SeqST-GAN: Seq2Seq Generative Adversarial Nets for Multi-step Urban Crowd Flow Prediction

SENZHANG WANG, Nanjing University of Aeronautics and Astronautics & The Hong Kong Polytechnic University, China
JIANNONG CAO, The Hong Kong Polytechnic University, Hong Kong, China
HAO CHEN and HAO PENG, Beihang University, China
ZHIQIU HUANG, Nanjing University of Aeronautics and Astronautics, China

Citywide crowd flow data are ubiquitous nowadays, and forecasting the flow of crowds is of great importance to many real applications such as traffic management and mobility-on-demand (MOD) services. The challenges of accurately predicting urban crowd flows come from both the nonlinear spatial-temporal correlations of the crowd flow data and the complex impact of the external context factors, such as weather, holidays, and POIs. It is even more challenging for most existing one-step prediction models to make an accurate prediction across multiple future time slots. In this article, we propose a sequence-to-sequence (Seq2Seq) Generative Adversarial Nets model named SeqST-GAN to perform multi-step Spatial-Temporal crowd flow prediction of a city. Motivated by the success of GAN in video prediction, we for the first time propose an adversarial learning framework by regarding the citywide crowd flow data in successive time slots as "image frames." Specifically, we first use a Seq2Seq model to generate a sequence of future "frame" predictions based on previous ones. Then, by integrating the generation error with the adversary loss, SeqST-GAN can avoid the blurry prediction issue and make more accurate predictions. To incorporate the external contexts, an external-context gate module called EC-Gate is also proposed to learn region-level representations of the context features. Experiments on two large crowd flow datasets in New York demonstrate that SeqST-GAN improves the prediction performance by a large margin compared with the existing state-of-the-art.

CCS Concepts: • **Information systems** → **Location based services**; **Data stream mining**;

Additional Key Words and Phrases: Generative adversarial nets, crowd flow prediction, deep learning

ACM Reference format:
Senzhang Wang, Jiannong Cao, Hao Chen, Hao Peng, and Zhiqiu Huang. 2020. SeqST-GAN: Seq2Seq Generative Adversarial Nets for Multi-step Urban Crowd Flow Prediction. *ACM Trans. Spatial Algorithms Syst.* 6, 4, Article 22 (June 2020), 24 pages.
https://doi.org/10.1145/3378889

This work is supported by Hong Kong RGC Collaborative Research Fund (Grant No. C5026-18G), Hong Kong Innovation and Technology Fund (ITP/024/18LP), NSF of Jiangsu Province (Grant No. BK20171420), Key Laboratory of Safety-Critical Software (Nanjing University of Aeronautics and Astronautics), Ministry of Industry and Information Technology (Grant No. NJ20170007), Hong Kong Scholar Program, and CCF-Tencent Open Research Fund.

Authors' addresses: S. Wang, Nanjing University of Aeronautics and Astronautics & The Hong Kong Polytechnic University, 29 Jiangjun Rd, Nanjing, Jiangsu, 211106, China; email: szwang@nuaa.edu.cn; J. Cao, PQ819, Mong Man Wai Building, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong; email: jiannong.cao@polyu.edu.hk; H. Chen, Beihang University, 37 Xueyuan Rd, Beijing, China; email: chh@buaa.edu.cn; H. Peng, Beihang University, 37 Xueyuan Rd, Beijing, China; email: penghao@act.buaa.edu.cn; Z. Huang, Nanjing University of Aeronautics and Astronautics, 29 Jiangjun Rod, Nanjing, Jiangsu, 211106, China; email: zqhuang@nuaa.edu.cn.

2374-0353/2020/06-ART22 $15.00
https://doi.org/10.1145/3378889

1 INTRODUCTION

The urban crowd flow data such as taxi trajectory data, sharing bike trip data, subway check-in/out data, and location-based social network (LBSN) data are becoming ubiquitous nowadays. Crowd flow data prediction, which aims to build a fitting model with the historical data to predict their future trend, is of great importance to various location-based services and has attracted increased research interest recently [3]. For example, accurately forecasting the traffic flow of a road network can facilitate a more effective traffic management and help drivers plan their travel routes in advance [9]; the passenger pickup/dropoff demand prediction is of great importance towards better vehicle distribution for the emerging mobility-on-demand (MOD) services [48].

As a hot and practically important research topic, various crowd flow prediction methods have been investigated. The most classical methods are statistics-based time serious analysis models. For example, autoregressive integrated moving average (ARIMA) [21, 31] and Kalman filtering [30] are among the most popular models for urban traffic prediction. However, a major limitation for such statistics-based prediction methods is that they are usually location-specified, and there lacks a unified model to predict all the areas of a city as a whole. Another limitation is that the spatial-temporal correlations of the urban crowd flow data are not fully explored. Some recent works tried to utilize the Bayesian network model to capture the spatial-temporal correlations for predicting the traffic data of a large road network [15, 37, 38]. However, these studies are essentially still statistical-based methods. They cannot effectively capture the complex non-linear spatial-temporal dependency of the crowd flow data, either.

Recently, with the advances of deep learning techniques, deep leaning models such as convolutional neural network (CNN) and recurrent neural network (RNN) have enjoyed considerable success in various machine learning tasks due to their powerful hierarchical feature learning ability and have been widely applied in many areas including computer vision [14], natural language processing [10], recommendation [33], and time series data prediction [43]. This inspires some recent work to adopt deep learning models for various spatial-temporal data prediction tasks. Zhang et al. [45] proposed a deep learning model ST-ResNet to collectively forecast the inflow and outflow of crowds in each region of a city. Yao et al. [41] proposed a Spatial-Temporal Dynamic Network (STDN) model for road network–based traffic prediction. Zhou et al. [48] proposed to use the attention-based neural network, which combined encoder-decoder framework and ConvLSTM to predict the passenger pickup/dropoff demands for the mobility-on-demand services. Cheng et al. [9] proposed the DeepTransport model, which combined CNN and RNN to capture the spatial-temporal traffic data within a transport network. Compared with traditional time series analysis–based prediction models, deep learning models usually can achieve remarkable performance improvement due to their powerful hierarchical feature learning ability from big data.

Although deep learning–based crowd flow prediction models are much more effective than traditional statistics-based models, there are still several unresolved challenging issues that might hinder us from going a further step on this research. First, most previous deep models are essentially one-step prediction [9, 41, 45], which means they focus on making the prediction for the next time slot. There still lacks a multi-step prediction model, which is more useful and challenging in practice. Reference [48] proposed a ConvLSTM-based encoder-decoder framework to predict multi-step citywide passenger demands in mobility-on-demand services. However, the two issues discussed below are still not well addressed. Second, existing deep models for crowd flow prediction suffer from the *blurry prediction* issue [16, 25]: averaging all possible futures into one single. The *blurry prediction* issue is caused by the standard mean squared error used in deep models. To apply CNN to process the data conveniently, the crowd flow data in a city are usually first modeled as image-like matrices by dividing a city into cell regions. Directly applying CNN on the crowd flow matrices cannot address the *blurry prediction* issue. Third, different from images and videos, the crowd flow data can be significantly affected by some external contexts such as weather, holiday, and Points of Interest (POIs) [34]. Although some previous models incorporated the external contexts [45, 48], they considered them as external features and simply concatenated them with the learned latent features from the historical crowd flow data. It is difficult for these models to quantitatively model the region-level impact of the external contexts on a citywide crowd flow prediction and incorporate them for further improving the prediction performance.

To address the above issues, in this article, we propose a sequence-to-sequence deep generative model SeqST-GAN for more effectively performing multi-step urban crowd flow prediction. Generative Adversarial Nets (GAN) is currently a powerful generative framework with an adversarial process [13] and has been widely used in various domains including image generation [11], video prediction [25], and text classification [26]. Previous works [16, 25] showed that adversarial training can effectively address the *blurry prediction* issue in video prediction by introducing the adversarial loss. Motivated by this, we propose an adversarial learning framework for citywide crowd flow prediction. Specifically, we consider the snapshot of the crowd flow data of a city in a time slot as a “frame image” and the crowd flow data in all the time slots as a “video.” Different from SeqGAN [44] that focuses on generating a sequence of discrete tokens like words, SeqST-GAN aims to generate a sequence of crowd flow “frames” that have high dependency in spatial and temporal dimensions. To implement the framework, we propose to apply the sequence-to-sequence (Seq2Seq) model as the generator to perform a multi-step prediction. Seq2Seq model is much more effective in predicting a sequence of future “frame images” due to the usage of attention mechanism and the consideration of the temporal correlations among the predicted “frame images.” To capture the external contexts, we also design an external-context gate module named EC-Gate. Different from previous works [45, 48] that simply concatenate different context features, EC-Gate learns a unified region-level representation for all the contexts. The learned context feature representations are then considered as “gate” to amplify or reduce the generated future “video frames.” We conduct extensive experiments on two large datasets that are widely used in crowd flow prediction, and the results demonstrate the superiority of SeqST-GAN by comparison with existing models.

Our main contributions are summarized as follows:

- We for the first time propose an adversarial learning framework for multi-step urban crowd flow prediction. The framework integrates Seq2Seq prediction model and adversarial learning by combining the prediction error and the adversarial loss to effectively address the blurry prediction issue. The proposed framework is general and can be easily extended and adapted to other prediction tasks in different domains.
- The auto-encoder framework is utilized to implement the Seq2Seq prediction model. By modeling the crowd flow in a city as an “image” and the crowd flow sequence as a “video,” CNN and LSTM components are applied to capture the spatial-temporal correlations of the crowd flow data sequence.
- External context features are also learned in a fine-grained manner through a carefully designed EC-Gate. We perform extensive experiments on two large crowd flow datasets of New York, and the results demonstrate that our proposal improves the prediction result in terms of both single step and multiple steps urban crowd flow prediction by a large margin compared with state-of-the-art models.

The remainder of this article is organized as follows: Section 2 will review related works. In Section 3, we will give a formal definition of the studied problem and show the framework of our solution. Section 4 will introduce our methodology. Evaluations are given in Section 5. Finally, we will conclude this article in Section 6.

2 RELATED WORK

In this section, we will review works that are closely related to ours from the aspects of crowd flow prediction and Adversarial Generative Nets.

Crowd flow prediction. As an important research topic in spatial-temporal data mining, crowd flow prediction including traffic prediction [9, 15, 18, 36, 37], taxi demand-supply prediction [42, 48], individual movement prediction [12, 32], and trajectory prediction [1, 29] have been extensively studied in recent years. Traffic prediction has been studied for many years in both intelligent transportation systems and data mining communities. However, the difference between traffic prediction and our study is that usually traffic prediction focuses on predicting the traffic on the road segments or a road network [15] rather than over cell regions. Traditional traffic prediction models are mostly statistics-based such as ARIMA and SVR models, and they mainly focus on predicting the traffic of one single road or a small set of road segments. Lee et al. [17] and Williams [40] used ARIMA model to predict the short-term traffic flow. Reference [40] showed that SARIMA model was considered to be superior to the neural network models. Mecit and Gurcan [7] put forward an ARIMA prediction model that included two kinds of traffic incident detection algorithms. Experiment result shows that this method is better than the fixed parameter ARIMA models. Chen et al. [8] used the switching ARIMA model to study the change rules of the traffic flow and introduced the turning proportion matrix to describe the traffic flow state of the road network to achieve the accurate prediction of the short-term traffic flow of urban roads. Lippi et al. [20] compared SVR model and SARIMA model and concluded that the proposed seasonal SVR model is in fierce competition during the most crowded period of prediction. The major limitation of the above models for road segment–level traffic prediction is that the correlations among the road segments are largely ignored or not fully explored, and the learned parameters of a road segment cannot be generalized to other roads. Some recent works tried to utilize the Bayesian network model to capture the spatial-temporal correlations for predicting the traffic data of a large road network [15, 37]. However, these studies are essentially still statistical-based methods. They cannot effectively capture the complex and non-linear spatial-temporal dependency of the crowd flow data, either. For example, References [37, 39] only considered the correlations among the neighbor road links, but ignored the impact of traffic on a road link from farther road links.

Recently, with the advances of deep learning techniques, deep leaning models such as convolutional neural network (CNN) and recurrent neural network (RNN) have enjoyed considerable success in various machine learning tasks [35]. Due to the powerful hierarchical feature learning ability of deep learning models in both spatial and temporal domains, various deep learning models are proposed for spatial-temporal data mining tasks. A line of studies applied CNN to capture the spatial correlation by treating the traffic data of the entire city as images. Ma et al. [22] utilized CNN on images of traffic speed for the speed prediction problem. Zhang et al. [45, 46] proposed to use residual CNN on the images of traffic flow. These methods simply use CNN on the whole city and use all the regions for prediction. The major limitation of these methods is that although they used historical traffic images in previous time slots for prediction, they did not explicitly model the temporal sequential dependency. Another limitation is that as they used CNN to model the traffic images, the *blurry prediction* issue that existed in image prediction cannot be addressed. Another line of research is combining CNN model and RNN model to capture both spatial and temporal correlations. Yao et al. [41] proposed a Spatial-Temporal Dynamic Network (STDN) model for road network–based traffic prediction. Cheng et al. [9] proposed the DeepTransport model, which combined CNN and RNN to capture the spatial-temporal traffic data within a transport network. All these models are basically designed for one-step prediction, which means they focus on predicting the traffic data in the next time slot, such as 20 minutes or half an hour. Reference [48] was the first recent work that studied the problem of multi-step taxi passenger demand prediction. [48] proposed to use the attention-based neural network, which combined encoder-decoder framework and ConvLSTM to predict the passenger pickup/dropoff demands for the mobility-on-demand services. [18] proposed the Diffusion Convolutional Recurrent Neural Network (DCRNN) to model the traffic flow as a diffusion process on a directed road graph, which is a deep learning framework for traffic forecasting that incorporates both spatial and temporal dependency in the traffic flow. Following this work, Reference [47] also proposed to combine GraphCNN and seq2seq model for road network–level traffic flow prediction. However, there still lacks an effective and general model that can accurately perform a multi-step crowd flow prediction and at the same time avoid the *blurry prediction* issue.

Generative Adversarial Nets. GAN is a powerful generative framework with an adversarial process [13] proposed recently and has been widely used in various domains including image generation [11], video prediction [25], and text classification [26]. The general idea of GAN is that it simultaneously trains two models: a generative model G that captures the data distribution and a discriminative model D that estimates the probability that a sample comes from the training data rather than G. This framework corresponds to a minimax two-player game. A major issue of the initial GAN model is that its training is rather unstable and prone to appear mode collapse. To address this issue, many improved GAN models are proposed, such as WGAN [2], DCGAN [28], and LSGAN [23]. Although GAN is initially proposed for samples generation, recently a bunch of models employed the adversarial leaning framework for video prediction and achieved promising performance [25]. In Reference [25], several loss functions including the adversarial training loss, the standard mean squared error (MSE) loss, and image gradient difference loss are integrated to achieve more promising prediction performance. Particularly, Reference [25] showed that generative adversarial training can be successfully employed for next frame prediction and helped preserve the sharpness of the frames by addressing the *blurry prediction* issue. Following the pioneer work, several works are done to further improve the performance of video prediction under the adversarial learning framework. Liang et al. [19] developed a dual motion GAN architecture that learned to explicitly enforce future-frame predictions to be consistent with the pixel-wise flows in the video through a dual-learning mechanism. Bhattacharjee and Das [6] proposed a multi-stage GAN framework for future frames prediction. Their method used a two-stage GAN to generate a crisp and clear set of future frames. Motivated by the successful application of GAN model in video prediction, in this article, we try to use it for crowd flow prediction, where the crowd flow data of a city in successive time intervals can be considered as "frames" in a video. However, different from the above works for video prediction, the model for crowd flow prediction not only needs to capture the high spatial-temporal correlations of the crowd flow data, but also needs to consider the complex impact of external context features such as weather, holidays, and social events on the crowd flow data.

3 PROBLEM DEFINITION

In this section, we will first give some definitions to help us state the studied problem and then will give a formal problem definition.

Definition 1. Cell Region. In this study, we partition a city into an $m \times n$ grid map based on the longitude and latitude. Each grid is defined as a cell region, and all the grids form a cell region set $R = \{r_{1,1}, \ldots r_{i,j}, \ldots r_{m\times n}\}$, where $r_{i,j}$ is the cell region in the *ith* row and *jth* column of the grid map.

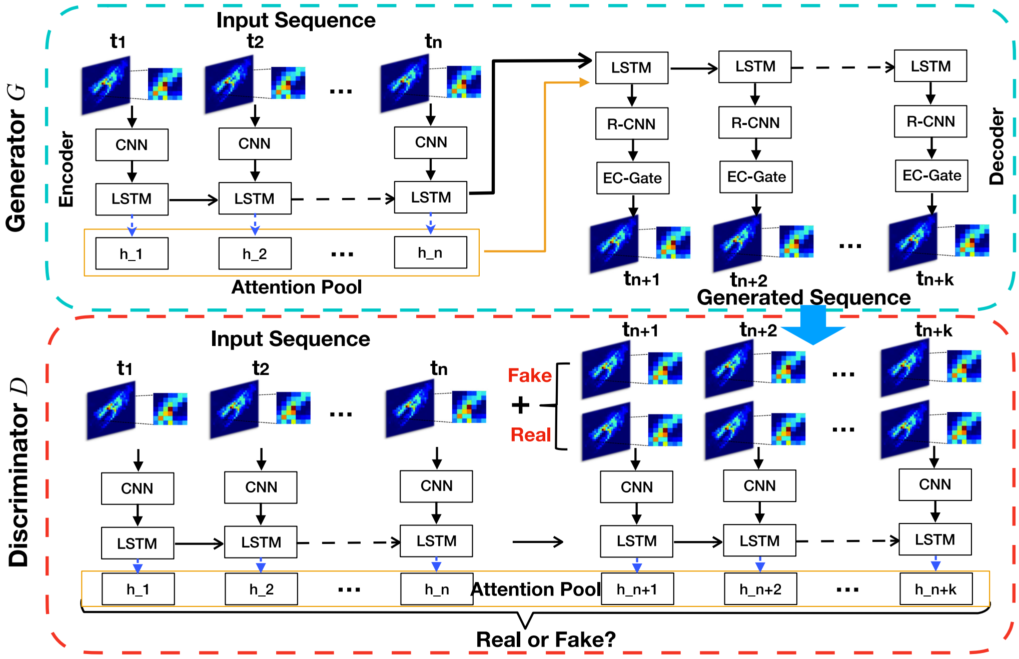


Fig. 1. The framework of SeqST-GAN model.

Definition 2. Inflow/Outflow [46]. Let $\mathcal{P}$ be a collection of crowd flow trajectories at the time slot t. For a cell region $r_{i,j}$, the inflow and outflow of the crowds at t are defined, respectively, as

$$x^t_{i,j,in} = \sum_{Tr \in \mathcal{P}} |\{k > 1 | g_{k-1} \notin r_{i,j} \wedge g_k \in r_{i,j}\}|,$$

$$x^t_{i,j,out} = \sum_{Tr \in \mathcal{P}} |\{k \geq 1 | g_k \in r_{i,j} \wedge g_{k+1} \notin r_{i,j}\}|,$$

where $Tr : g_1 \rightarrow g_2 \rightarrow \cdots \rightarrow g_{Tr}$ is a trajectory in $\mathcal{P}$, and g_k is the geospatial coordinate; $g_k \in r_{i,j}$ means g_k lies within the region $r_{i,j}$ and vice versa; $|\cdot|$ denotes the cardinality of a set.

Following previous works [45, 46], we denote the inflow and outflow in all the cell regions in time slot t as a crowd flow tensor $\mathbf{X}^t \in \mathcal{R}^{m \times n \times 2}$, where $(\mathbf{X}^t)_{i,j,0} = x^t_{i,j,in}$, $(\mathbf{X}^t)_{i,j,1} = x^t_{i,j,out}$. Based on the above definitions, we give a formal definition of the studied problem as follows:

Problem Definition 1. ***Multi-step Crowd Flow Prediction***. *Given the crowd flow tensors* $\{\mathbf{X}^t | t = 1, \ldots n\}$ *in the cell regions R over the previous n time slots, our goal is to predict the crowd flow tensors* $\{\mathbf{X}^t | t = n+1, \ldots n+k\}$ *for the next k time slots simultaneously.*

4 METHODOLOGY

In this section, we will introduce the proposed methodology in detail. We will first briefly introduce the framework of SeqST-GAN. Then, we will introduce the overall objective of SeqST-GAN. Next, we will elaborate on the generator of SeqST-GAN that is implemented as a seq2seq model, and its discriminator, to build up the adversarial learning framework.

Figure 1 shows the training framework of SeqST-GAN. One can see that SeqST-GAN contains two parts: the generator G and the discriminator D. For the generator G, we propose an encoder-decoder–based Seq2Seq framework to generate the future k crowd flow tensor sequences

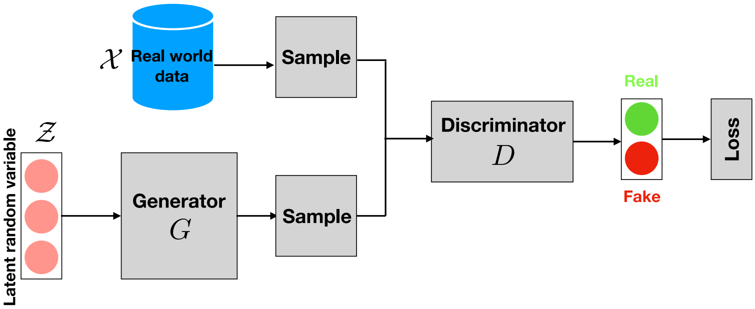


Fig. 2. Illustration of Generative Adversarial Nets (GAN).

$\{\hat{\mathbf{X}}^t | t = t_{n+1}, \ldots t_{n+k}\}$, and the input is the historical data $\{\mathbf{X}^t | t = t_1, \ldots t_n\}$. In this framework, the input historical crowd flow tensors are first encoded into a latent space vector with a CNN layer and an LSTM layer, and then a Seq2Seq attention is applied to capture the weights of the tensors in different time slots for predicting the future tensors. Next a decoder is used to decode the latent space vector and generate the future crowd flow tensors. The decoder contains an LSTM layer and a deconvolutional neural network R-CNN to transform a vector back to a crow flow tensor. Note that a external-context gate called EC-Gate is also designed to capture the external contexts such as weather, date, and POIs. For the discriminator D, the generated tensors and the real tensors are both concatenated with the previous crowd flow tensors and form a pair of inputs $\hat{\mathcal{X}} = \{\mathbf{X}^t, \hat{\mathbf{X}}^{\hat{t}} | t = t_1, \ldots t_n, \hat{t} = t_{n+1}, \ldots t_{n+k}\}$, $\mathcal{X} = \{\mathbf{X}^t | t = t_1, \ldots t_{n+k}\}$ for D. D tries to distinguish $\{\hat{\mathcal{X}}, \mathcal{X}\}$ which one is real and which one is generated by G. The generator G and discriminator D are trained iteratively, and finally the generated data sequence by G is so similar to the real one such that D cannot distinguish them. Thus, the generated data sequence can be used as the prediction. Next, we will introduce SeqST-GAN in detail.

4.1 The Overall Objective of SeqST-GAN

Before elaborating on our model, we first briefly introduce GAN [13]. GAN is a framework for estimating generative models via an adversarial process. As shown in Figure 2, GAN contains two components: a generative model G and a discriminative model D. It simultaneously trains the generative model G to capture the data distribution and the discriminative mode D to estimate the probability that a sample comes from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. Thus, GAN can be considered as a minimax two-player game whose objective function is as follows:

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

where $\mathcal{X}$ is a real data sample (the crowd flow tensor in our case), $p_{data}(\mathcal{X})$ is the real data distribution of $\mathcal{X}$, $D(\mathcal{X})$ represents the probability that $\mathcal{X}$ comes from the data distribution, $\mathcal{Z}$ is random noise, $p_{\mathcal{Z}}(\mathcal{Z})$ is the prior distribution on input noise $\mathcal{Z}$, $G(\mathcal{Z})$ is a generator that generates "fake" data from $\mathcal{Z}$, and $D(G(\mathcal{Z}))$ represents the probability that the generated "fake" sample comes from the data distribution. The goal of the objective function is to maximize the probability of assigning the correct label to both real and fake samples from G and minimize $log(1 - D(G(\mathcal{Z})))$.

In this article, instead of using the initial GAN, we utilize the Wasserstein GAN (WGAN) due to the following advantages [2]: First, training WGAN does not require maintaining a careful balance in training the discriminator and the generator, and thus solves the mode collapse problem. Second, it does not require a careful design of the network architecture. Third, WGAN can improve the stability of training, and thus is much easier to train. The major difference between WGAN and initial GAN is that the Wasserstein distance is used to measure the distance and divergence between the real data distribution and the learned data distribution by the generator.

By using WGAN as our adversarial learning framework, the objective function is as follows:

$$\min_{\theta} \max_{w} \mathbb{E}_{\mathcal{X} \sim p_{data}}[f_w(\mathcal{X})] - \mathbb{E}_{\mathcal{Z} \sim p(\mathcal{Z})}[f_w(g_\theta(\mathcal{Z}))], \tag{2}$$

where w is the critic parameters, θ is the generator's parameters, and $f_w(\mathcal{X})$ is a parameterized family of functions $\{f_w\}_{w \in \mathcal{W}}$ that are all K-Lipschitz for K.

Given N pairs of real and generated crowd flow tensors $\{(\mathcal{X}_i, \hat{\mathcal{X}}_i)\}_{i=1}^{N}$, the objective function under WGAN framework can be rewritten as follows:

$$\min_{\theta} \max_{w} \sum_{i=1}^{N}(f_w(\mathcal{X}_i)) - \sum_{i=1}^{N}(f_w(g_\theta(\hat{\mathcal{X}}_i))). \tag{3}$$

Note that in the multi-step crowd flow prediction task, the generated sequence $\{\hat{\mathbf{X}}^t | t = t_{n+1}, \ldots t_{n+k}\}$ by G is conditioned on the input sequence $\{\mathbf{X}^t | t = t_1, \ldots t_n\}$ as shown in the generator in Figure 1. Thus, in our model there is variability in the output of the generator even without noise, which means noise is not necessary anymore in the prediction model [25]. In addition, GAN and WGAN are both designed for new data generation rather than future data prediction. To make the generated future crowd flow tensors follow the real data distribution, and at the same time they are close to the real ones as much as possible, we add a mean square error loss to the generator as follows:

$$\mathcal{L}_p(G(\hat{\mathcal{X}}_i), \mathcal{X}_i) = ||\mathcal{X}_i - \hat{\mathcal{X}}_i||_p^p. \tag{4}$$

Here, we use the Euclidean norm and set $p = 2$. By combining Equations (3) and (4), the final objective function of SeqST-GAN is as follows:

$$\min_{\theta} \max_{w} \sum_{i=1}^{N}[f_w(\mathcal{X}_i) - f_w(g_\theta(\hat{\mathcal{X}}_i))] + \lambda \sum_{i=1}^{N} ||\mathcal{X}_i - \hat{\mathcal{X}}_i||_2, \tag{5}$$

where λ is a parameter to balance the importance of the adversary loss and the mean square error. By combining the adversary loss and the mean square error in Equation (5), the *blurry prediction* issue can be well addressed, because an average of several possible future predictions optimized by the mean square error will increase the adversary loss. Therefore, given a sequence of historical crowd flow data, if there are several possible future trends, the adversarial loss will help the model select the most likely future prediction rather than simply averaging them.

Model optimization. To find the function f in the objective function (5), usually a neural network parameterized with weights w is trained to approximate f. Similarly, another neural network parameterized with weights θ is trained to approximate the generator function g. The two neural networks will be introduced in detail in the next section. To train the objective function, the standard minibatch stochastic gradient descent training method can be applied.

The stochastic gradients of the generator G and discriminator D are as follows:

$$\nabla_\theta \left\{ -\frac{1}{m} \sum_{i=1}^{m} f_w(g_\theta(\hat{\mathcal{X}}_i) + \frac{\lambda}{m} \sum_{i=1}^{m} ||\mathcal{X}_i - \hat{\mathcal{X}}_i||_2) \right\}, \tag{6}$$

$$\nabla_w \left\{ \frac{1}{m} \sum_{i=1}^{m} f_w(\mathcal{X}_i) - \frac{1}{m} \sum_{i=1}^{m} f_w(g_\theta(\hat{\mathcal{X}}_i)) \right\}, \tag{7}$$

where m is the batch size in minibatch optimization. We follow the standard training process of WGAN. The procedure of the SeqST-GAN algorithm is shown in Algorithm 1. Note that *RMSProp*

ALGORITHM 1: Training of SeqST-GAN

```
Input: The learning rate $\alpha = 0.001$, the clipping parameter $c = 0.01$, the batch size $m = 32$, the number of
        iterations of the critic per generator iteration $n_{critic}$, the historical crowd flow tensor
        $\{\mathcal{X}^t | t = 1, \dots n\}$
Output: The model parameters $w$ and $\theta$.
while the algorithm does not converge do
    for $t = 0, \dots, n_{critic}$ do
        Sample $\{\mathcal{X}_i\}_{i=1}^m \sim p_{data}$ a batch from the real data.
        Sample $\{\hat{\mathcal{X}}_i\}_{i=1}^m \sim p_z$ a batch from the generated data with the seq2seq model.
        $g_w \leftarrow \nabla_w \{\frac{1}{m} \sum_{i=1}^m f_w(\mathcal{X}_i) - \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(\hat{\mathcal{X}}_i))$
        $w \leftarrow w + \alpha \cdot RMSProp(w, g_w)$
    end
    Sample $\{\hat{\mathcal{X}}^{(i)}\}_{i=1}^m \sim p_z$ a batch from the generated data with the seq2seq model.
    $g_\theta \leftarrow -\nabla_\theta \{-\frac{1}{m} \sum_{i=1}^m f_w(g_\theta(\hat{\mathcal{X}}_i) + \frac{\lambda}{m} \sum_{i=1}^m ||\mathcal{X}_i - \hat{\mathcal{X}}_i||_2\}$
    $\theta \leftarrow \theta - \alpha \cdot RMSProp(\theta, g_\theta)$
end
```

optimization method is used in SeqST-GAN, which is known to perform well even on very non-stationary problems.

4.2 Generator G: Seq2Seq-based Encoder-decoder for Multi-step Prediction

In this subsection, we introduce how we build the generator G. Seq2Seq was initially designed for machine translation [5]. Due to its advantage in handling sequential data, currently it has also been widely applied in many other applications, such as event prediction [27] and human motion prediction [24]. Given a word sequence, Seq2Seq model can align it with another word sequence or predict the following word sequence through an encoder-decoder learning framework. In our work, we aim to use a sequence of historical crowd flow tensors $\{\mathbf{X}^t | t = t_1, \dots t_n\}$ to predict a sequence of future tensors $\{\hat{\mathbf{X}}^t | t = t_{n+1}, \dots t_{n+k}\}$, whose scenario is similar to machine translation or sentence prediction. Motivated by this idea, we design a Seq2Seq-based encoder-decoder model as the generator G to generate a sequence of future crowd flow data for the multi-step prediction, as shown in the upper part of Figure 1.

The architecture of the Seq2Seq-based G is designed as follows: It contains the encoder part (the left part of the generator G in Figure 1) and the decoder part (the right part of the generator G in Figure 1). The encoder learns compact vector representations of the input sequential crowd flow tensors, while the decoder tries to generate a sequence of future tensors with the learned compact vector representations. In the encoder, the input tensor sequence $\{\mathbf{X}^t | t = t_1, \dots t_n\}$ is first fed into a CNN layer to learn the spatial features. Then, the outputs of the CNN layer are input into the LSTM layer to learn the temporal dependency among the crowd flow tensors in different time slots.

Note that the crowd flow tensors in different time slots contribute differently to the prediction of the future. For example, to predict the traffic flow in 3:00pm, the traffic flow in the last hour 2:00pm is more important than that in the hour 10:00am. To take this into account, we also propose to add an attention mechanism to the encoder. Attention mechanism is widely used in natural language processing [5] and imaging processing [4]. It allows the decoder to attend to different parts of the input data sequence at each step of the output generation. Such a Seq2Seq model with attention mechanism makes our multi-step crowd flow prediction more accurate compared with previous deep learning models [42, 45]. The Seq2Seq decoder model with attention mechanism is illustrated in Figure 3.

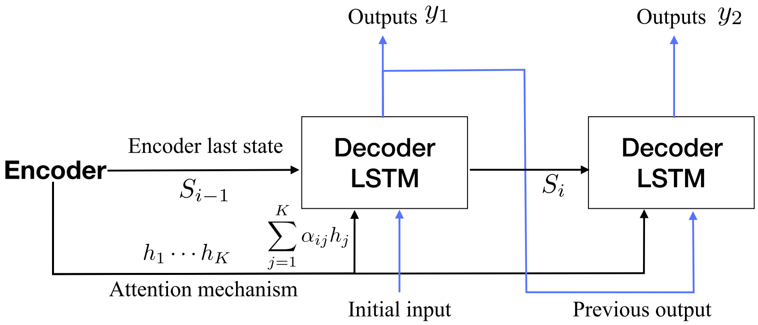


Fig. 3. Illustration of the Seq2Seq model with attention.

As shown in Figure 3, given the input crowd flow tensors $\{\mathbf{X}^t | t = t_1, \dots t_n\}$ and the previous output $\{y^1, \dots y^{i-1}\}$ of the LSTM layer in the decoder, the i-step prediction of y^i can be calculated by

$$p(y_i | y_1, \dots y_{i-1}, En(\mathbf{X}^1, \dots \mathbf{X}^{t_n})) = g(y_{i-1}, s_i, c_i), \tag{8}$$

where $En(\mathbf{X}^1, \dots \mathbf{X}^{t_n})$ is the output of the encoder, c_i is a context vector, and s_i is the hidden state of LSTM for time i, which is computed by

$$s_i = f(s_{i-1}, y_{i-1}, c_i), \tag{9}$$

$$c_i = \sum_{j=1}^{n} \alpha_{ij} h_j. \tag{10}$$

The weight α_{ij} of each annotation h_j is computed by

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})}, \tag{11}$$

where $e_{ij} = a(s_{i-1}, h_j)$ is an alignment model that scores how well the input around position j and the output at position i match. The score is based on the LSTM hidden state s_{i-1} and the jth annotation h_i of the input crowd flow tensor sequence $\{\mathbf{X}^t | t = t_1, \dots t_n\}$. The output of the LSTM layer in the decoder is $\{y_1, \dots y_k\}$, and then it is input into a deconvolution layer R-CNN to generate the tensors $\hat{\mathbf{X}}^t_{LSTM}$ with the same shape as the input $\mathbf{X}^t$.

EC-Gate to learn the external context features. Note that the crowd flow data such as traffic data and human mobility data can be remarkably affected by many external contexts such as weather, POIs, social events, and holidays. To take the external contexts into account, some previous works [45, 48] extracted external context features and concatenated them with the features of the crowd flow data. The limitation of such methods is that they do not distinguish the different impacts of the external features to different regions. In this article, we design a gate called EC-Gate to learn a fine-grained representation of the external contexts for each region in each time slot. Instead of simply concatenating the external context features, the proposed EC-Gate can learn a unified impact weight of all the external context features for each region in each time slot. The structure of EC-Gate is shown in Figure 4. Three types of external context features are extracted, *weather*, *day & hour*, and *POIs*. The extracted context features form a external feature tensor whose three dimensions are cell region *id*, cell region *id*, and feature type. The external feature tensor is then input into a CNN model. Note that, in the CNN model, we do not use the pooling layer to keep the shape of the output the same as the input. The final output of CNN is two matrices. One corresponds to the inflow gate and the other corresponds to the outflow gate. By incorporating

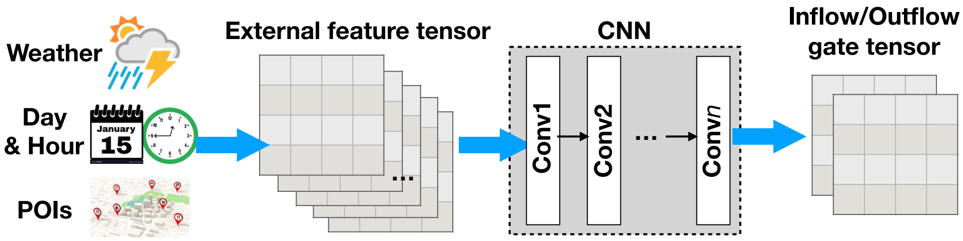


Fig. 4. ST-Gate to learn external context features.

EC-Gate, the final output of the decoder can be calculated by

$$\hat{\mathbf{X}}^t = \hat{\mathbf{X}}^t_{LSTM} \odot \mathbf{G}^t, \tag{12}$$

where $\hat{\mathbf{X}}^t_{LSTM}$ is the output of LSTM layer, $\mathbf{G}^t$ is the normalized output of the EC-Gate, and $\odot$ is the Hadamard product. Note that EC-Gate is jointly learned with SeqST-GAN, and thus the parameters of EC-Gate are updated together with the parameters of SeqST-GAN in each iteration to learn the region-level impact weights of the external contexts.

4.3 Discriminator *D*

Given a pair of real and generated crowd flow tensors $(\mathcal{X}, \hat{\mathcal{X}})$, the discriminator D tries to distinguish which one is real and which one is generated. The prediction error of discriminator D will be back-propagated to the generator G to guide it to generate more real data. We also design the discriminator as a deep neural network, and the architecture of D is shown in the lower part of Figure 1. First, the observed previous crowd flow tensors $\{\mathbf{X}^t | t = t_1, \dots t_n\}$ are merged with the real future tensors $\{\mathbf{X}^t | t = t_{n+1}, \dots t_{n+k}\}$ and the generated tensors $\{\hat{\mathbf{X}}^t | t = t_{n+1}, \dots t_{n+k}\}$ to form a pair of training samples $(\mathcal{X}, \hat{\mathcal{X}})$. Then, $(\mathcal{X}, \hat{\mathcal{X}})$ is in turn input into a CNN layer and an LSTM layer to learn the spatial and temporal latent features. Next the attention mechanism is also applied on the output of LSTM. Note that the D in WGAN uses a new loss function derived from the Wasserstein distance, and thus no logarithm is needed anymore. D here does not play as a direct critic but a helper for estimating the Wasserstein distance metric between the real and the generated data distributions. Thus, the *Sigmoid* layer is not needed in the final layer.

5 EVALUATION

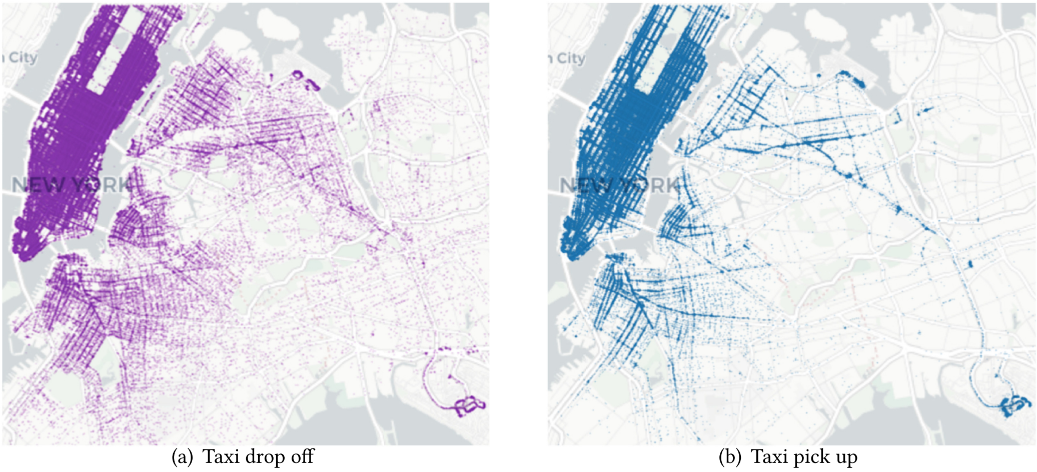
In this section, we will conduct extensive experiments to evaluate the proposed SeqST-GAN on two large urban crowd flow datasets: the taxi trip dataset and the bike trip dataset in New York. We will first introduce the datasets, the baselines, and the experiment settings. Then, we will show the experiment results on both single-step and multi-step prediction on the two datasets. Next, we will perform parameter study to show the effectiveness of SeqST-GAN in addressing the *blurry prediction* issue. Finally, we will evaluate the effectiveness of the proposed EC-Gate in external context features learning.

5.1 Dataset

The details of the two datasets used for evaluation are described as follows:

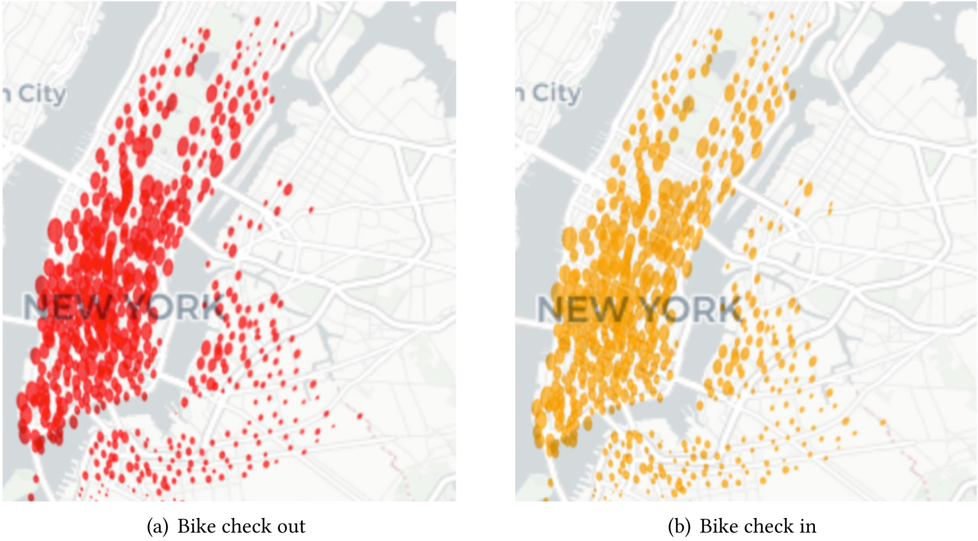
- **TaxiNYC dataset.**[1] TaxiNYC dataset contains 1.19B taxicab trip records in New York from January 2009 to December 2015. On average, there are 170M trip records collected in each year. Each taxi trip record includes fields capturing pickup and dropoff dates/times, pickup and dropoff locations, trip distances, itemized fares, rate types, payment types, and

[1] http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.



(a) Taxi drop off (b) Taxi pick up

Fig. 5. The heat maps of the TaxiNYC dataset.



(a) Bike check out (b) Bike check in

Fig. 6. The heat maps of the CitiBikeNYC dataset.

driver-reported passenger counts. We use four-year records from 2009 to 2013 as training data, while the records in 2014 and 2015 are used as validation and test sets, respectively.

- **CitiBikeNYC dataset.**[2] CitiBikeNYC dataset contains more than 28M bike trips in NewYork from July 2013 to June 2016. In total, CitiBike has established over 600 stations and 10K bikes in New York. Each bike trip contains the trip duration, start/end station IDs, start/end timestamps, station Lat/Long, and bike ID. We use the data from 2013 to 2015 as training and validation data and the data in 2016 as testing data.

Figure 5 shows the heat maps of the dropoff and pickup locations of the taxis in New York, and Figure 6 shows the heat maps of the check-out and check-in locations of the CitiBike bikes in New

[2] https://www.citibikenyc.com/system-data.

Table 1. Statistics of the Datasets

Dataset	TaxiNYC	CitiBikeNYC
Data type	Taxi GPS trip	Bike rent trip
Time Span	Jan. 2009–Dec. 2015	July 2013–June 2016
# of trips	1.19B	28M
Grid map size	(64, 64)	(16, 16)
Time interval	1 hour	1 hour
Point Of Interest (POI) data		
# of POIs	18,912	
Types of POIs	residential(16.7%), education(20%), culture(3%), transportation(6.1%), social services(8.7%), recreational(17.2%), commercial(5.5%), government(4.5%), religious institution(8.4%), water(1.6%), public safety(3.3%), health services(1.5%), miscellaneous(3.5%)	
Weather data		
Weather	rainy, snowy, sunny, visibility, etc.	
Temperature	[−30°C, 40°C]	
Wind speed	[0, 50mph]	

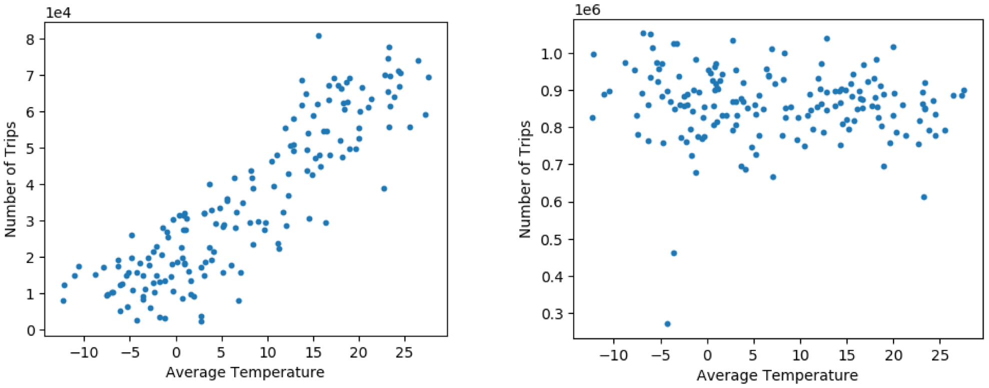
York. One can see the two datasets are both not evenly distributed in New York. There are a large number of taxi and bike trips in Manhattan, but the data are sparse in other areas of New York. As the locations of bike stations in New York are fixed and the bike trips can be only from one bike station to another, the covering areas of the bike trips is much smaller than the areas covered by the taxi trips. Note that in some areas there are no bike or taxi flow data at all, such as the ocean areas. Thus, although we still include such areas in our grid maps, we do not evaluate the prediction results for these regions, as the values are always zero.

We also use the publicly available POI data of New York.[3] In this dataset, there are 18,912 POIs in total, and it includes the POIs of the following facility domains: *residential*, *education facility*, *culture facility*, *recreational facility*, *social services*, *transportation facility*, *commercial*, *government facility*, *religious institution*, *health services*, *public safety*, *water,* and *miscellaneous*. We also use weather information, including the weather conditions (rainy, snowy, sunny, etc.), temperature, wind speed, and so on. The detailed statistical information of the two datasets, POI dataset, and the used weather information are given in Table 1.

5.2 Data Analysis

In this subsection, we conduct data analysis on the two datasets to show how the external factors affect the crowd flow of taxis and bikes. Figure 7(a) and Figure 7(b) show the number of bike trips and taxi trips varying with different average temperatures of a day, respectively. Each point in the figure represents the trip number of a day under a particular average temperature. One can see that with the increase of the temperature, the usage of bikes presents a significant increase trend. However, the effect of temperature on the usage of taxis is less significant than that of bikes, and the number of taxi trips does not increase or decrease remarkably with the change of temperature.

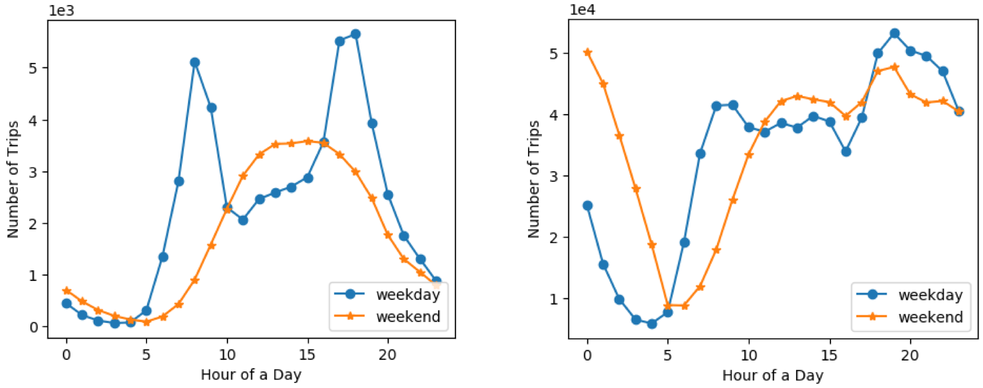
[3] https://data.cityofnewyork.us/City-Government/Points-Of-Interest/rxuy-2muj/data.



(a) Bike trip number vs. temperature

(b) Taxi trip number vs. temperature

Fig. 7. The effect of temperature to the trips of bikes and taxis.



(a) Bike trip number vs. hour

(b) Taxi trip number vs. hour

Fig. 8. Trip numbers in different hours of a day.

This is reasonable, because in cold days people will be less likely to take a bike for travel, and they might choose to take taxis, buses, or subways instead. For the taxi trips, however, cold weather will not decrease the possibility of people choosing taxi for traveling. On the contrary, people may be more likely to take a taxi when there is extreme weather. From this figure, one can see that the external factor of weather does have impact on the crowd flow data, but for different types of data the impact can be significantly different.

We next investigate how the usage of bikes and taxis varies in different hours of a day. Figure 8(a) and Figure 8(b) show the results. As people's travel patterns can be quite different on weekdays and weekends, we show the trip number curves in weekday and weekend for each dataset, respectively. One can see that, for the bike trip data, there are two remarkable peaks in the rush hours of weekdays. This is probably because on weekdays people tend to take bike to their work places or to the subway station for transferring. On weekends, although there are no significant peaks in a day, one can see that the largest number of bike trips appears in the interval from about 11:00am to 4:00pm. This is mainly because on weekends people take bikes for tour rather than for work. For the taxi trip dataset, there are also peaks for the curve on weekday, but they are not as large as that of the bikes. Overall, the usage of taxis in the daytime is always large, no matter on weekdays or weekends. This is probably because most people do not choose to take taxi to work and thus there is no big difference between the two curves of workdays and non-workdays. One can also see that a big difference between the usage patterns of bikes and taxis is that their curves in the night are quite different. In the night from 8:00pm to 12:00am (midnight), the bike trip number is always small, but the taxi trip number is still large. This is mainly because, in the late night, people are more likely to take a taxi home due to safety issues.

Both data analysis results show that external factors including weather, hours of a day, and holidays can affect the crowd flow data, and thus they should be carefully considered when building a mode for predicting the future trend of the crowd flow.

5.3 Baselines and Evaluation Metrics

We compare our model with the following baselines, including both traditional statistics-based models and the recent state-of-the-art deep learning models:

- **Historical Average (HA):** The historical average crowd flows are used as the prediction of the corresponding future crowd flows. For example, we use the average crowd flow in the past one year in the time slot 10:00am–11:00am as the prediction of the crowd flow in the same time slot.
- **ARIMA:** Auto-Regressive Integrated Moving Average (ARIMA) is a widely used regression model for time series data prediction. The spatial correlation among the grid regions is not explored in this model.
- **Ridge regression:** Ridge regression is also a widely used linear regression model for time-series data forecasting. Here, we use an l_2-norm regularization of Ridge regression.
- **ST-ResNet [45]:** ST-ResNet is a deep spatial-temporal residual network model for one-step crowd flow prediction. In this model, a residual netural network framework is proposed to model the temporal closeness, period, and trend properties of crowd traffic. The external context features considered in ST-ResNet include weather, holiday events, and day of a week. Two fully connected layers are stacked to learn the latent features for the external contexts, and then the latent features are fused with the crowd flow latent features through a *tanh* function.
- **AttConvLSTM [48]:** AttConvLSTM is a state-of-the-art model for multi-step passenger demands prediction in the mobility-on-demand services. It employs an encoder-decoder framework based on convolutional LSTM to capture the spatial-temporal features.
- **DMVST-Net [42]:** DMVST-Net considers the spatial and temporal relations as two-view data. It is a deep multi-view spatial-temporal network model to incorporate the temporal view, spatial view, and the semantic view for one-step taxi demand prediction. DMVST-Net also considers external context features including weather conditions and holidays. Such features are directly concatenated with the learned latent features of the taxi demand by CNN.
- **Seq2Seq:** To study whether the GAN framework is more effective, we also compare with the Seq2Seq model, which is used as the generator of SeqST-GAN.
- **SeqST-GAN-Con:** To study whether the proposed ST-gate can more effectively learn the external context features, we compare with SeqST-GAN-Con, which is also a variant of SeqST-GAN. SeqST-GAN-Con directly concatenates the latent features learned from the external factors through an MLP structure with the learned latent vectors of the encoder.

We use the widely adopted Mean Average Error (MAE) and Rooted Mean Square Error (RMSE) defined as follows as the evaluation metrics:

$$MAE = \frac{1}{N * k} \sum_{i=1}^{N} \sum_{t=n+1}^{n+k} |\hat{y}_t^i - y_t^i|,$$

$$RMSE = \sqrt{\frac{1}{N * k} \sum_{i=1}^{N} \sum_{t=n+1}^{n+k} (\hat{y}_t^i - y_t^i)^2},$$

where N is the number of testing samples, $\hat{y}_t^i$ is the prediction of sample i at time slot t, and y_t^i is the ground truth.

5.4 Experiment Results

Parameters Setting. We implement our model with Tensorflow framework on a 2×GTX 1080Ti GPU. The parameters in the model are set as follows: The input data size is 64×64×2 for taxi data, and 16×16×2 for bike data, where 64 and 16 are the row sizes of the divided cell regions, 2 is the number of channels. The previous time slot length n is set to 24, which means that we use the crowd flow data of the previous one whole data for prediction. The learning rate α is set to 0.001. Batch size m is set to 32. The CNN model of TaxiNYC contains 4 layers whose structure is 64×64×8, 32×32×16, 16×16×16, and 8×8×32. The CNN model of CitiBikeNYC also contains 4 layers whose structure is 16×16×16, 8×8×32, 4×4×64, and 2×2×128. We use 8 kernels, and the kernel size is 5×5. R-CNN has the reversed structure corresponding to the CNN. The output of CNN is flattened to a 512-dimensional vector. The size of LSTM's hidden state is set to 200.

The parameters of baseline methods are set based on the original papers. Following Reference [45], convolutions of *Conv1* and all residual units in ST-ResNet model use 64 filters of size 3×3, and *Conv2* uses a convolution with 2 filters of size 3×3. The batch size is 32. The hyperparamer p in ST-ResNet is fixed to one-day, and q is fixed to one-week. Following Reference [48], the parameters of the AttConvLSTM model are set as follows: The encoder consists of 2-layer CNN and 2-layer ConvLSTM. The kernel size of the 2-layer CNN is set to 3×3 with a stride of 2 and the number of features is set to 8 for the lower layer and 16 for the higher layer. For the 2-layer ConvLSTM, the kernel size of all convolutional operations is 3×3 with a stride of 1, and both layers contain 64 hidden states for each grid. The decoder is composed of 2-layer ConvLSTM and 2-layer DCNN, with the same configuration as encoder. For the attention model, the number of nodes in the single hidden layer of MLP is set to 1,024. For model training, we adopt the mini-batch learning method with a batch size of 16. Adam optimizer is used with a constant learning rate of 0.0002. Following Reference [42], the parameters of DMVST-Net model are set as follows: The number of layers of the spatial view is set to 3, the kernel size is set to 3×3, number of kernels is set to 64, and the dimension of the output is set to 64. For the temporal view, the sequence length is set to 8 for LSTM, and the output dimension of graph embedding is set to 32. The output dimension for the semantic view is set to 6. The batch size during model training is set to 64, and early-stop is used in the experiment.

Convergence Analysis. As GAN is generally hard to train, we first study whether the proposed SeqST-GAN can converge quickly. Figure 9 shows the training loss curves of generator G and discriminator D of SeqST-GAN on TaxiNYC dataset. One can see that the losses become stable when the training epoch reaches 400. It shows that SeqST-GAN can converge quickly. This is mainly because we use WGAN instead of the original GAN, and WGAN is more stable in training. For the CitiBikeNYC dataset, it needs smaller number of epochs to converge. In the following experiment, we train our model with 400 epochs on both datasets.

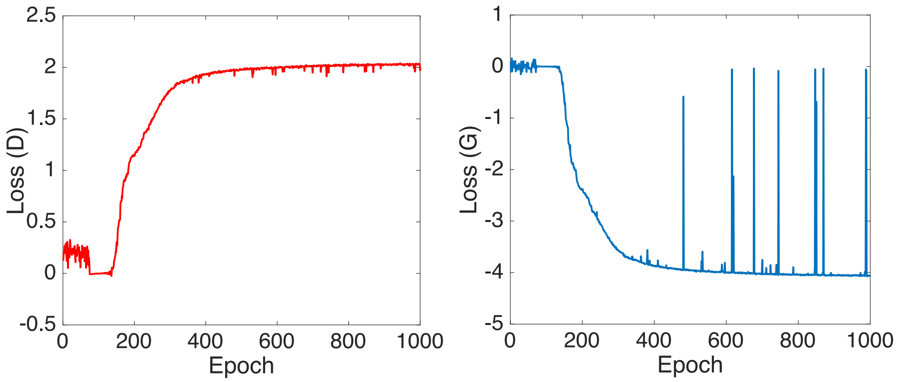


Fig. 9. The loss curves of *G* and *D* of SeqST-GAN.

Table 2. Comparison of Different Methods on *MAE* and *RMSE* for One-step Prediction

Method	TaxiNYC		CitiBikeNYC	
	MAE	*RMSE*	*MAE*	*RMSE*
HA	10.82	38.67	6.42	10.10
ARIMA	12.45	67.5	6.78	13.56
Ridge	11.23	46.56	6.25	11.25
ST-ResNet	9.76	43.97	5.84	10.58
AttConvLSTM	7.56	26.91	4.78	7.76
DMVST-Net	6.58	21.64	4.46	7.85
Seq2Seq	6.12	19.42	4.15	7.68
SeqST-GAN	**5.83**	**18.35**	**3.79**	**7.35**

Comparison on One-step Prediction. As most previous models focus on one-step prediction, for a fair comparison, we first evaluate the performance of various methods on one-step prediction on the two datasets. In this experiment, each training sample $\mathcal{X} = \{\mathbf{X}^t | t = 1, 2, \ldots 25\}$ is composed by the observed crowd flow matrices in the first 24 hours and the crowd flow matrix in the future one hour for prediction. Table 2 shows the result, and the best results are highlighted with bold font. One can see that SeqST-GAN outperforms all the baselines on both MAE and RMSE on the two datasets. HA, though, is a naive prediction model that totally relies on the history average, and it outperforms the other two statistics-based methods ARIMA and Ridge regression. This implies that the two crowd flow datasets present obvious periodicity, and history average can provide a rather good prediction on the future. However, HA is inferior to all the deep learning models, demonstrating that deep models are more powerful in learning spatial-temporal features from crowd flow data. In addition, HA does not consider the effect of external factors such as weather and holidays. Deep learning–based methods including ST-ResNet, AttConvLSTM, and DMVST-Net outperform the shallow models but are inferior to SeqST-GAN. This verifies the effectiveness of the proposed adversarial learning framework in one-step prediction, and incorporating the adversarial learning loss does improve the performance. On average, SeqST-GAN reduces MAE by 13.2% and RMSE by 10.5% on the two datasets compared with the strong baseline DMVST-Net. Compared with AttConvLSTM, the average reductions of MAE and RMSE are 21.8% and 18.5%, respectively. To further investigate whether adversarial learning helps in our studied problem, we compare SeqST-GAN with Seq2Seq. One can see that Seq2Seq actually works even better than the

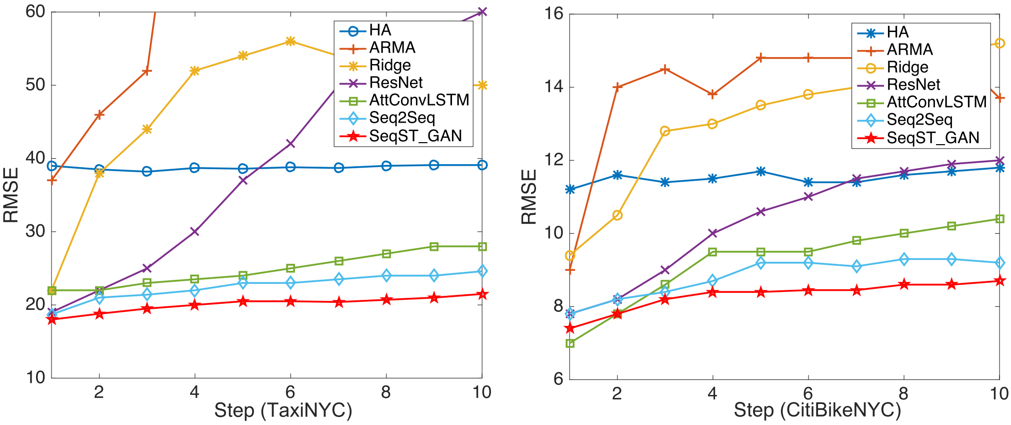


Fig. 10. RMSE of different methods on 10-step prediction.

strong baseline DMVST-Net, which shows the proposed Seq2Seq-based prediction model works well. Seq2Seq model is inferior to SeqST-GAN in both datasets. Thus, one can conclude that adversarial learning does improve the prediction performance.

Comparison on Multi-step Prediction. We next evaluate the performance of the methods on multi-step prediction. We set the step size k as 10 to predict the crowd flows in the future 10 hours simultaneously. In this experiment, each training sample $\mathcal{X} = \{\mathbf{X}^t | t = 1, 2, \ldots 34\}$ is composed by the observed crowd flow matrices in the first 24 hours and the future 10 hours for prediction. Figure 10 shows the RMSE curves of different methods on the two datasets. The experiment result on MAE is similar to RMSE, and thus, we do not show the MAE curves for simplicity. As DMVST-Net cannot perform multi-step prediction, we do not compare with it. One can see that our proposal SeqST-GAN consistently and significantly outperforms all the baselines. The prediction results of ARIMA, Ridge regression, and ST-ResNet are rather unstable, and the RMSE of them increases significantly with the increase of the step number. This implies these methods cannot capture a long-term dependency among the crowd flow data in different hours, and thus they are not suitable for a multi-step prediction. HA can make much more stable predictions, but the performance is poor with much higher RMSE values. The RMSE achieved by HA is nearly 40 over TaxiNYC dataset and larger than 11 over the CitiBike dataset. AttConvLSTM, Seq2Seq, and SeqST-GAN all give stable predictions on the future 10 time slots, and the performance is much better than other models. The three models all use Seq2Seq learning framework for multi-step prediction, which verifies the effectiveness of such a learning framework in the studied problem. The RMSE of SeqST-GAN ranges from around 18 to 22 over the TaxiNYC dataset and from 7.5 to 8.5 over the CitiBike dataset, which are the best results among all the methods. It shows that the three models successfully capture the periodicity of the crowd flow data, and thus their predictions on a longer future are still stable. AttConvLSTM achieves comparable performance with Seq2Seq model. Seq2Seq is slightly better than AttConvLSTM probably due to the usage of the ST-Gate to incorporate the external context features.

Note that the RMSE at $k = 1$ shown in Figure 10 is different from the RMSE shown in Table 2 for SeqST-GAN. This is mainly because the training samples and experiment settings are different in the two experiments. For one-step prediction, only the prediction error for the future 1 hour is minimized; while for 10-step prediction in this experiment, the prediction error for the future 10 hours are all minimized. Therefore, the final optimal solutions of SeqST-GAN in the two experiments are different, leading to different experiment results at $k = 1$.

Table 3. *MAE* of Four Methods under Different Steps k

method	dataset	$k = 1$	$k = 3$	$k = 5$	$k = 8$	$k = 10$	$k = 15$	$k = 20$
HA	CitibikeNYC	6.42	6.73	6.54	6.83	7.20	6.89	6.75
	TaxiNYC	10.82	11.42	11.45	11.26	11.26	11.28	11.27
AttConLSTM	CitibikeNYC	4.78	6.48	7.12	9.13	10.20	11.13	13.55
	TaxiNYC	7.56	8.12	10.14	12.15	13.26	15.44	17.82
Seq2Seq	CitibikeNYC	4.15	6.15	7.35	8.56	9.42	10.46	11.75
	TaxiNYC	6.12	7.82	9.57	11.56	12.56	13.87	15.21
SeqST-GAN	CitibikeNYC	3.79	5.18	5.42	7.16	8.35	9.42	11.75
	TaxiNYC	5.83	6.36	8.45	9.78	11.42	13.24	14.56

Parameter Study on Future Steps *k*. To study the effect of the future steps k to the model performance, we further evaluate the prediction performance of HA, AttConLSTM, Seq2Seq, and SeqST-GAN under different future steps k. Table 3 shows the result with the k set to 1, 3, 5, 8, 10, 15, and 20, respectively. >From this table, one can draw the following conclusions: First, with the increase of the future steps k, the average MAE of the methods AttConLSTM, Seq2Seq, and SeqST-GAN keeps increasing. For example, for the CitiBikeNYC dataset, the MAE achieved by SeqST-GAN is 3.79 when the step k is 1, which is a small value, but it increases to 8.35 when k increases to 10. The result of the TaxiNYC dataset presents the similar trend. This is reasonable, because a farther future (larger k) is always harder to predict than a nearer future (smaller k). Second, the prediction performance on the CitibikeBikeNYC dataset is consistently better than that on the TaxiNYC dataset for all the methods. This shows that the taxi data are harder to predict than the bike data in New York, which is consistent with our previous experiment results. Third, SeqST-GAN consistently outperforms the two baselines by achieving lower MAE on both datasets, demonstrating the effectiveness of the proposed model. Finally, by comparing AttConLSTM with Seq2Seq, one can see that the two models achieve comparable performance, but overall Seq2Seq is slightly better than AttConLSTM. This also verifies the effectiveness of the proposed Seq2Seq model in multi-step crowd flow prediction. Seq2Seq is inferior to SeqST-GAN, which demonstrates that adding the adversarial loss does help to improve the prediction performance.

By comparing HA with the other three methods, one can see that for the long-term prediction (say, $k = 10$), HA outperforms the other methods including our proposed SeqST-GAN. When k is smaller than 10, *HA* is inferior to SeqST-GAN, but when k is larger than 10, *HA* performs better than SeqST-GAN. This is because SeqST-GAN and other prediction models AttConLSTM and Seq2Seq only use the crowd flow data in the previous 24 hours for prediction, while HA makes the prediction based on the average of all the historical data. Therefore, SeqST-GAN, Seq2Seq, and AttConvLSTM perform much better than HA for short-term prediction; while for long-term prediction, HA performs better.

Parameter Study on λ. The parameter λ is used to balance the importance of mean square error and adversary loss for addressing the *blurry prediction* issue. A larger λ means that the mean square error is more important in the final loss function of Equation (5). When λ is too large, the effect of the adversary loss to the objective function is negligible and the model degenerates to most existing models that only use the square error loss. When $\lambda = 0$, it means only the adversary loss is considered in the objective function, while the square error is totally ignored.

To study the effect of λ on the model performance, we test the performance of SeqST-GAN with different value settings of λ ranging from 0.001 to 100 on the two datasets. The five-step prediction result on the MAE metric is shown in Figure 11. One can see that the MAE value varies remarkably with the increase of the λ on both datasets, which demonstrates that λ does have a remarkable

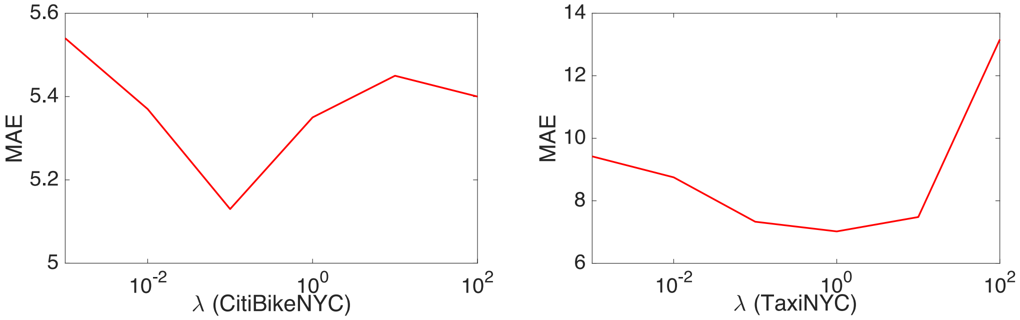
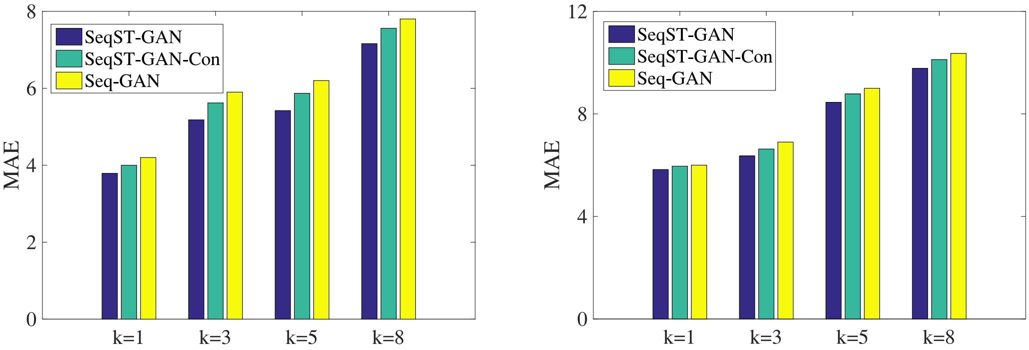


Fig. 11. The effect of λ on the model performance.



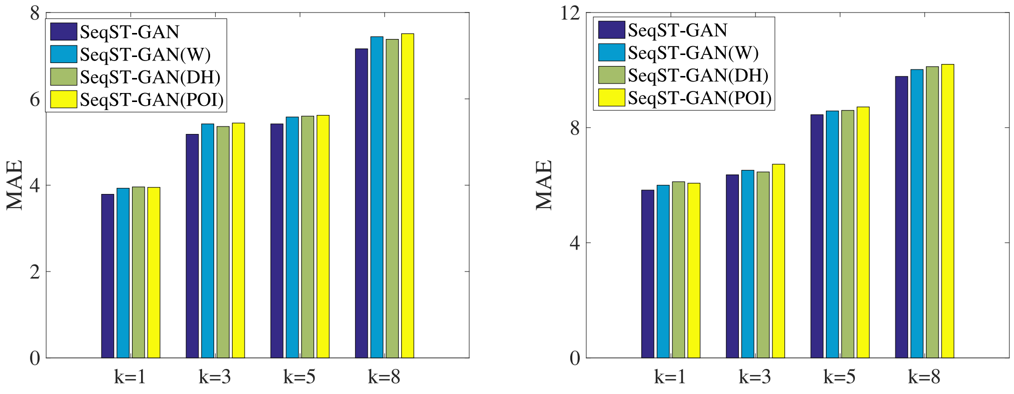
(a) CitiBikeNYC

(b) TaxiNYC

Fig. 12. MAE comparison of SeqST-GAN, SeqST-GAN-Con, and Seq-GAN.

impact on the model performance. On the CitiBikeNYC dataset, the MAE is below 5.2 when $\lambda = 0.1$, while MAE increases to nearly 5.6 when $\lambda = 0.001$. The effect of λ on the TaxiNYC dataset is even more significant, and the MAE varies from around 7 to nearly 13. One can see that a too large or small λ will hurt the performance and the best λ for CitiBikeNYC is 0.1, and 1 for TaxiNYC. When only the mean square error is considered with very large λ values, the performance decreases. This is probably due to the *blurry prediction* issue. Previous works [16, 25] showed that *blurry prediction* issue is common in image prediction and generation. To address this issue, adversary training framework was widely used. On the CitiBikeNYC dataset, the MAE is about 5.4 when $\lambda = 100$, which is lower than the best result 5.2 when a more suitable λ value is set as 0.1. On the TaxiNYC dataset, the effect of *blurry prediction* is even more significant, MAE increasing from 8 to 13. This result demonstrates that a suitable λ can help improve the performance of SeqST-GAN by reducing the negative effect of *blurry prediction* in crowd flow prediction.

Effectiveness of EC-Gate in External Context Feature Learning. Finally, we study whether and to what extent the fine-grained region-level representation features of the external contexts learned by EC-Gate—including weather, date, and POIs—help our prediction model. To this purpose, we compare the complete version of SeqST-GAN with the incomplete version of SeqST-GAN named Seq-GAN, which removes the EC-Gate. To examine whether EC-Gate can better integrate the external context features than simply concatenating the features, we also compare it with SeqST-GAN-Con. Figure 12 shows the MAE comparison of the three methods under different future steps. It shows that on both datasets SeqST-GAN outperforms the two baselines in all the cases, which verifies that the external context features are useful and incorporating them does improve



(a) CitiBikeNYC

(b) TaxiNYC

Fig. 13. MAE comparison of SeqST-GAN, SeqST-GAN(W), SeqST-GAN(DH), and SeqST-GAN(POI).

the model performance. One can also see that the performance improvement of SeqST-GAN on the CitibikeNYC dataset is more significant compared with that on the TaxiNYC dataset. On average, the MAE reduction on the CitibikeNYC dataset is around 5%, but the number is only less than 3% for the TaxiNYC dataset. This is mainly because, as shown in Figure 8 in our data analysis section, the taxi trip data are less sensitive to the external weather data. Comparing SeqST-GAN-Con with Seq-GAN, one can see that SeqST-GAN-Con consistently outperforms Seq-GAN with smaller MAE values. This verifies that the external contexts features are useful, and even simply concatenating the features with the encoded latent features of the crowd flow tensors, the performance can be improved.

However, SeqST-GAN-Con is consistently inferior to SeqST-GAN on the two datasets. It shows that EC-gate can better learn fine-grained features of the external contexts for different regions. This is mainly because for different regions the effect of the external context features on the crowd flow could be quite different. For example, the effects of weekday and weekend on a region of a mall and a region of a university could be different. The effects of weather on a region of park area and an office building area could be also different. For the feature concatenation method SeqST-GAN-Con, it assumes that the external context features such as weather and holidays have the same effect on all the regions, which is not reasonable.

We further investigate whether the three types of external features: weather, POIs, and day & hour are all helpful to the prediction. We remove the three types of features separately and then test the performance. We use SeqST-GAN(W) to denote that the weather features are removed, SeqST-GAN(DH) to denote that the day & hour features are removed, and SeqST-GAN(POI) to denote that the POIs are removed. The experiment results are shown in Figure 13. From this figure one can see that MAE will increase when any type of features are removed in both datasets, which means that the three types of features are all helpful to the crowd flow prediction task.

6 CONCLUSION AND FUTURE WORK

In this article, we proposed a novel model named SeqST-GAN that integrated Seq2Seq model and adversarial learning framework for forecasting multi-step urban crowd flow data. Specifically, a Seq2Seq model was first applied to generate the future crowd flow "frames" step-by-step. To capture the external context features, an EC-Gate module was also designed to learn a unified region-level representation of the features to help tune the initially generated future "frames." Then, through an adversary learning framework, the mean square error and the adversarial loss were combined to handle the *blurry prediction* issue. We evaluated our proposal on two large crowd flow datasets of New York. The results showed that it significantly outperformed several strong baselines.

For the future, it would be interesting to further study whether the proposed framework can be applied to other spatial-temporal data mining tasks, such as trajectory prediction and POI recommendation. Similar to the crowd flow data studied in this work, the trajectory data in an area and the POIs in a city can be also modeled as "images" or "videos," and thus the proposed framework can be applied to perform related prediction tasks. Another potential future research direction that we are particularly interested in is how to extend the current model for urban crowd flow prediction under the scenario of sparse data. As GAN model can produce high quality samples that are very similar to the real ones, we can use GAN to generate more training samples to address the data sparsity issue before training.

REFERENCES

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social LSTM: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[2] Martin Arjovsky, Soumith Chintala, and Leon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the International Conference on Machine Learning.*

[3] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. 2017. Spatio-temporal data mining: A survey of problems and methods. *arXiv:1711.04710v2.* (2017).

[4] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. 2015. Multiple object recognition with visual attention. In *Proceedings of the International Conference on Learning Representations.*

[5] Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations.*

[6] Prateep Bhattacharjee and Sukhendu Das. 2017. Temporal coherency based criteria for predicting video frames using deep multi-stage generative adversarial networks. In *Proceedings of the International Conference on Advances in Neural Information Processing Sytems.*

[7] Mecit Cetin and Gurcan Comert. 2006. Short-term traffic flow prediction with regime switching models. *Transport. Res. Rec.: J. Transport. Res. Board* 1965 (2006).

[8] Y Chen and D Xiao. 2009. Traffic network flow forecasting based on switching model. *Contr. Decis.* 24, 8 (2009), 1177–1180.

[9] Xingyi Cheng, Ruiqing Zhang, Jie Zhou, and Wei Xu. 2017. Deep transport: Learning spatial-temporal dependency for traffic condition forecasting. In *Proceedings of International Joint Conference on Neural Networks (IJCNN'18).*

[10] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning.*

[11] Emily L. Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015.Deep generative image models using a Laplacian pyramid of adversarial networks. In *Proceedings of the Conference on Neural Information Processing Systems.*

[12] Zipei Fan, Xuan Song, Ryosuke Shibasaki, and Ryutaro Adachi. 2015. CityMomentum: An online approach for crowd behavior prediction at a citywide level. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing.*

[13] Ian J. Goodfellow, Jean Pouget-Abadiey, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozairz, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the International Conference on Neural Information Processing Systems.*

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[15] Aude Hofleitner, Ryan Herring, and Pieter Abbeel. 2012. Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network. *IEEE Trans. Intell. Transport. Syst.* 13, 4 (2012), 1679–1693.

[16] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. 2018. Stochastic adversarial video prediction. *arXiv:1804.01523v1* (2018).

[17] Sangsoo Lee and Daniel Fambro. 1999. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transport. Res. Rec.: J. Transport. Res. Board* 1678 (1999).

[18] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *Proceedings of the International Conference on Learning Representations (ICLR'18).*

[19] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P. Xing. 2017. Dual motion GAN for future-flow embedded video prediction. In *Proceedings of the IEEE International Conference on Computer Vision.*

[20] M. Lippi, M. Bertini, and Paolo Frasconi. 2013. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Trans. Intell. Transport. Syst.* 14, 2 (2013), 871–882.

[21] Marco Lippi, Matteo Bertini, and Paolo Frasconi. 2013. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Trans. Intell. Transport. Syst.* 14, 2 (2013), 871–882.

[22] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang. 2017. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* 17, 4 (2017), 818.

[23] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision.*

[24] Julieta Martinez, Michael J. Black, and Javier Romero. 2017. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[25] Michael Mathieu, Camille Couprie, and Yann LeCun. 2015. Deep multi-scale video prediction beyond mean square error. *arXiv:1511.05440* (2015).

[26] Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *Proceedings of the International Conference on Learning Representations.*

[27] Dai Quoc Nguyen, Dat Quoc Nguyen, Cuong Xuan Chu, Stefan Thater, and Manfred Pinkal. 2017. Sequence to sequence learning for event prediciton. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics.*

[28] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the International Conference on Learning Representations.*

[29] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. 2016. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Proceedings of the European Conference on Computer Vision.*

[30] Sujit K. Sahu and Kanti V. Mardia. 2005. A Bayesian kriged Kalman model for short-term forecasting of air pollution levels. *J. Roy. Statist. Soc. Series C: Appl. Statist.* 54, 1 (2005), 223–244.

[31] Shashank Shekhar and Billy Williams. 2008. Adaptive seasonal time series models for forecasting short-term traffic flow. *Transport. Res. Rec.: J. Transport. Res. Board* 2024, 116–125 (2008).

[32] Xuan Song, Quanshi Zhang, Yoshihide Sekimoto, and Ryosuke Shibasaki. 2014. Prediction of human emergency behavior and their mobility following large-scale disaster. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining.*

[33] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining.*

[34] Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. 2018. Crowd flow prediction by deep spatio-temporal transfer learning. *arXiv:1802.00386.* (2018).

[35] Senzhang Wang, Jiannong Cao, and Philip S. Yu. 2019. Deep learning for spatio-temporal data mining: A survey. *arXiv preprint arXiv:1906.04928.* (2019).

[36] Senzhang Wang, Lifang He, Leon Stenneth, Philip S. Yu, and Zhoujun Li. 2015. Citywide traffic congestion estimation with social media. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.*

[37] Senzhang Wang, Fengxiang Li, Leon Stenneth, and Philip S. Yu. 2016. Enhancing traffic congestion estimation with social media by coupled hidden Markov model. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases.*

[38] Senzhang Wang, Xiaoming Zhang Fengxiang Li, Philip S. Yu, and Zhiqiu Huang. 2019. Efficient traffic estimation with multi-sourced data by parallel coupled hidden Markov model. *IEEE Trans. Intell. Transport. Syst.* 20, 8 (2019), 3010–3023.

[39] Senzhang Wang, Xiaoming Zhang, Jianping Cao, Lifang He, Leon Stenneth, Philip S. Yu, Zhoujun Li, and Zhiqiu Huang. 2017. Computing urban traffic congestions by incorporating sparse GPS probe data and social media data. *ACM Trans. Inf. Syst.* 35, 4 (2017), 40:1–40:30.

[40] Billy Williams. 2001. Multivariate vehicular traffic flow prediction: Evaluation of ARIMAX modeling. *Transport. Res. Rec.: J. Transport. Res. Board* 1776 (2001).

[41] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, Yanwei Yu, and Zhenhui Li. 2018. Modeling spatial-temporal dynamics for traffic prediction. *arXiv preprint arXiv:1803.01254.* (2018).

[42] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

[43] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. 2017. DeepSense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the World Wide Web Conference (WWW'17).*

[44] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

[45] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

[46] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiqing Zhang, and Xiuwen Yi. 2016. DNN-based prediction model for spatio-temporal data. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.*

[47] Yuxuan Zhang, Senzhang Wang, Bing Chen, and Jiannong Cao. 2019. GCGAN: Generative adversarial nets with graph CNN for network-scale traffic prediction. In *Proceedings of the International Joint Conference on Neural Networks.* 1–8.

[48] Xian Zhou, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. 2018. Predicting multi-step citywide passenger demands using attention-based neural networks. In *Proceedings of the ACM International Conference on Web Search and Data Mining.*

Received December 2018; revised November 2019; accepted December 2019