GACNet: A Generative Adversarial Capsule Network for Regional Epitaxial Traffic Flow Prediction

Jinyuan Li[1], Hao Li[1], Guorong Cui[1], Yan Kang[1,*], Yang Hu[1] and Yingnan Zhou[2]

Abstract: With continuous urbanization, cities are undergoing a sharp expansion within the regional space. Due to the high cost, the prediction of regional traffic flow is more difficult to extend to entire urban areas. To address this challenging problem, we present a new deep learning architecture for regional epitaxial traffic flow prediction called GACNet, which predicts traffic flow of surrounding areas based on inflow and outflow information in central area. The method is data-driven, and the spatial relationship of traffic flow is characterized by dynamically transforming traffic information into images through a two-dimensional matrix. We introduce adversarial training to improve performance of prediction and enhance the robustness. The generator mainly consists of two parts: abstract traffic feature extraction in the central region and traffic prediction in the extended region. In particular, the feature extraction part captures nonlinear spatial dependence using gated convolution, and replaces the maximum pooling operation with dynamic routing, finally aggregates multidimensional information in capsule form. The effectiveness of the method is evaluated using traffic flow datasets for two real traffic networks: Beijing and New York. Experiments on highly challenging datasets show that our method performs well for this task.

Keywords: Regional traffic flow, adversarial training, feature extraction, nonlinear spatial dependence, dynamic routing.

1 Introduction

With the rapid pace of deep learning, these related methods have further fostered the development of traffic prediction [Yi, Jung and Bae (2017)]. A lot of traffic data is obtainable through the effective channel that extensive deployment of traffic sensors. Simultaneously, advanced data processing technology is convenient for converting raw trajectory data into traffic flow. The study of traffic prediction has also changed from statistical model based on limited traffic data to research model based on data-driven deep learning methods. More complex architectures are used to achieve better results than traditional methods. Nevertheless, these attempts still focus primarily on traffic forecasts

[1] National Pilot School of Software, Yunnan University, Kunming, 650091, China.

[2] School of Engineering and Applied Science, The George Washington University, Washington DC, 20052, USA.

[*] Corresponding Author: Yan Kang. Email: kangyan@ynu.edu.cn.

Received: 25 January 2020; Accepted: 28 February 2020.

for road networks or small areas [Park, Li, Murphey et al. (2011); Huang, Song, Hong et al. (2014); Lv, Duan, Kang et al. (2015); Ma, Yu, Wang et al. (2015)]. Few studies have considered an entire transportation network and directly estimated large-scale traffic evolution. What's more, time series forecasts of traffic variables ignored spatial characteristics and the spatial information cannot be effectively exploited this way.

However, with the rapid development of global urbanization, the trend of urban expansion to surrounding areas is obvious, especially in metropolitan areas. This expansion trend will generate new situations and problems in urban management and traffic control. In terms of data collection, due to the influence of daily life and economic activities, the traffic data information in the central area is easy to obtain and cover widely. In terms of data collection, the influence of daily life and economic activities help central area cover more trajectory information. On the contrary, it is difficult to obtain the data of an area far from the center, as the coverage is limited, and the cost of data collection is increasing. This two-level differentiation phenomenon brings challenges to the intelligent traffic management, and proposes new research directions and tasks, i.e., regional extension prediction task of short-term traffic flow. This task of traffic flow prediction is a novel work, which make an epitaxial prediction from central area to surrounding area. As far as we know, this task has not been studied yet. The research of this paper has an essential meaning to future urban traffic calculation.

In order to explore spatial correlation of urban transportation in detail, we use geographic information to describe the spatial structure. As shown in Fig. 1, we try to divide the city into many regions in a grid form. As a reference, latitude and longitude is necessary. Each mesh represents traffic flow of different regions. Traffic flow includes two types: inflow and outflow. These can be estimated by the number of pedestrians, cars, and buses which enter or go out a region within a specific time interval. The flow between the regions always interacts and changes. In order to explore what kind of spatial dependence existing between regions, we select 9 regions centering on A (including region A), which are four neighbor regions B, C, D and E, respectively, and F, G, H and I far from A. The nine regions selected correspond to the urban regional network of Beijing and New York respectively. It can be seen in Fig. 2(a), the corresponding relations of Beijing are [16, 16]-A, [15, 16]-B, [16, 15]-C, [16, 17]-D, [17, 16]-E, [16, 0]-F, [1, 16]-G, [16, 31]-H, [31, 16]-I, respectively; In fig 2(b), the corresponding relations of New York are [8, 4]-A, [7, 4]-B, [8, 3]-C, [8, 5]-D, [9, 4]-E, [8, 0]-F, [0, 4]-G, [8, 7]-H, [15, 4]-I, respectively. Taking the inflow as an example, Figs. 2(a) and 2(b) respectively show the interactions of inflows between different regions in the trajectory data of Beijing and New York over a certain time interval. It can be seen from the two figures that the time regularity of traffic flow during a short period of time is weakened or even negligible, but the spatial correlation performance is particularly prominent. Specifically, in the same time intervals, the traffic flows of the adjacent regions B, C, D, and E and the far-distance regions F, G, H, and I have the same tendency to change with the region A. It is worth to note that this trend indicates that if there is traffic congestion in a certain region, the spatial dependence or the influence of congestion propagation in the spatial regions will not only quickly affect its neighbors, but also the regions far from it. The reason is the adjacency between regions doesn't necessarily mean the adjacency in the road network, such as the existence of subways. Therefore, how to establish an intrinsic relationship model of map attributes and spatial correlation types is of great significance for urban traffic forecasting, which can help with real-time traffic control and induction.

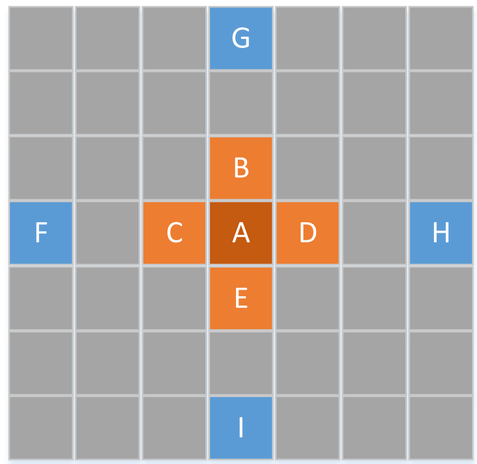


Figure 1: Urban regional traffic gird diagram

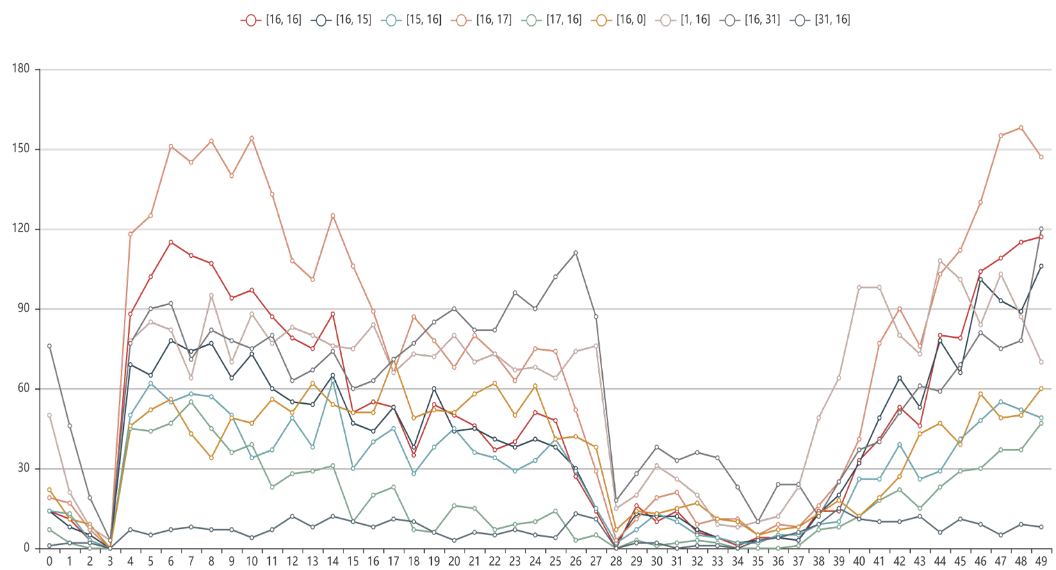


Figure 2(a): The different distance zones' traffic inflow of adjacent time period trend in Beijing. The abscissa contains fifty time intervals, and every time interval has thirty minutes. The ordinate is value of traffic inflow

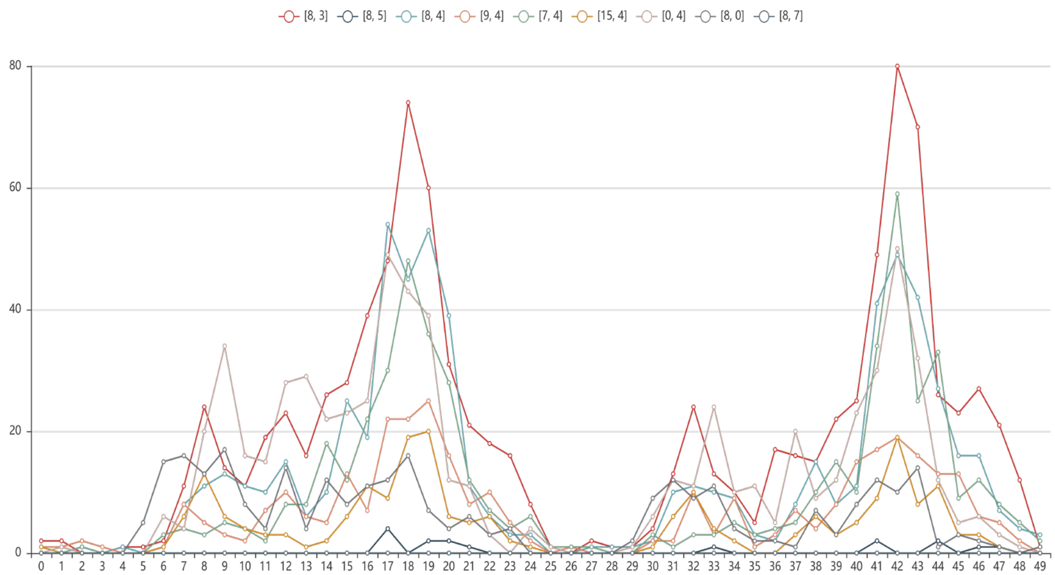


Figure 2(b): The different distance zones' traffic inflow of adjacent time period trend in New York. The abscissa contains fifty time intervals, and every time interval has one hour. The ordinate is value of traffic inflow

In response to the tasks mentioned above, we propose a generative adversarial capsule network model called GACNet. This method is aimed to predicts traffic flow in the surrounding areas by capturing the nonlinear spatial dependence of the near and far regions. In order to better extract spatial features and achieve good predictive performance, we constructed a model between adversarial capsule network based on the idea of confrontation. In addition, we use gated convolution to capture nonlinear spatial correlation, and then the maximum merge operation is replaced by dynamic routing to aggregate multidimensional information in a capsule form. What we design can improve the robustness of the prediction in an effective manner. Finally, this approach would be evaluated on a public dataset in a large real-world world.

The contributions of the paper can be summarized as follows:

● In view of the trend of urban expansion to surrounding areas in the process of urbanization development, we propose a new task of epitaxial regional traffic flow forecasting, i.e., to predict the traffic flow of surrounding areas according to the central area. This task is original and has important research significance for future traffic prediction.

● We design an adversarial capsule network model to solve the proposed epitaxial regional traffic flow prediction task. In order to better extract the spatial correlation between the near and far regions, we collect multidimensional information through the dynamic routing of the capsule layer.

● We use the method of adversarial training, exploiting the discriminant function of the discriminator to improve predictive ability of the generator and enhance robustness of the model. The approach proposed is verified on real-world data.

The rest of this paper is structured as follows: The data definitions and problem of our research is described in Section 2. The architecture of our model is described in Section 3. The experimental results and analysis of the proposed method are shown in Section 4. Finally, Section 5 will draw conclusions and sketch out future research directions.

2 Related work

2.1 Traffic prediction method

Great success deep learning achieved in many challenging tasks means that these techniques can be solve the related problems of traffic forecasts. Original method of studying traffic prediction was to study the time characteristics and construct a complex network structure to simulate time series. This type of method decomposes the traffic time series into periodic terms, trend terms and other parts, and strives to analyze multiple factors in the real situation, and pursues prediction accuracy. Later, some researchers gradually shifted their attention to spatial characteristics, and creatively proposed a method of learning traffic as images [Ma, Dai, He et al. (2017)] proposed the deep learning architecture of Convolutional Neural Network (CNN) to extract temporal and spatial traffic features contained in images, and achieved good prediction performance. It is noteworthy that spatio-temporal features can be captured by convolutional network better. In complicated scene, therefore, Kim et al. proposes that adopting deep learning approach to solve traffic forecasts for road networks [Kim, Wang, Zhu et al. (2018)]. Differ from other manners, this way focuses on the temporal features of traffic flow, and not treats the time dimension as multi-dimension channel that like images. However, what focus on long-term forecasting tasks based on past traffic information would not be our emphases for research. In this paper, we will study how to solve one short-term forecasting tasks that named regional epitaxial traffic flow prediction.

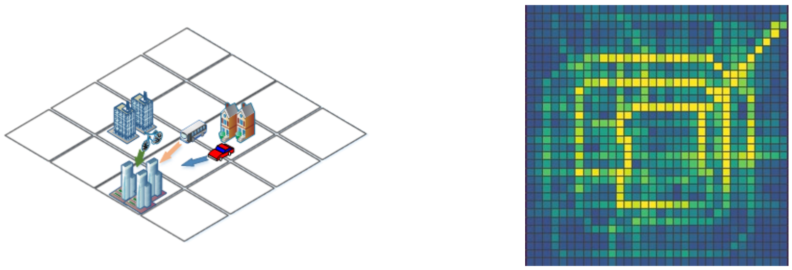
2.2 Generative adversarial network

The appearance of Generative Adversarial Network (GAN) provides a new idea for solving deep learning problems [Li, Liang, Zhao et al. (2019)]. It consists of a generator and a discriminator. The important point of Generative Adversarial Network is adversarial training between the generator and the discriminator. The discriminator distinguishes false and real samples to learn iteratively in a supervised way. Through competition of two models, generator of network gets continually optimization. This optimization come from discriminant loss of the discriminator, finally its purpose is that indistinguishable pseudo-examples are produced by the generator. The great potential of Generative Adversarial Network has verified in many applications [Li, Jiang and Cheslyar (2018)]. As shown in Eq. (1), it is objective function of Generative Adversarial Network.

$$\min_{G} \min_{D} V(D,G) = E_{x\sim p_{data}(x)}[\log \mathrm{D}(\mathrm{x})] + E_{z\sim \mathrm{p(z)}}[\log(1-D(G(z)))] \tag{1}$$

3 Preparation materials

3.1 Data definition



(a) (b)

Figure 3: Urban regional flow processing graph

As seen as Fig. 3(a), we divide whole city into several grids, and each mesh represents traffic flow of different regions. Inflow and outflow can be estimated by the number of pedestrians, cars, and buses which enter or go out a region within a specific time interval. Let P be the trajectory set of time interval t^{th}. For the grid (i,j) is the i^{th} row and j^{th} column, the inflow and outflow of the specified time interval are calculated as follows:

$$x_t^{in,i,j} = \sum_{Tr \in P} \left| \left\{ k > 1 \mid g_{k-1} \notin (i,j) \wedge g_k \notin (i,j) \right\} \right| \tag{2}$$

$$x_t^{out,i,j} = \sum_{Tr \in P} \left| \left\{ k \geq 1 \mid g_k \in (i,j) \wedge g_{k+1} \notin (i,j) \right\} \right| \tag{3}$$

where $Tr: g_1 -> g_2 -> ....-> g_{|Tr|}$ is anyone trajectory which belong to P; the space coordinate $g_k \in (i,j)$ indicates the position in the grid region (i,j). In the t^{th} time interval, inflow and outflow in whole regions $\mathbf{I} \times \mathbf{J}$ can be expressed as $\mathbf{X}_t \in R^{2 \times I \times J}$, where $(\mathbf{X}_t)_{0,i,j} = x_t^{in,i,j}$, $(\mathbf{X}_t)_{1,i,j} = x_t^{out,i,j}$.

The inflow and outflow of each region we get is displayed in the form of images, as shown in Fig. 3(b), the depth of yellow represents the flow rate.

3.2 Problem statement

In this article, we will identify clearly the research goal. And given the central area $\left\{ (\mathrm{i}/4, 3i/4) \times j \right\}$, predict traffic flow of both sides area, as shown in Fig. 4.

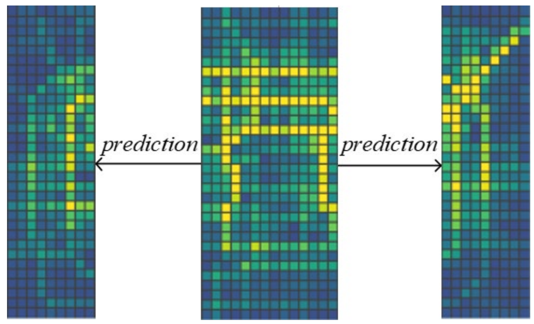


Figure 4: Region extension flow prediction graph

4 Description of our approach

The GACNet we proposed aims to address the previously mentioned problem that predicts epitaxial regional traffic flow. And Fig. 5 illustrates the architecture of GACNet, which contains two components: a generator and a discriminator. We introduce a gating mechanism in the convolutional layer of the generator's feature extraction part and add different levels of multidimensional features extracted by the capsule layer at the back end. The discriminator is trained to distinguish between predicted surrounding area traffic and real results. In this section, the architectural details of the generator and discriminator are given.

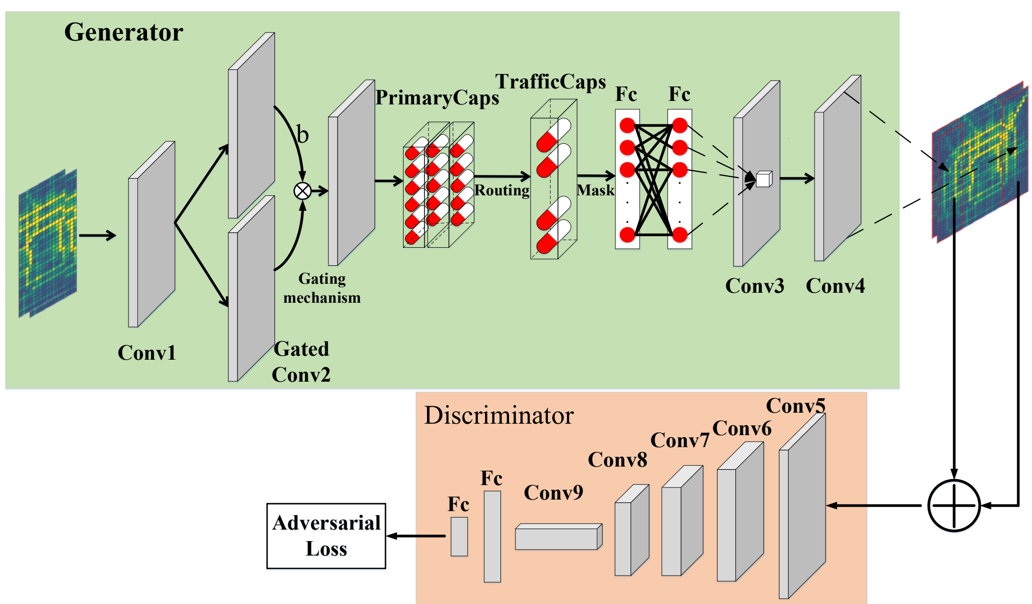


Figure 5: Architecture of regional extended traffic prediction

4.1 Generator

Due to existence of some special means of transportation such as subway, therefore the adjacency in the roads does not represent adjacency between various regions of city

[Shaw, Fang, Lu et al. (2014)]. In this case, although the CNN acquires some spatial features, and the maximum pooling operation is used to construct higher-order features, the largest collection loses valuable information by picking the way with the most activated neurons. This structure, which uses only one vector to represent relational features, can result in impaired accuracy of multi-region association extraction. Even with different convolution sizes, the effect on the extraction of multidimensional features is not obvious. The proposed capsule network was used to improve the representative limitations of CNN [Sabour, Frosst and Hinton (2017); Hinton, Sabour and Frosst (2018)]. A capsule is a group of neurons, and each layer contains a number of capsules, each of which represents a different property of the same object. One of the main features of the capsule is that it has a carrier form, in which the vector output is provided by the squash activation function and the artificial neurons are operated by scalars. Unlike convolutional network, the use of capsules can extract more local features, such as the relative relationship between non-linear spatial traffic features, and high-level capsules can be gathered from lower-level capsules through a transformation matrix.

Inspired by the above-described method, we specially designed the feature extraction part to put at the front end of the generator, whose main components are Conv1, Gated Conv2, PrimaryCaps, and TrafficCaps. The first convolutional layer, Conv1, has a kernel size of 4×4, 8 channels, and step size is set to 1. It initially combines the inflow and outflow as regional traffic information for dual channel inputs. Then, we introduce a gating mechanism in the second layer of the model. Specifically, we use the higher-level activations to only process relevant information by filtering out the rest since start. The output in the Gated Conv2 layer is defined as follows:

$$u = (X * W + a) \otimes b(X * V + c) \tag{4}$$

where X is the output of convolutional layer, W and V represent different convolution kernels. And parameters a and c can be learned continuously. $\otimes$ represents elementwise product between matrices, and sigmoid function is represented as b.

Since the size of our input image is small, the size of the output is kept constant in the design of the first two layers of convolution and the number of channels of the feature map is increased. The purpose is to increase the amount of information obtained.

According to deal with images of traffic flow, The Gated Conv2 layer transform it to local feature which is used as inputs of PrimaryCaps. Actually, it is deformation of the convolutional layer that has 3×3 kernel size and four channels. PrimaryCaps layer has several capsules with 8-dimensional vector. It's worth noting that capsules share their weights with each other.

Dynamic routing performs 3 iterations between PrimaryCaps and TrafficCaps to capture important spatial-hierarchical relationships among all capsules in the PrimaryCaps layer and each capsule in TrafficCaps layer. After acquiring the high-level capsule feature, we perform a nonlinear squash operation on the high-level capsule, ensuring the direction of the vector is constant. This operation can be defined as:

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2}\frac{s_j}{\|s_j\|} \tag{5}$$

At the back end of the generator, two fully connected layers are designed to decode, then reconstruct the regional spatial relationships by Conv3. In order to refine this detailed information, we finally add a convolutional layer Conv4 with 2 channels and output regional epitaxial traffic map predicted.

4.2 Discriminator

The discriminator accepts the results of the extended prediction and the real image to determine whether the input is true or false. It consists of five convolution layers and two fully connected layers. The first convolutional layer uses the 32 convolution kernel size, while the other four convolutional layers all have 64 kernel size. After convolutional layer, we add relu activation. Finally, our fully connected layer with sigmoid activation is added to obtain a probability for binary classification. It is worth noted that both the generator and the discriminator add an instance normalization (IN) [Ulyanov, Vedaldi and Lempitsky (2016)] layer to alleviate the vanishing gradient problem. We use the discriminator to train once, while the generator trains twice to balance the training.

As illustrated in Algorithm 1, our approach is implemented as following:

By taking traffic flow dataset set as input, the original data is preprocessed and obtains two subdata sets including the central region dataset and the surrounding region dataset. Instead of using batch training, take the way to train by entering a single piece of data. Firstly, enter the data of the central area into the generator to get the preliminary prediction results of the surrounding area. Then, train discriminator to distinguish between the real and predicted data. After that, keep discriminator parameters fixed, and use the BP algorithm to adjust the generator parameters. Finally, output generator (predictor).

4.3 Loss function

To train our model better, we follow the training loss function of Least Squares GAN [Mao, Li, Xie et al. (2017)]. We mark the generated predicted traffic graph as 0, and the real graph as 1. The discriminator is used to distinguish fake and real samples. We adopt the mean squared error loss function to approximate the distance between result predicted and real value. The way training the generator is effective and achieves the goal of improving the quality of result generated.

The loss function of the discriminator and generator is as follows:

$$\min_D L_{GAN}(D) = \frac{1}{2}E_{x\sim p_{data}(x)}\left[(D(x)-b)^2\right] + \frac{1}{2}E_{z\sim p_z(z)}\left[\left(D\left(G(z)\right)-a\right)^2\right] \tag{6}$$

In the above equation, we choose b=1, to indicate that it is real data, and a=0, to indicate that it is forged data.

$$\min_G L_{GAN}(G) = \frac{1}{2}E_{z\sim p_z(z)}[D(G(x)-1)]^2 \tag{7}$$

Algorithm 1: The training procedure of the regional epitaxial traffic flow predictor in association with GAN.

Input: The given dataset U, discriminator D and generator G.

Output: Final generator (predictor) G.

Initialization: Preprocessing data U obtain $X=\left\{x^{(1)},x^{(2)},....,x^{(n)}\right\}$ and $Y=\left\{y^{(1)},y^{(2)},....,y^{(n)}\right\}$; $P=\varnothing$; the initial learning rate of discriminator α; the initial learning rate of generator β; the initial parameters of discriminator $\left(W_1,b_1\right)$; the initial parameters of generator $\left(W_2,b_2\right)$; the training iterations is k.

Training:

1: for $t=1$ to k do
2: for each instance $x^{(i)}$ in X
3: $p^{(i)} \leftarrow G\left(x^{(i)}\right)$
4: Add $p^{(i)}$ to P
5: Distinguish the truth and prediction by $D\left(y^{(i)},p^{(i)}\right)$
6: for $j=1$ to T do
7: Keep discriminator parameters $\left(W_1,b_1\right)$ fixed, and use the BP algorithm to adjust the generator parameters $\left(W_2,b_2\right)$.
8: end for
9: end for
10: end for

5 Empirical study

5.1 Data description

Experiments on two real-world datasets that will verify the effectiveness of the proposed network. Zhang et al. [Zhang, Zheng, Qi et al. (2018)]. The details of our experimental datasets are shown in Tab. 1.

TaxiBJ: We obtain taxi flows from Beijing's taxicab trajectories, which include four time periods: June 1st, 2013 to August 30th, 2013, May 1st, 2014 to June 30th, 2014, May 1st, 2015 to June 30th, 2015, December 1st, 2015 to April 10th, 2016. Beijing city is divided to 32×32 grids, then we get hourly taxi flows for each grid.

BikeNYC: The New York's trajectory data is from the New York bicycle system, which spans from April 1st to September 30th, 2014. One piece of data includes: travel time, ID of the start and end sites, start time, and end time. New York City is divided to 16×8 grids, then we get hourly taxi flows for each grid.

According to the above data definition, we obtain two types of traffic flow. The data is divided into non-overlapping training and test data at a ratio of 8:2.

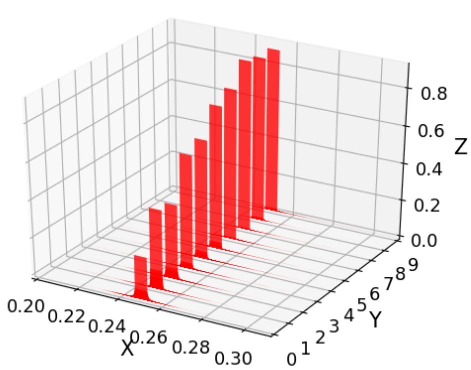
Table 1: Experimental datasets

Dataset	Taxi BJ	Bike NYC
Data type	Taxi GPS	Bike rent
Location	Beijing	New York
Time Span	2013/7/1~2013/10/30 2014/3/1~2014/6/30 2015/3/1~2015/6/30 2015/11/1~2016/4/10	2014/4/1~2014/30/9
Time interval	30 minutes	1 hour
Size of grid	(32, 32)	(16, 8)
Average sampling rate(s)	~60	\
taxis/bikes	34000+	6800+
Available time interval	22459	4392

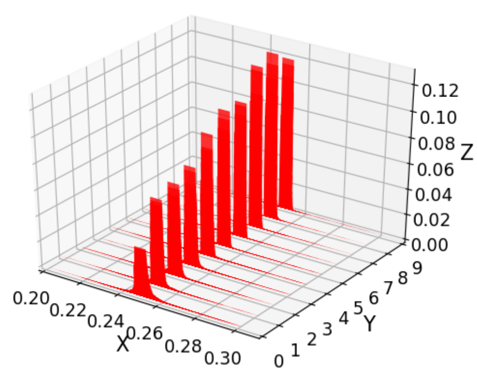
5.2 Implementation details

In regard to the efficiency performance, it takes 10 epochs for training. Scale data into the range [-1,1], before feeding it into our network. The learning rate of the generator and the discriminator are initially set to 0.001 and 0.0001 respectively. And the dropout rate is 0.5.

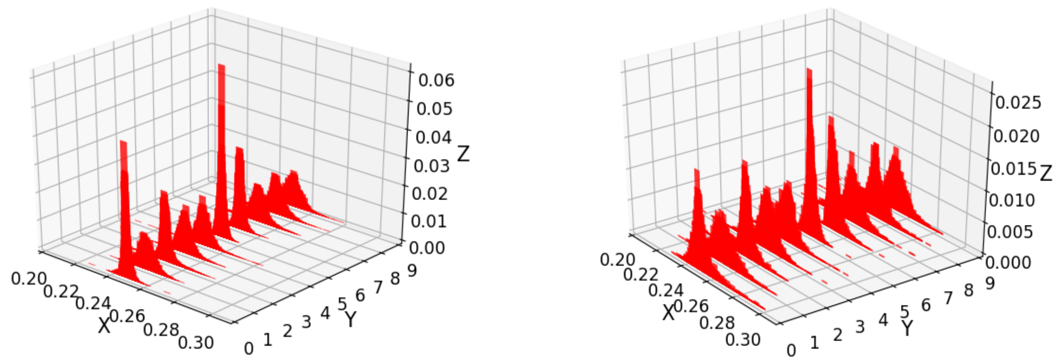
5.3 Results and comparison



(a)



(b)

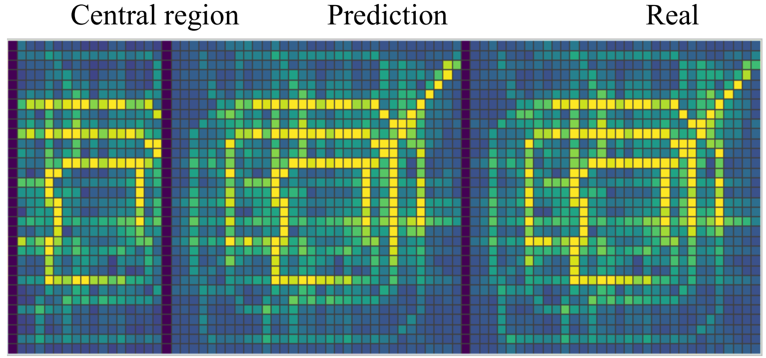


(c) (d)

Figure 6: The training process of the discriminator and generator on TaxiBJ and BikeNYC dataset. We divide the loss value interval from 0.2 to 0.3 into 1000 equal parts. The X-axis represents the interval of the loss value, the Y-axis represents the epoch of GACNet training, and the Z-axis represents the probability that the loss value belongs to a certain interval

Fig. 6 shows the training processes of the discriminator and generator on the TaxiBJ and BikeNYC training sets, respectively. In order to better show convergence of the discriminator and the generator in the model training process, we set the loss value interval to 0.2-0.3. We divide the average into 1000 segments between 0.2 and 0.3. As shown in Figs. 6(a) and 6(b) (respectively), the discriminator and generator of GACNet converge very fast on the TaxiBJ dataset, and as the number of epochs increases, the loss value remains stable around 0.25. On the BikeNYC dataset, the discriminator and generator of GACNet also converge very fast, and gradually stabilize at around 0.25, as shown in Figs. 6(c) and 6(d). The training process described above shows that our training method performs well on the two training sets.

Central region Prediction Real



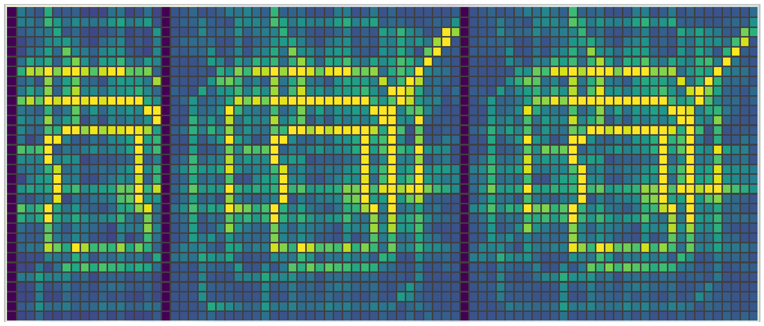


Figure 7: Traffic prediction on TaxiBJ dataset

We present some samples to show intuitively the effect of prediction for on TaxiBJ dataset. As showed as Fig. 7, inflow and outflow are visualized. Concretely speaking, samples show traffic condition in different areas. We choose two color to represent the condition of traffic flow. For high traffic areas, it's yellow is deeper. On the contrary, the deeper the green is, the smaller the traffic flow is. Real the images of traffic flow are added to contrast, and it's obvious that our method proposed is close to real results. The visualization has a good performance.

The visualization shown in Fig. 7 are just one kind form of effect on experiment. However, its form has not shown complete performance on so large data set. And, mean absolute error (MAE) and root mean squared error (RMSE) [Miyato, Kataoka, Koyama et al. (2018); Heusel, Ramsauer, Unterthiner et al. (2018); Lucic, Kurach, Michalski et al. (2018)] are intuitive metrics to assess the performance of traffic flow prediction. The two performance metrics are defined as:

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

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|(y_i - \hat{y}_i)\right| \tag{9}$$

Five baselines as comparison are chosen to test the performance of our algorithm proposed. CNN [Wang, Zhang, Yang et al. (2015)] is good for extracting features near-distance spatial for traffic flow. Deformable CNN [Dai, Qi, Xiong et al. (2017)] flexibly extracts feature information for important regions by adopting the irregular shape of the convolution. Separable CNN [Chollet (2017)] adopt convolution layers to extract features then merge, after input data from several channel. Gated CNN [Dauphin, Fan, Auli et al. (2017)] uses the gated mechanism in the convolution layer to extract important feature information. NonLocalResNet adopt non-local means methods to extract long-range dependencies [Wang, Girshick, Gupta et al. (2018)]. Tab. 2 shows the results for the proposed GACNet model and the alternative algorithms on the two types of test datasets over the 10 epochs. The results show that GACNet is better than other comparison algorithms for the regional extension traffic flow prediction task. Compared with the Gated CNN, our method provides 5.54% and 6.28% improvement in RMSE and MAE on TaxiBJ dataset. In addition, ours provides 37.5% and 35.7% improvement in RMSE and

MAE on TaxiBJ dataset. The result shows that adversarial learning can not only improve the prediction accuracy, but also stably train. Compared with other models, GACNet has a significant advantage in regional traffic flow prediction tasks.

Table 2: Performance (RMSE, MAE) of GACNet and other algorithms

Model	RMSE		MAE	
	TaxiBJ	BikeNYC	TaxiBJ	BikeNYC
CNN	36.89	12.47	16.32	6.67
NonLocalResNet	55.66	10.28	34.59	5.92
Deformable CNN	19.62	4.68	11.57	2.80
Separable CNN	16.61	4.78	9.95	2.71
Gated CNN	15.68	6.45	9.87	3.64
GACNet	14.81	4.03	9.25	2.34

6 Conclusion

In this paper, we have presented a new deep learning architecture, called GACNet for regional extension traffic flow prediction task. This approach is absorbed in the design of the information extraction structure of the generator, particularly adding the capsule network layer with dynamic routing mechanism, which automatically improves spatial traffic feature extraction capability. The results show that our model has better predictive effect than comparison models on TaxiBJ and BikeNYC datasets. In future work, we will present some interesting extensions to this task, such as a further extended forecast in space based on the predicted results.

Acknowledgement: This work was funded by the National Natural Science Foundation of China under Grant (Nos. 61762092 and 61762089).

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Chollet, F. (2017): Xception: deep learning with depthwise separable convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251-1258.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G. et al. (2017): Deformable convolutional networks. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 764-773.

Dauphin, Y. N.; Fan, A.; Auli, M.; Grangier, D. (2017): Language modeling with gated convolutional networks. *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 933-941.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B. (2018): GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems*, pp. 6626-6637.

Hinton, G. E.; Sabour, S.; Frosst, N. (2018): Matrix capsules with EM routing.

Huang, W.; Song, G.; Hong, H.; Xie, K. (2014): Deep architecture for traffic flow prediction: deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191-2201.

Kim, Y.; Wang, P.; Zhu, Y.; Mihaylova, L. (2018): A capsule network for traffic speed prediction in complex road networks. *Sensor Data Fusion: Trends, Solutions, Applications*, pp. 1-6.

Li, C.; Jiang, Y.; Cheslyar, M. (2018): Embedding image through generated intermediate medium using deep convolutional generative adversarial network. *Computers, Materials & Continua*, vol. 56, no. 2, pp. 313-324.

Li, X.; Liang, Y.; Zhao, M.; Wang, C.; Jiang, Y. (2019): Few-shot learning with generative adversarial networks based on WOA13 data. *Computers, Materials & Continua*, vol. 60, no. 3, pp. 1073-1085.

Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F. Y. (2015): Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865-873.

Lucic, M.; Kurach, K.; Michalski, M.; Gelly, S.; Bousquet, O. (2018): Are GANs created equal? A large-scale study. *Advances in Neural Information Processing Systems*, pp. 700-709.

Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y. et al. (2017): Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, vol. 17, no. 4, pp. 818.

Ma, X.; Yu, H.; Wang, Y.; Wang, Y. (2015): Large-scale transportation network congestion evolution prediction using deep learning theory. *PLos One*, vol. 10.

Mao, X.; Li, Q.; Xie, H.; Lau, R. Y.; Wang, Z. et al. (2017): Least squares generative adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794-2802.

Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. (2018): Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957.

Park, J.; Li, D.; Murphey, Y. L.; Kristinsson, J.; McGee, R. et al. (2011): Real time vehicle speed prediction using a neural network traffic model. *IEEE International Joint Conference on Neural Networks.*

Sabour, S.; Frosst, N.; Hinton, G. E. (2017): Dynamic routing between capsules. *Proceedings of Advances in Neural Information Processing Systems*, pp. 3859-3869.

Shaw, S. L.; Fang, Z.; Lu, S.; Tao, R. (2014): Impacts of high-speed rail on railroad network accessibility in China. *Journal of Transport Geography*, vol. 40, pp. 112-122.

Ulyanov, D.; Vedaldi, A.; Lempitsky, V. (2016): Instance normalization: the missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022.

Wang, X.; Girshick, R.; Gupta, A.; He, K. (2018): Non-local neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794-7803.

Wang, C.; Zhang, H.; Yang, L.; Liu, S.; Cao, X. (2015): Deep people counting in extremely dense crowds. *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 1299-1302.

Yi, H.; Jung, H. J.; Bae, S. (2017): Deep neural networks for traffic flow prediction. *IEEE International Conference on Big Data and Smart Computing.*

Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; Yi, X. et al. (2018): Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence*, vol. 259, pp. 147-166.