Graph Neural Network for Robust Public Transit Demand Prediction

Can Li, Lei Bai, Wei Liu, Lina Yao, *Member, IEEE*, and S Travis Waller

***Abstract*— Understanding and forecasting mobility patterns and travel demand are fundamental and critical to efficient transport infrastructure planning and service operation. However, most existing studies focused on deterministic demand estimation/prediction/analytics. Differently, this study provides confidence interval based demand forecasting, which can help transport planning and operation authorities to better accommodate demand uncertainty/variability. The proposed Origin-Destination (OD) demand prediction approach well captures and utilizes the correlations among spatial and temporal information. In particular, the proposed Probabilistic Graph Convolution Model (PGCM) consists of two components: (i) a prediction module based on Graph Convolution Network and combined with the gated mechanism to predict OD demand by utilizing spatio-temporal relations; (ii) a Bayesian-based approximation module to measure the confidence interval of demand prediction by evaluating the graph-based model uncertainty. We use a large-scale real-world public transit dataset from the Greater Sydney area to test and evaluate the proposed approach. The experimental results demonstrate that the proposed method is capable of capturing the spatial-temporal correlations for more robust demand prediction against several established tools in the literature.**

***Index Terms*— Probabilistic demand prediction, public transit, graph convolution network, Bayesian inference.**

I. Introduction

With the development of information and communication technology, growing big data sources (e.g., smart public transit cards with location information, social media platform data) provide new opportunities to identify and understand travel patterns (e.g., travel demand profile [1], travel time distribution [2], and travel patterns analysis [3]). In particular, smart transit card usage records are continuously generated in many cities around the world (e.g., Opal card in Sydney, Octopus card in Hong Kong, Oyster card in London), which provide valuable data sources for estimating/forecasting public transit demand. The travel demand and mobility patterns are often the most critical inputs to transport infrastructure/services planning and operation optimization problems (e.g., dedicated bus lanes, bus/rail lines, scheduling), which further determine efficiency, reliability, and attractiveness of public transit services.

Travel demand and mobility patterns vary from time to time and exhibit a strong level of uncertainty. Understanding, forecasting, and incorporating the demand uncertainty are both important and necessary to optimally determine, e.g., bus fleet size, vehicle size, public transit lines and networks, and scheduling. Besides, uncertainties in public transport can result in bus bunching and unbalanced waiting flows across different stations or stops. Although there have been fruitful studies on predicting passenger demand for public transport [4], existing studies often focused on deterministic demand forecasting or mobility analytics. The confidence interval or precision level of the estimates has received much fewer attention [5]. This paper aims to explicitly consider the confidence interval for demand prediction or alternatively speaking, quantify the precision level of demand prediction. The proposed "confidence interval" is an estimated interval for the demand, which is associated with a target confidence level and computed based on the observed data. It can be helpful in future service planning and operation optimization problems since it provides a more reliable representation of demand, which directly relates to and affects the robustness of the planning and operation decisions. Robust service design is useful to reduce the negative effects of demand variations and service disturbance [6].

In particular, this study proposes a Probabilistic Graph Convolution Model for forecasting Origin-Destination (OD) demand in the public transit system with a target confidence interval. The proposed model is able to capture and take advantage of the underlying spatio-temporal correlations in the dataset. As mentioned earlier, we propose the "confidence interval" for demand prediction, which incorporates the variability of travel demand and reflects the precision level of the demand prediction.

Specifically, we divide the whole city into multiple regions based on the postcodes (which define administrative regions). The setting of origins and destinations is based on this partitioning or division of the city. Note that different sizes or types of regions can be readily considered (e.g., we also test station-to-station demand prediction). Our numerical experiments

Manuscript received December 16, 2019; revised July 11, 2020, September 30, 2020, and November 22, 2020; accepted November 23, 2020. This work was supported by the Australian Research Council under Grant DE200101793. The Associate Editor for this article was Y. Kamarianakis. *(Corresponding author: Wei Liu.)*

Can Li, Lei Bai, and Lina Yao are with the School of Computer Science and Engineering, UNSW Sydney, Sydney, NSW 2052, Australia (e-mail: can.li4@student.unsw.edu.au; lei.bai@student.unsw.edu.au; lina.yao@unsw.edu.au).

Wei Liu is with the School of Computer Science and Engineering, UNSW Sydney, Sydney, NSW 2052, Australia, and also with the Research Centre for Integrated Transport Innovation, School of Civil and Environmental Engineering, UNSW Sydney, Sydney, NSW 2052, Australia (e-mail: wei.liu@unsw.edu.au).

S Travis Waller is with the Research Centre for Integrated Transport Innovation, School of Civil and Environmental Engineering, UNSW Sydney, Sydney, NSW 2052, Australia (e-mail: s.waller@unsw.edu.au).

Digital Object Identifier 10.1109/TITS.2020.3041234

mainly consider the administrative region level demand predictions, which can be useful in practice for policy making at the region-level since many relevant planning and management activities are administrative-region-based.

Given the setting of origins and destinations, we can define the OD demand vector, where one OD pair corresponds to one element of this vector. Then we construct a graph-based network with nodes and edges, where a node in the graph corresponds to one OD pair, and an edge is added between two nodes (two nodes in the graph correspond to two different OD pairs) if the demand patterns for these two OD pairs exhibit sufficient similarities. A series of gated graph convolution layers are then applied to capture the spatial and temporal correlations in the demand profile simultaneously. Moreover, we utilize the Bayesian-based module to quantify the uncertainty associated with the proposed graph-based model. The graph-based model uncertainty is the uncertainty due to imperfections made in the model formulations [7]. And the model uncertainty can often be linked to prediction uncertainty [8], which can be utilized to produce the proposed "confidence interval" for demand prediction with a target level of confidence.

The main highlights of this paper are summarized in the following:

1) This study adopts a graph convolution deep network to capture the spatial-temporal correlations for public transit OD demand forecasting. Most existing methods often utilized Recurrent Neural Network (RNN) based models or its variants to analyze time-series features for demand prediction. However, this often consumes a large number of iterations and can yield large accumulative errors in the forecasting process since the previous time steps often directly influence the prediction in the next time step [9].
2) This study applies a Bayesian-based method to capture the model uncertainty and produce an interval of the predicted demand. This provides a powerful tool to incorporate demand variability and helps address planning and operation issues related to demand uncertainty and decision robustness. To the best of our knowledge, this is the first work to predict the demand interval through measuring the uncertainty of the graph-based model.
3) This study tests and evaluates the developed method on a large-scale real-world public transit system dataset collected in a large metropolitan area (Greater Sydney area) and demonstrates the effectiveness of the method against several baseline methods and state-of-the-art strategies.

The rest of this paper is organized as follows. Section II introduces related works in the literature and Section III defines the OD demand prediction problem. Then, Section IV presents the Probabilistic Graph Convolution Model and corresponding techniques. The test and evaluation of the proposed method and comparison with other methods are presented in Section V. Finally, Section VI concludes the paper.

II. Related Work

In this section, we review relevant works on demand prediction/estimation and Bayesian-based models.

A. Demand Prediction

Conventional OD matrices estimation by transport scientists often focused on demand prediction with limited observations or no direct observation of OD demand [10]. For example, many studies used traffic counts on a small number of road links to infer OD demand, where one may adopt Maximum Entropy approach [11], Generalized Least Squares approach [12], or Bayesian updating approach [13]. These approaches often assumed that travellers followed certain rules (e.g., shortest path) when choosing routes and then computed the OD demand patterns that will likely result in the observed link flow patterns. Differently, this paper focuses on forecasting OD demand based on the true observations of OD demand.

With real demand observations, traditional time series regression models such as Autoregressive Integrated Moving Average (ARIMA), Kalman Filter, and their variants [4], [14]–[17] have been widely used for forecasting passenger demand. For example, Kalman-filter-based models [14], [15] were employed to estimate the traffic flow under various traffic situations. Similarly, Seasonal ARIMA (SARIMA) model coupled with the Kalman filter was adopted in [16], which helped achieve good-quality demand predictions. More recently, an Interactive Multiple Model-based Pattern Hybrid (IMMPH) approach was utilized in [4] to predict short-term passenger demand based on different temporal relevant pattern time series. A Local Ensemble Transformed Kalman Filter by extending the Kalman Filter theory was proposed by [17] for dynamic demand forecasting. However, these strategies were often less capable of capturing the non-linear temporal and/or spatial correlations in the data for demand prediction purpose.

Recently, deep neural networks are adopted to better capture the non-linear effects/correlations in the data for passenger demand forecasting. RNN, Long Short-Term Memory Model (LSTM) and their variants were used to capture non-linear temporal correlations and predict demand [18]–[20]. For instance, LSTM was applied to predict future taxi requests in [18], where additional relevant information (i.e weather, time) is incorporated. These studies often focused on capturing temporal relations for demand prediction but ignored spatial information, which is related to the demand distribution and can be helpful in demand forecasting. Therefore, in more recent studies, Convolution Neural Network (CNN) was further utilized to capture the spatial correlations and combined with temporal models for demand forecasting [1], [21]–[25]. For example, an end-to-end multi-task model for demand prediction was proposed in [21], where CNN is used to extract spatial correlations and external factors such as weather conditions are incorporated to enhance the prediction accuracy. A hexagon-based ensemble mechanism with CNN was developed in [1] to enhance demand prediction performance for on-demand services. The spatio-temporal recurrent convolutional networks (SRCNs) were also proposed for predicting long-term and short-term large-scale network traffic, which utilized LSTM to learn the temporal dynamics and CNN to learn the spatial dependencies of network-wide traffic [23]. More recently, a contextualized spatial-temporal network for

OD demand forecasting by utilizing CNN to learn spatial dependencies and Convolutional LSTM to analyze demand evolution was proposed in [24].

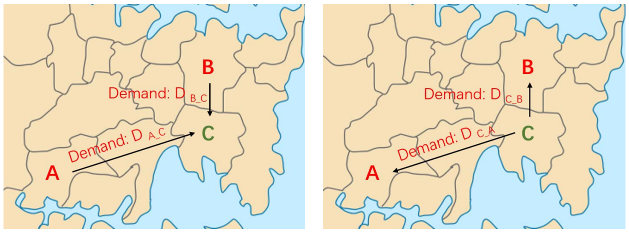
The CNN-based models often assume that the adjacent areas have similar demand features so they only modeled the Euclidean relationships among regions. Motivated by the proposal of graph convolution for signal processing in frequency domain [26] and vertex domain [27], Graph-Convolution-based models were further developed to capture the non-Euclidean spatial correlations [28]–[35]. For instance, based on a directed graph, a data-driven method was proposed for OD matrix estimation, which combined the PCA method to constrain the solution space for large networks and the neural network to achieve a robust prediction accuracy [29]. The grid embedding method for both geographical and semantic neighborhoods was illustrated to capture spatial correlations by [31]. The model was also combined with LSTM to capture the temporal trends for predicting OD demand. The combination of Graph Convolution Network (GCN) and GRU for road traffic forecasting was explored in [30]. Similarly, [33] proposed a framework named Spatio-Temporal Graph Convolutional Networks (STGCN) for traffic forecasting tasks that modeled the traffic network by a general graph and employed a fully convolutional structure on the time axis to handle the inherent deficiencies of recurrent networks.

Although previous works showed encouraging performance in terms of demand forecasting, OD demand predictions coupled with confidence intervals have received very limited attention in the literature. This motivates the current study. Moreover, as mentioned earlier, different from previous deep-learning-based studies that relied on RNN-based models to capture temporal correlations, this study adopts a set of gated graph convolution layers to model the spatial and temporal correlations simultaneously. Doing so avoids potential large accumulative errors associated with RNN-based models.

B. Bayesian-Based Networks

This work aims to capture the variability/uncertainty of the neural network model, in order to utilize this information to provide confidence interval based demand predictions. In recent years, different algorithms were proposed to estimate model uncertainty based on Bayesian Neural Networks (BNN) [36]–[39]. The Bayesian method provides an uncertainty estimation in the form of a probability distribution [40]. Bayes by back-propagation was proposed in [37] to regularize the weights of neural networks by minimizing a compression cost and thus improve the predictive uncertainty. Probabilistic back-propagation (PBP) was presented for learning BNN by a product of Gaussians to approximate the posterior for weights in BNN, which can deal with large network sizes [36]. More recently, the multiplicative normalizing flows (MNFs) were introduced to interpret multiplicative noise for BNN for augmenting the approximated posterior [38].

The aforementioned strategies often rely on different specific training methods, which should be tested in order to provide useful forecasting or predictions. In this study, in order to approximate the uncertainty of the model to obtain the confidence interval of the demand prediction without relying on the training methods, we adopt Bayesian approximation through Monte Carlo (MC) dropout training, which was firstly proposed by [39]. As discussed in [39], when dropout is applied before every weight layer, a neural network with arbitrary depth and non-linearities will be mathematically equivalent to an approximation to the probabilistic deep Gaussian process.



(a) Morning Demand (b) Evening Demand

Fig. 1. Illustration of regions and OD demand.

III. Preliminary

In this section, we describe the basic settings for the demand prediction problem, including city partition, the definition of the OD demand, and the approximation of prediction interval with a target level of confidence.

A. City Partition

Many existing studies on demand forecasting divided the whole city/area in concern into multiple squares, triangles, or hexagons grids, e.g., [1]. This can be readily done with the model proposed in this study. However, differently, in the experiments of this study, we consider the administrative areas for demand prediction based on the postcodes. Smaller region sizes or types can be considered (e.g., we also test station-to-station demand for three regions in Section V-D). In practice, many relevant planning and management activities are indeed administrative-region-based. Estimating the demand at the administrative region level can be useful in many occasions and can provide evidence for policy making at the region-level. For example, demand information at the administrative region level will be useful for inter-region bus service network design.

It is also noteworthy that administrative regions often have their unique features or functionalities (e.g., land use and demographic attributes). In practice, urban features partially govern traffic patterns [24], and thus further governs Origin-Destination demand patterns. In particular, the city/area in concern is partitioned into Q regions based on postcodes. Figure 1 shows an example of OD demand based on administrative areas, where the land use and its functions largely affect the demand pattern. In this figure, Region A and Region B are residential regions while Region C is an industrial region. This means that Region B has similar land use attribute to Region A rather than adjacent Region C. As for Region C, in the morning rush hours as shown in Figure 1(a), the demand D_{A-C} (from A to C) and D_{B-C} (from B to C) are high. On the contrary, in the afternoon rush hours as shown in Figure 1(b), the demand D_{C-A} (from C to A) and D_{C-B} (from C to B) will be high.

B. Origin-Destination Demand

The Origin-Destination demand is defined as the number of trips from one region (the origin) to another region (the destination) in each time step. We define the OD pair set as $R = \{r_1, r_2, \cdots, r_n, \cdots, r_N\}$, where N is the total number of OD pairs. Then, we denote the demand of OD pair n at time step i as a scalar x_i^n. Thus, time-dependent OD demand can be represented as $X \in \mathbf{R}^{T\times N\times d_{in}}$, where T is the total number of time steps (for the whole time horizon where have demand data) and d_{in} is the dimension of OD and time-step specific data. For example, if the OD and time-step specific data includes OD demand value and average trip distance for this OD, then $d_{in} = 2$. In this study, we only use the OD demand value, and thus $d_{in} = 1$. When $d_{in} = 1$, we can simply consider $X \in \mathbf{R}^{T\times N}$.

In the demand forecasting process, a demand sequence of the observed OD pairs $\{X_1, X_2, \cdots, X_i, \cdots, X_I\}$ is used instead of all historical information where $X_i = \{x_i^1, x_i^2, \cdots, x_i^n, \cdots, x_i^N\}$ represents the demand of N OD pairs at time step i and I is the number of time steps that are utilized. The OD demand prediction problem is then formulated as learning a prediction function $\Gamma(\cdot)$ to estimate the demand of each OD pair in the future time step $t+1$: $\hat{X}_{t+1} = \Gamma\left(X_{t-I+1}, \cdots, X_{t-1}, X_t\right)$.

C. Prediction Interval Approximation

As described in Section I, the predicted demand interval with a target level of confidence provides a more robust way to represent demand and better captures demand variability and uncertainty. However, the "true" demand confidence interval may never be obtained. We approximate the "true" interval with the α-level prediction interval based on a normal distribution. Specifically, the interval can be formulated as: $[\hat{\mu} - z_{\alpha/2}\hat{\eta}, \hat{\mu} + z_{\alpha/2}\hat{\eta}]$, where $\hat{\mu}$ and $\hat{\eta}$ are the mean and standard deviation of the estimated demand $\hat{X}_{t+1}$, respectively, and $z_{\alpha/2}$ is a coefficient for the standard deviation such that $\hat{\mu} - z_{\alpha/2}\hat{\eta}$ and $\hat{\mu} + z_{\alpha/2}\hat{\eta}$ correspond to the lower and upper $\alpha/2$ quantiles of a standard Normal distribution, respectively. These values can be obtained based on quantifying the graph-based model uncertainty through Mote Carlo dropout training.

IV. PGCM Framework

In this section, we propose a Probabilistic Graph Convolution Model (**PGCM**) for OD demand prediction and demand interval approximation. The architecture of the proposed model consists of two components: the Origin-Destination Demand Prediction Module based on Graph Neural Network (i.e., a set of gated graph convolution layers to capture spatio-temporal correlations for further forecasting) and the Bayesian Approximation Module based on Monte Carlo dropout. The overall structure of the proposed model is illustrated in Figure 2 and these modules will be explained in detail in the following.

A. Origin-Destination Demand Prediction Module

The OD demand prediction module is composed of a series of Gated Graph Convolution Layers (GGCLs) to extract the spatial-temporal correlations for demand forecasting. Before formulating the GGCL, the construction of the graph based on OD demand is introduced first. Then, how to utilize the GGCLs to construct the prediction module will be introduced.

Some previous studies [1], [21] assumed that spatially adjacent areas may have similar OD demand patterns. However, OD demand is not only related to the geographical location (i.e latitude and longitude) but also relies heavily on the non-Euclidean features such as demographic attributes (i.e income, age distribution) and land use (i,e Point of Interest) of origins and destinations. Graph-based deep models are able to process the data with non-Euclidean structure and effectively extract the features from OD demand data. Following [9], we define an undirected graph $\mathcal{G} = (V, E, A)$ (as will be introduced below, an edge in the graph reflects the correlation between two OD pairs and has no direction, which has nothing to do with the OD directions in Figure 1) and adopt a gated mechanism for the graph convolution layers, where V is the set of nodes, E is the set of edges that does not change with time, and A is the adjacency matrix. The adjacency matrix is used to indicate the adjacent relationship between nodes (reflecting whether there is an edge between any two nodes).

An OD pair is represented by a node in the graph constructed, i.e., a node in the graph defined (as a part of the deep learning model) is different from the "node" usually defined in a physical transport network. An edge between two nodes in the graph is used to indicate the correlation/similarity between the demand patterns of two OD pairs. If the demand patterns of two OD pairs (two nodes in the graph) are correlated to or above a certain extent (a threshold is defined), we add an edge between the two nodes in concern and set the corresponding value in the adjacency matrix as one (otherwise zero).

More specifically, in the graph, $v_n \in V$ is denoted as a node and $e_{n_1n_2} = (v_{n_1}, v_{n_2}) \in E$ is denoted as an edge. Then, we use Pearson Correlation Coefficient to measure the similarity $s_{n_1n_2} = Pearson(x_{0\sim t}^{n_1}, x_{0\sim t}^{n_2})$ between the historical demand patterns of two OD pairs (OD pair n_1 and OD pair n_2) where $x_{0\sim t}^{n}$ represents the historical passenger demand sequence for OD pair n from time period 0 to t. The adjacency matrix $A \in \mathbf{R}^{N\times N}$ is determined by the demand similarity between two OD pairs:

$$A_{n_1n_2} = \begin{cases} 1 & s_{n_1n_2} > \epsilon \\ 0 & s_{n_1n_2} \le \epsilon \end{cases} \tag{1}$$

where ϵ is a threshold used to decide whether a sufficient strong correlation exists between two OD pairs. The threshold governs the sparsity of the adjacency matrix. In practice, the threshold values chosen for different datasets and the corresponding sparseness of adjacency matrix may be different. We will conduct a sensitivity analysis on the value of the threshold and sparsity.

To capture the spatial-temporal dependency, graph convolution layers are adopted in the module based on the graph $\mathcal{G}$. We first denote the graph Laplacian matrix as $L \in \mathbf{R}^{N\times N}$

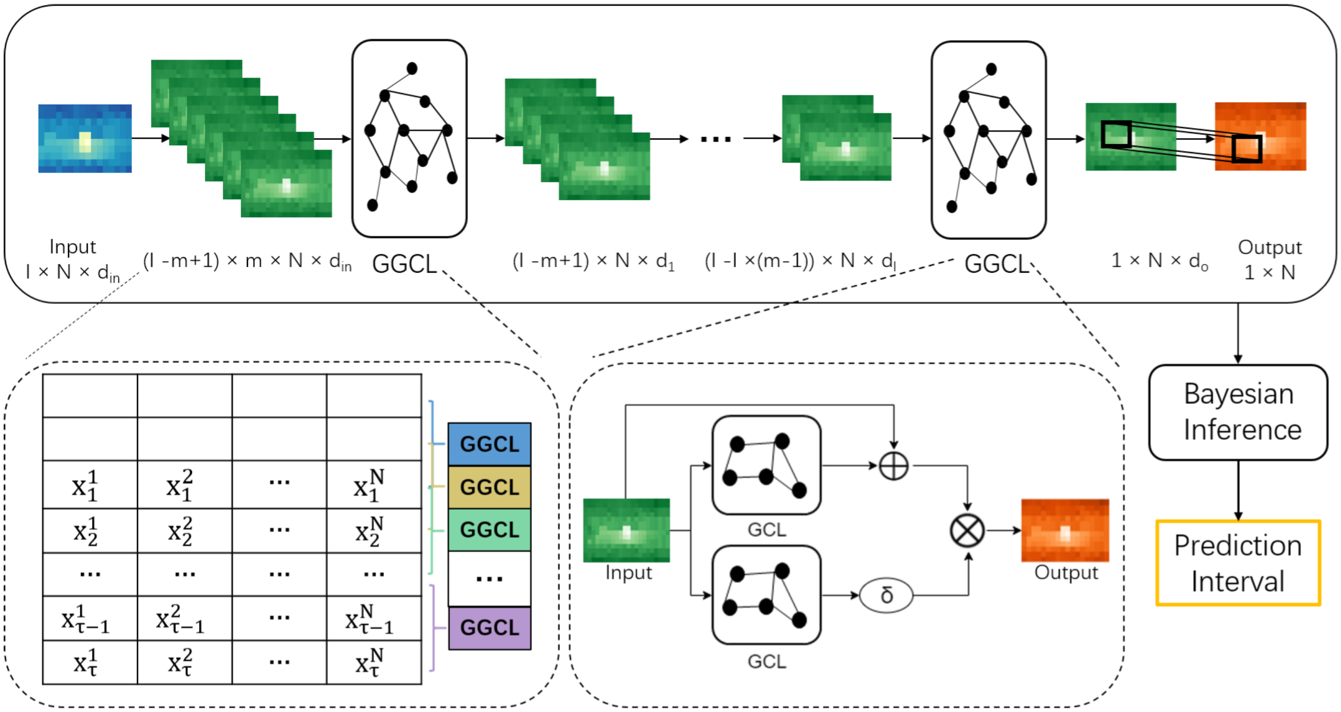


Fig. 2. The Architecture of Probabilistic Graph Convolution Model (PGCM). x_i^j is the demand of j^{th} OD pair at i^{th} time step. I is the number of time steps, N is the total number of OD pairs, d_{in} is the dimension of the input data, d_l is the dimension of the data in emitted from l^{th} GGCL, d_o is the dimension of the output data emitted from the last GGCL, m is the number of time steps used in one gated graph convolution layer, δ is the sigmoid function. The number of gated graph convolution layers is $\frac{I-1}{m-1}$. And the number of learnable GGCL parameters is $\frac{I-1}{m-1} \times 2$ (bias is omitted for simplicity).

where $L = U\Lambda U^T$. $U \in \mathbf{R}^{N\times N}$ is the matrix for eigenvectors and $\Lambda \in \mathbf{R}^{N\times N}$ is the diagonal matrix for eigenvalues of L. Then, based on the graph convolution in frequency domain [26], following [30], [41], a signal $x \in \mathbf{R}^N$ filtered by a filter g_θ can be written as:

$$x * g_\theta = U g_\theta(\Lambda) U^T x \tag{2}$$

Suggested by [42], we approximate the filter function $g_\theta(\Lambda)$ with Chebyshev polynomials $T_k(x)$. Following [41], the order is set to $K = 1$, Equation (2) then can be approximated by:

$$x * g_\theta = \theta(I_N + M^{-\frac{1}{2}} A M^{-\frac{1}{2}})x \tag{3}$$

where M represents the diagonal matrix of node degrees and $M_{ii} = \sum_j A_{ij}$. Therefore, the result of Graph Convolution Layer (GCL) can be calculated as:

$$X^{l+1} = (\tilde{M}^{-\frac{1}{2}} \tilde{A} \tilde{M}^{-\frac{1}{2}}) X^l W \tag{4}$$

where $I_N + M^{-\frac{1}{2}} A M^{-\frac{1}{2}}$ is replaced by $\tilde{M}^{-\frac{1}{2}} \tilde{A} \tilde{M}^{-\frac{1}{2}}$, and $\tilde{A} = A + I_n$, $\tilde{M}_{ii} = \sum_j \tilde{A}_{ij}$. X^l denotes the demand-like input in the l^{th} GCL. Note that the step in Eq. (4) does not contain any non-linear operation.

Then, gating mechanism in [43] is adopted in the GCL to capture the non-linearity and decide which part(s) of the linear transformation can be passed through the gate and thus contribute to the prediction. Moreover, residual learning in [44] is utilized to reduce the vanishing gradient problem. These actions complete the construction of the proposed Gated Graph Convolution Layer (GGCL). And the structure of GGCL is shown in Figure 2. The result of l^{th} layer is formulated as:

$$X^{l+1} = ((\tilde{M}^{-\frac{1}{2}} \tilde{A} \tilde{M}^{-\frac{1}{2}}) X^l W_1 + X^l) \odot \delta((\tilde{M}^{-\frac{1}{2}} \tilde{A} \tilde{M}^{-\frac{1}{2}}) X^l W_2) \tag{5}$$

where δ denotes the sigmoid function and is defined as:

$$\delta(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

To predict the OD demand, a series of GGCL as described above has been applied along the temporal axis to capture the spatial and temporal correlations simultaneously, which requires less temporal iterative operations and avoids the error accumulation problem for those using RNN, as discussed in [9]. Also, following many existing studies, e.g., [43], [45], adopting sigmoid function allows the proposed GGCL with the gated units to capture the non-linearity. And adopting several GGCL for demand prediction means that the model contains more than one non-linear operations.

In detail, one GGCL has limited capability to capture long-term temporal relations in the total number of I time steps. Thus, we define another number m that is smaller than I. Based on $X \in \mathbf{R}^{I\times N\times d_{in}}$, we then can define $I - m + 1$ different $X_{new} \in \mathbf{R}^{m\times N\times d_{in}}$, where each X_{new} involve less time steps than X. For example, if $I = 5$, i.e., we have five time steps in consideration ($i = 1, 2, 3, 4, 5$) and $X \in \mathbf{R}^{5\times N\times d_{in}}$, and $m = 3$, we then include time steps $i = 1, 2, 3$ into $X_{new}^{(1)}$, time steps $i = 2, 3, 4$ into $X_{new}^{(2)}$ and time steps $i = 3, 4, 5$ into $X_{new}^{(3)}$, i.e., $I - m + 1 = 5 - 3 + 1 = 3$ different $X_{new} \in \mathbf{R}^{3\times N\times d_{in}}$. The matrix in the lower-left corner in Figure 2 also provides an example of X_{new} for illustration.

Such an operation discussed in the above means that one GGCL acts on m time steps of the demand data rather than a single time instant. It could extract the spatial correlation among all regions within m time steps. Moreover, a number of $I-m+1$ different $X_{new} \in \mathbf{R}^{m\times N\times d_{in}}$ will be sent into one GGCL sharing the same group of biases and weights to capture the correlations of the data in I time steps used for prediction. The output of l^{th} layer is $\hat{Y}_l \in \mathbf{R}^{I-l\times(m-1)\times N\times d_l}$.

Therefore, only $\frac{I-1}{m-1}$ gated graph convolution layers are needed to capture the temporal dependencies in I time steps, which decreases the number of iterations and improves the accuracy when compared with RNN-based models. Two parameters W_1 and W_2 should be learned in each GGCL. Thus, the total number of GGCL parameters need to be learnable is $\frac{I-1}{m-1}\times 2$ (bias is omitted for simplicity). The output of the set of GGCLs is $\hat{Y} \in \mathbf{R}^{1\times N\times d_o}$, where d_o is the output dimension of the last gated graph convolution layer.

The proposed model is able to capture the spatial and temporal features by the several gated graph convolution layers before the last output layer. Thus, it is not necessary to also use GGCL as the final prediction layer. Due to the weights sharing mechanism of CNN as discussed in [46], CNN needs fewer parameters than the fully connected layer. Using the fully connected layer with more parameters will be less efficient in terms of computation. Thus, we choose CNN as the output layer in our model to extract the dependencies among d_o dimensions' data for prediction. The convolution layer is composed of a filter equalling 1×1 to sweep the input matrix and produce the prediction demand $\hat{X}_{t+1}$.

B. Bayesian Approximation Module

This section discusses how to provide a confidence interval or a metric for prediction accuracy. As introduced by [47], model uncertainty is used to indicate the accuracy of the prediction in neural networks. The graph-based model uncertainty in our work could be applied to approximate the forecasting demand interval by Bayesian-based methods. For the Bayesian methods, there are two main types of uncertainty: Aleatoric Uncertainty and Epistemic Uncertainty. Aleatoric Uncertainty measures the noise inherent in observations that mainly exists in data collection methods, such as the noises caused by the card reader machines of buses. Even if more data is collected, the Aleatoric Uncertainty cannot be reduced and is hard to be detected or measured by machine learning algorithms. On the contrary, Epistemic Uncertainty is caused by the deep neural network itself and can be measured by suitable Bayesian Neural Networks (BNN), which is the main focus of this paper and will be explained in the following.

Motivated by [39], Bayesian deep learning assumes that each weight and bias should obey a certain distribution instead of a certain value, which leads to an interval of the demand that associates with a certain level of confidence. The Monte Carlo dropout only involves minor changes in the neural network and does not change the overall structure of the network, so the prediction results are still compatible.

Given a set of values estimated by the OD demand prediction module $\{\hat{X}_{T+1}, \hat{X}_{T+2}, \cdots, \hat{X}_{T+I},\}$ and true values $\{X_{T+1}, X_{T+2}, \cdots, X_{T+I}\}$, according to the Bayesian Theorem, the posterior distribution $P(W|X,\hat{X})$ is used to measure the probability of the parameters over the model. As verified by Gal *et al.* [39], the approximation of the Gaussian process [48] is equivalent to a neural network with dropout. Thus, we use Gaussian distribution $q_\theta(W|X,\hat{X})$ to measure the posterior distribution of our model by minimizing the Kullback-Leibler divergence between them. The Gaussian process in this paper is:

$$W|X,\hat{X} \sim N(\hat{\mu},\hat{\eta}^2) \tag{7}$$

where $\hat{\mu}$ denotes the mean value of the demand estimation and $\hat{\eta}$ denotes the standard error. Then the demand intervals are given by the Bayesian Inference.

C. Prediction Model Training and Bayesian Inference

In the training process of the OD demand forecasting module, the objective is to minimize the error between the true OD demand and the predicted values. The loss function is defined as the mean squared error for I time steps formulated as follows:

$$L(\theta) = \sum_{i=T+1}^{T+I} ||\hat{X}_i - X_i|| \tag{8}$$

where θ denotes all the learnable parameters in the prediction model. It is solved via back-propagation and Adam optimizer.

In order to check whether it is reasonable to approximate the OD demand interval by Bayesian Inference as Gaussian distribution, we visualize the distribution of demand at a certain clock time on weekdays (or weekends) for a specific OD pair. Note that we only have one demand observation for each OD pair at a specific time step on a specific date, but the demand patterns on weekdays (or weekends) may be similar (e.g., May 09, 2017 is a weekday, and demand during 8:00 am - 8:30 am on weekdays might be similar to that on May 09, 2017). We find that the distributions of demand at a certain clock time on weekdays (or weekends) for a specific OD pair can be well approximated by the Gaussian distribution.

In the Bayesian Inference of the approximation module, the algorithm consists of three steps. First, we randomly dropout some neural units with probability p before each layer, conduct the forward passes through the network, and obtain the demand prediction $\hat{X}^s_{t+1}$. Then, we repeat the first step for S times, and we can get a set of predicted results $\{\hat{X}^1_{t+1}, \hat{X}^2_{t+1}, \cdots, \hat{X}^S_{t+1}\}$. At last, the average value $\hat{\mu}$ and standard error $\hat{\eta}$ are used to approximate the demand interval:

$$\begin{aligned} \hat{\mu} &= \bar{\hat{X}}_{t+1} = \frac{1}{S}\sum\nolimits_{s=1}^{S} \hat{X}^s_{t+1} \\ \hat{\eta}^2 &= \frac{1}{S}\sum\nolimits_{s=1}^{S} (\hat{X}^s_{t+1} - \hat{\mu})^2 \end{aligned} \tag{9}$$

The Bayesian inference based on MC dropout was listed in Algorithm 1. With the help of Standard Normal Distribution table, we can obtain the demand interval $[\hat{\mu} - z_{\alpha/2}\hat{\eta}, \hat{\mu} + z_{\alpha/2}\hat{\eta}]$ with a target level of confidence α.

Algorithm 1 Bayesian Inference Algorithm

```
Require:
    Mobility Forecasting Module $\Gamma$
    Number of Iteration $S$
    Dropout Probability $p$
    True Demand $X$
Ensure:
    Mean Value $\hat{\mu}$
    Standard Error $\hat{\eta}$
for $s$ in $S$ do
    Dropout units based on $p$ of Model $\Gamma$
    Conduct the forward passing of $\Gamma$ and otain the predicted demand value $\hat{X}^s$
end for
Calculate $\hat{\mu} = \bar{\hat{X}} = \frac{1}{S}\sum_{s=1}^{S} \hat{X}^s$
Calculate $\hat{\eta}^2 = \frac{1}{S}\sum_{s=1}^{S}(\hat{X}^s - \hat{\mu})^2$
end procedure
```

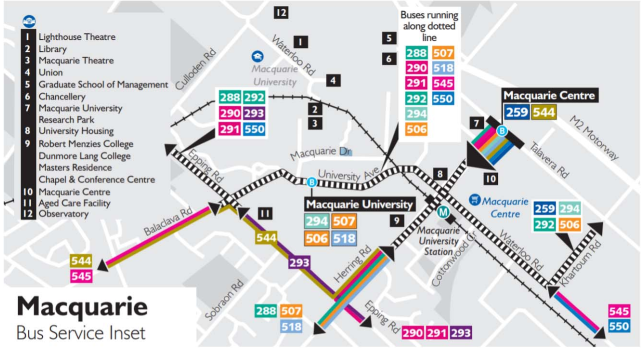


Fig. 3. An example of the bus lines and stations in sydney.

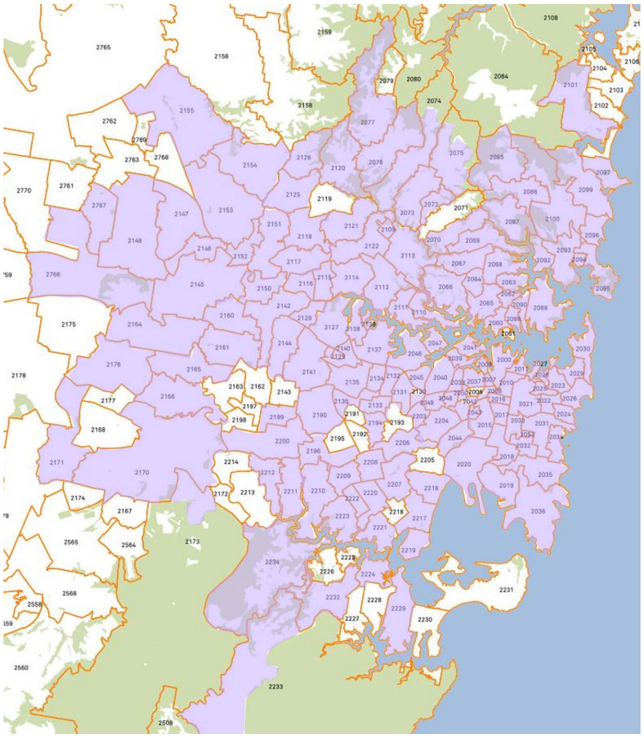


Fig. 4. City partition based on postcodes.

V. Performance Evaluation

In this section, we first introduce the dataset used in our experiments to test the proposed model. Then, experiment settings are given. In the next, we present the experimental results from two perspectives: the prediction accuracy and prediction confidence interval coverage ratio among existing strategies and our proposed model based on three different time resolutions for demand predictions (1 hour, 30 minutes, and 15 minutes). The prediction accuracy is reflected by three evaluation metrics (MAE, RMSE, and MAPE). The prediction confidence interval coverage ratio is defined as the proportion of the true values falling into the demand prediction interval (i.e., the proposed "confidence interval") for the testing samples. Moreover, we also test and evaluate our method for station-to-station demand forecasting based on the station-level data. The last part in this section focuses on parameter sensitivity analysis on a number of factors or settings (e.g., the learning rate, the threshold, and the sparsity of the adjacency matrix).

A. Dataset

The dataset is collected from Sydney covering main public transportation services (buses, trains, ferries, and light rails) from 01/Apr/2017 to 30/Jun/2017 covering 6.37 million users. We focus on forecasting the demand of buses in this study. To have a more intuitive understanding of the bus operation, we give an example of the bus lines and stations in a region which is shown in Figure 3 (https://transportnsw.info/travel-info/ways-to-get-around/bus/bus-operator-maps). The partition of Sydney based on postcode is shown in Figure 4 and the areas shaded in purple are the study units (as origin/destination) of our study. Different regions have different sizes and the average value is about $7.8km^2$. In order to protect the privacy of users, the data does not involve personal information that can be used to identify the individuals.

B. Experimental Setting

1) Dataset Setting: In the experiments, we choose all bus lines information including tap-on and tap-off location, time, corresponding administrative area, and the number of passengers getting on and off. We divide the whole dataset into three mutually exclusive sub-sets, i.e., the first seventy days' data are used for training (training set), the last ten days' data are used for testing (testing set), and the rest for validation (validation set).

The demand data is normalized by Min-Max normalization for training and re-scaled for evaluating the prediction accuracy. We implement the model in Python with Pytorch 1.1.0. In the experiments, we have tested three different lengths of time step for demand prediction, i.e., 1 hour, 30 minutes, and 15 minutes. Furthermore, we exclude the OD pairs with extremely small demand since they are not the focus of demand prediction in this paper. In particular, we exclude the OD pairs with an average demand less than 3 persons per time step when the time step length is one hour or 30 minutes,

TABLE I
OVERALL COMPARISON OF PREDICTION ACCURACY

time step		1 hour			30 minutes			15 minutes		
Index	**Method**	**RMSE**	**MAE**	**MAPE**	**RMSE**	**MAE**	**MAPE**	**RMSE**	**MAE**	**MAPE**
1	**ARIMA**	29.6429	20.4692	0.6065	20.3203	15.2411	0.5221	14.4282	12.5180	0.5892
2	**HA**	13.4791	12.1329	0.4630	10.0264	8.9477	0.4396	9.3335	7.8171	0.4086
3	**LR**	12.9000	10.2313	0.4011	9.5403	8.3992	0.3843	8.7148	7.6929	0.4012
4	**GRU**	10.1630	4.6386	0.2499	6.5868	3.5475	0.3119	4.5135	2.6724	0.3594
5	**LSTnet**	8.9854	**4.0135**	0.2515	7.0769	3.4815	0.3388	4.9903	**2.6446**	0.4330
6	**GCRN**	8.4710	4.1026	0.2765	6.8699	3.5766	0.3418	4.6497	2.7804	0.3877
7	**STGCN**	8.2523	4.0916	0.2882	6.5819	3.5372	0.3487	4.5968	2.6897	0.3694
8	**Our**	**7.8741**	**4.0412**	**0.2441**	**6.3196**	**3.4804**	**0.3109**	**4.4219**	**2.6680**	**0.3475**

and exclude the OD pairs with an average demand less than 2 persons per time step when the time step is 15 minutes. Thus, when the time step set to 1 hour, the number of OD pairs/nodes in the experiment is 634. When the time step set to 30 minutes and 15 minutes, the number of OD pairs/nodes in the experiment are 266 and 94, respectively. The number of time steps I is set to 12.

2) Network Implementation: In all deep-learning based model, the batch size is set to 64. And they are tuned with the learning rate from 0.001 to 0.011 with a step size of 0.001. In the proposed model, the size of hyperparameter m is 3. If the value of m is too small, too many layers have to be generated. The over-deep neural network will easily lead to gradient disappearance or gradient explosion. On the contrary, if the value of m is too large, it will yield too few layers which might not be able to well capture the temporal correlations. Moreover, the number of Gated Graph Convolution Layers is six. It is tuned with the number of hidden units (16, 32, 64) for six layers. The chosen output dimensions of the hidden layers are 32, 32, 64, 64, 64, and 64. In the Bayesian Approximation Module, the dropout probability p is set to 0.15. And the amount of repeat times is set to 1000 for Bayesian Inference.

3) Evaluation Metrics: To test the effectiveness of the proposed OD demand forecasting model, three evaluation metrics are used to evaluate it: Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). For Bayesian Approximation, we calculate the range of the prediction interval and the proportion of the true value falling into the interval.

C. Overall Comparison

This subsection includes two parts, the comparison of the predicted accuracy among the proposed method **PGCM** and various previous studies and the coverage proportion of predicted demand interval.

1) Accuracy of Prediction: To test the effectiveness of the proposed model, we first compare the proposed method with three traditional baselines and four deep-learning-based methods, which are briefly summarized in the following.

- **Autoregressive Integrated Moving Average (ARIMA)**: It is one of the most common statistic models used for time series prediction, which combines the autoregressive components and the moving average method. In detail, the order of the autoregressive model equals 2, the degree of differencing equals zero, and the order of the moving-average model equals one. A separate model for each OD pair is developed.
- **Linear Regression (LR)**: Ordinary least squares Linear Regression is tested, which minimizes the sum of the squares of the errors. We regard the date and time of data as one-hot input features and develop a separate model for each OD pair.
- **Historical Average (HA)**: It utilizes the average values of historical demand values at the same time step of every day as the predicted demand.
- **Gate Recurrent Unit (GRU)** [49]: It has a similar structure with LSTM but has fewer parameters than LSTM such as the output gate, which may yield better performance in some circumstances. It is tuned with the number of hidden units (32, 64, 100, 128), where 100 is chosen as it yields better performance.
- **Graph Convolutional Recurrent Network (GCRN)** [28]: It combines CNN on graphs to identify spatial structures and RNN to find temporal patterns for precise demand forecasting. It is tuned with the number of hidden units (32, 64, 100, 128) and the threshold of the adjacency matrix (from 0.001 to 0.01 with a step size of 0.01). And the chosen number of units in the graph convolution layer is 64 while the chosen number of units in LSTM layer is 128. The chosen threshold is 0.005. These chosen values produce better performance than others.
- **Long- and Short-term Time-series network (LSTnet)** [50]: It leverages a novel recurrent structure, i.e., recurrent-skip network to capture very long-term dependence patterns and convolutional layers to discover the local dependency patterns for forecasting. The window size is 12 while the skip length is 3. It is tuned with the number of hidden dimensions (32, 64, 100, 128). The chosen hidden dimensions of the Recurrent layer and Convolutional layer are 100.
- **Spatio-Temporal Graph Convolutional Networks (STGCN)** [33]: It combines graph convolutional layers and convolutional sequence learning layers to model spatial and temporal dependencies for traffic flow forecasting. According to their hyperparameter setting, both the graph convolution kernel size and temporal convolution kernel size in our experiments are set to 3. It is tuned with the number of channels of three layers (32, 64, 100, 128)

TABLE II
COMPARISON OF PREDICTION INTERVAL PERFORMANCE

Confidence Level		**94%**		**95%**		**96%**		**97%**		**98%**	
time step	**Method**	**Span**	**Proportion**	**Span**	**Proportion**	**Span**	**Proportion**	**Span**	**Proportion**	**Span**	**Proportion**
1 hour	**GRU**	11.4551	50.24%	11.9545	52.59%	12.4937	55.45%	13.2014	59.09%	14.1520	63.76%
	LSTnet	8.6212	46.43%	8.9854	49.09%	9.4091	52.59%	9.9421	57.03%	10.6580	62.73%
	GCRN	7.8988	46.58%	8.2313	48.62%	8.6252	50.81%	9.1138	53.57%	9.7702	62.93%
	STGCN	8.0900	53.66%	8.4306	55.38%	8.8340	57.36%	9.3344	59.84%	10.0066	63.06%
	Our	**7.0105**	**56.41%**	**7.3056**	**57.91%**	**7.6551**	**59.66%**	**8.0888**	**61.72%**	**8.6712**	**64.27%**
30 minutes	**GRU**	5.7325	34.52%	5.9738	35.95%	6.2597	37.63%	6.6143	39.91%	7.0905	**43.10%**
	LSTnet	5.1590	32.73%	5.4059	34.74%	5.7121	37.46%	6.1234	41.95%	4.9506	31.17%
	GCRN	4.5875	32.23%	4.7806	33.24%	5.0093	34.28%	5.2931	35.71%	4.6742	35.12%
	STGCN	4.3225	38.41%	4.4860	39.73%	4.6768	41.13%	5.0134	42.88%	5.2311	45.16%
	Our	**4.1310**	**39.84%**	**4.3287**	**41.42%**	**4.5739**	**43.30%**	**4.9033**	**45.63%**	**4.9642**	**38.50%**
15 minutes	**GRU**	2.4089	20.99%	2.5103	22.07%	2.6304	23.37%	2.7794	25.36%	2.9795	28.46%
	LSTnet	2.3836	20.49%	2.4839	21.43%	2.6028	22.91%	2.7502	25.65%	2.9483	30.35%
	GCRN	2.1749	22.68%	2.2665	23.62%	2.3749	24.61%	2.5094	26.10%	2.6902	28.05%
	STGCN	1.9151	34.31%	1.9958	25.22%	2.0913	26.43%	2.2097	27.78%	2.3688	29.67%
	Our	**1.7725**	**25.27%**	**1.8471**	**26.14%**	**1.9355**	**27.25%**	**2.0452**	**28.83%**	**2.1924**	**30.72%**

and the threshold of the adjacency matrix (from 0.001 to 0.01 with a step size of 0.01). The chosen numbers of channels for three layers are all set to 64 and the chosen threshold is 0.003.

Table I summarizes the results regarding prediction accuracy for the proposed method and the aforementioned tools in the literature under different time resolutions for demand forecasting. We test different settings of hyperparameters for the compared strategies based on the validation set. The setting of hyperparameters that yields the best performance is then utilized to conduct testing and comparison based on the testing set (which has not been used before in training or validation process).

Several observations are made based on the results, which are discussed below.

First, the three traditional machine learning methods (ARIMA, LR, and HA) have relatively large RMSE, MAE, and MAPE. This is partially due to their inability to capture non-linear relations among spatial and temporal information for the OD demand forecasting. These results imply that there exist non-linear spatio-temporal relationships in the demand dataset that non-deep learning methods can not capture well.

Second, ARIMA yields relatively poor performance. A possible explanation is that ARIMA involves several parameters (the order of the autoregressive model, the degree of differencing, and the order of the moving-average model). It is not straightforward to obtain the optimal setting of these parameters (i.e., the optimal setting is not always guaranteed). These indeed have been reported in some previous works [51], where ARIMA yields relatively poor performance.

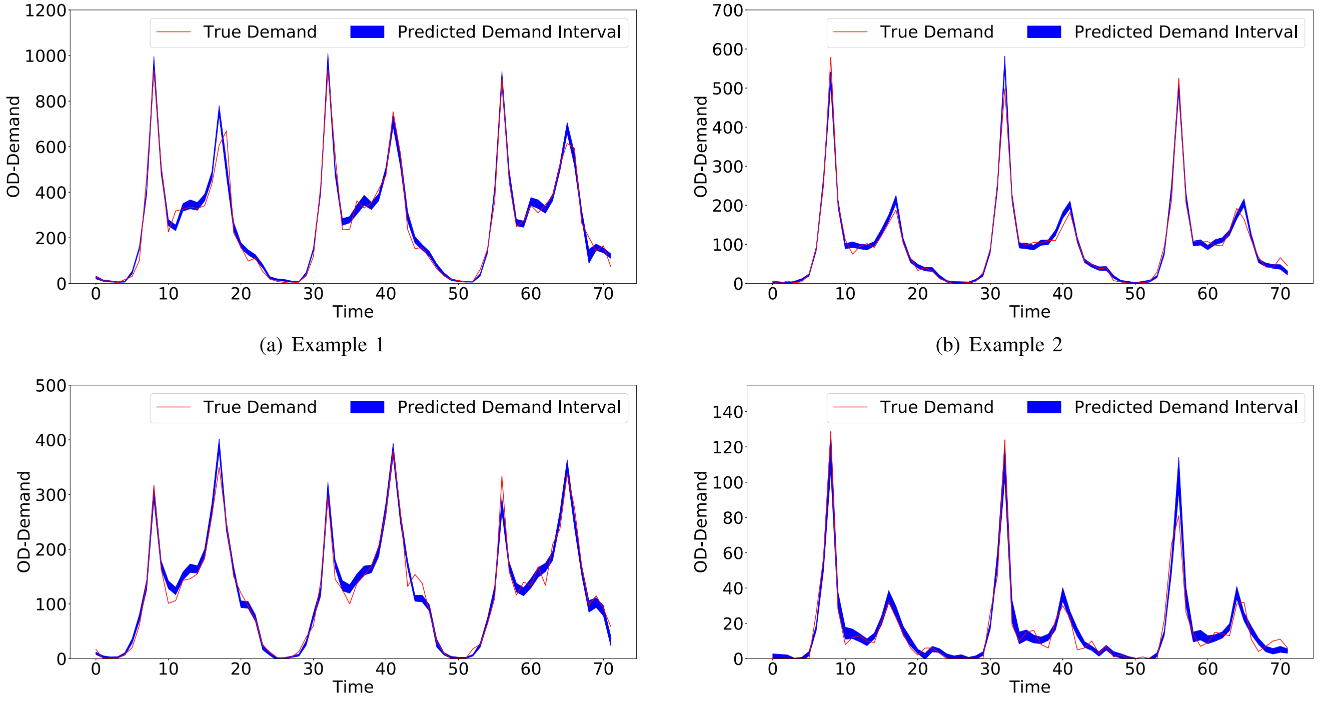
Third, the listed remaining neural network-based strategies including GRU, GCRN, LSTnet, and STGCN are able to predict the OD demand more precisely which are more able to capture the non-linear relations. Moreover, the good performance of LSTnet is partially due to the skip connection. The operation of the skip connection could also be adopted in other models to enhance the forecasting performance. However, GRU and LSTnet are less interpretable, e.g., in terms of correlations among the OD pairs since it did not explicitly analyse the spatial correlations. Compared to them, our model based on the graph convolution network is more interpretable, where the correlations among the OD pairs (nodes) based on the adjacency matrix are identified and more intuitive. As for GCRN and STGCN, they were modeled based on graph convolution operation but were not designed for OD demand forecasting which have less ability to capture the spatial correlations among OD pairs than our model.

Fourth, for the proposed model, when the time step is set to one hour and 15 minutes, although MAE values are slightly larger than LSTnet, RMSE and MAPE are smaller than all methods, which means that **PGCM** yields fewer big errors.

In general, the results indicate that the Probabilistic Graph Convolution Model can capture spatio-temporal correlations more accurately than others when predicting OD demand. In addition, the proposed model achieves more precise results under a larger time step.

2) Prediction Interval: First, we have checked that the predicted values $\{\hat{X}^1_{t+1}, \hat{X}^2_{t+1}, \cdots, \hat{X}^S_{t+1}\}$ can be well approximated as a Gaussian distribution. Then, we collect statistical results about the range of prediction interval with a certain level of confidence, and the proportion of true values falling into the interval based on three time steps. Table II compares the results of the proposed model and four deep-learning-based strategies which could combined with Bayesian approximation: GRU, LSTnet, GCRN, and STGCN. In the experiments, the level of confidence value α changes from 94% to 98%. Specifically, $\alpha = 95\%$ means that the estimated values fall within the range of 1.96 standard deviations of the mean value. The results show that over three time periods and five confidence levels, the proposed **PGCM** achieves a higher coverage proportion within a shorter prediction interval compared to other strategies. This means that the proposed model can give more accurate predictions in a smaller range.

Then, in Figure 5, we provide several examples with different demand patterns to illustrate the predicted OD demand intervals (when $time\ step = 1\ hour$). The red lines represent true values and the blue shaded parts are the prediction intervals. These results indicate that the predicted intervals can cover true values well for various ranges of demand data and only in a small number of occasions that the true demand is not covered by the predicted intervals.



(a) Example 1

(b) Example 2

(c) Example 3

(d) Example 4

Fig. 5. Examples of predicted OD demand interval.

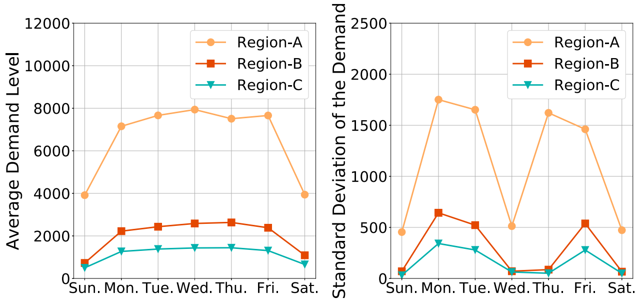


Fig. 6. Demand distribution of the selected regions.

D. Comparison With Station-Level Prediction

The results from Section V-C.1 show that our model achieves good performance at the region-level based on administrative regions. In this subsection, we further illustrate the effectiveness of the proposed method at the micro-level, i.e., the transit-stop level (station-level).

We choose three administrative areas with totally different demand intervals and predict the OD demand between each transit stop within one area (when the time step is one hour). The statistical demand distributions of three chosen areas are shown in Figure 6 (Region-A: Postcode 2000; Region-B: Postcode 2170; Region-C: Postcode 2200). They have different average values and distributions for the demand. Region-A has a much lager OD demand level than the other two regions. And the geographical distribution of average demand based on all historical data in the three regions are shown in Figure 7, which reflects the spatial distribution of demand. Then we compare the prediction results of our model with GRU, LSTnet, GCRN, and STGCN. The RMSE, MAE, and MAPE values are shown in Table III. Though LSTnet obtains slightly smaller MAE value than the proposed model

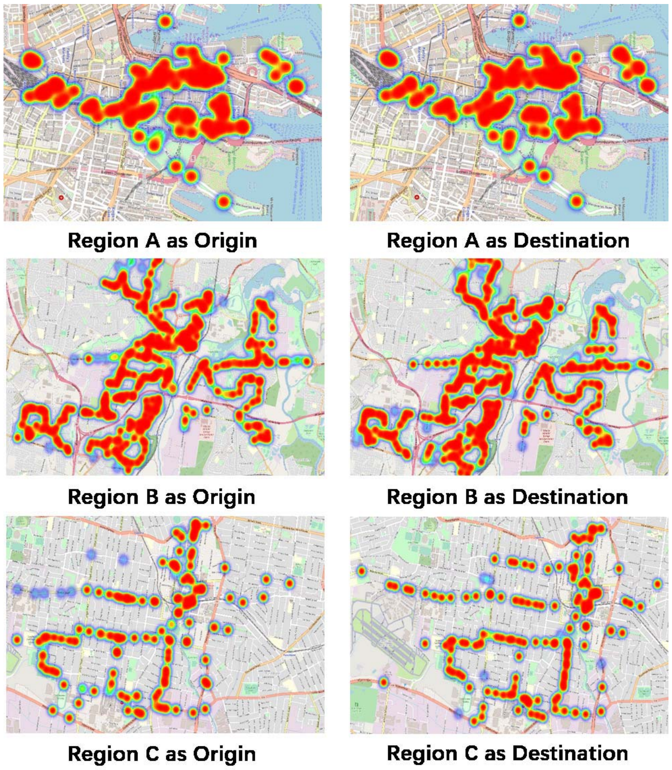
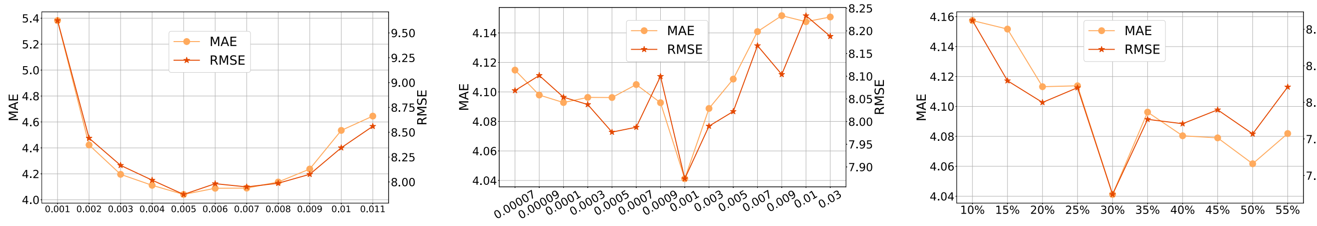


Fig. 7. Heatmap of the selected regions.



(a) Learning Rate

(b) Threshold

(c) Sparsity of the Adjacency Matrix

Fig. 8. Performance on parameter tuning.

TABLE III
COMPARISON OF PREDICTION ACCURACY AT THE MICRO-LEVEL

		Method				
	Region	**GRU**	**LSTnet**	**GCRN**	**STGCN**	**Our**
RMSE	**Region A**	12.1331	11.6914	11.4835	11.5926	**11.1668**
	Region B	7.5565	5.5670	5.1012	4.8637	**4.4474**
	Region C	5.9944	4.3636	4.0685	3.9729	**3.7338**
MAE	**Region A**	6.5966	4.3352	4.4431	4.8935	**4.2912**
	Region B	4.4878	**3.6252**	3.7265	3.8305	**3.6405**
	Region C	4.0148	3.0719	3.1069	3.0959	**2.8152**
MAPE	**Region A**	0.2511	0.2409	0.2761	0.24432	**0.2352**
	Region B	0.2340	0.2579	0.2680	0.2555	**0.2502**
	Region C	0.2591	0.2514	0.2735	0.2501	**0.2474**

over Region-B, other results indicate that the proposed model still has better forecasting performance at the micro-level (transit-stop level). Moreover, based on the proposed model, the stations with larger demand achieve relatively smaller MAPE. When comparing the predictions at the station-level and region-level, based on MAPE, one can see that the region-level prediction with large demand is more accurate. This is possibly due to that stations with a smaller level of demand involve more randomness, which can hardly be fully captured.

E. Parameter Sensitivity

We now evaluate the performance of the proposed method under various hyperparameters tuning based on the validation set, including learning rate, the threshold, and sparsity of adjacency matrix A based on a time step of one hour. And the results listed below are computed based on the test set.

Figure 8(a)shows RMSE and MAE under different learning rates from 0.001 to 0.011. When the learning rate is 0.05, the model yields the best performance.

The threshold introduced in Equation (1) is used to determine whether there is a sufficiently large correlation/similarity between two nodes in the graph (correspond to the demand patterns of two OD pairs). We test various values of the threshold in this subsection from 0.00007 to 0.03. Figure 8(b) gives the values of MAE and RMSE under different thresholds. When the threshold value is set to 0.001, the proposed model achieves the best performance. If the threshold value is too large, the adjacency matrix A will be relatively sparse, which means that most OD pairs are considered to have no similarity. The proposed model is then unable to fully utilize the correlation/similarity among OD pairs to enhance prediction accuracy. On the contrary, if the threshold value is too low, those not related OD pairs might be considered to have similar demand patterns, which results in the addition of noise in the model when predicting the demand.

Moreover, the threshold further governs the sparsity of the adjacency matrix defined. We then test different sparsity of the adjacency matrix which are shown in Figure 8(c). It illustrates RMSE and MAE under different sparsity of adjacency matrix A from 10% to 55%. Specifically, when the sparsity of A is set to 100%, the prediction achieves MAE= 4.4256, RMSE= 8.286, and MAPE= 0.2571, which are worse than others. An over-sparse adjacency matrix means that the potential relationship between two OD pairs is ignored while an over-dense matrix means that two unrelated OD pairs are considered to be related. An appropriate sparsity is important for the proposed approach. The results indicate that a sparsity of 30% yields the best performance.

VI. CONCLUSION

This paper proposes a novel deep learning framework based on a graph for confidence interval-based OD demand forecasting through exploring and utilizing the relevance among temporal and spatial information of the public transit data. Specifically, in the graph constructed, we define the node as one OD pair, the edge as the correlation between two OD pairs. One may define the nodes in the graph as zones in the real world and the edges as the demand between the origin and destination intuitively. However, in this case, the graph constructed will not contain any information regarding correlations between OD pairs. In order to utilize the correlations between OD pairs for forecasting, one may have to integrate an additional module to define the correlations between OD pairs. Treating each OD pair as a node to construct the graph is a more concise (while less intuitive) way to make full use of the graph network structure, which covers information regarding correlations among OD pairs for prediction.

In particular, within the proposed approach, (i) the Probabilistic Graph Convolution Model employs the demand forecasting module based on a series of gated graph convolution layers to extract spatio-temporal correlations; and (ii) the Bayesian Approximation Module is proposed to measure the model uncertainty and further provide the confidence interval for the demand prediction. The proposed approach is compared with several benchmark algorithms in the literature including ARIMA, LR, HA, GRU, and LSTnet, GCRN, and STGCN. The experiments on the real-world dataset show that the proposed approach outperforms other state-of-the-art methods. In general, the proposed approach is able to achieve lower

RMSE, MAE, MAPE, and yields more precise demand intervals under different time resolutions for demand forecasting.

This study makes the first attempt to combine Bayesian Neural Network with OD demand forecasting for predicting demand confidence interval. It provides a reliable way to represent the demand and measure the uncertainty of demand. The outputs can be used for public transit planning and operation optimization problems. The developed techniques may also be applied to other domains such as human mobility uncertainty analysis and traffic flow variability.

This study can be further extended in many ways. Firstly, the current work can be extended by adding more information into the neural network, such as the demographic attributes and PoI (Point of Interest) of the areas. This may further improve the prediction. Secondly, a future study may also explore the time series mutation problems based on demand forecasting. Last but not least, we may utilize the graph network in alternative manners in order to predict demand. For example, we may consider nodes in the graph as zones in the real world and the edges in the graph as the demand between the origin and destination and develop further related techniques. One then can also compare these alternative approaches with the method proposed in this paper.

Acknowledgment

The authors would like to thank the handling editor and the referees for their constructive comments, which helped to improve both the technical quality and exposition of the manuscript substantially.

References

[1] J. Ke *et al.*, "Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 11, pp. 4160–4173, Nov. 2019.

[2] M. Yildirimoglu and N. Geroliminis, "Experienced travel time prediction for congested freeways," *Transp. Res. B, Methodol.*, vol. 53, pp. 45–63, Jul. 2013.

[3] X. Ma, Y.-J. Wu, Y. Wang, F. Chen, and J. Liu, "Mining smart card data for transit riders' travel patterns," *Transp. Res. C, Emerg. Technol.*, vol. 36, pp. 1–12, Nov. 2013.

[4] Z. Ma, J. Xing, M. Mesbah, and L. Ferreira, "Predicting short-term bus passenger demand using a pattern hybrid approach," *Transp. Res. C, Emerg. Technol.*, vol. 39, pp. 148–163, Feb. 2014.

[5] M. L. Hazelton, "Statistical inference for time varying origin–destination matrices," *Transp. Res. B, Methodol.*, vol. 42, no. 6, pp. 542–552, Jul. 2008.

[6] J. Liang, J. Wu, Z. Gao, H. Sun, X. Yang, and H. K. Lo, "Bus transit network design with uncertainties on the basis of a metro network: A two-step model framework," *Transp. Res. B, Methodol.*, vol. 126, pp. 115–138, Aug. 2019.

[7] Y. Bai and W.-L. Jin, "Random variables and uncertainty analysis," in *Marine Structural Design*, 2nd ed. San Diego, CA, USA: Elsevier, 2015, pp. 615–625.

[8] K. J. Rothman and S. Greenland, "Planning study size based on precision rather than power," *Epidemiology*, vol. 29, no. 5, pp. 599–603, Sep. 2018.

[9] L. Bai, L. Yao, S. S. Kanhere, X. Wang, and Q. Z. Sheng, "STG2Seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1981–1987.

[10] H. Yang, T. Sasaki, Y. Iida, and Y. Asakura, "Estimation of origin-destination matrices from link traffic counts on congested networks," *Transp. Res. B, Methodol.*, vol. 26, no. 6, pp. 417–434, Dec. 1992.

[11] P. Robillard, "Estimating the O-D matrix from observed link volumes," *Transp. Res.*, vol. 9, nos. 2–3, pp. 123–128, Jul. 1975.

[12] L. J. LeBlanc and K. Farhangian, "Selection of a trip table which reproduces observed link flows," *Transp. Res. B, Methodol.*, vol. 16, no. 2, pp. 83–88, Apr. 1982.

[13] M. J. Maher, "Inferences on trip matrices from observations on link volumes: A Bayesian statistical approach," *Transp. Res. B, Methodol.*, vol. 17, no. 6, pp. 435–447, Dec. 1983.

[14] G. Vigos, M. Papageorgiou, and Y. Wang, "Real-time estimation of vehicle-count within signalized links," *Transp. Res. C, Emerg. Technol.*, vol. 16, no. 1, pp. 18–35, Feb. 2008.

[15] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: A general approach," *Transp. Res. B, Methodol.*, vol. 39, no. 2, pp. 141–167, Feb. 2005.

[16] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[17] S. Carrese, E. Cipriani, L. Mannini, and M. Nigro, "Dynamic demand estimation and prediction for traffic urban networks adopting new data sources," *Transp. Res. C, Emerg. Technol.*, vol. 81, pp. 83–98, Aug. 2017.

[18] J. Xu, R. Rahmatizadeh, L. Boloni, and D. Turgut, "Real-time prediction of taxi demand using recurrent neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2572–2581, Aug. 2018.

[19] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.

[20] C. Li, L. Bai, W. Liu, L. Yao, and S. T. Waller, "Knowledge adaption for demand prediction based on multi-task memory neural network," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 715–724.

[21] L. Bai, L. Yao, S. S. Kanhere, Z. Yang, J. Chu, and X. Wang, "Passenger demand forecasting with multi-task convolutional recurrent neural networks," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2019, pp. 29–42.

[22] J. Ke, H. Zheng, H. Yang, and X. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 591–608, Dec. 2017.

[23] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, p. 1501, Jun. 2017.

[24] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin, "Contextualized Spatial–Temporal network for taxi origin-destination demand prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3875–3887, Oct. 2019.

[25] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2588–2595.

[26] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[27] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[28] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2018, pp. 362–373.

[29] P. Krishnakumari, H. van Lint, T. Djukic, and O. Cats, "A data driven method for OD matrix estimation," *Transp. Res. C, Emerg. Technol.*, vol. 113, pp. 38–56, Apr. 2020.

[30] L. Zhao *et al.*, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[31] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, "Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1227–1235.

[32] Y. Huang, Y. Weng, S. Yu, and X. Chen, "Diffusion convolutional recurrent neural network with rank influence learning for traffic forecasting," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 678–685.

[33] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3634–3640.

[34] L. Ruiz, F. Gama, and A. Ribeiro, "Gated graph convolutional recurrent neural networks," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2019, pp. 1–5.

[35] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1–12.

[36] J. M. Hernández-Lobato and R. Adams, "Probabilistic backpropagation for scalable learning of Bayesian neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1861–1869.

[37] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, vol. 3, 2015, pp. 1613–1622.

[38] C. Louizos and M. Welling, "Multiplicative normalizing flows for variational Bayesian neural networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 2218–2227.

[39] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[40] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter, "Bayesian optimization with robust Bayesian neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4134–4142.

[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, 2017, pp. 1–14.

[42] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[43] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, JMLR, 2017, pp. 933–941.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[45] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883–4894, Nov. 2020.

[46] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[47] C. R. Fox and G. Ülkümen, "Distinguishing two dimensions of uncertainty," in *Perspectives on Thinking, Judging & Decision-Making*. Oslo, Norway: Universitetsforlaget, 2011, pp. 21–35.

[48] A. Damianou and N. Lawrence, "Deep Gaussian processes," in *Artificial Intelligence and Statistics*. Scottsdale, AR, USA: PMLR, 2013, pp. 207–215.

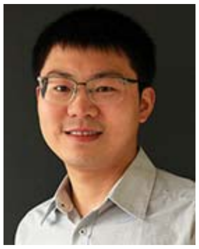
[49] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: https://arxiv.org/abs/1412.3555

[50] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling Long- and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 95–104.

[51] X. Zhou, Y. Shen, L. Huang, T. Zang, and Y. Zhu, "Multi-level attention networks for multi-step citywide passenger demands prediction," *IEEE Trans. Knowl. Data Eng.*, early access, Oct. 17, 2019, doi: 10.1109/TKDE.2019.2948005.

Can Li received the M.S. degree in computer and science engineering from Rutgers University. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, University of New South Wales (UNSW). She is currently supervised by Dr. Wei Liu, Prof. Travis Waller, and Dr. Lina Yao. Her research interests lie in deep learning for smart transport based on big data.

Lei Bai is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, University of New South Wales (UNSW). He is currently supervised by Dr. Lina Yao and Prof. Salil Kanhere. His research interests lie in deep learning and spatial–temporal data mining for smart cities applications, human pattern recognition (e.g. EEG, activity), and the IoT analytics.



Wei Liu received the B.Eng. degree in civil engineering from Tsinghua University, China, and the Ph.D. degree in transport system engineering from The Hong Kong University of Science and Technology. He is currently a Senior Lecturer in Transport Engineering with the University of New South Wales (UNSW), Sydney. He mainly works in transport system modeling and optimization; large-scale traffic modeling, simulation, and computing; transport economics; and urban big data analytics.

Lina Yao (Member, IEEE) received the Ph.D. degree in computer science from The University of Adelaide, Australia. She is currently a Scientia Associate Professor with the School of Computer Science and Engineering, University of New South Wales (UNSW). Her research interests lie in machine learning and data mining with applications to the Internet of Things, brain-computer interface (BCI), information filtering and recommending, and human activity recognition. She is a member of the ACM.

S Travis Waller received the B.S. degree in electrical engineering from The Ohio State University and the M.S. and Ph.D. degrees in industrial engineering and management sciences from Northwestern University. He was on the faculty at the Department of Civil, Architectural & Environmental Engineering, University of Texas at Austin. He is the Advisian Chair of Transport Innovation, the Deputy Dean (Research) of Engineering, and the Executive Director for the Research Centre for Integrated Transport Innovation (rCITI) at the University of New South Wales (UNSW).