CityTraffic: Modeling Citywide Traffic via Neural Memorization and Generalization Approach

Xiuwen Yi[1,2], Zhewen Duan[3,1], Ting Li[1], Tianrui Li[4], Junbo Zhang[1,4,+], Yu Zheng[1,4]
[1]JD Intelligent Cities Research & JD Intelligent Cities Business Unit, China
[2]Department of Computer Science and Technology, Tsinghua University, China
[3]School of Computer Science and Technology, Xidian University, China
[4]Institute of Artificial Intelligence, Southwest Jiaotong University, China
xiuwenyi@foxmail.com,{duanzhewen,liting30}@jd.com,trli@swjtu.edu.cn,{msjunbozhang,msyuzheng}@outlook.com

ABSTRACT

With the increasing vehicles on the road, it is becoming more and more important to sense citywide traffic, which is of great benefit to the government's policy-making and people's decision making. Currently, traffic speed and volume information are mostly derived from GPS trajectories data and volume sensor records respectively. Unfortunately, speed and volume information suffer from serious data missing problem. Speed can be absent at arbitrary road segment and time slot, while volume is only recorded by limited volume sensors. For modeling citywide traffic, inspired by the observations of missing patterns and prior knowledge about traffic, we propose a neural memorization and generalization approach to infer the missing speed and volume, which mainly consists of a memorization module for speed inference and a generalization module for volume inference. Considering the temporal closeness and period properties, memorization module takes advantage of neural multi-head self-attention architecture to memorize the intrinsic correlations from historical traffic information. Generalization module adopts neural key-value attention architecture to generalize the extrinsic dependencies among volume sensors by exploiting road contexts. We conduct extensive experiments on two real-world datasets in two cities, Guiyang and Jinan, and the experimental results consistently demonstrate the advantages of our approach. We have developed a system on the cloud, entitled CityTraffic, providing citywide traffic speed and volume information and fine-grained pollutant emission of vehicles in Guiyang city.

CCS CONCEPTS

• **Applied computing** → **Transportation**; • **Information systems applications** → **Spatial-temporal systems**;

KEYWORDS

Traffic modeling; Deep learning for spatio-temporal data; Neural attention; Urban computing

+ Junbo Zhang is the corresponding author.

CIKM '19, November 3–7, 2019, Beijing, China

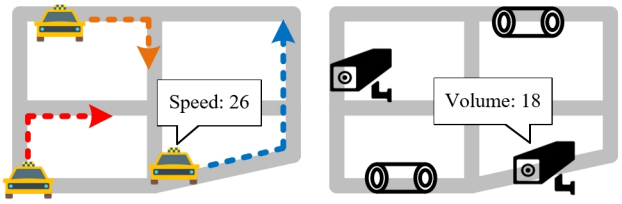
ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357822

ACM Reference format:
Xiuwen Yi[1,2], Zhewen Duan[3,1], Ting Li[1], Tianrui Li[4], Junbo Zhang[1,4,+], Yu Zheng[1,4]. 2019. CityTraffic: Modeling Citywide Traffic via Neural Memorization and Generalization Approach. In *Proceedings of The 28th ACM International Conference on Information and Knowledge Management, Beijing, China, November 3–7, 2019 (CIKM '19),* 7 pages.
https://doi.org/10.1145/3357384.3357822

1 INTRODUCTION

With the increasing vehicles on the road, it is becoming more and more important to model citywide traffic, which is of great benefit to many urban applications, especially for the government's policy-making and people's decision making [15]. For example, transportation agencies can perform traffic control and quickly respond to traffic accidents with citywide traffic information. Also, people can make fast-driving routes as well as avoid traffic congestion during the driving. Moreover, the traffic information can be used to estimate the spatial and temporal fine-grained pollutant emission of vehicles, which is a crucial problem for mobile emission inventory in environment domain [14].

For sensing traffic information, GPS trajectories, loop detectors, and surveillance cameras are three widely used sensors in the transportation domain, which can collect traffic speed and volume on the road. As shown in Figure 1 a), GPS trajectories record the driving routes of vehicles, e.g. taxicabs and online car-hailing, which can be used to estimate the travel speed with location and time [20]. However, it is insufficient to sense citywide traffic speed and directly infer traffic volume as it is only a small sample of entire traffic. As shown in Figure 1 b), loop detectors and surveillance cameras are volume sensors, which can record the actual number of vehicles traversing the road [11]. However, the coverage of volume sensors is limited because of the high installation and maintenance overheads. Thus, the collected speed and volume information can not cover the whole road network.



a) GPS trajectories b) Volume sensors

Figure 1: Traffic speed and volume sensors

For modeling citywide traffic speed and volume, we can first estimate the speed and volume on covered road segments with collected GPS trajectory data and volume sensor records and then infer the missing values of speed and volume on uncovered road segments. However, this work faces two challenges.

First, the data suffers from a serious missing problem. For example, in Jinan, only 2% road segments are monitored by surveillance cameras and about 69% of road segments are traversed by taxicabs in one hour. Moreover, the absence patterns of speed and volume are significantly different. Speed can be absent at arbitrary road segment and time slot, while volume is only recorded by a few limited volume sensors. Thus, it is difficult to predict speed and volume simultaneously in a multi-task framework.

Second, traffic has dynamic temporal dependencies and complex spatial correlations. Traffic speed is correlated with historical speed information, both near and far. For instance, traffic during morning rush hours may be similar on consecutive workdays. Also, traffic flow is correlated with adjacent road segments as vehicles traversed on the road. Moreover, road context features, such as speed limitation and road level, have an effect on the traffic as they reveal the characteristics of a road.

To address these challenges, for inferring citywide traffic speed and volume, we propose a neural memorization and generalization approach, which mainly consists of a memorization module for speed inference and a generalization module for volume inference. Our approach is inspired by the observations of speed and volume missing patterns and prior knowledge about traffic, which can help to design the model structure with more interpretations. Our contributions are listed as below:

- We propose a memorization module (CT-Mem) for citywide speed inference, which can memorize the intrinsic correlations from historical traffic information with neural multi-head self-attention architecture.
- We propose a generalization module (CT-Gen) for citywide volume inference, which can generalize the extrinsic dependencies among existing volume sensors with neural key-value attention architecture.
- We evaluate our approach on two real-world traffic datasets in two cities, Guiyang and Jinan. Extensive experiments demonstrate the advantages of our approach for both speed and volume inference.
- We develop an online traffic modeling system on the cloud, entitled CityTraffic, providing citywide traffic speed and volume information and fine-grained pollutant emission of vehicles in the Guiyang city of China.

2 OVERVIEW

Our approach mainly consists of three parts: preprocessing module for data transformation, memorization module for speed inference, and generalization module for volume inference. More specifically, in the preprocessing module, we first map the GPS trajectories onto road network with a storm-based map-matching method [4] and then calculate the average travel speed for the road segments covered by matched trajectories [14]. Also, we aggregate traffic volume by counting the number of vehicles detected by loop detectors or surveillance cameras in a given time interval [11]. Moreover, for

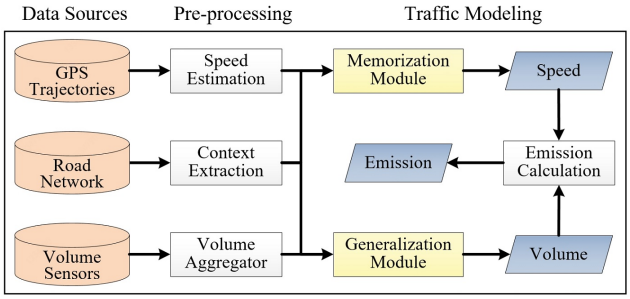


Figure 2: Framework of our approach

each road segments, we extract road context features, e.g. road level and speed limitation. For memorization module, considering the temporal closeness and period properties, we firstly select correlated historical speed records and then exploit neural multi-head self-attention architecture to memorize the intrinsic correlations for citywide speed inference. For generalization module, we firstly select adjacent and correlated road segments with existing volume sensors as candidates and then leverage neural key-value attention architecture to generalize the underlying dependencies among volume sensors for citywide volume inference. Thus, we estimate citywide speed and volume with GPS trajectory data and volume sensor records. Moreover, we can estimate the spatial and temporal fine-grained pollutant emission of vehicles based on the estimated traffic information using an existing emission estimation equation from environmental science [11].

Figure 3 illustrates the sensed speed and volume information after data preprocessing. Speed and volume are stored in the form of a matrix, where a row stands for a time slot t and a column refers to a road segment r. An entry in the matrix refers to the reading of speed or volume, while ? means the data is missing. For modeling citywide traffic speed and volume, the task can be formulated as filling the missing values of speed and volume matrix. From the left part of Figure 3, we can observe that the absence of speed can occur on an arbitrary road segment at any time slot. In other words, the missing pattern of speed is dynamic and unpredictable. On the contrary, as shown in the right part of Figure 3, the absence of volume is constant and stable over time as the number of volume sensors is limited. Thus, it is infeasible to infer the absent volume of a road segment without volume sensor by its historical records. Inspired by the above observations, considering the spatial and temporal neighbors, we design a memorization module for speed inference and a generalization module for volume inference respectively, which will be detailed in Section 3.

	Speed (*km/hour*)						Volume (*vehicle/minute/lane*)					
	r_1	r_2	r_3	⋯	r_{n-1}	r_n	r_1	r_2	r_3	⋯	r_{n-1}	r_n
t_1	17	?	21	⋯	?	40	?	12	?	⋯	?	22
t_2	?	24	29	⋯	?	?	?	17	?	⋯	?	29
t_3	19	?	?	⋯	33	47	?	24	?	⋯	?	37
⋮												
t_m	?	19	?	⋯	26	42	?	15	?	⋯	?	26

Figure 3: Illustration of speed and volume information

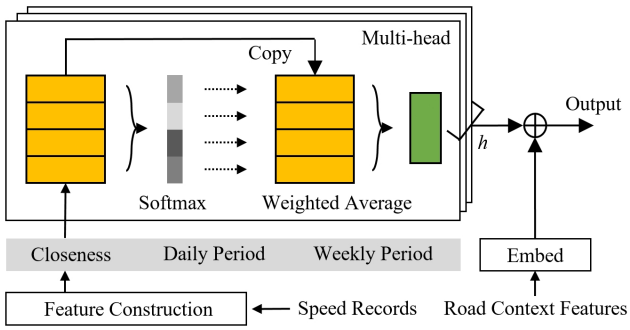


Figure 4: Speed inference method

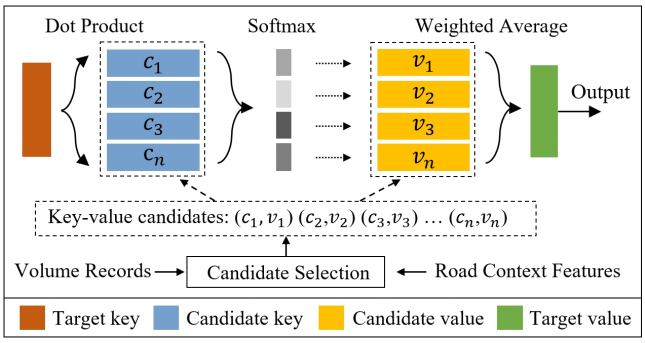


Figure 5: Volume inference method

3 METHODOLOGY

In this section, we elaborate our proposed memorization module for citywide speed inference and generalization module for citywide volume inference, which each module is implemented with a neural attention architecture.

3.1 Memorization Module: CT-Mem

For citywide speed inference, as shown in Figure 4, we propose a neural memorization module, entitled CT-Mem, which consists of a feature construction component and a multi-head self-attention network. The former component selects correlated historical speed records as inputs and the latter network learns intrinsic correlations from these records. Also, we integrate road context features for the final speed prediction.

As we know, traffic speed is correlated with historical information, both near and far. For example, the speed of a road is similar to recent time slots, and the traffic during morning rush hours is similar to that of consecutive workdays. Thus, in feature construction component, we select some relevant historical speed records by jointly considering the records with adjacent time slots within two hours and the records with the same time slot on daily and weekly periods. Here, we do not incorporate speed information from spatial neighbors. The reason behind it is the dynamic and unpredictable missing pattern of speed, which we can not construct a consistent input for the further network.

To memorize the correlations from historical traffic records, we adopt a multi-head self-attention architecture based neural network [3, 17], where self-attention can learn a better intrinsic relation from one feature vector and multi-head structure can extract more useful information than single self-attention. More specifically, we aggregate the selected historical speed records together using a concatenate layer and then feed it into a self-attention network, which consists of weight calculation and weight average.

For weight calculation, it first takes the selected speed records $\mathbf{x}$ as input, and calculates a vector of attention weights with a Softmax function on the given input itself:

$$\boldsymbol{\alpha} = Softmax(\Phi(\mathbf{x}) \cdot \mathbf{w}) \tag{1}$$

where $\Phi(\mathbf{x})$ is the projected feature vectors of $\mathbf{x}$ with a dense layer, and $\mathbf{w}$ is a weight parameter vector. As the original dimension of speed record is 1-dim, here, we up-sampling it to 2-dim for a value-enhanced representation with $\Phi(\mathbf{x})$.

For weight average, the result is computed as a weighted sum of the calculated attention weights dot producing with the projected feature vectors $\Phi(\mathbf{x})$:

$$\mathbf{s}_h = \Phi(\mathbf{x}) \cdot \boldsymbol{\alpha} \tag{2}$$

Finally, we can gather h-layer self-attention results with the multi-head structure.

In addition to the speed information, we also consider the road level, speed limitation, time of day, day of week as external features. For learning the intra-dynamics of each influential factors, we use embedding layer first before fusion. Combine external features and outputs of multi-head self-attention together, we make the final prediction with a full-connected layer.

3.2 Generalization Module: CT-Gen

For citywide volume inference, As shown in Figure 5, we propose a neural generalization module, entitled CT-Gen, which consists of a candidates selection component and a key-value attention network. The former component selects related road segments with existing volume sensors as candidates and the latter network learns the extrinsic dependencies among volume sensors.

Due to the missing pattern of volume is constant and stable over time, it is infeasible to infer volume from historical records. As vehicles traversed on the road, volume records from adjacent roads can provide lots of useful information. For modeling these information, we design a candidates selection component to select correlated road segments with existing volume sensors. The selection is based on following two considerations. First, adjacent roads may be more likely to have a similar volume. Second, road segments with the same road characteristics may share similar volume pattern. Thus, we first filter the volumes sensors with road characteristics, i.e., road level and road speed limitation. Then, we rank the adjacent road segments by road-network distance in descending-order and select the top k road segments as inputs.

To generalize the dependencies among volume sensors, we leverage a key-value attention network [12], which can learn the importance degrees of existing volume sensors and well fit the key-value data storage format. More specifically, we divide road segments into two categories, target and candidate. For each road segment, we can construct a key-value pair, which the key refers to the road segment id, time slot id, and road context features including length, lanes, level, speed limitation and average speed, while the value is the

volume. Thus, the data can fall into 4 groups, candidate key $\mathbf{C}$, candidate value $\mathbf{V}$, target key $\mathbf{t}$, and target value $\mathbf{o}$, where target value needs to be predicted. With the candidate selection component, we can construct a list of key-value candidates.

For learning the importance degrees of existing volume sensors, we calculate the similarity scores between each candidate key and target key with a dot-product operation, and then normalize the scores as the relevance probability using a Softmax function:

$$\boldsymbol{\beta} = Softmax(\Phi(\mathbf{t}) \odot \Phi(\mathbf{C})) \tag{3}$$

where Φ is the feature projection operation, $\odot$ denotes dot product operation, and $\boldsymbol{\beta}$ represents the relevance probabilities of all candidates. Taking the result of candidates values dot producing the calculated probabilities, we can estimate the target value $\mathbf{o}$, which is the final prediction.

$$\mathbf{o} = \Phi(\mathbf{V}) \cdot \boldsymbol{\beta} \tag{4}$$

3.3 Model training and prediction

Both CT-Mem and CT-Gen are trained via back propagation to minimize mean squared error between the predicted value and the ground truth value. The pseudo code of the training and prediction process is presented in Algorithm 1.

Algorithm 1: CT-Mem and CT-Gen Algorithms

Input: Observed speed records X_s; Observed volume records X_v; Absent speed records $\mathcal{R}_s$; Absent volume records $\mathcal{R}_v$; Road context features X_r;
Output: Predicted values of absent speed Y_s and volume Y_v.

```
// Memorization module;
```
$\mathcal{D}_s \longleftarrow \emptyset$;
for each r in $\mathcal{X}_s$ **do**
  $\mathcal{D}_s \longleftarrow FeatureConstruction(r, X_s, X_r)$;
end
train CT-Mem model $\mathcal{M}_s$ by minimizing the loss with $\mathcal{D}_s$;
$Y_s \longleftarrow M_s(\mathcal{R}_s)$;
```
// Generalization module;
```
$\mathcal{D}_v \longleftarrow \emptyset$;
for each r in $\mathcal{X}_v$ **do**
  $\mathcal{D}_v \longleftarrow CandidateSelection(r, X_v, X_r)$;
end
train CT-Gen model $\mathcal{M}_v$ by minimizing the loss with $\mathcal{D}_v$;
$Y_v \longleftarrow M_v(\mathcal{R}_v)$;

4 SYSTEM

Figure 6 illustrates the online process of CityTraffic architecture on the cloud. With the data receiver implemented by web API, the cloud can continuously receive GPS trajectories of floating cars and volume records from loop detectors and surveillance cameras, and then caches them into Redis, which is an in-memory database for fast data changing. A virtual machine (VM) on the cloud pulls the data from the Redis, and then infer citywide traffic speed and volume with the learned CT-Mem and CT-Gen models from the offline training process. The results from VM are pushed into Redis and then visualized in the web through the web services. As the cache size of Redis is only a little GB (e.g. 6GB), we also store the data and results in the storage (e.g. Hive). For VM, we adopt the machine with 2 cores and 3.5 GB memory, which is enough to model citywide traffic for a city.

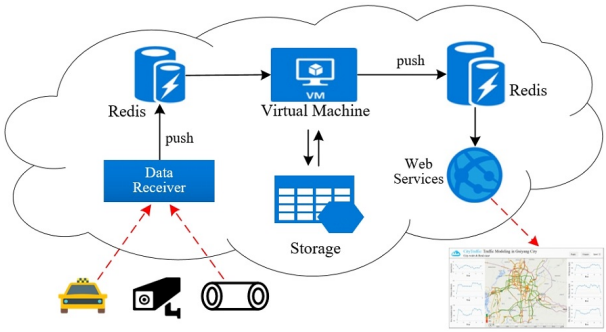


Figure 6: CityTraffic architecture on the cloud

Figure 7 shows the website of CityTraffic. The main part is a geo-map presenting the inferred citywide traffic information over the whole road network, which can zoom-in and zoom-out for different scales. The color of each road segment is determined in accordance with the value of speed, volume, and emission, e.g. red means congestion and green means unimpeded. Different from traditional online map provider, we not only provide speed information but also provide volume information and pollutant emission information. The user can view the speed or volume or emission via clicking the name in the top-right of the web. The left and right parts visualize the citywide traffic information with line chart, which the horizontal axis is the hour in past 24 hours and the vertical axis is citywide information, consisting of average traffic speed and volume for each road segment, and aggregated CO, NO_x, and $PM_{2.5}$ emission for whole city. Moreover, the user can watch the movie-style historical information by clicking the "Replay" button at the right-top of the website. Also, the bottom of map shows a few sequential time slots, which we can click the associated time slot to see the detailed traffic information at that time slot, as shown in the bottom of Figure 7.

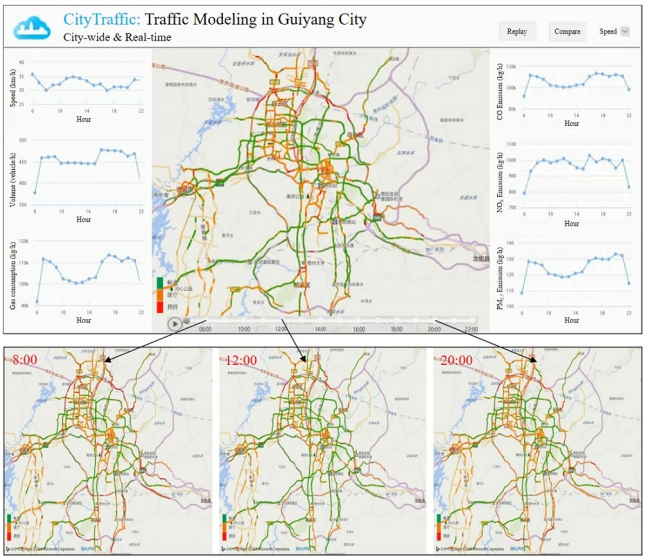


Figure 7: Web interface of CityTraffic

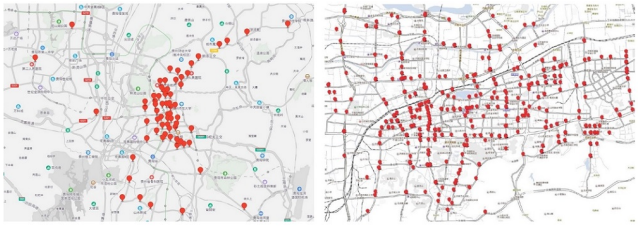
Table 1: Data statistic of Guiyang and Jinan datasets

Dataset	Guiyang	Jinan
Time Spans	2016/03/16-2016/04/01	2017/09/01-2017/09/30
Time Slots (minute)	20	60
Spatial Range (km)	25×32	28×18
# Road Segments	12,647	18,727
# Taxicabs	6,918	13,269
% covered/time slot	58.5%	69.2%
Volume Sensor Type	Loop detector	Surveillance camera
# Volume Sensors	155	354

5 EXPERIMENTS

5.1 Settings

5.1.1 Datasets. We evaluate our approach on two real-world traffic datasets in Guiyang and Jinan cities, which table 1 details the statistical results. We collect GPS trajectories data and volume records data from March 16, 2016 to April 1, 2016 in Guiyang and from September 1, 2017 to September 30, 2017 in Jinan. There are 6,918 taxicabs traversed in 12,647 road segments in Guiyang covering 25 × 32 km spatial range, and 13,269 floating cars traversed in 18,727 roads in Jinan covering 28 × 18 km spatial range. Here, we set 20 minutes and 60 minutes as a time slot for Guiyang and Jinan, respectively. % covered/time slot is the proportion of road segments traveled by at least once in a given time slot, which denotes the missing rate of citywide speed information. There are 155 loop detectors in Guiyang and 354 surveillance cameras in Jinan, respectively. The geo-distribution of volume sensors are illustrated in Figure 8, which each red icon represents a volume sensor. We can see that the loop detectors are mainly deployed in the old downtown in Guiyang city, while the surveillance cameras are relatively dispersed over Jinan city.



a) Guiyang b) Jinan

Figure 8: Geo-distribution of Volume sensors

5.1.2 Ground Truth. For speed inference, we randomly remove 30% of non-zero observed speed records and predict these values using different methods. The removed entries are then used as ground truth to measure the accuracy of the predicted values.

For volume inference, as the task is to predict the volume of road segments without volume sensors, we divide all observed volume data by volume sensors into training and test sets with the proportion of 7:3. Thus, we can avoid using historical volume data to infer current volume information for the same location.

5.1.3 Baselines.

- **K-Nearest Neighbor (KNN)** averages the nearest top k speed or volume values as the prediction.
- **Historical Average (HA)** averages the speed of historical records at the same time slots.
- **Contextual Average (CA)** averages the volume values of road segments with the same road context.
- **Gradient Boosting Regression Tree (GBRT)** is a powerful and widely used ensemble regression model.
- **Feedforward Neural Network (FNN)** flattens all the features and then feeds them together into a multi-layer fully-connected network.
- **Attentional Deep Air quality Inference Network (ADAIN)** combines FNN and RNN to capture static and sequential features and leverages an attention layer to learning different weights of features [5].
- **Context-aware Matrix Factorization (CMF)** constructs multiple related matrices, historical speed matrix and road context matrix, and then perform collective matrix factorization to infer the missing values [14].
- **Spatio-Temporal Semi-Supervised Learning (ST-SSL)** applies semi-supervised learning into spatio-temporal domain to infer citywide traffic volume with loop detector data and taxi trajectories [11].

5.1.4 Model Details.

- Preprocessing. We use min-max normalization to normalize the continuous features into [0, 1], e.g. speed, and use one-hot encoding to transform discrete features, e.g. weekend/weekday. In the evaluation, we rescale the predicted values back to the normal values.
- Hyperparameters. (1) the number of candidates in memorization module and generalization module are 20 and 15; (2) the dimension of volume key embedding is 10 and volume value embedding is 10; (3) the dimension of road context embedding is 5. We select 10% of the training data as the validation set and allow training to be early stopped according to the validation score.
- Activation & Optimization. For activation function, we use Exponential Linear Unit for fully-connected layers. We apply Adam to train the parameters with learning rate is 0.001 and batch size is 512. To prevent over-fitting, we employ dropout with probability 0.5 on the last layer.
- Experimental environment. We train the models on a GPU server with Tesla K40m GPU and programming environment is Keras with TensorFlow as backend.

5.1.5 Evaluation Metrics. We use Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) for evaluation, which are defined as follow:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_i - \hat{y_i}|, MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|y_i - \hat{y_i}|}{\hat{y_i}} \quad (5)$$

where y_i and $\hat{y_i}$ are ground truth and the corresponding predicted value, and n is the total number of all available ground truth. Note that MAE is more affected by larger values, while MAPE receives more punishments from smaller values.

5.2 Results

For evaluation, we repeat the experiments 5 times and take the average as final results.

5.2.1 Results on speed inference. The performances of different approaches for speed inference are presented in table 2. We find that the HA performs better than MF and CMF. The reason behind it is that speed has strong temporal dependencies, and which does not change much within the closeness and the same daily period and weekly period. Our CT-Mem approach achieves the highest accuracy in both datasets as it memorizes the intrinsic dependencies from historical traffic information, which can capture the temporal correlations effectively. To further investigate the effectiveness of closeness/daily/weekly features, we also compare CT-Mem with its variants. By combining closeness, weekly period, daily period together, we can see a clear decrease in MAE and MAPE. Also, we can find the speed have a stronger daily correlation then closeness and weekly. For CT-Mem(With Spatial), which add the speed information from its adjacent road segments, we can find a worse performance. The reason behind it is the spatial correlations over the whole road network are really complex and the dynamic and unpredictable missing pattern of speed.

Table 2: Comparison results on speed inference

Methods	Guiyang		Jinan	
	MAE	MAPE	MAE	MAPE
KNN	1.91	0.282	2.49	0.331
HA	0.99	0.143	1.25	0.160
MF	1.66	0.218	1.89	0.235
CMF	1.06	0.146	1.29	0.163
FNN	0.96	0.151	1.22	0.169
CT-Mem(Only Closeness)	1.13	0.175	1.38	0.180
CT-Mem(Only Weekly)	0.99	0.153	1.23	0.163
CT-Mem(Only Daily)	0.93	0.149	1.21	0.161
CT-Mem(With Spatial)	1.21	0.184	1.49	0.191
CT-Mem	**0.89**	**0.140**	**1.16**	**0.153**

5.2.2 Results on volume inference. Table 3 shows the performance of the proposed approach CT-Gen with other competing baselines. CT-Gen achieves the highest accuracy in both datasets as it generalizes the extrinsic dependencies among volume sensors by exploiting road contexts and spatial properties, which can capture the spatial correlations effectively. Also, we observe that the performance improvement of Guiyang is higher than that of Jinan, as Jinan dataset focuses on the downtown while Guiyang dataset contains both downtown and suburbs, which demonstrates that our model performs better for the road with low visiting frequency. To illustrate the advantage of key-value attention, we compare CT-Gen with its variant CT-Gen(self-attention), which adopts the self-attention architecture. CT-Gen(self-attention) performs worst that CT-Gen. The reason behind it is the context features and volume are two different kinds of information. Key-value attention can capture complex feature interactions and encode prior knowledge about traffic flexibility.

Table 3: Comparison results on volume inference

Methods	Guiyang		Jinan	
	MAE	MAPE	MAE	MAPE
KNN	1.27	0.642	1.49	0.801
CA	1.14	0.623	1.47	0.712
GBRT	1.23	0.643	1.42	0.722
ST-SSL	1.06	0.581	1.37	0.692
FNN	1.33	0.688	1.51	3.853
ADAIN	1.23	0.643	1.42	0.722
CT-Gen(self-attention)	1.06	0.531	1.39	0.665
CT-Gen	**0.98**	**0.395**	**1.32**	**0.618**

5.2.3 Parameter turning. Figure 9 shows the impact of candidate number of generalization module in Jinan dataset, which controls the spatial range of adjacent neighbors. We observe that when varying the number of candidate, both MAE and MAPE first decreases and then increases. When the candidate number is 15, CT-Gen achieves the best performance consistently on all metrics. This is because a small number may only capture the spatial correlation in a small range, but a large number may fail to capture the spatial correlation over a huge road network.

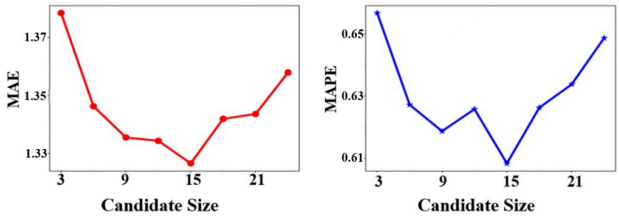
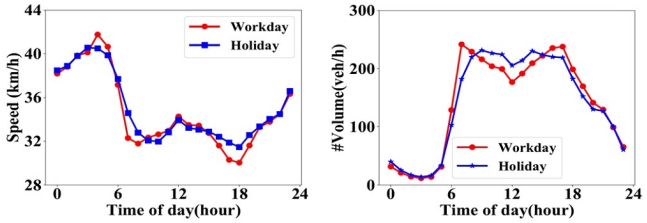


Figure 9: Impact of candidate number in Jinan

5.2.4 Citywide traffic speed and volume. Figure 10 shows the citywide average traffic speed and volume in Jinan respectively. We can observe that, before 6 am, the speed is fast while the volume is low. During 6-9am (morning rush hours) and 4-7pm (afternoon rush hours), the traffic volume has a similar increasing pattern. Also, we can find the traffic of 12 am is the most unimpeded during the daytime. Comparing workdays and holidays, the overall speed and volume trend is similar and the congestion of workday come earlier and heavier than the holiday.



a) Traffic speed b) Traffic volume

Figure 10: Citywide traffic speed and volume of Jinan

6 RELATED WORK

6.1 Traffic Inference

There are many works focus on future traffic prediction, such as [6, 10] propose CNN/RNN and attention based neural network to short-term traffic speed prediction. However, future prediction task is different with filling missing values task, due to the dynamic missing pattern of traffic. Travel speed inference methods mainly consider historical traffic data [7]. Matrix completion models [1] are widely used by obtaining a suitable low-rank approximation of the incomplete matrix. [14] proposed a context-based matrix factorization to model real-time and historical speed patterns simultaneously. Different from them, we propose a multi-head self-attention based neural model to memorize the intrinsic dependencies from historical traffic information.

Traffic volume inference methods mainly consider the spatio-temporal correlations and road contexts to infer absent volumes [18]. [13] learns the relationships for highways traffic density estimation. [14, 20] adopt an unsupervised bayesian model with GPS trajectories and a few volume data. [2] explored the possibility of learning a regression model with floating cars' occurrence. [16] propose a deep reinforcement learning based approach with surveillance camera data. [11] estimate volume with a graph-based semi-supervised model based on loop detector and taxi trajectories. Different from them, we propose a key-value attention based model to generalize the extrinsic dependencies among existing volume sensors for capturing complex feature interactions and encoding prior knowledge about traffic flexibility.

6.2 Deep Learning for Spatio-Temporal Data

Currently, many works show the strength of DNN in solving spatio-temporal prediction problems. To predict citywide crowd flows, [21] proposed a residual based CNN network to learn both spatial and temporal features. [19] proposed a deep distributed fusion network to predict future air quality for each monitoring stations. [9] proposes a deep neural network-based method for spatial fine-grained urban flow inference problem. Recently, attention mechanism is wildly used in general sequence-to sequence problems [17]. [8] developed a RNN-based multi-level attention network to forecast geo-sensor readings. For inferring the air quality of locations without monitoring stations, [5] proposes a attention based neural network, which concatenates dynamic features and static features to learn the importance of different stations.

7 CONCLUSION

In this work, we propose a neural memorization and generalization approach for citywide traffic inference. Based on the observations of missing patterns and prior knowledge about traffic, CT-Mem takes advantage of multi-head self-attention architecture to memorize the intrinsic correlations for speed inference and CT-Gen adopts key-value attention architecture to generalize the extrinsic dependencies for volume inference. Experimental results on two real-world datasets, Jinan and Guiyang, consistently demonstrate the effectiveness of our approach. We have developed a system on the cloud, entitled CityTraffic, providing citywide traffic speed and volume modeling services in Guiyang city.

ACKNOWLEDGMENTS

This work was supported by China Postdoctoral Science Foundation and National Natural Science Foundation of China Grant (61773324).

REFERENCES

[1] Muhammad Tayyab Asif, Nikola Mitrovic, Justin Dauwels, and Patrick Jaillet. 2016. Matrix and tensor based methods for missing data estimation in large traffic networks. *IEEE Transactions on intelligent transportation systems* 17, 7 (2016), 1816–1825.

[2] Javed Aslam, Sejoon Lim, Xinghao Pan, and Daniela Rus. 2012. City-scale traffic estimation from a roving sensor network. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 141–154.

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

[4] Jie Bao, Ruiyuan Li, Xiuwen Yi, and Yu Zheng. 2016. Managing massive trajectories on the cloud. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 41.

[5] Weiyu Cheng, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. 2018. A neural attention model for urban air quality inference: Learning the weights of monitoring stations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[6] Zhixiang He, Chi-Yin Chow, and Jia-Dong Zhang. 2018. STANN: A Spatio–Temporal Attentive Neural Network for Traffic Prediction. *IEEE Access*.

[7] Huiqi Hu, Guoliang Li, Zhifeng Bao, Yan Cui, and Jianhua Feng. 2016. Crowdsourcing-based real-time urban traffic speed estimation: From trends to speeds. In *IEEE 32nd International Conference on Data Engineering*. 883–894.

[8] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. 2018. GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction.. In *IJCAI*. 3428–3434.

[9] Yuxuan Liang, Kun Ouyang, Lin Jing, Sijie Ruan, Ye Liu, Junbo Zhang, David S Rosenblum, and Yu Zheng. 2019. UrbanFM: Inferring Fine-Grained Urban Flows. *arXiv preprint arXiv:1902.05377* (2019).

[10] Zhongjian Lv, Jiajie Xu, Kai Zheng, Hongzhi Yin, Pengpeng Zhao, and Xiaofang Zhou. 2018. LC-RNN: A Deep Learning Model for Traffic Speed Prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. 3470–3476.

[11] Chuishi Meng, Xiuwen Yi, Lu Su, Jing Gao, and Yu Zheng. 2017. City-wide Traffic Volume Inference with Loop Detector Data and Taxi Trajectories. In *Proceedings of 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*.

[12] Alexander H. Miller, Adam Fisch, Jesse Dodge, AmirHossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-Value Memory Networks for Directly Reading Documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1400–1409.

[13] Laura Muñoz, Xiaotian Sun, Roberto Horowitz, and Luis Alvarez. 2003. Traffic density estimation with the cell transmission model. In *Proceedings of the 2003 American Control Conference, 2003*, Vol. 5. IEEE, 3750–3755.

[14] Jingbo Shang, Yu Zheng, Wenzhu Tong, Eric Chang, and Yong Yu. 2014. Inferring Gas Consumption and Pollution Emissions of Vehicles throughout a City. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1027–1036.

[15] Jack Snowdon, Olga Gkountouna, Andreas Zufle, and Dieter Pfoser. 2018. Spatiotemporal Traffic Volume Estimation Model Based on GPS Samples. In *Proceedings of the Fifth International ACM SIGMOD Workshop on Managing and Mining Enriched Geo-Spatial Data*. 1–6.

[16] Xianfeng Tang, Boqing Gong, Yanwei Yu, Huaxiu Yao, Yandong Li, Haiyong Xie, and Xiaoyu Wang. 2019. Joint Modeling of Dense and Incomplete Trajectories for Citywide Traffic Volume Inference. *arXiv preprint arXiv:1902.09255* (2019).

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, and ÅĄukasz Kaiser. 2017. Attention Is All You Need. In *Proceedings of 31st Conference on Neural Information Processing Systems*.

[18] Yang Wang, Yiwei Xiao, Xike Xie, Ruoyu Chen, and Hengchang Liu. 2018. Real-time Traffic Pattern Analysis and Inference with Sparse Video Surveillance Information. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. 3571–3577.

[19] Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. 2018. Deep Distributed Fusion Network for Air Quality Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*.

[20] Xianyuan Zhan, Yu Zheng, Xiuwen Yi, and Satish V. Ukkusuri. 2017. Citywide Traffic Volume Estimation Using Trajectory Data. *IEEE Transactions on Knowledge and Data Engineering* 29, 2 (February 2017), 272–285.

[21] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, Xiuwen Yi, and Tianrui Li. 2018. Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence* 259 (2018), 147–166.