (with different threshold λ). By introducing the traffic correlation matrix $\boldsymbol{A}$ in dynamic convolutional layer, a dynamic filter is created and applied to the feature matrix $\boldsymbol{H}$, where the size and the shape of the dynamic filter is controlled by $\boldsymbol{A}$ and the threshold λ, when λ changes, the dynamic filter for the same grid cell could be different. The filters are marked in yellow, and the blue block represents the target grid. The filter of a standard convolutional layer (Fig. 7(a) has fixed size, *e.g.*, a 3×3 square, which cannot naturally captures the traffic correlations along the road networks and would include some grids with no roads or some girds having low traffic correlations. In contrast, the dynamic filters created by the traffic correlation matrix (in Fig. 7(b)-7(c)) align with the road network very well. Moreover, comparing Fig. 7(b) and Fig. 7(c), it is clear that a smaller threshold λ leads to a larger range of dynamic filter, and vice versa.

Besides the dynamic filter, another filter $\boldsymbol{W}$ in Eq. 3 performs convolution on traffic features in each grid cell. Moreover, the corresponding dynamic de-convolutional layer is the same structure as the dynamic convolutional layer as shown in Fig 6. This is because the matrix operation of the dynamic convolutional layer and dynamic de-convolutional layer is invariant. We omit the detailed proof for brevity.

4.4.2 TrafficGAN Architecture

To tackle the challenge of conditioned traffic distribution generation, we introduce conditional generative model structure in designing TrafficGAN. Fig. 8 shows the overall structure of TrafficGAN. The goal of the generator G is to generate traffic distributions with respect to the region location loc and travel demand d. The input of the generator

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3014511, IEEE Transactions on Big Data



(a) Generator of TrafficGAN (b) Discriminator of TrafficGAN

Fig. 9: TrafficGAN architecture

G includes three parts, i) a low-dimensional code vector $\boldsymbol{z}$, randomly sampled from Gaussian distribution, ii) condition vector $\boldsymbol{c} = [\text{loc}, d]$, defining the desired region and travel demand and iii) a traffic correlation matrix $\boldsymbol{A}_{\text{loc}}$. The discriminator D takes three inputs, i) a traffic distribution $\boldsymbol{M}$, ii) condition information $\boldsymbol{c} = [\text{loc}, d]$ and iii) a traffic correlation matrix $\boldsymbol{A}_{\text{loc}}$. D outputs a scalar indicating whether the traffic distribution $\boldsymbol{M}$ is real and whether the input $\boldsymbol{M}$ and $\boldsymbol{c}$ are matched. The detailed structures of generator G and discriminator D are detailed in Fig. 9a and Fig. 9b, len($\boldsymbol{c}$) represents the number of conditions in $\boldsymbol{c}$. In generator, $\boldsymbol{c}$ is concatenated into $\boldsymbol{z}$ such that the generator is conditioned by $\boldsymbol{c}$, which means the generator G builds the mapping from distribution $p_z(\boldsymbol{z})$ to a traffic distribution $G(\boldsymbol{c}, \boldsymbol{A}_{\text{loc}}, \boldsymbol{z})$. In our case, $N = 100$, $F_0 = 1$ since the only traffic feature is taxi inflow.

4.4.3 TrafficGAN Loss Function

In TrafficGAN, the generator G aims to generate "like-real" traffic distributions so that the discriminator D cannot distinguish the generated traffic distributions from the real traffic distributions well. For the discriminator D, it aims to rise the score of real traffic distributions, lower down the score of generated traffic distributions, and lower down the score of mismatched pairs of traffic distributions and conditions. As a result, the loss function of TrafficGAN is in the form of Eq. 4, modeled as a MinMax game. (See more details in [23].)

$$\min_G \max_D V(D,G) = E_{\boldsymbol{M}\sim p_{data}(\boldsymbol{M})}[\log D(\boldsymbol{c}, \boldsymbol{A}_{\text{loc}}, \boldsymbol{M})] + E_{z\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{c}, \boldsymbol{A}_{\text{loc}}, \boldsymbol{z})))]. \quad (4)$$

4.4.4 Training Process

During the training process, we apply batch gradient descent. The detailed training process is shown in Algorithm 1, where the discriminator D and the generator G are updated in line 3 – 7 and line 8, respectively. In each training iteration, we update the parameters $\boldsymbol{\theta}_D$ of D with Eq. 5 and Eq. 6, where η_D is the learning rate.

$$\tilde{V}_D = \frac{1}{m}\sum_{i=1}^{m}\Big(\log(1 - D(\boldsymbol{c}^i, \boldsymbol{A}^i_{\text{loc}}, \tilde{\boldsymbol{M}}^i)) + \log D(\boldsymbol{c}^i, \boldsymbol{A}^i_{\text{loc}}, \boldsymbol{M}^i) + \log(1 - D(\boldsymbol{c}^i, \boldsymbol{A}^i_{\text{loc}}, \hat{\boldsymbol{M}}^i))\Big), \quad (5)$$

$$\boldsymbol{\theta}_D = \boldsymbol{\theta}_D + \eta_D \triangledown \tilde{V}_{\boldsymbol{\theta}_D}(\boldsymbol{\theta}_D). \quad (6)$$

Algorithm 1 TrafficGAN Training Process

Input: Training iteration K, a training set $\mathcal{Z} = \{(\boldsymbol{c}^1, \boldsymbol{A}^1_{\text{loc}}, \boldsymbol{M}^1), \cdots, (\boldsymbol{c}^n, \boldsymbol{A}^n_{\text{loc}}, \boldsymbol{M}^n)\}$, initialized G and D.

Output: Well trained G and D.

1: In each training iteration $iter$:
2: **repeat**
3: Sample $\mathcal{Z}_0 = \{(\boldsymbol{c}^1, \boldsymbol{A}^1_{\text{loc}}, \boldsymbol{M}^1), \cdots, (\boldsymbol{c}^m, \boldsymbol{A}^m_{\text{loc}}, \boldsymbol{M}^m)\}$ from training set $\mathcal{Z}$, where $m < n$.
4: Sample $\mathcal{N} = \{\boldsymbol{z}^1, \boldsymbol{z}^2, \cdots, \boldsymbol{z}^m\}$ from Gaussian distribution.
5: Generate $\tilde{\mathcal{T}} = \{\tilde{\boldsymbol{M}}^1, \cdots, \tilde{\boldsymbol{M}}^m\}$ with G, where $\tilde{\boldsymbol{M}}^i = G(\boldsymbol{c}^i, \boldsymbol{A}^i_{\text{loc}}, \boldsymbol{z}^i)$.
6: Sample $\hat{\mathcal{T}} = \{\hat{\boldsymbol{M}}^1, \hat{\boldsymbol{M}}^2, \cdots, \hat{\boldsymbol{M}}^m\}$ from training set $\mathcal{Z}$, where each $\hat{\boldsymbol{M}}^i$ is mismatched with $(\boldsymbol{c}^i, \boldsymbol{A}^i_{\text{loc}})$.
7: Update D with Eq. 6 to maximize Eq. 5.
8: Update G with Eq. 8 to maximize Eq. 7.
9: **until** $iter > K$.

Then, we update the parameters $\boldsymbol{\theta}_G$ of G with Eq.7 and Eq.8, where η_G is the learning rate.

$$\tilde{V}_G = \frac{1}{m}\sum_{i=1}^{m}\log D(G(\boldsymbol{c}^i, \boldsymbol{A}^i_{\text{loc}}, \boldsymbol{z}^i)), \quad (7)$$

$$\boldsymbol{\theta}_G = \boldsymbol{\theta}_G + \eta_G \triangledown \tilde{V}_{\boldsymbol{\theta}_G}(\boldsymbol{\theta}_G). \quad (8)$$

5 Stage 3: Urban Plan Evaluation

The generator G obtained from Stage 2 can be used by urban planners to evaluate urban construction plans at various locations, and search for more appropriate plans. To do so, given an urban deployment plan, the generator G takes (i) the expected travel demand $\hat{d}_R$, (ii) the location of the target region R, (iii) traffic correlation matrix of R, and (iv) random code vector $\boldsymbol{z}$, as inputs to generate traffic distributions for the plan to be evaluated.

Note that future traffic distributions hinge on many factors such as weather, *etc*. To capture the entire distribution of what the future traffic will look like over all potential (hidden) factors, we randomize a large number L of random code vectors to regenerate the traffic distributions for the urban plan. All L generated traffic distributions $[\tilde{\boldsymbol{M}}^1, \cdots, \tilde{\boldsymbol{M}}^L]$ are used to capture the future traffic distributions. The urban planners can summarize and evaluate various statistics of their interests using the L generated traffic distributions, for example, the mean, variance, minimum, maximum of L traffic distributions as outlined below.

Traffic mean distribution. The average of L generated traffic distributions reflects the average traffic status in the target region R after the plan is deployed. The positive or negative impacts of the urban construction plan on the local traffic status in R can be analyzed with the average generated distribution.

Traffic variance distribution. Similarly, we can take the variance of traffic status (*e.g.*, inflow) in each grid cell in R to obtain a traffic variance distribution, which indicates the fluctuation of the generated traffic status in each grid cell.

6 Evaluations

We conduct experiments to evaluate our proposed TrafficGAN with baseline approaches using large scale real world taxi GPS data.

6.1 Experiment Design

We performed two sets of experiments: (i) Generate traffic distributions in a target region R that was "seen" by TrafficGAN in the training set but under other travel demands. To do this, we remove the traffic distribution $\boldsymbol{M}_R$ of R under a specific travel demand d_R from the training set, and train our model with the rest of the data. Then we let TrafficGAN generate the traffic distribution $\tilde{\boldsymbol{M}}_R$ under d_R and compare them with the removed $\boldsymbol{M}_R$. (ii) Generate traffic distributions for an "unseen" target region R with a specific target travel demand d_R, where R was not included in the training set, therefore, TrafficGAN has never seen traffic distributions of R under any travel demand during training process. We extract the real traffic distributions of R from the original taxi GPS dataset and treat them as the ground truths, then we use the well-trained generator of TrafficGAN to generate the same number of traffic distributions and compare them with the ground truths. Obviously, the second task is more challenging.

In this paper, **Euclidean distance** and **mean absolute percentage error (MAPE)** are used to evaluate the quality of a generated traffic distribution against the ground truth traffic distribution of a target region R. Euclidean distance is defined as follows. For the ground-truth vector $\boldsymbol{V} = (v_1, \cdots, v_n)$ and the predicted vector $\hat{\boldsymbol{V}} = (\hat{v}_1, \cdots, \hat{v}_n)$, the Euclidean distance and MAPE between $\boldsymbol{V}$ and $\hat{\boldsymbol{V}}$ is:

$$\|\hat{V} - V\|_2 = \sqrt{\sum_{i=1}^{n} (\hat{v}_i - v_i)^2}. \tag{9}$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} |v_i - \hat{v}_i| / v_i, \tag{10}$$

We define statistics P_1—P_4 (measured by Eq. 9), P_1'—P_4' (measured by Eq. 10) to measure and evaluate the difference between the generated traffic distribution and the ground-truth traffic distribution.

•$\boldsymbol{P_1}$ and $\boldsymbol{P_1'}$: For each R and d_R pair, we calculate the average traffic distribution using real traffic distributions and refer to it as "true average distribution". We also calculate the average of generated traffic distributions and refer to it as "generated average distribution". The "true average distribution" and "generated average distribution" can be reshaped into two vectors, the smaller Euclidean distance and MAPE between the two vectors (denoted with P_1 and P_1', respectively) reflect that the mean of the generated data are similar to the mean of the true data.

•$\boldsymbol{P_2}$, $\boldsymbol{P_3}$ and $\boldsymbol{P_2'}$, $\boldsymbol{P_3'}$: Under the condition of target region R and target travel demand d_R, for each grid cell $s \in R$, we calculate the Euclidean distance and MAPE of s between real traffic distributions and generated distributions so that we have $N = \ell^2$ Euclidean distances and MAPEs for all $s \in R$. The mean of them (denoted as P_2 and P_2') indicate on average the similarity between the generated data and the true data for each grid cell. The standard deviation of these Euclidean distances and MAPEs are denoted as P_3 and P_3'.

•$\boldsymbol{P_4}$ and $\boldsymbol{P_4'}$ refer to the Euclidean distance and MAPE between real traffic distributions and generated traffic distributions with various travel demands. We combine all the real/generated traffic distributions with different travel demands as two huge matrices and reshape them into two vectors and calculate the Euclidean distance and MAPE between them, which indicate whether the traffic distributions conditioned on different travel demands are realistic or not.

6.2 Baseline Models

We compare our TrafficGAN with four baseline approaches below.

Standard cGAN [23]. Without deep convolutional layers, the generator and discriminator are both composed of four fully-connected layers and the first three layers are activated by ReLU, output of the generator is activated by hyperbolic tangent function, and the output of discriminator is fed to Sigmoid function.

Conditional DCGAN [20]. The generator and discriminator of cDCGAN are composed of four transposed convolutional/convolutional layers and the first three layers are batch normalized and activated by leaky ReLU, the output of the generator is activated by hyperbolic tangent function, and the output of discriminator is activated by Sigmoid.

Spatial smoothing approach with neighboring regions [24]. This method uses the traffic distributions of 9 closest regions under the same travel demand to compute a mean distribution as the resulting estimation. Note we only use available data in the training set to estimate and we will ignore a neighboring region if its data is not available for this travel demand.

Regression [25]. Ridge regression is applied to estimate the taxi inflow of each grid cell with the location of the grid cell and the travel demand as predictors.

6.3 Experiment Settings

In the experiments, we obtain $122,472$ traffic distributions of Shenzhen, China from Jul 1st to Dec 31st in 2016. We train TrafficGAN, cGAN and cDCGAN both for 200 epochs, and randomly sample code $\boldsymbol{z}$ from a standard normal distribution with $\mu = 0, \sigma = 1$. All models are trained using Adam with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and a learning rate of 2×10^{-5} for the first 10 epochs and linearly decayed to 2×10^{-6}. In the training process, we use batch stochastic gradient descent with a batch size of 128.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3014511, IEEE Transactions on Big Data

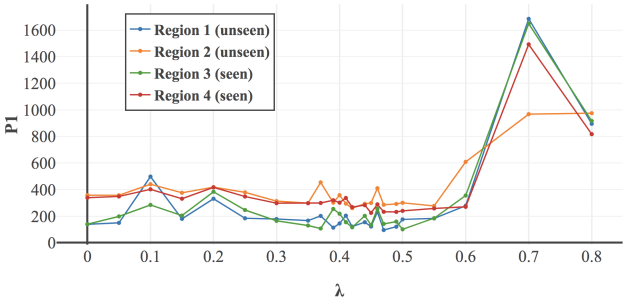


Fig. 10: P_1 value changes with λ

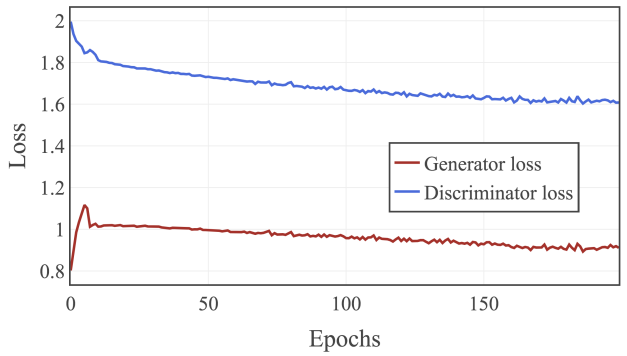


Fig. 11: TrafficGAN Loss with $\lambda = 0.47$

6.4 Evaluation Results

6.4.1 λ selection

As we illustrated in Fig. 7, as a parameter of TrafficGAN, the correlation threshold λ would influence the performance of the generation. First we test the impact of different λ on the results. In Fig. 10, we pick two "seen" regions and two "unseen" regions with specific travel demands to see how their P_1 performances change with λ. We can see when λ moves from 0 to 0.4, the performance slightly improves (P_1 decreasing) but with big fluctuations. P_1 becomes more stable with smaller fluctuations when $0.4 < \lambda < 0.5$. When $\lambda > 0.6$, the P_1 increases drastically indicating bad performance. Based on our test when $\lambda = 0.47$, almost all regions have the lowest P_1. Thus, in this paper, we pick 0.47 as the proper value of λ, all the following experiments are conducted with $\lambda = 0.47$. Moreover, with $\lambda = 0.47$, we show the convergence of our TrafficGAN with the loss plot in Fig. 11. In Fig. 11, the TrafficGAN approaches to convergence after 150 epochs.

6.4.2 Statistics comparisons with four baselines

With $\lambda = 0.47$, we have the P_1 values for 4 "seen" and 4 "unseen" regions in Fig. 12, where TrafficGAN always has the lowest P_1 indicating the mean of the generated data with TrafficGAN is the closest to the true data. Other statistics also have similar results.

We pick two representative regions (seen and unseen) as target regions with a specific travel demand. All the statistics are shown in Table. 1 and Table. 2. Since smoothing and regression method can only provide one estimated traffic distribution for a region based on a specific travel demand, only P_1 and P_1' can be calculated. For both "seen" and "unseen" regions, TrafficGAN presents the lowest error in all statistics, which indicates the generated traffic distributions with TrafficGAN are much closer to the real ones. Compared with cGAN and cDCGAN, our TrafficGAN model brings down the P_1 error by up to 93.79% and 39.12% on the "seen" region and up to 93.32% and 34.14% on the "unseen" region.

TABLE 1: Statistics Comparisons for an "Unseen" Region

	TrafficGAN	cGAN	cDCGAN	smoothing	regression
P_1	**956.78**	14321.60	1452.82	1178.62	55302.89
P_1'	**3.55**	1710.41	3.71	21.27	220.79
P_2	**420.50**	7096.76	523.37	-	-
P_2'	**0.65**	253.48	1.25	-	-
P_3	**314.21**	1914.90	539.78	-	-
P_3'	**0.87**	400.30	3.09	-	-
P_4	**5249.24**	73505.63	7519.95	-	-
P_4'	**1.68**	223.64	2.26	-	-

TABLE 2: Statistics Comparisons for a "Seen" Region

	TrafficGAN	cGAN	cDCGAN	smoothing	regression
P_1	**896.95**	14436.03	1473.42	1418.74	57792.57
P_1'	**4.35**	1669.10	6.43	207.04	527.47
P_2	**361.74**	6393.57	455.25	-	-
P_2'	**0.73**	247.53	2.13	-	-
P_3	**277.67**	1837.80	512.83	-	-
P_3'	**0.89**	403.26	5.04	-	-
P_4	**4560.26**	66524.57	6857.47	-	-
P_4'	**3.78**	287.66	4.23	-	-

6.4.3 Spatial pattern visualization

In this part, we visualize the generated/estimated traffic distributions and compare them with the real one. Here the traffic distributions are normalized to the same scale. Fig. 13 shows the visualizations of spatial patterns of 9 connected "unseen" regions, where each region has a corresponding travel demand. Fig. 13a marks the locations of selected 9 regions with red color on the whole city map. Fig. 13b shows the zoomed-in road map of the 9 regions. Fig. 13c shows the true average distribution and Fig. 13d shows the generated average distribution with TrafficGAN. Fig. 13e - 13h show the generated/estimated average traffic distribution of the baselines. Obviously, the generated average distribution with TrafficGAN captures the structure of the underlying road networks of all 9 "unseen" regions. TrafficGAN clearly outperforms all the baselines which cannot accurately learn the spatial patterns of "unseen" regions and they usually overestimate or underestimate the value in each grid cell.

6.4.4 Traffic condition visualization

Moreover, in the traffic estimation problem, we focus more on estimating the traffic conditions in roads. Fig. 14 shows the traffic conditions in all roads in an "unseen" region under a specific travel demand, where the roads in red indicate congestion, the yellow roads indicates slight congestion, and the green ones represent no traffic congestion. Fig. 14a shows the location of this "unseen" region. Fig. 14b is the road map of the "unseen" region with high travel demand locations marked. Fig. 14c shows actual traffic condition in the data. Fig. 14d shows the generated traffic condition with TrafficGAN, which is highly similar to the ground truth. Fig. 14e - 14h show the results of the baselines. Clearly, the results of TrafficGAN outperforms all baselines. Results on the "seen" regions also suggest the same trend, where TrafficGAN can better capture the road networks and generate more realistic traffic distributions than all baselines. Due to space limit, we only present the results on "unseen" regions since it is a harder task.

In conclusion, TrafficGAN is a success in traffic estimation, which can not only capture the shared patterns across

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3014511, IEEE Transactions on Big Data

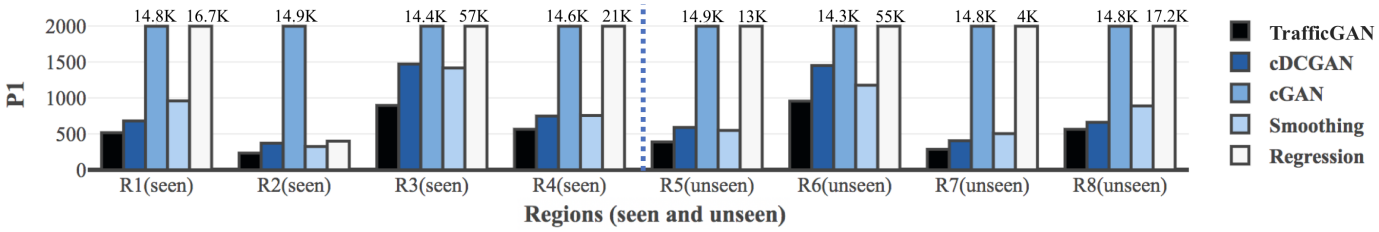
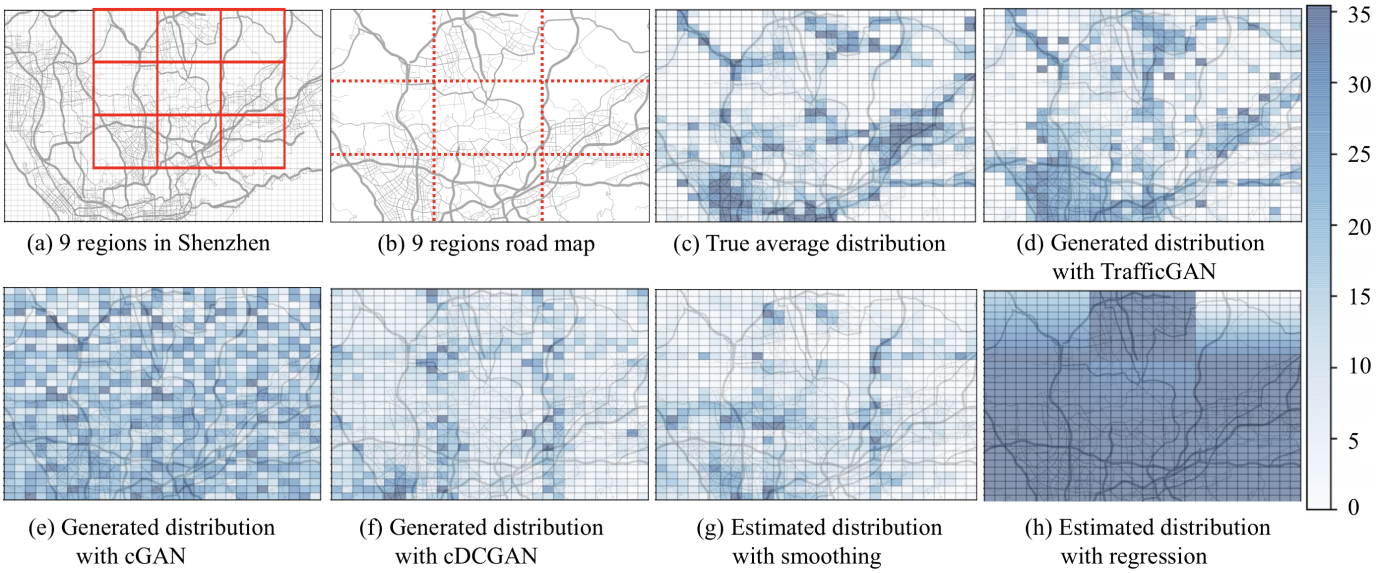


Fig. 12: P_1 comparisons over 8 target regions (seen and unseen)



(a) 9 regions in Shenzhen (b) 9 regions road map (c) True average distribution (d) Generated distribution with TrafficGAN

(e) Generated distribution with cGAN (f) Generated distribution with cDCGAN (g) Estimated distribution with smoothing (h) Estimated distribution with regression

Fig. 13: Spatial patterns of 9 "unseen" regions

spatial regions of how traffic conditions evolve according to travel demand changes and underlying road network structures, but also provide realistic estimation of the traffic conditions in roads based on different travel demands.

6.5 Case Studies

To further utilize our TrafficGAN, we look into real traffic condition evaluation cases in urban planning. As we mentioned earlier in this paper, the traffic condition always changes with the travel demand and rural areas usually have lower travel demands than urban areas. Therefore, it is a good opportunity to apply TrafficGAN in practice to forecast the possible traffic conditions under a not-yet-observed travel demand in an area.

For example, in 2018, Shenzhen government announced the plan to expropriate residential building. A large part of residential buildings to be expropriated are located in Longgang District. The goal of the expropriation is to build new residential and commercial areas in Longgang. The urban off-deployment traffic estimation can be performed before the expropriation and construction.

Longgang District is mainly located in the region marked with red box in Fig. 15a. The current average travel demand is 192. Fig. 15b shows the current traffic conditions. If new residential and business areas are built in Longgang, the travel demand would grow rapidly to 800 [14]. Fig. 15c shows the predicted traffic conditions of nearby roads under this expected travel demand. Compared with the current traffic conditions in Fig. 15b, apparently the overall traffic inflow is higher, the average traffic inflow increases drastically in two places marked in Fig. 15c. Fig. 15d illustrates possible reasons for the traffic congestion after the construction, *i.e.*, compacted roads and the poor design of lanes in these areas.

7 Related Work

Urban planning is a technical and political process concerning with urban data analysis, urban data mining, off-deployment evaluation and urban design. The off-deployment evaluation problem is a vital and difficult part among all the urban problems. Related works are summarized below.

Traffic volume prediction. Some previous works focus on traffic volume prediction from different perspectives. For example, [26] proposes a hybrid framework that integrates both state-of-art machine learning techniques and well-established traffic flow theory to estimate citywide traffic volume. In [27] and [28], the authors develop models to predict the road traffic volume and crowd flows in subway stations. These work assume unchanged urban settings and predict the traffic volume over time and locations. However, in this work we aim to generate the traffic distributions under various travel demands, which are significant changes to the urban settings.

Graph Convolutional Networks (GCN). [29] is usually used to classify the nodes in a graph. It can be seen as a generalization of neural network models like CNN to graphs and networks. GCN applies graph convolutional layers inside the model with a feature matrix and an adjacency matrix as inputs, where each row of the feature matrix

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3014511, IEEE Transactions on Big Data

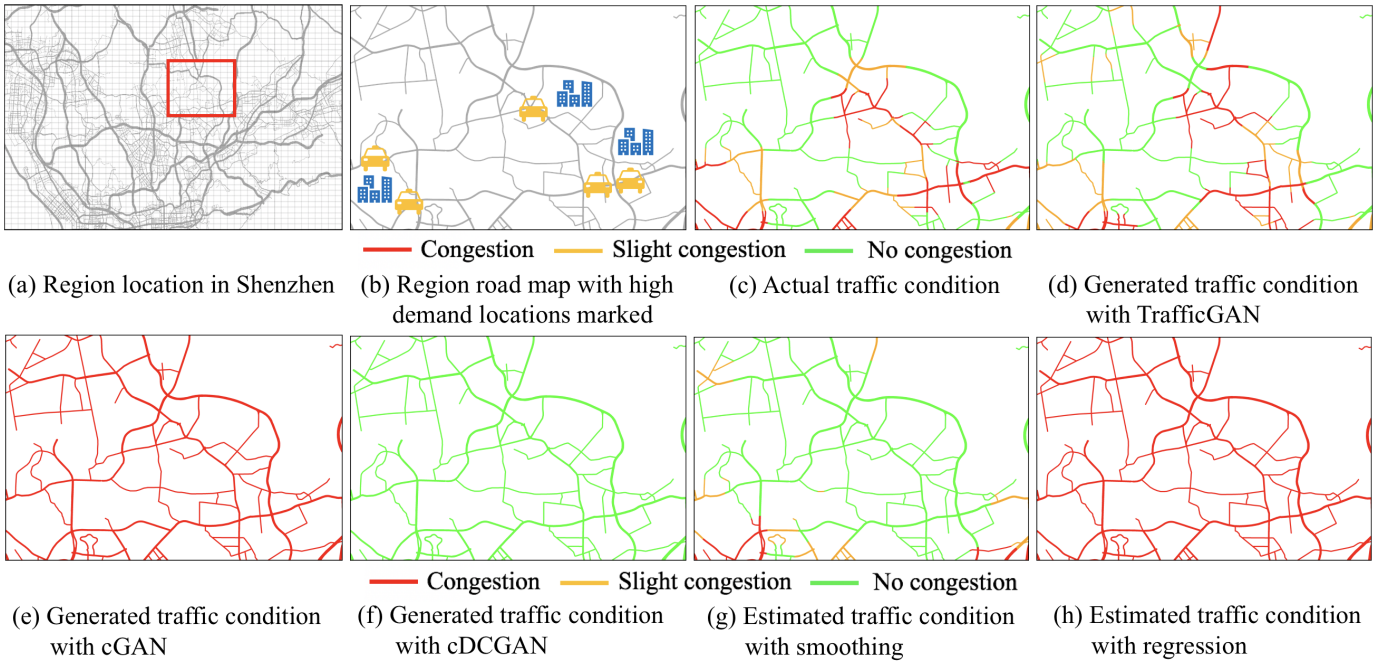


Fig. 14: Traffic conditions of an "unseen" region

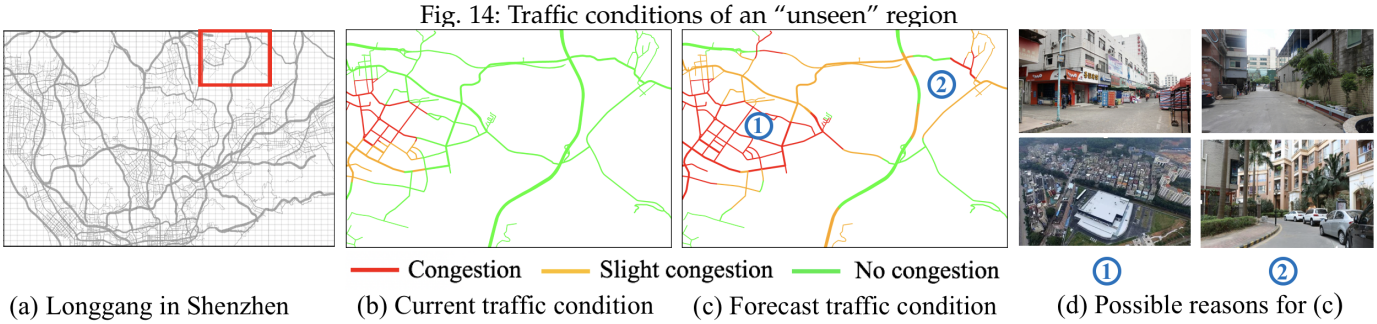


Fig. 15: Traffic condition forecast. (a) is the target region covering the Longgang District; (b) is the actual traffic condition with current travel demand in the region; (c) is the forecast traffic condition with a higher expected travel demand, where more congestion appears; (d) indicates two possible reasons for (c);

contains the features of one node, and the adjacency matrix is a representative description of the graph structure. Even though GCN takes the graph structure and the correlation between two nodes into consideration in convolution process, it is not a generative model which is in need to solve the traffic estimation problem.

Deep Learning for Urban Computing. Urban Computing is a general research area which integrates urban sensing, data management and data analysis as a unified process to explore, analyze and solve crucial problems related to people's everyday life [30]–[33]. With the recent rapid development of deep learning techniques, many researchers have made attempts to use deep learning models to solve urban computing problems. For example, Yuan et al. [33] propose to use a variation of the ConvLSTM model to predict traffic accidents. Wu et al use recurrent neural networks (RNN) to predict trajectories. Huang et al. [31] employ a deep attention model to predict crimes. Li et al. [32] employ a reinforcement learning method to dynamically reposition shared bikes. These work, however, do not use a generative model and they are very different from our problem.

Other Generative Models have been discussed in Section 4.2 when motivating the TrafficGAN model. They do not capture irregular spatial structures of the road networks and the traffic correlations and thus could not effectively solve our problem.

8 Conclusion

This paper proposed and investigated a novel *off-deployment traffic estimation* problem, namely, estimating the impact on regional traffic conditions before an urban construction plan is deployed. Solving this problem is crucial to potentially avoid traffic issues caused by an urban construction plan. In this paper, a novel generative model - TrafficGAN was proposed. Using traffic data (*e.g.*, taxi inflow) from all regions under different travel demands, TrafficGAN is trained to capture the fundamental patterns of how traffic condition evolves with respect to the travel demand changes and underlying road network structures. With such knowledge, the obtained generator is capable of generating realistic traffic conditions within a region for a not-yet-observed travel demand. Evaluation results on a large-scale real taxi dataset demonstrate that TrafficGAN can generate meaningful and accurate traffic distributions on the road network under various travel demands, and outperforms all the baselines.

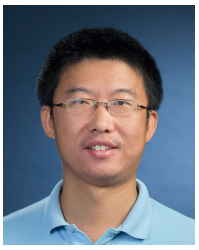
ACKNOWLEDGMENT

Yanhua Li and Yingxue Zhang were supported in part by NSF grants CNS-1657350 and CMMI-1831140, and a research grant from DiDi Chuxing Inc. Xun Zhou was partially supported by NSF grant IIS-1566386. Xiangnan Kong was supported in part by NSF grant IIS-1718310.

REFERENCES

[1] (2014) World urbanization prospects 2014. [Online]. Available: https://esa.un.org/unpd/wup/Publications/Files/WUP2014-Highlights.pdf
[2] J. Hooker, *Beijing Announces Traffic Plan for Olympics*. The New York Times, 2008.
[3] H. Su and S. Yu, "Hybrid GA based online support vector machine model for short-term traffic flow forecasting," in *APPT*, 2007, pp. 743–752.
[4] R. Herring, A. Hofleitner, P. Abbeel, and A. M. Bayen, "Estimating arterial traffic conditions using sparse probe data," in *ITSC*, 2010, pp. 929–936.
[5] P. S. Castro, D. Zhang, and S. Li, "Urban traffic modelling and prediction using large scale taxi GPS traces," in *Pervasive*, 2012, pp. 57–72.
[6] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: driving directions based on taxi trajectories," in *GIS*, 2010, pp. 99–108.
[7] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *KDD*, 2011, pp. 316–324.
[8] B. Karima, S. Ellagoune, H. Seridi, and H. Akdag, "Agent-based modeling for traffic simulation," June 2019.
[9] J. Taplin, "Simulation models of traffic flow," 08 2008.
[10] Y. Zhang, Y. Li, X. Zhou, X. Kong, and J. Luo, "TrafficGAN: Off-deployment traffic estimation with traffic generative adversarial networks," in *ICDM*, 2019.
[11] A. Singh Lemar, "Zoning as Taxidermy: Neighborhood Conservation Districts and the Regulation of Aesthetics," *Indiana Law Journal*, vol. 90, pp. 1525–1590, 09 2015.
[12] N. Mukai and N. Yoden, "Taxi demand forecasting based on taxi probe data by neural network," in *IIMSS*, 2012, pp. 589–597.
[13] E. J. Gonzales, C. J. Yang, E. F. Morgul, and K. Ozbay, "Modeling taxi demand with gps data from taxis and transit," Mineta National Transit Research Consortium, Tech. Rep., 2014.
[14] M. G. McNally, *The Four-Step Model*, ch. 3, pp. 35–53.
[15] Y. Li, M. Steiner, J. Bao, L. Wang, , and T. Zhu, "Region sampling and estimation of geosocial data with dynamic range calibration," in *ICDE*, 2014, pp. 1096–1107.
[16] Y. Li, J. Luo, C.-Y. Chow, K.-L. Chan, Y. Ding, and F. Zhang, "Growing the charging station network for electric vehicles with trajectory data analytics," in *ICDE*, 2015, pp. 1376–1387.
[17] B. Zhu and X. Xu, "Urban principal traffic flow analysis based on taxi trajectories mining," in *ICSI*, 2015, pp. 172–181.
[18] J. Guo, X. Li, Z. Zhang, and J. Zhang, "Traffic flow fluctuation analysis based on beijing taxi gps data," in *KSEM*, 2018, pp. 452–464.
[19] 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, pp. 2672–2680.
[20] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," 2015.
[21] J. Hauke and T. Kossowski, "Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data," *Quaestiones Geographicae*, vol. 30, no. 2, pp. 87 – 93, 2011.
[22] W. R. Tobler, "A computer movie simulating urban growth in the detroit region," *Economic geography*, vol. 46, no. sup1, pp. 234–240, 1970.
[23] M. Mirza and S. Osindero, "Conditional generative adversarial nets," in *CoRR abs/1411.1784*, 2014.
[24] A. Getis, "A history of the concept of spatial autocorrelation: A geographer's perspective," *Geographical Analysis*, vol. 40, pp. 297–309, 07 2008.
[25] I. Alam, D. M. Farid, and R. J. F. Rossetti, "The prediction of traffic flow with regression analysis," in *IEMIS*, 2019, pp. 661–671.
[26] X. Zhan, Y. Zheng, X. Yi, and S. Ukkusuri, "Citywide traffic volume estimation using trajectory data," *TKDE*, vol. 29, no. 2, pp. 272–285, 2017.
[27] X. Liu, X. Kong, and Y. Li, "Collective traffic prediction with partially observed traffic history using location-base social media," in *CIKM*, 2016, pp. 2179–2184.
[28] 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, pp. 112–128.
[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
[30] H. Wu, Z. Chen, W. Sun, B. Zheng, and W. Wang, "Modeling trajectories with recurrent neural networks," in *IJCAI*, 2017, pp. 341–350.
[31] C. Huang, J. Zhang, Y. Zheng, and N. V. Chawla, "DeepCrime: Attentive hierarchical recurrent networks for crime prediction," in *CIKM*, 2018, pp. 1423–1432.
[32] Y. Li, Y. Zheng, and Q. Yang, "Dynamic bike reposition: A spatio-temporal reinforcement learning approach," in *KDD*, 2018, pp. 1724–1733.
[33] Z. Yuan, X. Zhou, and T. Yang, "Hetero-ConvLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data," in *KDD*, 2018, pp. 984–992.

Yingxue Zhang is currently a Ph.D. student in the Department of Data Science at Worcester Polytechnic Institute. She received her BS degree in Computer Science from Shanghai Jiao Tong University and received her MS degree in Financial Engineering from Stevens Institute of Technology. Her research interests include deep learning and urban big data analysis.



Yanhua Li received two Ph.D. degrees in electrical engineering from Beijing University of Posts and Telecommunications, Beijing in China in 2009 and in computer science from University of Minnesota at Twin Cities in 2013, respectively. He has worked as a researcher in HUAWEI Noah's Ark LAB at Hong Kong from Aug 2013 to Dec 2014, and has interned in Bell Labs in New Jersey, Microsoft Research Asia, and HUAWEI research labs of America from 2011 to 2013. He is currently an Assistant Professor in the Department of Computer Science at Worcester Polytechnic Institute (WPI) in Worcester, MA. His research interests are big data analytics and urban computing in many contexts, including urban network data analytics and management, urban planning and optimization.

Xun Zhou is currently an Assistant Professor in the Department of Management Sciences at the University of Iowa. He received a PhD degree in Computer Science from the University of Minnesota, Twin Cities in 2014. His research interests include big data management and analytics, spatial and spatio-temporal data mining, and Geographic Information Systems (GIS). He has published over 30 papers in these areas and has received three best paper awards. He also served as a co-editor-in-chief of the Encyclopedia of GIS, 2nd Edition.

Xiangnan Kong received his Ph.D. degree in Computer Science from the University of Illinois at Chicago. He received his BS and MS degrees in CS from Nanjing University. He is an Assistant Professor in Computer Science at the Worcester Polytechnic Institute in the US. Dr. Kong's research interests include data mining and machine learning. He has published more than 80 papers in refereed journals and conferences.

Jun Luo is a principal researcher at Lenovo Machine Intelligence Center, Lenovo Group Limited, in Hong Kong. He received his PhD degree in computer science from the University of Texas at Dallas, USA, in 2006. His research interests include big data, machine learning, spatial temporal data mining and computational geometry. He has published over 90 journal and conference papers in these areas.