C^3-GAN: Complex-Condition-Controlled Urban Traffic Estimation through Generative Adversarial Networks

Yingxue Zhang*, Yanhua Li*, Xun Zhou†, Zhenming Liu‡, Jun Luo§
*Worcester Polytechnic Institute
†University of Iowa
‡College of William & Mary
§Lenovo Group Limited
yzhang31@wpi.edu; yli15@wpi.edu; xun-zhou@uiowa.edu; zliu@cs.wm.edu; jluo1@lenovo.com

***Abstract*—Given historical traffic distributions and associated urban conditions observed in a city, the conditional urban traffic estimation problem aims at estimating realistic future projections of the traffic under a set of new urban conditions, *e.g.*, new bus routes, rainfall intensity and travel demands. The problem is important in reducing traffic congestion, improving public transportation efficiency, and facilitating urban planning. However, solving this problem is challenging due to the strong spatial dependencies of traffic patterns and the complex relations between the traffic and urban conditions. In this paper, we tackle the challenges by proposing a novel Complex-Condition-Controlled Generative Adversarial Network (C^3-GAN) for urban traffic estimation of a region under various complex conditions. C^3-GAN features the following three novel designs on top of standard cGAN model: (1) an embedding network mapping the complex conditions to a latent space to find representations of the urban conditions; (2) an inference network to enhance the relations between the embedded latent vectors and the traffic data. Extensive experiments on real-world datasets demonstrate that our C^3-GAN produces high-quality traffic estimations and outperforms state-of-the-art baseline methods.**

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

I. Introduction

Given the road network of a city and the historical traffic status (*e.g.*, volume, speed) over the network under various complex conditions (*e.g.*, travel demands, constructions, transit line designs) in the city, the problem of *conditional urban traffic estimation problem* aims at generating realistic traffic distribution projections of the city under new, previously unseen environmental conditions.

The urban traffic estimation problem has long been an important issue in various aspects of urban planning, including bus route planning, traffic management, land use design, *etc*. Accurate urban traffic estimation can not only help to reduce traffic congestion and improve the public transportation efficiency, but also provide insights for new urban constructions. For example, as illustrated in Figure 1, since the taxi demand greatly influenced the local traffic in Shenzhen, China, new subway stations were planned to be built to reduce the local taxi demand and thus release the traffic burden. Before the deployment, traffic estimations were conducted aiming to find the the best locations for new subway stations. Therefore, urban traffic estimation is an important step when evaluating an urban construction plan.

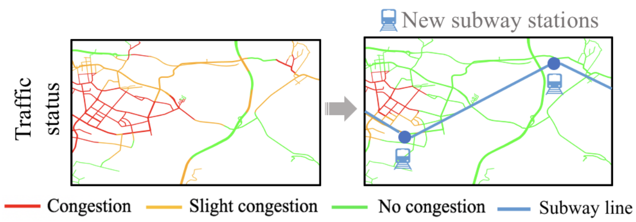


Fig. 1: Traffic before & after building subway stations.

Challenges. Realistic and accurate urban traffic estimation is usually sophisticated and challenging due to the following reasons:

(1) *Complex conditions.* The urban conditions that affect traffic distributions are usually complex and unstructured in representation, such as multi-dimensional tensors or matrices (*e.g.*, subway routes) instead of simple labels or numeric measures. The complexity of the conditions leads to difficulties in building a strong connection between the conditions and the traffic distribution, making it hard to capture the traffic changes caused by these factors.

(2) *Complex traffic spatial dependencies*[1]. The traffic status at a location is usually correlated with the traffic status in nearby locations. Such traffic dependencies are hard to capture since the underlying complex road networks usually lead to diverse traffic patterns.

The urban traffic estimation problem has received a lot of attentions in recent years. While most of the works addressed the second challenge above, *the first challenge is still unaddressed*. Some works [2], [3], [8] try to solve this problem with classical machine learning models. However, when estimating traffic status regarding to complex conditions, they typically cannot get good performance since they are incapable of dealing with complex conditions and accurately capturing traffic changes.

[1]In this paper, we focus on getting the estimation of the average traffic under specific conditions so do not consider the temporal dependencies here.

TABLE I: Notations

Notations	Descriptions
i, j	Locations (coordinates in a grid world)
$\mathcal{S} = \{s_{ij}\}$	Grid cells within a city
$\boldsymbol{h} \in \mathbb{R}^{m\times n}$	Traffic condition
$\boldsymbol{x} \in \mathbb{R}^{m\times n}$	Traffic distribution
$\boldsymbol{z} \in \mathbb{R}^{v}$	Randomly sampled noise
$\boldsymbol{c} \in \mathbb{R}^{u}$	Embedded latent vector

Recently, many works have focused on applying deep neural networks to solve traffic estimation problem. For example, stacked autoencoder [11] and ConvLSTM [17], [18] are used to predict travel demands and traffic accidents, respectively. These models greatly improve the prediction accuracy, however, they did not consider the impact of conditions and thus fail to solve the conditional urban traffic estimation problem. Besides, TrafficGAN [19] and Curb-GAN [20] take simple conditions into account and estimate traffic with advanced GAN models. However, both of them cannot handle complex conditions, which usually lead to model collapse or instability.

Contributions. In this paper, we aim to solve the conditional urban traffic estimation problem and tackle both of the aforementioned challenges from a traffic data generation perspective. We propose a novel model — Complex-Condition-Controlled Generative Adversarial Network (C^3-GAN), which can successfully estimate traffic of an area based on complex urban conditions. Figure 2 is our solution framework. Our C^3-GAN features an embedding network and an inference network on top of the standard conditional GAN model. Our main contributions can be summarized as follows:

- We formulate the conditional urban traffic estimation problem as a traffic data generation problem, and propose a novel model — Complex-Condition-Controlled Generative Adversarial Network (C^3-GAN). C^3-GAN handles complex urban conditions through an fixed embedding network which transforms the complex conditions to latent vectors, and an inference network which enhances the connections between the embedded vectors and the traffic data. (See Section III-B.)
- We perform extensive experiments on real-world datasets to evaluate our C^3-GAN. The experimental results prove that C^3-GAN can significantly improve the urban traffic estimation performance 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.

Definition 1 (Grid cells). A whole city is divided into $m \times n$ grid cells, which have equal side-length in latitude and longitude. We denote the set of grid cells in the city as $\mathcal{S} = \{s_{ij}\}$, where $1 \leq i \leq m$ and $1 \leq j \leq n$.

Definition 2 (Urban conditions). Urban conditions (*e.g.*, travel demands, time of the day, *etc*) usually have strong correlations with road traffic. In this paper, we only consider complex urban conditions, *e.g.*, bus routes, rainfall intensity, *etc*. These complex conditions are usually presented in matrix form, thus, we denote a matrix $\boldsymbol{h} \in \mathbb{R}^{m\times n}$ as one feature map of the city in a period of time, where each element $h_s \in \mathbb{R}$ of the matrix indicates the corresponding condition in a specific grid cell $s \in \mathcal{S}$.

For example, if we use travel demand of a city as an urban condition, $\boldsymbol{h}$ will be a $m \times n$ travel demand matrix, each entry of the matrix indicates the average travel demand of a grid cell during a specific time slot.

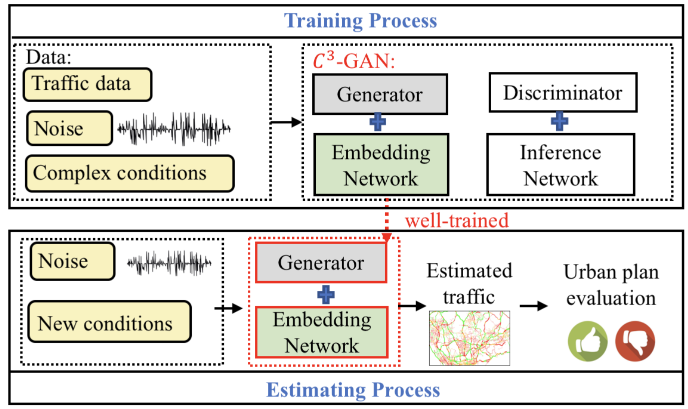


Fig. 2: Solution framework.

Definition 3 (Traffic status and traffic distribution). Traffic status indicates the basic knowledge of the road traffic, which can be measured by different measurements, *e.g.*, traffic speed, traffic volume, *etc*. We denote x_s as the average traffic status of a grid cell $s \in \mathcal{S}$ within a period of time, and a matrix $\boldsymbol{x} \in \mathbb{R}^{m\times n}$ as the traffic distribution of the city.

Problem Statement: A city area is partitioned into grid cells $\mathcal{S}$, given historical samples of complex urban conditions $\mathcal{H} = \{\boldsymbol{h}_t\}$ and traffic distributions $\mathcal{X} = \{\boldsymbol{x}_t\}$ over a time span $1 \leq t \leq T$, we aim to estimate the future traffic distributions $\tilde{\boldsymbol{x}}$ given a set of new features $\tilde{\boldsymbol{h}}$.

III. Methodologies

Built upon the state-of-the-art (SOTA) literatures in generative models (See Section III-A), we propose C^3-GAN for the conditional urban traffic estimation problem. C^3-GAN address the two challenges we mentioned in Section I with its unique designs:

(1) *Complex conditions challenge:* the proposed C^3-GAN introduces an randomly chosen embedding network and an inference network on top of the original cGAN to extract high-quality representations of the complex conditions and produce good generation results (See Section III-B).

(2) *Complex traffic spatial dependencies challenge:* C^3-GAN applies convolutional layers inside each model component to capture the traffic spatial dependencies.

A. SOTA of Deep Generative Models

Various generative adversarial networks (GANs) have been proposed to build mappings from simple distributions to data corpuses, *e.g.*, images, texts, *etc* [7], [13]. The general idea of conditional GANs matches the problem of urban traffic estimation very well. Below, we briefly introduce two GAN models that are relevant to our C^3-GAN, namely, the conditional GAN [12] and InfoGAN [4], and discuss the technical gaps for them to solve our problem.

Conditional GAN. The conditional generative adversarial network (cGAN) is a deep generative model proposed by Mirza *et al.* [12]. The generation process of cGAN is governed by conditions, which tackles a min-max game as shown in Eq. 1. The goal of the generator G is to learn a distribution matching the real data distribution p_{data} using random noises $\boldsymbol{z} \sim p_{\boldsymbol{z}}$ and conditions $\boldsymbol{h}$, the discriminator D aims to distinguish the true data pairs from the generated ("fake") ones.

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

<u>*Limitation of cGAN.*</u> Since our goal is to generate urban traffic estimations $\boldsymbol{x}$ using complex conditions $\boldsymbol{h}$ (*e.g.*, bus routes, travel demands), our intuition is to apply cGAN framework. However, cGAN usually deals with simple conditions (*e.g.*, discrete or continuous numbers), once the conditions become more complex (*e.g.*, multi-dimensional tensors or matrices), it is hard for standard cGAN to build strong connections between $\boldsymbol{x}$ and $\boldsymbol{h}$ and generate reasonable results due to the high-dimensionality of $\boldsymbol{h}$.

InfoGAN. InfoGAN proposed by Chen *et al.* [4] is an extension of GAN model [7], which adds a mutual information based regularizer to enable disentangled representations. To learn the semantic features of data, InfoGAN first splits the latent code into two parts — the disentangled code vector $\boldsymbol{c}$ and the remaining code vector $\boldsymbol{z}$, and then maximizes the mutual information $I(\boldsymbol{c}; G(\boldsymbol{c}, \boldsymbol{z}))$ and thus realize the goal of distinguishing data in an unsupervised fashion. The objective is given by the following expression:

$$\min_G \max_D \mathcal{L}_{\text{GAN}}(G, D) - \lambda I(\boldsymbol{c}; G(\boldsymbol{c}, \boldsymbol{z})), \tag{2}$$

where $\mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D(G(\boldsymbol{z})))]$.

<u>*Limitation of InfoGAN.*</u> Even though InfoGAN enables disentangled representations, it cannot be used to deal with conditional traffic estimation problem with complex urban conditions (*e.g.*, bus routes, travel demands). InfoGAN learns data representations from unlabeled data in a unsupervised learning paradigm, however, once the dimension of the disentangled code vector is too high, InfoGAN would fail since it is hard to learn the relations between the data and the disentangled code. In this paper, we are inspired by InfoGAN loss (especially the mutual information based regularizer) which can help to build strong connections between the generated data and the conditions, and design a new model C^3-GAN to better deal with the complex conditions, the details are elaborated in Section III-B.

B. Objective

Given the limitations of SOTA works of generative models to our traffic estimation problem, we propose a novel model C^3-GAN to tackle the conditional urban traffic estimation problem. The overview of C^3-GAN is shown in Figure 3(a). In C^3-GAN, we first focus on solving the complex condition challenge, thus, we propose to transform high-dimensional $\boldsymbol{h}$ to low-dimensional vector $\boldsymbol{c} \in \mathbb{R}^u$ with an randomly chosen and fixed embedding network E. The embedded latent vectors $\boldsymbol{c}$ should reflect key characteristics of the corresponding urban conditions $\boldsymbol{h}$. Once we use the embedded vector $\boldsymbol{c}$ and noise $\boldsymbol{z} \sim p_{\boldsymbol{z}}$ to generate the urban traffic $\boldsymbol{x} \sim G(\boldsymbol{z}, \boldsymbol{c})$, we need to ensure that the generated $\boldsymbol{x}$ is like real and matches the original condition $\boldsymbol{h}$, and also guarantee that the embedded latent vector $\boldsymbol{c}$ can accurately infer the corresponding traffic $\boldsymbol{x}$.

Since we use the embedding $\boldsymbol{c}$ of urban condition $\boldsymbol{h}$ for generation, the objective can be written as:

$$\begin{aligned}\min_G \max_D V(G, D) = & \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D(\boldsymbol{x}, \boldsymbol{h})] \\ &+ \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D(G(\boldsymbol{z}, \boldsymbol{c}), \boldsymbol{h}))].\end{aligned} \tag{3}$$

Eq.3 alone is certainly not good enough to produce good generation results, since the connections between $\boldsymbol{c}$ and $\boldsymbol{x}$ are not guaranteed. Borrowing the idea from InfoGAN, maximizing the mutual information $I(\boldsymbol{c}; (G(\boldsymbol{z}, \boldsymbol{c}), \boldsymbol{h}))$ can help to build strong connections between $\boldsymbol{c}$ and generated $\boldsymbol{x}$.

In information theory, mutual information between two random variables X and Y measures the "amount of information" learned from Y about X. The mutual information between X and Y and be written as:

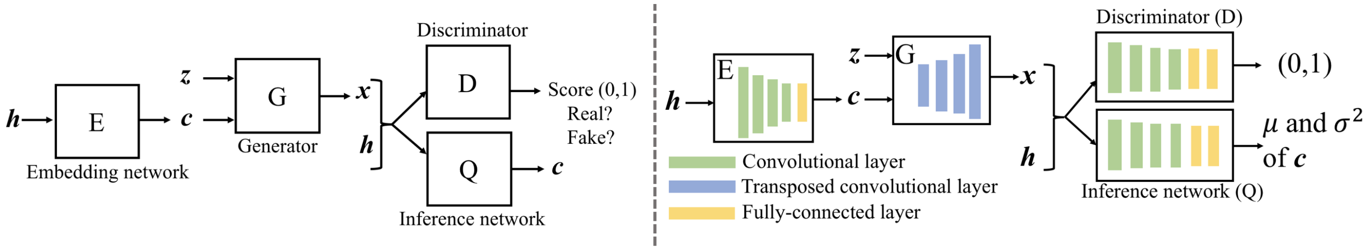
$$I(X; Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X). \tag{4}$$

Based on Eq.4, the mutual information $I(X; Y)$ can be interpreted as the reduction of uncertainty in X when Y is provided. $I(X; Y) = 0$ represents X and Y are independent and knowing one variable reveals nothing about the other; by contrast, maximizing $I(X; Y)$ means Y can provide the most information about X.

Hence, to enhance the connections between embedded vectors $\boldsymbol{c}$ and the generated traffic $\boldsymbol{x}$, we propose to maximize $I(\boldsymbol{c}; (G(\boldsymbol{z}, \boldsymbol{c}), \boldsymbol{h}))$ instead of $I(\boldsymbol{c}; G(\boldsymbol{z}, \boldsymbol{c}))$, since both of $\boldsymbol{x} \sim G(\boldsymbol{z}, \boldsymbol{c})$ and $\boldsymbol{h}$ contain the information of $\boldsymbol{c}$, which indicates $G(\boldsymbol{z}, \boldsymbol{c})$ and $\boldsymbol{h}$ together have stronger mutual information with $\boldsymbol{c}$ than $G(\boldsymbol{z}, \boldsymbol{c})$ alone does. Thus, maximizing $I(\boldsymbol{c}; (G(\boldsymbol{z}, \boldsymbol{c}), \boldsymbol{h}))$ can not only enhance the control of $\boldsymbol{c}$ over generated $\boldsymbol{x} \sim G(\boldsymbol{z}, \boldsymbol{c})$ but also potentially accelerate the convergence. Therefore, we add a mutual information regularizer to Eq.3:

$$\begin{aligned}&\min_G \max_D V_I(G, D) = V(G, D) - \lambda I(\boldsymbol{c}; (G(\boldsymbol{z}, \boldsymbol{c}), \boldsymbol{h})); \\ &\text{where } \boldsymbol{c} = E(\boldsymbol{h}).\end{aligned} \tag{5}$$

In practice, the mutual information term $I(\boldsymbol{c}; (G(\boldsymbol{z}, \boldsymbol{c}), \boldsymbol{h}))$ is hard to characterize analytically, since we do not have the access to the posterior distribution $P(\boldsymbol{c}|(G(\boldsymbol{z}, \boldsymbol{c}), \boldsymbol{h}))$. Instead, we can calculate the lower bound of $I(\boldsymbol{c}; (G(\boldsymbol{z}, \boldsymbol{c}), \boldsymbol{h}))$ and use the an auxiliary distribution $Q(\boldsymbol{c}|(G(\boldsymbol{z}, \boldsymbol{c}), \boldsymbol{h}))$ to approximate



(a) Model overview

(b) Detailed architecture

Fig. 3: C^3-GAN structure.

$P(\boldsymbol{c}|(G(\boldsymbol{z},\boldsymbol{c}),\boldsymbol{h}))$. We denote $\hat{\boldsymbol{x}} = (G(\boldsymbol{z},\boldsymbol{c}),\boldsymbol{h})$ for simplicity, the lower bound of $I(\boldsymbol{c};(G(\boldsymbol{z},\boldsymbol{c}),\boldsymbol{h}))$ is as follows:

$$\begin{aligned}
&I(\boldsymbol{c};(G(\boldsymbol{z},\boldsymbol{c}),\boldsymbol{h})) = H(\boldsymbol{c}) - H(\boldsymbol{c} \mid (G(\boldsymbol{z},\boldsymbol{c}),\boldsymbol{h}))\\
&= \mathbb{E}_{\boldsymbol{x}\sim G(\boldsymbol{z},\boldsymbol{c}),\boldsymbol{h}\sim p_{\text{data}}}\left[\mathbb{E}_{\boldsymbol{c}'\sim P(\boldsymbol{c}|\hat{\boldsymbol{x}})}\left[\log P\left(\boldsymbol{c}' \mid \hat{\boldsymbol{x}}\right)\right]\right] + H(\boldsymbol{c})\\
&= \mathbb{E}_{\boldsymbol{x}\sim G(\boldsymbol{z},\boldsymbol{c}),\boldsymbol{h}\sim p_{\text{data}}}[\underbrace{D_{\text{KL}}(P(\cdot \mid \hat{\boldsymbol{x}})\|Q(\cdot \mid \hat{\boldsymbol{x}}))}_{\geq 0}\\
&+ \mathbb{E}_{\boldsymbol{c}'\sim P(\boldsymbol{c}|\hat{\boldsymbol{x}})}\left[\log Q\left(\boldsymbol{c}' \mid \hat{\boldsymbol{x}}\right)\right]] + H(\boldsymbol{c})\\
&\geq \mathbb{E}_{\boldsymbol{x}\sim G(\boldsymbol{z},\boldsymbol{c}),\boldsymbol{h}\sim p_{\text{data}}}\left[\mathbb{E}_{\boldsymbol{c}'\sim P(\boldsymbol{c}|\hat{\boldsymbol{x}})}\left[\log Q\left(\boldsymbol{c}' \mid \hat{\boldsymbol{x}}\right)\right]\right] + H(\boldsymbol{c})\\
&= L_I(G,Q),
\end{aligned} \tag{6}$$

where Q is the auxiliary distribution, and we can treat Q as a inference neural network which uses $\hat{x}$ to infer c just as illustrated in Figure 3(a). Moreover, we can simply omit $H(\boldsymbol{c})$ in $L_I(G,Q)$ since it is a constant when $\boldsymbol{c}$ is sampled from a fixed distribution. As a result, our final objective is as Eq.7:

$$\begin{aligned}
&\min_{G,Q}\max_{D} V(D,G,Q) = V(G,D) - \lambda L_I(G,Q);\\
&\text{where } \boldsymbol{c} = E(\boldsymbol{h}).
\end{aligned} \tag{7}$$

C. C^3-GAN Architecture

To tackle the complex spatial dependencies challenge, we design a unique architecture for C^3-GAN. Figure 3(b) shows the detailed architecture of C^3-GAN, which contains an embedding network E, a generator G, a discriminator D and an inference network Q. C^3-GAN applies convolutional layers inside each model component to capture the traffic spatial dependencies. Next, we will introduce our model architecture in detail.

The embedding network E is a randomly chosen and fixed network, which aims to find a latent representation for the high-dimensional urban condition $\boldsymbol{h}$. The input of E is an original condition $\boldsymbol{h}$, the output is a low-dimensional latent vector $\boldsymbol{c}$. Inside E, we have several convolutional layers, which help to capture the spatial patterns of urban conditions, each one is followed by a batch normalization and activated by Leaky ReLU [15], the final fully-connected layer is activated by hyperbolic tangent function.

The generator G aims to generate like-real traffic distributions with respect to an embedded latent vector $\boldsymbol{c}$. 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) an embedded latent vector $\boldsymbol{c}$. G outputs the generated traffic distribution $\boldsymbol{x} \sim G(\boldsymbol{z},\boldsymbol{c})$. Inside the generator G, $\boldsymbol{c}$ and $\boldsymbol{z}$ are concatenated together and go through some convolutional layers, where all the layers but the last one are batch normalized and activated by ReLU, the last layer is activated by hyperbolic tangent.

The discriminator D aims to distinguish the real data from the generated data by giving high score if the input $\boldsymbol{x}$ is from real data and matches the urban condition $\boldsymbol{h}$. It yields a low score if the input $\boldsymbol{x}$ is "fake" or does not match the input $\boldsymbol{h}$. Its input includes i) traffic distribution $\boldsymbol{x}$, 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 $\boldsymbol{x} \sim G(\boldsymbol{z},\boldsymbol{c})$, and ii) the urban condition $\boldsymbol{h}$. Inside the discriminator D, it contains 4 different convolutional layers and two fully-connected layers, each layer is batch normalized and activated by leaky ReLU function, the last fully-connected layer is activated by Sigmoid and outputs a score between 0 and 1 indicating the extent to which the data is real.

The inference network Q aims to recover the distribution of latent vector $\boldsymbol{c}$ using $(\boldsymbol{x},\boldsymbol{h})$ pairs, so Q takes the same input as D, which includes $\boldsymbol{x}$ and $\boldsymbol{h}$. Q has the similar architecture to the discriminator. Inside the inference network Q, it contains 4 different convolutional layers and two fully-connected layers, each convolutional layer is batch normalized and activated by leaky ReLU function, the final outputs of the inference network include the mean and variance of the $\boldsymbol{c}$ distribution.

IV. Evaluation

In this section, we will first introduce the real-world datasets we use, and then present baselines and evaluation metrics. At last, we provide our experimental results.

A. Dataset and Experiment Descriptions

Dataset Descriptions. We validate our model on real-world datasets: (1) traffic speed, (2) taxi inflow, (3) bus routes and (4) travel demand. The detailed information of the dataset is shown in Table II.

- **Traffic speed.** The hourly average traffic speed is extracted from GPS records collected from taxis in Shenzhen, China from Jul 1st to Dec 31st, 2016. The whole city is partitioned into 40×50 grid cells with a side-length $l_1 = 0.0084°$ in latitude and $l_2 = 0.0126°$ in longitude, and the traffic status in each grid cell is measured by average traffic speed. The data size is (1944, 1, 40, 50), which means there are 1944 traffic distributions in total, and each traffic distribution is a 40×50 matrix.
- **Taxi inflow.** The data is collected from taxis in Shenzhen, China from July 1st to Dec. 31st, 2016. Taxi inflow is the

TABLE II: Dataset descriptions.

Dataset	Timespan	Data size	Dimension
Traffic speed	07/01/2016-12/31/2016	(1944, 1, 40, 50)	40×50
Taxi inflow	07/01/2016-12/31/2016	(1944, 1, 40, 50)	40×50
Bus routes	07/01/2016-12/31/2016	(1944, 20, 40, 50)	$20 \times 40 \times 50$
Travel demand	07/01/2016-12/31/2016	(1944, 1, 40, 50)	40×50

count of all taxis that stayed or arrived at each grid cell within one hour. The data size is (1944, 1, 40, 50), which indicates there are 1944 traffic distributions (of 1-hour slot) in total, each traffic distribution is a 40×50 matrix.

- **Bus routes.** The bus data is collected from 20 different bus routes in Shenzhen, China from July 1st to Dec 31st, 2016. Since there are 990 bus routes in total in Shenzhen City, and only a few of them got updated during July 1st to Dec 31st, 2016, thus, we randomly sample 20 bus routes for simplicity which includes both the unchanged and updated ones. For each bus route, we have the bus route map which is also divided into 40×50 grid cells, and the value of each grid cell indicates the number of buses passing this area within one hour. The data size is (1944, 20, 40, 50), which indicates there are 1944 time slots (1-hour), each time slot has 20 bus route maps, and each bus route map is a 40×50 matrix, so the data dimension for each time slot is $20 \times 40 \times 50$.
- **Travel demand.** The travel demand data is collected from taxis GPS records in Shenzhen, China from July 1st to Dec. 31st, 2016. To extract the travel demands, in each time slot of a day, *i.e.*, one hour, we count the total taxi pickup events within each grid cell. In general, it is hard to obtain the total travel demand including all transport modes. In this work, we use the demand for taxis to represent the local travel demand, where many studies have shown that taxi demands represent the total demands quite well [6], [14], [20]. The data size is also (1944, 1, 40, 50), which indicates there are 1944 travel demand maps (in 1-hour slot), and each travel demand map is a 40×50 matrix.

Experiment Descriptions. We introduce all different traffic estimation experiments we conducted below.

- **Task 1: traffic speed and taxi inflow estimation based on bus route changes.** In this task, we study how the bus route changes (as urban condition) influence the traffic, thus, we estimate the traffic distributions in Shenzhen City given the complex bus routes as conditions which should be $20 \times 40 \times 50$ tensors, and traffic speed and taxi inflow are both estimated. All the data including traffic speed, taxi inflow and bus route data is divided into training set (90% of data) and testing set (the remaining 10%).
- **Task 2: traffic speed and inflow estimation based on travel demand changes.** In this task, we study how the travel demand changes influence the traffic status, thus, we estimate the traffic distributions in Shenzhen City given the complex travel demands as conditions which are 40×50 matrices. All the data including traffic speed, taxi inflow and travel demand data is also divided into training set (90% of data) and testing set (the remaining 10%).

B. Baselines

To evaluate the effectiveness of our model, we compare our C^3-GAN with state-of-the-art methods. We first use the following two baselines to validate that standard cGAN cannot successfully estimate urban traffic based on complex urban conditions:

- **cGAN [12].** This is the standard conditional GAN, which applies convolutional layers inside both generator and discriminator.
- **cGAN+L1 [9].** This method uses standard conditional GAN structure. The objective of discriminator stays unchanged, while the generator is trained using both the adversarial loss and the L1 loss.

Then, we use two baseline methods to validate that cGAN with a single embedding network E or a Q network is not good enough to solve the traffic estimation problem with complex conditions:

- **cGAN+E [5].** This method uses a predefined embedding network E (namely, a randomly selected network) to transform the original complex conditions to low-dimensional conditions. With the embedded conditions in a latent space, the standard conditional GAN is applied.
- **InfoGAN [4], [16].** InfoGAN adds an encoder Q to the standard GAN structure, the details are explained in Section III-A.

Besides, we also have state-of-the-art GAN models for traffic estimation as baselines:

- **Curb-GAN [10], [20].** Curb-GAN applies self-attention and convolutional layers to deal with sequential data generation problem, here only convolutional layers are applied to generate average traffic estimations. The generator is trained with the adversarial loss and the L2 loss together.
- **TrafficGAN [19]** TrafficGAN solves the conditional traffic estimation problem, where the conditions are simple discrete values. TrafficGAN applies several dynamic convolutional layers inside generator and discriminator to capture spatial patterns of traffic.

C. Evaluation Metrics

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

$$\mathrm{MAPE} = \frac{1}{n_s}\sum_{i=1}^{n_s} |y_i - \hat{y}_i| / y_i,$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n_s}\sum_{i=1}^{n_s} (y_i - \hat{y}_i)^2}, \tag{8}$$

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

TABLE III: Performance on task 1: traffic speed and taxi inflow estimation based on bus route changes.

Methods		cGAN	cGAN+L1	Curb-GAN	TrafficGAN	cGAN+E	InfoGAN	C^3-**GAN**
Traffic speed	RMSE	49.08	36.81	44.52	108.78	104.41	80.94	29.05
	MAPE	6.68	2.92	3.32	62.43	60.76	42.38	2.48
Taxi inflow	RMSE	524.62	415.78	730.66	1579.56	1494.72	1771.25	251.23
	MAPE	65.67	15.38	64.66	670.33	649.39	631.32	14.58

TABLE IV: Performance on task 2: traffic speed and taxi inflow estimation based on travel demand changes.

Methods		cGAN	cGAN+L1	Curb-GAN	TrafficGAN	cGAN+E	InfoGAN	C^3-**GAN**
Traffic speed	RMSE	19.73	60.07	48.90	69.31	23.37	84.17	18.32
	MAPE	3.24	12.42	8.34	39.39	6.62	45.40	2.83
Taxi inflow	RMSE	170.09	300.15	758.76	1364.87	1570.29	1872.08	28.05
	MAPE	10.98	30.96	165.10	430.21	381.00	799.81	5.75

D. Results

1) Overall performance results: In this part, we present the overall performance of our C^3-GAN compared with all baseline models on two different traffic estimation tasks. For all deep models, we train and test all methods five times, and we pick the best trained model and show the testing results including RMSE and MAPE. We have the following observations based on all statistics from both tasks.

First, the overall performance of task 1 is presented in Table III, which includes both traffic speed and taxi inflow estimation based on bus route changes, we find the classic baseline models including cGAN, cGAN+L1, Curb-GAN and TrafficGAN present high testing errors (*i.e.*, high RMSE and high MAPE), which indicates these methods cannot deal with the conditional generation regarding to complex conditions very well. Besides, compared with our C^3-GAN, the two baseline models incluing cGAN+E and InfoGAN still present bad performance, which means it is not enough to only equip cGAN with a simple randomly pre-defined embedding network or an inference network. The performance of task 2 is shown in Table IV which presents similar results.

V. Conclusion

In this paper, we propose a novel Complex-Condition-Controlled Generative Adversarial Network (C^3-GAN) to estimate the regional urban traffic based on complex urban conditions, *e.g.*, new bus routes, rainfall intensity and travel demands. In C^3-GAN, we design *i)* an fixed embedding network to map the complex urban conditions to a latent space and extract representations of complex conditions, and *ii)* an inference network to enhance the relations between the embedded latent vectors and the traffic data. Our experimental results using real-world datasets demonstrate that our C^3-GAN outperforms state-of-the-art baselines in the traffic estimation with complex urban conditions, *e.g.,* bus route planning and travel demands.

Acknowledgment

Yingxue Zhang and Yanhua Li were supported in part by NSF grants IIS-1942680 (CAREER), CNS1952085, CMMI-1831140, and DGE-2021871. 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. Zhenming Liu is supported by NSF grants CRII-1755769, OAC-1835821, III-2008557.

References

[1] C^3-GAN. https://www.dropbox.com/sh/pjfxqev0vr3zpgo/AABEVV1T0WBIaXmPNYOwCH5ca?dl=0, 2020. [Online].

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

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

[4] 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.

[5] P. Chonwiharnphan, P. Thienprapasith, and E. Chuangsuwanich. Generating realistic users using generative adversarial network with recommendation-based embedding. *IEEE Access*, 2020.

[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] 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.

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

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

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

[11] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, et al. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 2015.

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

[13] O. Mogren. C-RNN-GAN: continuous recurrent neural networks with adversarial training. *CoRR*, 2016.

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

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

[16] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez. Invertible conditional gans for image editing. *CoRR*, 2016.

[17] 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.

[18] Z. Yuan, X. Zhou, and T. Yang. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. 2018.

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

[20] 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.