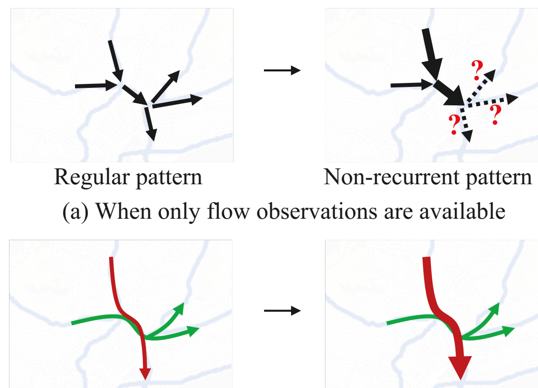
Traffic Flow Prediction with Vehicle Trajectories

Mingqian Li,[13*] Panrong Tong,[12] Mo Li,[12] Zhongming Jin,[3] Jianqiang Huang,[13] Xian-Sheng Hua[3]

[1] Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University, Singapore
[2] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[3] Alibaba Group
{mingqian001, tong0091, limo}@ntu.edu.sg, {zhongming.jinzm, jianqiang.hjq, xiansheng.hxs}@alibaba-inc.com

Abstract

This paper proposes a spatiotemporal deep learning framework, Trajectory-based Graph Neural Network (TrGNN), that mines the underlying causality of flows from historical vehicle trajectories and incorporates that into road traffic prediction. The vehicle trajectory transition patterns are studied to explicitly model the spatial traffic demand via graph propagation along the road network; an attention mechanism is designed to learn the temporal dependencies based on neighborhood traffic status; and finally, a fusion of multi-step prediction is integrated into the graph neural network design. The proposed approach is evaluated with a real-world trajectory dataset. Experiment results show that the proposed TrGNN model achieves over 5% error reduction when compared with the state-of-the-art approaches across all metrics for normal traffic, and up to 14% for atypical traffic during peak hours or abnormal events. The advantage of trajectory transitions especially manifest itself in inferring high fluctuation of flows as well as non-recurrent flow patterns.



Regular pattern — Non-recurrent pattern
(a) When only flow observations are available

Regular pattern — Non-recurrent pattern
(b) When vehicle trajectories are available

Figure 1: The challenge in predicting non-recurrent traffic flows and how vehicle trajectory information may help.

1 Introduction

Robust and accurate predictions of vehicular traffic conditions (e.g., flow, speed, density), either short-term or long-term, is necessary for transportation services such as traffic control and route planning. The challenge of traffic prediction primarily stems from the complex nature of spatiotemporal interactions among vehicles and the road network.

Data driven approaches have been extensively exploited in predicting vehicular traffic on the road network. Early attempts leverage time series analysis (Williams and Hoel 2003), which primarily models the temporal correlations of traffic. Conventional machine learning models (Sun, Zhang, and Yu 2006) are applied to learn the spatiotemporal correlations of traffic from historical data. Latest works apply deep learning to traffic prediction, and they typically follow a spatiotemporal framework, e.g., Graph Neural Network (GNN) (Li et al. 2018), which demonstrates superior capability in learning complicated spatiotemporal correlations.

According to studies in transportation domain (Skabardonis, Varaiya, and Petty 2003), the road traffic contains two parts: recurrent traffic, which often arises from periodic traffic demand such as daily commuters during morning and evening rush hours, and non-recurrent traffic, which is triggered by unexpected causes such as public transit disruptions or accidents. Existing approaches learn spatiotemporal traffic correlations from patterns that were seen in history, and thus are favourable in predicting recurrent traffic. In predicting non-recurrent traffic, however, existing approaches may fail to achieve the same level of accuracy, mainly due to insufficient observations of similar flow patterns in history. Figure 1(a) illustrates such an issue with an example - when only flow observations (i.e., the number of vehicles passing each road segment) are available, existing approaches may learn spatiotemporal correlations of recurrent flow patterns among different road segments across different time, which cannot effectively reason how a previously unseen part of the traffic is credited to future road traffic.

This paper targets at addressing the current challenge with non-recurrent traffic flow prediction - the challenge that historical flow data cannot provide insights on how non-recurrent traffic flows correlate in time and space. Thus,

[*]Mingqian Li is under the joint PhD Program with Alibaba Group and Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University, Singapore.

complementary to conventional spatiotemporal modeling using aggregated traffic flow observations, we also exploit intact vehicle trajectories to infer short-term traffic dependency. As suggested in Figure 1(b), trajectory data provide information of how each portion of a traffic flow transits from one road segment to another, and thus implies the dependency of upstream and downstream flows. The dependency embeds knowledge of how downstream traffic are caused by upstream traffic and thus may help infer traffic flow patterns that have not been seen before.

Incorporating trajectories to traffic flow prediction entails the following challenges: 1) trajectories are only observed from historical data; 2) traffic patterns at trajectory level need be aggregated properly to reflect the traffic patterns at flow level; and 3) future flows may deviate from observed patterns due to influence from the environment. To address those challenges, this paper proposes a novel model, Trajectory-based Graph Neural Network (TrGNN), which learns trajectory transition patterns from historical trajectories and incorporates that into a spatiotemporal graph-based deep learning framework. The main contributions of this paper are summarized as follows:

- We identify the challenge of predicting non-recurrent traffic flows, and we propose to incorporate vehicle trajectory data in traffic flow prediction. To the best of our knowledge, this is the first study that attempts to leverage vehicle trajectory data to mine the underlying causality of flows among roads.
- We design an end-to-end spatiotemporal graph-based deep learning model to predict traffic flows of the entire road network. Our model embeds trajectory transition into graph propagation along the road network to model the spatial traffic demand; it learns the temporal dependencies with an attention mechanism based on neighborhood traffic status; and finally it fuses multi-step predictions.
- We conduct extensive experiments [1] with real-world vehicle trajectory data and the results suggest that our model outperforms state-of-the-art approaches in terms of prediction errors across various scenarios (over 5% error reduction), and is especially superior in predicting non-recurrent flow patterns, e.g., during abnormal events (up to 14% error reduction).

The rest of the paper is organized as follows. Section 2 discusses related work on vehicular traffic prediction. Section 3 defines the problem and introduces some preliminary knowledge. The proposed model TrGNN is introduced in Section 4, and experimentally evaluated in Section 5. Section 6 concludes the paper.

2 Related Work

For decades, data driven approaches have been exploited to predict road traffic conditions, such as flow (Lv et al. 2014), speed (Li et al. 2018), density (Raj, Bahuleyan, and Vanajakshi 2016), accident rate (Sun and Sun 2015), and arrival time (Zhou, Zheng, and Li 2012).

Early attempts with time series analysis model the temporal correlations of traffic, such as SARIMA (Williams and Hoel 2003) and VAR (Chandra and Al-Deek 2009). Those approaches rely on strong assumptions of linearity and stationarity and often ignore the spatial impact from neighboring traffic. Another line of research focuses on studies of conventional machine learning models, such as k-NN (Davis and Nihan 1991), Bayesian network (Sun, Zhang, and Yu 2006), and SVR (Vanajakshi and Rilett 2004). In those models, spatiotemporal features are manually designed and extracted, and the models are often shallow in structure with limited learning capability.

Recent advances in deep learning have motivated its application in traffic prediction (Liu et al. 2018). Earlier neural network architectures include SAEs (Lv et al. 2014) and DBN (Jia, Wu, and Du 2016). State-of-the-art approaches typically follow a spatiotemporal framework: it models the spatial correlations by CNNs (Yu, Yin, and Zhu 2018) viewing the map as an image, or by GCNs (Li et al. 2018) viewing the road network as a graph; and it models the temporal evolution of traffic as a sequence of signals (Zhao et al. 2019). The spatiotemporal framework makes it flexible to embed auxiliary information such as weather conditions (Yao et al. 2018), accident data (Yu et al. 2017), map query records (Liao et al. 2018), and POIs (Geng et al. 2019). Similar to these works, our model follows a graph-based spatiotemporal deep learning framework; in addition, we incorporate vehicle trajectory data into the design to address the challenge of predicting non-recurrent flows.

Among deep learning approaches, Graph Wavenet (Wu et al. 2019) and SLC (Zhang et al. 2020) mine latent graph structures to capture long-range dependencies, and MRes-RGNN (Chen et al. 2019) designs a multiple hop scheme to capture long-term periodic patterns; the design goals of those methods deviate from our key objective of predicting non-recurrent traffic which is often caused by sudden disruptions locally. Other methods explore temporal building blocks and combine them with graph convolution, e.g., attention in ASTGCN (Guo et al. 2019) and GMAN (Zheng et al. 2020), gated recurrent unit in T-GCN (Zhao et al. 2019), temporal convolution in STGCN (Yu, Yin, and Zhu 2018). Most of these methods aim at reducing overall prediction errors without specific focus on non-recurrent traffic.

Few existing works leverage trajectory data in traffic flow prediction. Zhang et al. leverages trajectories in traffic state estimation, only to calibrate the embedding of road intersections (2019). In comparison, our work utilizes trajectories to mine the traffic flow transitions among road segments. Some work leverages trajectories for other purposes such as map generation (Ruan et al. 2020) which is less relevant to the topic of our study.

3 Preliminaries

We first define the problem of traffic flow prediction, and then introduce some preliminary knowledge of Graph Convolutional Networks (GCNs) and attention mechanism.

[1]Code and dummy data are available at https://github.com/mingqian000/TrGNN.

3.1 Problem Definition

In traffic flow prediction, the target is to predict future traffic flows given historical traffic flows on a static road network.

Definition 1 (Road Graph)**.** *The road network can be formulated into a directed road graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$. $\mathcal{V} = \{v_i\}_{i=1,2,...,M}$ is a finite set of nodes where each node v_i represents a road segment i, and $\mathcal{E} = \{e_{ij}\}$ is a set of directed edges where each edge $e_{ij} = (v_i, v_j)$ indicates that road segment i is an immediate upstream of road segment j, and $A \in [0,1]^{M\times M}$ represents the weighted road adjacency matrix. Each node v_i has a self-loop, i.e., $e_{ii} \in \mathcal{E}$.*

Definition 2 (Traffic Flows)**.** *Traffic flow is defined as the number of vehicles passing by a road segment during a specific time interval. Given a road graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, we use $X \in \mathbb{R}^{T\times|\mathcal{V}|}$ to represent the time series of traffic flows, where for each interval $t = 1, 2, ..., T$, $X_t \in \mathbb{R}^{|\mathcal{V}|}$ represents the traffic flows of all road segments in the road network during time interval t.*

Definition 3 (Trajectory)**.** *Given a road graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, we use $\mathcal{T} = (v_1, v_2, ..., v_I)$ to represent a trajectory of a vehicle, where each $v_i \in \mathcal{V}$ represents a road segment in the trajectory, satisfying $(v_i, v_{i+1}) \in \mathcal{E}, v_i \neq v_{i+1}, \forall i = 1, 2, ..., I-1$.*

Problem 1 (Traffic Flow Prediction)**.** *Given a road graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, find a prediction function $\hat{f}$ with parameter Θ such that given traffic flows $X_{(t-T_{in}+1):t}$ within a historical window period T_{in} up to time interval t, $\hat{f}$ estimates the most likely traffic flows $\hat{X}_{t+1}$ for the next time interval $t+1$, i.e.,*

$$\begin{aligned}\hat{X}_{t+1} &:= \hat{f}_\Theta(X_{t-T_{in}+1}, ..., X_t) \\ &\approx \underset{X_{t+1}}{\arg\max} \log p(X_{t+1}|X_{(t-T_{in}+1):t}),\end{aligned} \tag{1}$$

3.2 Graph Convolutional Networks (GCNs)

To model the spatial traffic demand, we leverage the idea of graph propagation from GCNs.

A GCN is defined over a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$. It applies convolutional operations on graph signals in spectral domain (Kipf and Welling 2017; Defferrard, Bresson, and Vandergheynst 2016). Given a graph signal $X \in \mathbb{R}^{|\mathcal{V}|\times N}$ where N is the number of features, a typical formulation of a K-hop graph convolutional layer is

$$GCN_\mathcal{G}(X; W, \theta) = \sigma(\sum_{k=0}^{K} \theta_k L^k X)W \tag{2}$$

where $L \in [0,1]^{|\mathcal{V}|\times|\mathcal{V}|}$ is the graph Laplacian, a variant of A, to control the graph propagation across nodes; $\theta \in \mathbb{R}^K$ controls the fusion of different hops; $W \in \mathbb{R}^{N\times M}$ (M is the output feature dimension) controls the fusion of different features; and σ is the activation function.

In this paper, to model traffic demand and traffic status, we adopt graph propagation (GraphProp), a simplified variant of GCN, with one single input feature (i.e., traffic flow) and thus ignoring the feature-wise parameters W:

$$GraphProp(X, A; K) := [X\|AX\|A^2X\|...\|A^KX]. \tag{3}$$

Instead of directly learning θ to fuse different hops of traffic demand, we define an attention mechanism to learn the temporal weights, as illustrated in Section 3.3.

3.3 Attention Mechanism

In learning a weighted sum of values, an attention mechanism (Vaswani et al. 2017; Veličković et al. 2018) replaces the weight parameters with a learning module in which the same set of parameters (called keys) are shared across all values in calculating the weights. A typical formulation of an attention mechanism given queries $Q \in \mathbb{R}^{N\times d_k}$, keys $K \in \mathbb{R}^{M\times d_k}$ and values $V \in \mathbb{R}^{M\times d_v}$ is

$$Attention(Q, K, V) := softmax(QK^\intercal)V. \tag{4}$$

In this paper, we apply the attention mechanism for a more flexible fusion of traffic demand based on traffic status.

4 Methodology

In this section, we introduce our proposed model, TrGNN, to address the problem of traffic flow prediction (Problem 1). The model follows a spatiotemporal framework, leveraging trajectory transition patterns. The extraction of trajectory transition is illustrated in Figure 2. An overview of the model architecture is illustrated in Figure 3. We elaborate each part of the architecture in the subsections below.

4.1 Trajectory Transition

Figure 1 illustrates the extra information gain from historical trajectory data when inferring non-recurrent traffic patterns. Compared to flows, trajectories provide essential knowledge about drives' origin and destination (O-D), and help infer their choices of routes at road intersections. Figure 1(b) visualizes a contrast between green and dark red trajectories, which indicates that vehicles coming from different upstream road segments (or origins) may differ in their distributions of downstream road segments (or destinations). Hence, for non-recurrent traffic patterns, we may infer flows on a trajectory basis: first obtain the origins of the existing vehicles, and then based on their origins, infer the distribution of their destinations. In Figure 1(b), for example, we may infer that the extra spike of flow is more likely caused by extra vehicle flow of dark red trajectories instead of green trajectories.

We utilize historical trajectories to explicitly model the transition of flows between upstream and downstream road segments. Figure 2 illustrates the extraction of trajectory transition. We view historical trajectories as Markov processes, and by aggregating trajectories of all vehicles, we may infer the transition of flows from one road segment to others. The rest of this subsection elaborates the extraction of trajectory transition.

The trajectory generation of a vehicle can be modeled as a first-order Markov process, assuming that the transition probability from each upstream road segment to each downstream road segment is stationary across days. We define a trajectory transition tensor $\mathcal{P} \in \mathbb{R}^{T_d\times|\mathcal{V}|\times|\mathcal{V}|}$, where each

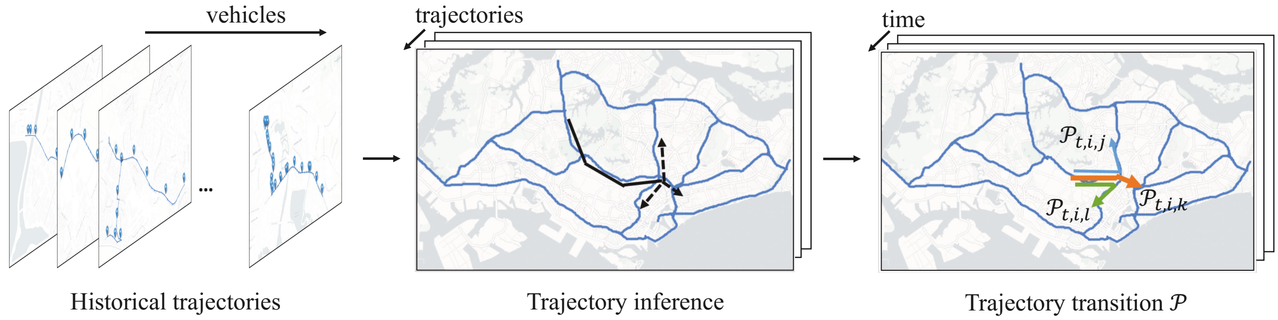


Figure 2: Learning trajectory transition from historical trajectories.

$\mathcal{P}_t \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ represents the trajectory transition probabilities for the t^{th} time interval of the day. The trajectory generation process can be represented as

$$\begin{aligned} P(\mathcal{T}|t) &= P((v_1, v_2, ..., v_I)|t) \\ &= \pi(v_1) \prod_{i=1}^{I-1} P(v_{i+1}|v_i; t) \\ &= \pi(v_1) \prod_{i=1}^{I-1} \mathcal{P}_{t,v_i,v_{i+1}}. \end{aligned} \tag{5}$$

Alternatively, $\mathcal{P}$ can be derived from a higher-order Markov process, which would require larger sample size and higher computational complexity.

To estimate tensor $\mathcal{P}$, we collect historical trajectories of all vehicles from the training set, and aggregate the cumulative transition probability with respect to time of day, upstream road segment, and downstream road segment:

$$\hat{\mathcal{P}}_{t,v_i,v_j} = \frac{\#vehicles\,(v_i \to v_j|t) + \mathbf{1}[v_j \in \mathbf{N}(v_i)]}{\#vehicles(v_i|t) + \|\mathbf{N}(v_i)\|}, \tag{6}$$

where t stands for the t^{th} time interval of day, and $\mathbf{N}(v_i)$ denotes the set of downstream neighbors of road segment v_i. To mitigate data sparsity, $\hat{\mathcal{P}}$ is smoothed out by adding a constant 1 for any pair of consecutive road segments.

The trajectory transition tensor $\mathcal{P}$ summarizes the probability distribution of drivers' choices of routes. In a macroscopic view, it approximates the transition of flows from upstream to downstream road segments in near future, and serves as a lookup table in the proposed TrGNN model.

4.2 Spatial Modeling of Traffic Demand

Based on trajectory transition, we design a graph propagation mechanism to infer the traffic demand in the spatial domain. Traffic demand refers to the short-range and long-range destinations of existing vehicles on the road network.

We leverage graph propagation from Graph Convolutional Networks (Section 3.2) to simulate the transition of vehicles along the road network. We perform graph propagation in d hops, resulting in a graph of traffic demand for each hop. For each input time interval t, we can derive traffic demand $D_t \in \mathbb{R}^{|\mathcal{V}|\times(d+1)}$ via graph propagation:

$$\begin{aligned} D_t &= GraphProp(X_t, \mathcal{P}_t^{\intercal}; d) \\ &= [X_t \,\|\, \mathcal{P}_t^{\intercal} X_t \,\|\, (\mathcal{P}_t^{\intercal})^2 X_t \,\|\, ... \,\|\, (\mathcal{P}_t^{\intercal})^d X_t], \end{aligned} \tag{7}$$

where $\|$ denotes concatenation, $\cdot^{\intercal}$ denotes matrix transpose, and parameter d stands for demand hop, controlling the farthest possible destination.

The graph propagation simulates the propagation of flows along the road network, and as a result, the traffic demand D is an aggregation of the short-range and long-range destinations (in different hops) of all vehicles in existing flows.

4.3 Temporal Modeling of Traffic Demand based on Traffic Status

The modeling of traffic demand in Section 4.2, however, does not consider the propagation speed of flows, which should depend on traffic status. Traffic status refers to the overall traffic volume in the neighborhood of each road segment. If the traffic status is congested around a road segment (i.e., high volume of flows in the neighboring road segments), the propagation of flows along that road segment should be slow, and vice versa.

A temporal module is thus designed to infer how each hop of traffic demand, from short range to long range, corresponds to the future traffic flow in the targeted time interval. This is done by assigning a weight to each hop of traffic demand via an attention mechanism (Section 3.3) based on traffic status. Thus, we first obtain traffic status, and then build an attention mechanism based on traffic status.

For each input time interval t, we obtain traffic status $S_t \in \mathbb{R}^{|\mathcal{V}|\times(2^{s+1}-1)}$ via graph propagation in s hops from neighboring road segments. The graph propagation is done via dual random walk to incorporate both upstream and downstream traffic:

$$\begin{aligned} S_t &= GraphProp_{dual}(X_t, \tilde{A}; s) \\ &= [X_t \,\|\, \tilde{A}^{\intercal} X_t \,\|\, \tilde{A} X_t \,\|\, \tilde{A}^{\intercal}\tilde{A}^{\intercal} X_t \,\|\, ... \,\|\, \tilde{A}^s X_t], \end{aligned} \tag{8}$$

where $\tilde{A}$ is a normalized variant of the weighted road adjacency matrix A, and parameter s stands for status hop, controlling the radius of the neighborhood.

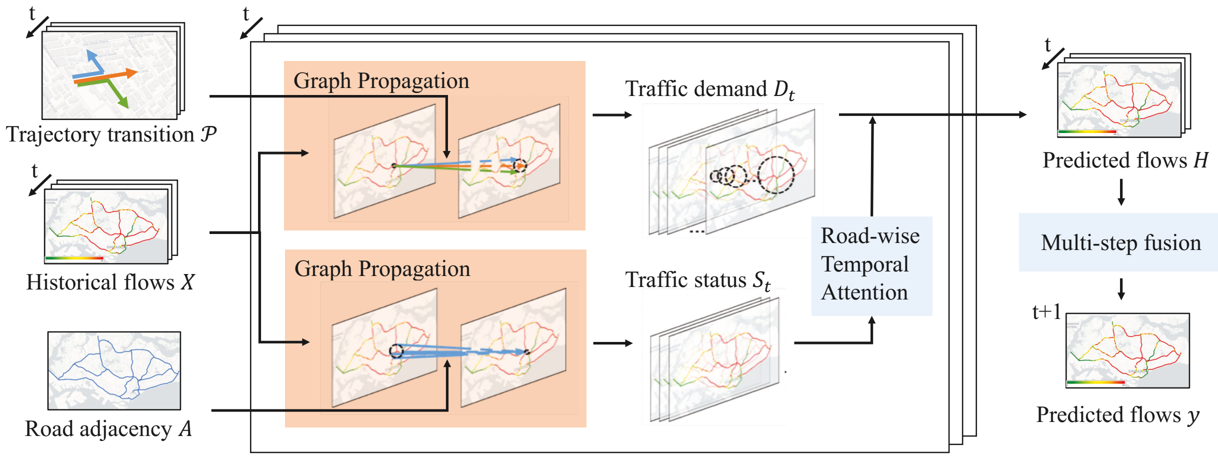


Figure 3: Trajectory-based Graph Neural Networks (TrGNN). The framework models spatial traffic demand via graph propagation based on trajectory transition, and models temporal dependencies via attention mechanism based on neighborhood traffic status. The final prediction is a fusion of multi-step prediction.

We apply a road-wise attention mechanism (referring to the dot-product attention in (Vaswani et al. 2017)) parameterized by keys $K \in \mathbb{R}^{|\mathcal{V}|\times(2^{s+1}-1)\times(d+1)}$, taking traffic status S_t as queries and traffic demand D_t as values, to assign weights $\alpha \in [0,1]^{|\mathcal{V}|\times(d+1)}$ to different hops in traffic demand D_t and take the weighted sum as an initial prediction of flows $H_t \in \mathbb{R}^{|\mathcal{V}|}$:

$$\begin{aligned} H_t &= Attention(S_t, D_t; K) \\ &= \sum_{i=0}^{d} \alpha_{:,i} \odot D_{t,:,i} \\ &= \sum_{i=0}^{d} [softmax(S_t \circ K)]_{:,i} \odot D_{t,:,i} \end{aligned} \tag{9}$$

where $softmax(\cdot)$ is applied over the dimension of demand hop, $\circ$ denotes road-wise matrix product, and $\odot$ denotes element-wise (or Hadamard) product.

4.4 Multi-Step Fusion

From a sequence of input flows $\{X_i\}_{i=t-T_{in}+1,\ldots,t}$, we obtain a sequence of initial predictions $H \in \mathbb{R}^{T_{in}\times|\mathcal{V}|}$. The final layer of the model is a temporal fusion of H. We adopt a road-wise fully connected layer. For each road segment v,

$$\begin{aligned} y_v &:= X_{t+1,v} \\ &= FullyConnected(H_{:,v}; \Theta) \\ &= \Theta^{\intercal} H_{:,v}. \end{aligned} \tag{10}$$

Alternatively, this layer can be replaced by any RNN cell such as LSTM (Hochreiter and Schmidhuber 1997) or GRU (Chung et al. 2014)), or sequence modeling (Sutskever, Vinyals, and Le 2014), for a longer-term prediction.

As a side note, the conservation of vehicles on the road network does not hold in practice. Future flows not only depend on trajectory transition within the road network, but also depend on new vehicles entering the road network (e.g., entering from the boundary of the region, or entering from a local road to an arterial road) and existing vehicles leaving the road network, which we call boundary flows. Since the boundary flows are strongly associated with drivers' O-D demand which is periodic, we embed some periodic features (e.g., time of day, is working day) into the multi-step fusion module to model the boundary flows.

5 Experimental Evaluation

5.1 Dataset Description

We evaluate our model with SG-TAXI, a real-world dataset comprising GPS mobility traces from over 20,000 taxis in Singapore. The dataset is provided by Singapore Land Transport Authority. We collect the GPS readings of all active taxis for a period of 8 weeks (14th Mar-8th May 2016). Each GPS reading comprises vehicle id, longitude, latitude, and timestamp. The road network comprises 2,404 road segments, covering all expressways in Singapore.

5.2 Data Preprocessing

We preprocess the SG-TAXI dataset in 4 steps:

1. **Road graph formulation.** For the road graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, we calculate the weighted road adjacency matrix A as the exponential decay of distance between roads.
2. **Map matching.** We apply the Hidden Markov map matching algorithm (Newson and Krumm 2009) to correct GPS readings to their corresponding road segments.
3. **Trajectory cleansing.** Given a sequence of mapped GPS points, we cleanse the vehicle's trajectory as follows: 1) eliminate duplicate records; 2) if GPS reading is off for over 10 minutes (e.g., the driver turns off the sensing device), split the trajectory; 3) if driver stays on the same road segment for over 2 minutes, split the trajectory; 4) if

no path exists between two consecutive GPS points (e.g., driver drives off the road network), split the trajectory; and 5) remove GPS points with extreme speed (i.e., speed derived from two consecutive GPS points exceeds 120 km/hr). Finally, we recover the full trajectory via Dijkstra's algorithm (Dijkstra et al. 1959).

4. **Flow aggregation.** We aggregate trajectories into flows per road segment per 15-minute interval. We calibrate flows to correct the daily fluctuation in taxi arrangement and better represent the overall traffic flows in Singapore.

5.3 Experiment Settings

The model is trained on the preprocessed SG-TAXI dataset. The train-validate-test split is 5-1-2 week. Each data point consists of input flows for 4 intervals (i.e., 1 hour) and output flows for 1 interval (i.e., 15 minute). Flows are normalized before being input into the model. For hyperparameters, the demand hop d is set to 75, i.e., the maximum number of road segments that a vehicle with a normal speed could traverse within a 15-minute interval, and the status hop s is set to 3.

The model is implemented in PyTorch (Paszke et al. 2019) on a single Tesla P100 GPU and is trained using Adam optimizer (Kingma and Ba 2014) to minimize MSE loss. The learning rate is initially set to 0.004 and is halved every 30 epochs. The maximum epochs to train is set to 100. Early stopping is applied on validation MAE. The training takes less than 4GB RAM and less than 1GB GPU memory.

5.4 Baseline Approaches

Our model TrGNN is compared to representative baseline methods of each type, including naive methods (HA, MA), time series analysis (VAR), conventional machine learning (RF), and deep learning (T-GCN, STGCN, DCRNN). Specifically, (i) **HA** (Historical Average) is the average flow of the same time on the same day in the past four weeks; (ii) **MA** (Moving Average) is the average flow of the previous 1 hour; (iii) **VAR** (Vector Auto-Regression) (Hamilton 1994) models the future flow as a linear combination of historical flows in 5-hop neighborhood, implemented in StatsModels (Seabold and Perktold 2010); (iv) **RF** (Random Forest) is a decision-tree-based ensemble method that fits a piece-wise function on historical flows in 5-hop neighborhood, implemented in Scikit-learn (Pedregosa et al. 2011) with 100 trees; (vi) **T-GCN** (Temporal Graph Convolutional Network) (Zhao et al. 2019) is a graph-based neural network that integrates GCN with GRU, implemented in Tensorflow [2]; (vii) **STGCN** (Spatio-Temporal Graph Convolutional Networks) (Yu, Yin, and Zhu 2018) is a graph-based neural network that models both spatial and temporal dependencies via convolution, implemented in Pytorch [3]; and (viii) **DCRNN** (Diffusion Convolutional Recurrent Neural Network) (Li et al. 2018) is a graph-based neural network that integrates diffusion convolution on graph with sequence learning, implemented in PyTorch [4].

[2]https://github.com/lehaifeng/T-GCN
[3]https://github.com/FelixOpolka/STGCN-PyTorch
[4]https://github.com/chnsh/DCRNN_PyTorch

5.5 Evaluation Metrics

We evaluate prediction results by three error metrics: MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), and RMSE (Root Mean Squared Error), same as in (Li et al. 2018). Lower errors indicate better performance.

5.6 Results and Analysis

Table 1 summarizes the evaluation of different approaches for traffic flow prediction on SG-TAXI dataset. The comparison covers overall testing as well as specific scenarios including peak hours, non-peak hours and MRT breakdown.

Overall Performance. According to Table 1, the overall prediction errors of our model TrGNN are 26.43/0.30/38.65 vph for MAE/MAPE/RMSE, and TrGNN achieves over 5% error reduction from baselines across all metrics. The naive baselines generally give high errors, as they consider only temporal correlations of flows; MA is more accurate than HA, indicating that near-past flows play a stronger role than periodicity. VAR and RF perform better than the naive baselines, as they incorporate neighborhood flows to model spatial correlations; in particular, RF performs better than VAR, implying that flows are not linearly correlated. For deep learning, DCRNN achieves the best results out of all baselines, indicating the capability of graph-based deep learning in capturing the spatiotemporal correlations. Finally, TrGNN outperforms all existing baselines in all metrics, which verifies the effectiveness of learning spatiotemporal transition of flows from trajectories.

The line plot in Figure 4 visualizes predicted flows of TrGNN and a few representative baselines. HA fits the worst to the ground truths, implying high variation of flows from week to week; while TrGNN and DCRNN are more sensitive to the real-time fluctuations of flows. If we look further into the peak hours indicated in the dashed box, TrGNN captures the fluctuations of flows slightly earlier than DCRNN.

Peak hours and non-peak hours. We select two typical periods for experiments: peak hours (8-10pm on working days, when public transport services become limited and the demand for taxis increases, thus with high fluctuation of flows); and non-peak hours (2-4pm on working days when people stay at offices and the demand for taxis stabilizes, thus with low fluctuation of flows). Results are summarized in Table 1 (under "Peak hours" and "Non-peak hours" column). In peak hours, absolute errors (MAE/RMSE) are consistently higher than in overall testing; while in non-peak hours, the results are the opposite. This meets our expectation, as predicting peak hour flows is more challenging due to higher fluctuation in traffic demand. In both peak hours and non-peak hours, TrGNN outperforms all baselines, and the error reduction of TrGNN is more significant during peak hours (6-13% reductions on the performance metrics).

Figure 4 displays the heatmap snapshots of the prediction errors of HA, DCRNN and TrGNN on the entire road network during selected peak hours. The color indicates increasing prediction error from green to red. A comparison of the heatmap snapshots suggests the robustness of TrGNN in capturing the periodic fluctuation of flows in peak hours.

Abnormal event: MRT breakdown. We analyze an abnormal event in Singapore, an MRT (Mass Rapid Transit)

Method	Overall			Peak hours			Non-peak hours			MRT breakdown		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
HA	33.74	0.34	52.58	36.83	0.25	55.02	32.53	0.28	48.67	**40.07**	0.27	**59.34**
MA	31.55	0.35	47.69	36.14	0.26	53.18	28.18	0.27	39.41	44.85	0.30	71.43
VAR	29.27	0.33	43.22	34.23	0.24	49.71	28.10	0.26	39.28	40.68	**0.27**	64.41
RF	29.26	0.33	43.38	34.13	**0.24**	49.75	**27.53**	**0.26**	**38.53**	42.28	0.28	66.53
T-GCN	31.12	0.35	45.69	36.57	0.27	52.91	30.03	0.29	41.53	42.38	0.30	67.39
STGCN	29.88	0.33	44.51	34.86	0.24	50.86	27.94	0.27	39.05	42.19	0.28	66.40
DCRNN	**29.01**	**0.31**	**43.12**	**33.74**	0.25	**48.88**	27.75	0.27	38.74	40.39	0.28	64.28
TrGNN-	27.34	0.31	40.05	31.35	0.23	45.11	26.61	0.26	37.20	38.57	0.27	59.53
TrGNN	**26.43**	**0.30**	**38.65**	**29.81**	**0.23**	**42.62**	**25.65**	**0.25**	**35.68**	**34.56**	**0.25**	**54.31**
%diff	-9%	-5%	-10%	-12%	-6%	-13%	-7%	-4%	-7%	-14%	-8%	-8%

Numbers in bold denote the best baseline performance and the best performance.
%diff denotes the error reduction of TrGNN from the best baseline performance.

Table 1: Performance of different approaches for traffic flow prediction on SG-TAXI dataset.

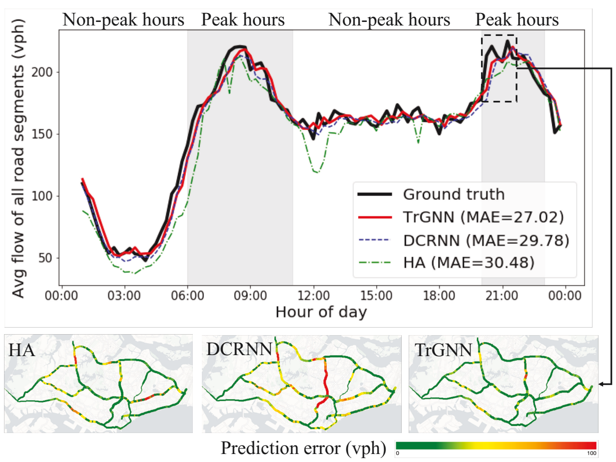


Figure 4: Line plot of predicted flows on the road network over a working day, and heatmap snapshots of prediction errors during peak hours.

breakdown, when train services were disrupted due to power fault (Chew 2016). The disruption falls on a Monday night lasting for more than one hour, and it affects 52 train stations on 4 train lines, covering the whole area of west Singapore. In Figure 5, the heatmap visualizes the abnormal spike of flows of the affected region due to the increase in taxi demand during the MRT breakdown period, and the line plot visualizes the predicted flows on a sample abnormal road segment - TrGNN fits the best to the ground truths.

We select road segments within a 3km neighborhood of any affected train station, and summarize their prediction results during the breakdown period in Table 1 (under “MRT breakdown” column). Compared to “Overall”, we observe a significant increase in MAE for all baselines, ranging from 19% to 44%, which demonstrates the performance drop in predicting abnormal flows. Nevertheless, TrGNN outperforms baselines by a significant error reduction of 14%. The result suggests the capability of TrGNN in capturing the spatiotemporal causality even for non-recurrent flow patterns,

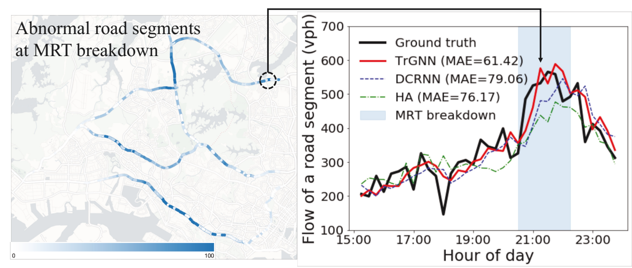


Figure 5: Heatmap of abnormal flows in west Singapore due to MRT breakdown, and line plots of abnormal flows and predicted flows on a road segment. In heatmap, the color scale indicates the amount of extra flow compared to that of a normal day.

instead of simply memorizing the historical flow patterns.

Component analysis: trajectory transition. To analyze the utility of trajectories, we build a variant of TrGNN, TrGNN-, that replaces trajectory transition by the static road network. Results in Table 1 show that compared to TrGNN-, TrGNN reduces the prediction errors on all metrics in all scenarios, especially in MRT breakdown where MAE drops from 38.57 vph to 34.56 vph. This verifies the effectiveness of trajectory transition in capturing flow dependency.

6 Conclusion and Future Work

This paper proposes a spatiotemporal deep learning model, Trajectory-based Graph Neural Network (TrGNN), to solve the traffic flow prediction problem. The architecture leverages historical trajectory transition as an input into the graph-based deep learning framework. TrGNN is evaluated on SG-TAXI dataset. Results show that TrGNN outperforms state-of-the-art approaches, especially being superior in predicting non-recurrent traffic flows such as in MRT breakdown event. Potential future work includes expansion to a higher-order Markov model, longer-term prediction, and optimization of computational complexity in extracting trajectories. Moreover, our work points out a promising direction in incorporating trajectory data into traffic prediction.

Acknowledgments

This research is supported, in part, by NRF Singapore under its grant SDSC-2019-001, Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), and Singapore MOE Tier 1 grant RG18/20. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of funding agencies.

References

Chandra, S. R.; and Al-Deek, H. 2009. Predictions of Freeway Traffic Speeds and Volumes Using Vector Autoregressive Models. *Journal of Intelligent Transportation Systems* 13(2): 53–72.

Chen, C.; Li, K.; Teo, S. G.; Zou, X.; Wang, K.; Wang, J.; and Zeng, Z. 2019. Gated Residual Recurrent Graph Neural Networks for Traffic Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 485–492.

Chew, H. M. 2016. Train Service Disruptions on Three MRT Lines and Bukit Panjang LRT Due to Power Fault: SMRT. *The Straits Times* URL https://www.straitstimes.com/singapore/transport/train-disruption-on-east-west-line-and-north-south-line-due-to-traction-power.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv preprint arXiv:1412.3555* .

Davis, G. A.; and Nihan, N. L. 1991. Nonparametric Regression and Short-term Freeway Traffic Forecasting. *Journal of Transportation Engineering* 117(2): 178–188.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 29, 3844–3852. Curran Associates, Inc.

Dijkstra, E. W.; et al. 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik* 1(1): 269–271.

Geng, X.; Li, Y.; Wang, L.; Zhang, L.; Yang, Q.; Ye, J.; and Liu, Y. 2019. Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3656–3663.

Guo, S.; Lin, Y.; Feng, N.; Song, C.; and Wan, H. 2019. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 922–929.

Hamilton, J. D. 1994. *Time Series Analysis*, volume 2. Princeton University Press Princeton, NJ.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-term Memory. *Neural computation* 9(8): 1735–1780.

Jia, Y.; Wu, J.; and Du, Y. 2016. Traffic Speed Prediction Using Deep Learning Method. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 1217–1222. IEEE.

Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* .

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations (ICLR)*.

Liao, B.; Zhang, J.; Wu, C.; McIlwraith, D.; Chen, T.; Yang, S.; Guo, Y.; and Wu, F. 2018. Deep Sequence Learning with Auxiliary Information for Traffic Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 537–546.

Liu, Z.; Li, Z.; Wu, K.; and Li, M. 2018. Urban Traffic Prediction from Mobility Data Using Deep Learning. *IEEE Network* 32(4): 40–46.

Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; and Wang, F.-Y. 2014. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems* 16(2): 865–873.

Newson, P.; and Krumm, J. 2009. Hidden Markov Map Matching Through Noise and Sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 336–343.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, 8026–8037.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12: 2825–2830.

Raj, J.; Bahuleyan, H.; and Vanajakshi, L. D. 2016. Application of Data Mining Techniques for Traffic Density Estimation and Prediction. *Transportation Research Procedia* 17: 321–330.

Ruan, S.; Long, C.; Bao, J.; Li, C.; Yu, Z.; Li, R.; Liang, Y.; He, T.; and Zheng, Y. 2020. Learning to Generate Maps from Trajectories. *Proceedings of the AAAI Conference on Artificial Intelligence* 34(01): 890–897.

Seabold, S.; and Perktold, J. 2010. Statsmodels: Econometric and Statistical Modeling with Python. In *9th Python in Science Conference*.

Skabardonis, A.; Varaiya, P.; and Petty, K. F. 2003. Measuring Recurrent and Nonrecurrent Traffic Congestion. *Transportation Research Record* 1856(1): 118–124.

Sun, J.; and Sun, J. 2015. A Dynamic Bayesian Network Model for Real-Time Crash Prediction Using Traffic Speed Conditions Data. *Transportation Research Part C: Emerging Technologies* 54: 176–186.

Sun, S.; Zhang, C.; and Yu, G. 2006. A Bayesian Network Approach to Traffic Flow Forecasting. *IEEE Transactions on Intelligent Transportation Systems* 7(1): 124–132.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*.

Vanajakshi, L.; and Rilett, L. R. 2004. A Comparison of the Performance of aAtificial Neural Networks and Support Vector Machines for the Prediction of Traffic Speed. In *IEEE Intelligent Vehicles Symposium*, 194–199. IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention Is All You Need. *Advances in Neural Information Processing Systems* 30: 5998–6008.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations (ICLR)*.

Williams, B. M.; and Hoel, L. A. 2003. Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results. *Journal of Transportation Engineering* 129(6): 664–672.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 1907–1913.

Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; Ye, J.; and Zhenhui, L. 2018. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*.

Yu, R.; Li, Y.; Shahabi, C.; Demiryurek, U.; and Liu, Y. 2017. Deep learning: A Generic Approach for Extreme Condition Traffic Forecasting. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, 777–785. SIAM.

Zhang, Q.; Chang, J.; Meng, G.; Xiang, S.; and Pan, C. 2020. Spatio-Temporal Graph Structure Learning for Traffic Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* 34(01): 1177–1185.

Zhang, X.; Xie, L.; Wang, Z.; and Zhou, J. 2019. Boosted Trajectory Calibration for Traffic State Estimation. In *2019 IEEE International Conference on Data Mining (ICDM)*, 866–875. IEEE.

Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; and Li, H. 2019. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems* .

Zheng, C.; Fan, X.; Wang, C.; and Qi, J. 2020. GMAN: A Graph Multi-Attention Network for Traffic Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence* 34(01): 1234–1241.

Zhou, P.; Zheng, Y.; and Li, M. 2012. How Long to Wait? Predicting Bus Arrival Time With Mobile Phone Based Participatory Sensing. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, 379–392.