2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS) | 978-1-6654-7177-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDCS54860.2022.00085

BikeCAP: Deep Spatial-temporal Capsule Network for Multi-step Bike Demand Prediction

Shuxin Zhong[§], Wenjun Lyu[§], Desheng Zhang[§], Yu Yang[‡]
[§]Rutgers University, New Brunswick, USA
[‡]Lehigh University, Bethlehem, USA
Email: {shuxin.zhong, wenjun.lyu}@rutgers.edu, desheng@cs.rutgers.edu, yuyang@lehigh.edu

***Abstract*—Given the recent global development of bike-sharing systems, numerous methods have been proposed to predict their user demand. These methods work fine for single-step prediction (i.e., 10 mins) but are limited to predicting in a multi-step prediction (i.e., more than 60 mins), which is essential for applications such as bike re-balancing that requires long operation time. To address this limitation, we leverage the fact that the demand for upstream transportation, e.g., subways, can assist the future demand prediction of downstream transportation, e.g., bikes. Specifically, we design a deep spatial-temporal capsule network called `BikeCAP` with three components: (1) a historical capsule that learns the demand characteristics for both the upstream (i.e., subways) and downstream (i.e., bikes) transportation systems, where a pyramid convolutional layer explores the simultaneous spatial-temporal correlations; (2) a future capsule that actively captures the dynamic spatial-temporal propagation correlations from the upstream to the downstream system, in which a spatial-temporal routing technique benefits to reduce the accumulated prediction errors; (3) a 3D-deconvolution decoder that constructs future bike demand considering the similar downstream demand patterns in neighboring grids and adjacent time slots. Experimentally, we conduct comprehensive experiments on the data of $30,000$ bikes and 7 subway lines collected in Shenzhen City, China, The results show that `BikeCAP` outperforms several state-of-the-art methods, significantly increasing the performance by 38.6% in terms of accuracy in multi-step prediction. We also conduct ablation studies to show the significance of `BikeCAP`'s different designed components.**

***Index Terms*—Distributed Transportation Systems, Demand Prediction, Bike Sharing**

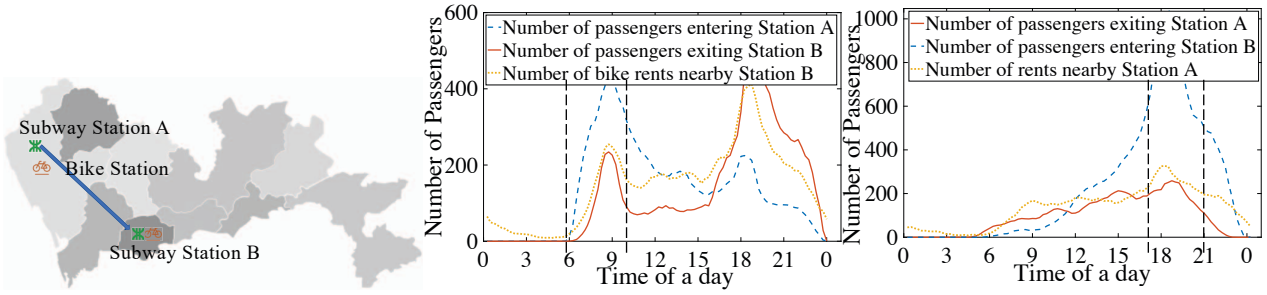
I. Introduction

Bike-sharing systems are an emerging transportation system in many cities such as New York City [1], [2], London [3], and Beijing [4]. These systems complement existing public transportation systems such as subways and buses and play an important role in solving the last mile problem on people's daily commute. For such a transportation system, due to the unbalanced demand distribution, user experience is significantly impacted by the bike shortage issue. To solve it, operators re-balance available bikes to improve user satisfaction [2], [5], which requires accurate bike demand prediction *ahead of time*.

The research community has been working on the bike demand prediction problem, and many works have achieved good performance in single-step prediction [6]–[8] (i.e., the demand in the next time slot). These works can be mainly divided into three categories according to how they capture spatial and temporal correlations. (1) Only temporal correlation: The first category of works [2] regarded the bike demand prediction problem as a time series prediction problem and solved it by linear regression models. (2) Asynchronous spatial and temporal correlation: The following works [5], [9] augmented the existing solution with multi-source heterogeneous data and captured the spatial and temporal correlations by spatial and temporal components, respectively. However, the separate spatial and temporal components cannot fully capture the synchronous spatial-temporal characteristics of bike flow propagation. For example, suppose a scenario that many bikes transport from grid A at t to grid B at $t+1$, which indicates that the passengers' flow propagation synchronously requires temporal (e.g., the impacts delayed due to the traffic propagation) and spatial (e.g., the impacts appears only between certain regions) patterns. Thus, the improvement is limited because the bike flows propagate along the synchronous spatial-temporal dimensions [10]. (3) Synchronous spatial and temporal correlation: Recent works [11], [12] addressed this limitation by considering synchronous spatial-temporal correlations with spatial-temporal graphs. However, those methods are only good at single-step prediction, i.e., predicting the check-ins or check-outs in the next time slot (i.e., 10 minutes) [12], which may fail to meet the time requirement when operators need to re-balance a large number of bikes far away, i.e., 60 minutes. Thus, the question we aim to answer is how to enable multi-step prediction with synchronous spatial-temporal correlations.

Our intuition is to leverage the fact that bike systems are generally for short-distance trips which compensate for other transportation systems (e.g., buses and subways) that are generally for long-distance trips. In this sense, if we can understand the relationship between short-distance trips and long-distance trips (e.g., predict the number of people who would use bike systems when they get on buses or subways), we can potentially better predict the demand of bikes ahead of time (i.e., multi-step bike demand prediction). For illustration purpose, we name these long-distance transportation systems *upstream transportation* and bike systems *downstream transportation*. For example, Fig. 1 shows an illustrative example with two subway stations and one bike station. During the rush hour in the morning (from 6 to 9 AM), the number of passengers (denoted by a blue dashed curve) entering station A located in a residential area increases ahead of the number of passengers exiting station B (denoted by a solid red curve) located in the CBD areas. Meanwhile, the number of bike

2575-8411/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDCS54860.2022.00085



(a) Illustration Example

(b) Correlations between demand of subway and bike stations

(c) Correlations between demand of subway and bike stations

Fig. 1: The number of subway and bike passengers over one day. As depicted on the map, station A is located in a residential area, and B is located in a CBD area. During the morning rush hours of the left figure, the number of entering passengers in station A (blue dashed curve) increases before that of exiting passengers in station B (solid red curve). The number of bike rentals shows a similar trend to that of outbound passengers nearby station B (yellow dotted curve). During the afternoon rush hours in the middle figure, the number of entering passengers in station B (blue dashed curve) increases before that of exiting passengers in station A (solid red curve). The number of bike rentals shows a similar trend to that of exiting passengers nearby station A (yellow dotted curve).

rentals nearby station B (denoted by a yellow dotted curve), e.g., within 200 meters, shows a similar trend to the number of passengers exiting station B (denoted by a solid red curve). In the middle figure, we found an opposite situation during the rush hour in the afternoon (from 15 to 21 PM), i.e., the number of passengers entering station B influence the number of bike rentals nearby station A. Based on these two figures, we argue that consolidating downstream bike data with other upstream transportation systems, e.g., subways, provides us a new opportunity to predict user demand *ahead of time* accurately, i.e., the multi-step bike demand prediction.

Though the idea of consolidating upstream transportation for bike prediction sounds straightforward, it is non-trivial to design such a prediction model because of two challenges.

- **Accumulated prediction errors in multi-step prediction.** A commonly used framework for multi-step prediction is based on autoregressive models that recursively leverage the predicted results as inputs to predict the next time step, which leads to accumulated prediction errors easily when the steps are long. In our scenario, the distance between two stations can be long, which leads to large accumulated prediction errors.
- **Time-specific correlations between upstream and downstream transportation systems**. We found that the correlations between two subway stations are temporally varied. For example, in Fig. 1, the number of passengers exiting station B is positively related to the number of passengers entering station A in the morning, whereas some are not related (i.e., at noon) or are negatively related (i.e., in the afternoon). Traditional methods based on Convolutional Neural Networks (CNNs) or Long-short Term Memories (LSTMs) adopt a fixed-size kernel with shared parameters cannot effectively explore such temporally-varied correlations between two stations [13].

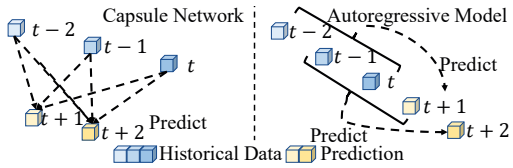


Fig. 2: Main Differences between Capsule Network and Autoregressive Model. The capsule network independently predicts each future time slot via historical time slots reconstruction. It reduces the accumulated errors caused by the autoregressive models, e.g., LSTM, which leverages the previously predicted results as inputs to predict the next time slot.

Motivated by the prevalence of capsule networks [14] in Computer Vision that effectively capture the location information of each object within an image (regarded as a *part-whole* relationship in the *spatial domain*, we design a novel spatial-temporal deep capsule network, named `BikeCAP`, to capture the **part-and-whole relationship** and reduce the accumulative errors in a temporal domain. As illustrated in Fig. 2, the "part" components in historical capsules (i.e., lower layers) present both the upstream and downstream transportation demand features in historical time slots; The "whole" component in future capsules (i.e., higher layers) indicates the future downstream transportation demand in each time slot. The connection between historical and future capsules is a dynamic routing mechanism, which leverages historical demand to re-construct that in each future time slot independently (as illustrated in Fig. 2). Different from autoregressive models such as LSTM that rely on previously predicted results as inputs to predict the next time slot, it effectively reduces the accumulated errors in the aforementioned *multi-step* series prediction. The main contributions of our work are as follows.

- To our knowledge, we make the first attempt in exploring

the time-specific spatial correlations between upstream transportation and downstream transportation for the multi-step demand prediction. In contrast, most of the existing works of demand prediction focus on single-step prediction without the synchronization between two transportation systems.

- Technically, we design a novel spatial-temporal capsule network that extends the previous capsule networks in the temporal domain. In particular, we design a pyramid convolutional layer in historical capsules to learn the spatial-temporal feature representation for both the upstream and downstream transportation demand along the propagation direction. Moreover, we introduce a spatial-temporal routing mechanism to actively capture the time-specific spatial connections between historical capsules and future capsules. Lastly, we adopt a 3D-deconvolution decoder to construct future bike demand considering the similar bike demand patterns in neighboring grids and adjacent time slots.
- We conduct extensive experiments on real-world datasets of $30,000$ bikes and 7 subway lines collected in Shenzhen City, China. The results show that `BikeCAP` outperforms the other 7 baselines by increasing the accuracy by 38.6% in multi-step prediction. Moreover, we also conduct ablation studies and parameters analysis to explain the effectiveness of `BikeCAP`.

II. Preliminaries

In this section, we introduce the background of the pyramid convolutional layer and the capsule network.

A. Pyramid-shape Design

Since we leverage the demand for upstream transport to assist the demand prediction of downstream transport, we need to explore the spatial-temporal correlations between historical transport (i.e., blue grid at $t-1$ and $t-2$) and the downstream transport (i.e., red grid at t), as shown in Fig. 3. The shape of such kind of spatial-temporal correlations motivates a pyramid-shape design, where the top of the pyramid (i.e., red grid) represents the demand at the target grid at the current time slot t, and other parts (i.e., blue grids) indicate the spatial-temporal correlated grids in historical time slots, i.e., $t-2$ and $t-1$. The more time slots we trace back, the larger areas we should consider (i.e., the shape of the kernel at time slot $t-2$, i.e., dark blue grids, is always bigger than that at $t-1$, i.e., light blue grids) because passengers can move farther distance within more time slots. Therefore, we design the pyramid-shape kernel to depict such spatial-temporal correlations along the flow propagation direction, which benefits learning the spatial-temporal correlations between proper grids with the target, ignoring the uncorrelated grids.

B. Capsule Network

In Computer Vision, Hinton and Sabour [14] borrowed the idea from neuroscience, where the brain is organized into modules. Then, they introduced capsules to act like modules

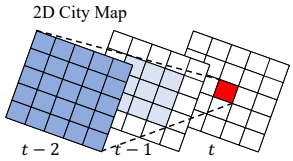


Fig. 3: The insight of Pyramid-shape Design. The top of the pyramid (denoted in red) represents the target grid at the current time slot t, and other parts (denoted in blue) indicate the related grids in historical time slots, i.e., $t-2$ and $t-1$.

(i.e., parts) to explore the features of different objects within an image (i.e., whole), especially for rotational relationships. i.e., position, size, orientation, deformation, and so on [14]. Based on that concept, they proposed the capsule network that incorporates a dynamic routing mechanism, which calculates the contribution between the **part-and-whole relationships** in a bottom-up manner.

Motivated by the prevalence of capsule networks in the *spatial domain*, we make an attempt to map its strength to model the **part-and-whole relationship** in the *temporal domain*. Specifically, as illustrated in Fig. 2, each capsule in the lower layer, i.e., historical capsules, is regarded as one module to capture the "part" features of traffic, including upstream and downstream, for each historical time slot. Meanwhile, each capsule in the higher layer, i.e., future capsules, represents the "whole" features of downstream transportation systems in every future time slot. The connection between historical capsules and future capsules is the dynamic routing mechanism, which benefits to learn the time-specific characteristics of passenger flow propagation from historical to future time slots in a bottom-up manner. It represents how the demand in upstream transportation systems reflects the future demand in the downstream transportation systems.

Compared with transformer [15] which is computed in a top-down manner and mainly focuses on the correlations within lower layers, we argue that **using capsule networks** is more suitable in our scenario where we aim to synchronously capture the time-specific spatial correlations between upstream and downstream transportation systems along the flow propagation direction. Moreover, different from autoregressive models such as LSTM, which recursively leverages the previously predicted results to predict the next time slot, the dynamic routing in future capsules re-constructs the demand features in future time slots independently, as illustrated in Fig. 2. Such an architecture is beneficial to effectively addressing the accumulated errors which appear in autoregressive models in the aforementioned *multi-step* series prediction.

III. Design

In this section, we introduce the formal problem definition. Then we briefly illustrate the overall architecture of `BikeCAP`, followed by detailed technical explanations, i.e., the designs in historical and future capsules.

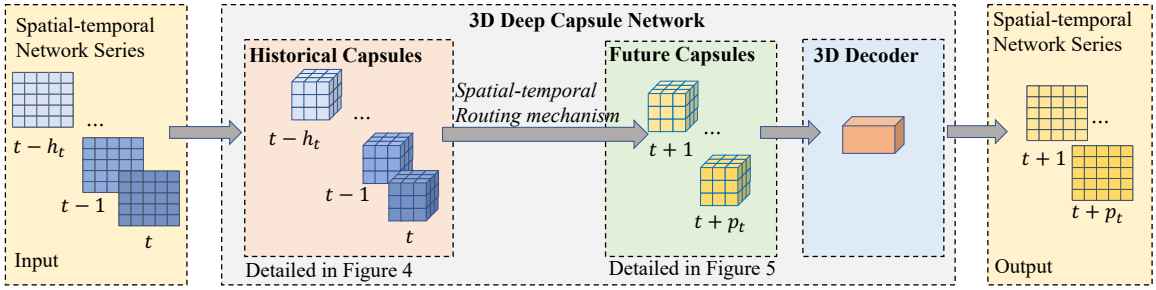


Fig. 4: The Architecture of `BikeCAP`. It consists of three components: Input, 3D Deep Capsule Network, and Output. The second component can be further divided into historical capsules and future capsules. We design a pyramid convolutional layer in historical capsules (detailed in Sec.2.3) for spatial-temporal correlations. We also design a spatial-temporal routing mechanism connecting the future capsules and historical capsules (detailed in Sec.2.4) for spatial-temporal correlations. At last, we adopt a spatial-temporal decoder for output (detailed in Sec.2.5).

A. Problem Definition

We target the multi-step bike demand prediction. It can be formulated as finding a function $\mathcal{F}$ that maps the historical demand series $\mathcal{X}_{[t-h:t]}$, including upstream and downstream, into the future downstream series $\mathcal{X}_{[t+1:t+p]}$.

$$\mathcal{X}_{[t+1:t+p]} = \mathcal{F}_\theta(\mathcal{X}_{[t-h:t]}) \tag{1}$$

where θ denotes all the learnable parameters and $\mathcal{X}_t \in \mathrm{R}^{(N_{g_1} \times N_{g_2})}$. The city is divided into $N_{g_1} \times N_{g_2}$ number of grids. The reason why we leverage the grid-based representation is that it is more flexible compared to graph-based representation and is adaptable in all scenarios [16]–[18]. For example, without a pre-defined graph structure, such as road interactions and road segments, constrains the deployment of graph-based representation, while the grid-based representation can be directly applied merely based on space partition.

Further, we use spatial-temporal network series to represent the demand data (a sample is provided in the input and output of Fig. 4). There are three types of correlations existing between grid i and j across different time steps: i) i and j indicate the neighboring nodes at the same time step t, named spatial correlation; ii) i is equal to j, which indicates the same node between the adjacent time steps, i.e., t and $t+1$, named temporal correlations; iii) different nodes, i.e., i and j, and different time steps represent the passengers' flow propagation direction, named synchronous spatial-temporal correlations. We aim to leverage these kinds of correlations to study the passengers' flow propagation and then predict the downstream demand in a multi-step scenario.

B. Overall Architecture

In Fig. 4, we present the architecture of `BikeCAP`, which contains three components: Input, 3D Deep Capsule Network, and Output. The second component can be further divided into historical capsules, future capsules, and 3D Decoder.

- The *historical capsules* are designed to learn the spatial-temporal demand features representation for both the upstream and downstream transportation systems in historical time slots. We implement the historical capsules as a *pyramid convolutional layer* with a 3D `squash` function. The rationale behind the pyramid convolutional layer is that the passengers' flow propagation requires time (e.g., delayed impacts) and has specific patterns (e.g., only between certain regions). This pyramid-based design is beneficial to consider the proper grids and time along with the space and time dimension, capturing the synchronous spatial-temporal correlations rather than separating spatial and temporal correlations.
- The *future capsules* adopt a *spatial-temporal routing mechanism* to learn the time-specific spatial-temporal correlations from demand in both upstream and downstream transportation systems to future demand in the downstream transportation system. It is a different type of computation compared to the autoregressive or one-dimension convolution methods, which depend on previous continuous time slots to predict the next time slot, i.e., it leverages $X^{t-h:t}$ to predict X^{t+1}. The dependency on previous time slots, which leverages the predicted results for the next prediction, leads to accumulated prediction errors. Compared with it, the spatial-temporal routing mechanism models how the upstream transportation demand in each historical time slot contributes to the downstream transportation demand for each future slot along the flow propagation direction independently.
- The *3D Decoder* consists of two 3D-deconvolution layers that utilize the similar bike demand existing in neighboring grids and adjacent time slots to increase the prediction ability further.

C. Historical Capsules

Historical capsules convert the spatial-temporal network series into the capsule domain while exploring the spatial-temporal demand representation for both the upstream and downstream transportation systems. The input of the historical capsules is a spatial-temporal demand flow network, as shown in Fig. 5. It is represented as a tensor $\mathbf{X} \in \mathbb{R}^{(N_{g_1}, N_{g_2}, h, f)}$ where f is the extracted features, and h is the number of historical time slots. Then, we adopt a pyramid convolutional

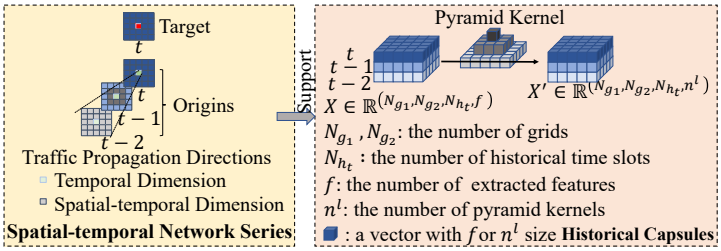


Fig. 5: The left part shows the propagation direction of the passengers' flow of the red node, i.e., target, in a spatial-temporal network. The green nodes indicate the same node in continuous time slots, representing the influence of the temporal dimension. The grey nodes indicate neighboring nodes, representing the impact across the synchronous spatial and temporal dimensions. These two correlations support the design of a pyramid convolutional layer.

layer to the demand features exploration for both upstream and downstream systems from $\mathbf{X}$ along the synchronous spatial-temporal dimensions. As illustrated in Fig. 5, the demand of the target grid (denoted in red at time slot t) can be influenced by the demand of itself (denoted in green) and its neighboring grids (denoted in grey) at previous time slots, i.e., $t-1$ and $t-2$. In order to capture the above influences, including spatial, temporal, and spatial-temporal correlations (detailed in Sec. III-A), simultaneously, we adopt a pyramid-shape kernel in convolution layers. Specifically, we use the number of kernel layers to indicate the number of historical time slots we traceback. The shape of the kernel in each layer depicts the correlated grids at that time slot. Intuitively, the longer time slots we trace back, the larger areas we should consider. For example, the shape of the kernel at time slot $t-2$ is always bigger than that at $t-1$ (denoted in grey in Fig. 5). Stacking the kernels in all the time slots constructs a pyramid-shaped kernel, and hence we name it a pyramid convolutional layer. Different from the standard convolutional layer that only considers the spatial correlation, such pyramid-shaped spatial-temporal kernels capture the synchronous spatial-temporal correlations along flow propagation directions for spatial-temporal network series. Moreover, the flow propagation requires time (e.g., delayed impacts) and also has specific patterns (e.g., only between certain regions). Such a design also benefits the elimination of the uncorrelated grids for correlations computation.

The propagation rule is given by $\Phi^l = f(X,K) = \delta(KXW^l)$, where $\Phi^l \in \mathbb{R}^{(N_{g_1},N_{g_2},h,n^l)}$, W^l is the weight tensor in layer l, and δ is an activation function. K is the pyramid kernel defined by:

$$K : \mathcal{X} \times \mathcal{X} \to \mathbf{R}, K(x,y) = \sum_{(x_i,y_i)\in\mathcal{X}_x\times\mathcal{X}_y} x_i y_i \qquad (2)$$

Specifically, a k-sized kernel K is set to be $K = [a_{t-k+1},\ldots,a_{t-1},a_t]$, with the size of a_t is 1×1, the size of a_{t-1} is 3×3, ..., and the size of a_{t-k+1} is $(2k+1)\times(2k+1)$. Intuitively, a larger k contains more spatial-temporal information and then improves the performance, but it may also cause higher computation cost (detailed in Sec. IV). Then, Φ^l will be passed through a 3D `squash` function defined as Eq. 3 to limit the length of the tensor, which represents the transportation demand in each historical time slot. For example, a longer tensor indicating closer demand correlations was shrunk to a length slightly below one, and a short tensor indicating uncorrelated relationships was shrunk to almost zero [14].

$$\hat{S}_{ijk} = \frac{||S_{ijk}||^2}{1+||S_{ijk}||^2}\cdot\frac{S_{ijk}}{||S_{ijk}||}, \qquad (3)$$

where $i \in N_{g_1}$, $j \in N_{g_2}$, and $k \in h$. The shape of outputs is $[N_{g_1}, N_{g_2}, h, n^l]$, where n^l is the dimension of each capsule in layer l.

D. Future Capsules

Future Capsules learn the spatial-temporal connections between historical demand features, including upstream and downstream, and demand in downstream systems. We design a spatial-temporal routing mechanism to model how the upstream and downstream transportation demand in each historical time slot contributes to the downstream transportation demand for each future slot independently, illustrated in Fig. 6. The main idea is to transform the demand for upstream and downstream transportation services in each historical capsule inside the historical capsules into one block to independently contribute to the downstream transportation demand in each future time slot. It benefits to explore the time-specific spatial correlations between upstream and downstream systems.

The key difference between **attention-based methods** and spatial-temporal routing is that the routing mechanism in the capsule network calculates the contribution from lower to higher layers, i.e., historical and future capsules, independently. It helps to model the passengers' flow propagation along the synchronous space and time dimensions in a bottom-up manner. Instead, attention is computed top-down, focusing on the relationships in the same layer, which cannot fully explore the flow propagation characteristics. Moreover, this approach breaks the dependencies between the previous continuous time slots and the following time slot and then reduces the accumulated multi-step prediction errors. In particular, we have three detailed steps as follows.

First, we reshape Φ^l into $\hat{\Phi}^l = (N_{g_1}, N_{g_2}, h\times n^l, 1)$ and convolve it with $(c^{l+1}\times n^{l+1})$ number of 3D convolutional kernels, whose strides are $(1,1,n^l)$. The purpose of keeping the

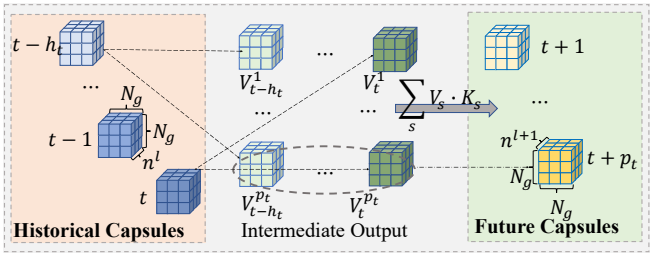


Fig. 6: Spatial-temporal Routing using 3D convolutions. From the high-level perspective, each capsule in historical capsules l predicts for c^{l+1} capsules in future capsules independently. In the first routing iteration, all the intermediate predictions $\mathbf{V}$ are equally weighted and summed for the final prediction $\mathbf{S}$. Then, in the following iterations, coupling coefficients $\mathbf{K}$ (i.e., the degrees of contribution) are updated according to the agreement with $\mathbf{S}$ and $\mathbf{V}$.

size of stride as n^l along the temporal dimension is to obtain the contribution of each historical capsule. The intermediate output, denoted as $\mathbf{V}$, has the shape of $(N_{g_1}, N_{g_2}, h, n^{l+1} \times p)$, which are reshaped to $(N_{g_1}, N_{g_2}, n^{l+1}, p, h)$ for the routing algorithm, where p is the length of predicted time slots.

Then, we initialize the logits $\mathbf{B}_s \in \mathbb{R}^{(N_{g_1}, N_{g_2}, p)}$ as $\mathbf{0}$, where $s \in [h]$. The corresponding coupling coefficients $\mathbf{K}_s$ are calculated by a 3D version of the existing `softmax` function Eq. 4 [19].

$$\begin{aligned} \mathbf{K}_s &= \texttt{softmax\_3D}(\mathbf{B}_s) \\ k_{ijks} &= \frac{exp(b_{ijks})}{\sum_x \sum_y \sum_z exp(b_{xyzs})} \end{aligned} \tag{4}$$

where $i, x \in N_{g_1}$, $j, y \in N_{g_2}$, and $k, z \in p$. Here, the logits are normalized among all predicted capsules from each capsule s in historical capsules l. It is because each predicted capsule in future capsules $l+1$ will receive N_h corresponding contributions from l. Each contribution is weighted with k_{ijks} to obtain the prediction outcome for one future time slot $\mathbf{S}_s$, i.e., $\mathbf{S}_s = \sum_s \mathbf{V}_s \cdot \mathbf{K}_s$. Thus, the $\mathbf{K}_s \in \mathbb{R}^{(N_{g_1}, N_{g_2}, p)}$ is the multi-step spatial-temporal connections between historical and future capsules. At last, we pass the $\mathbf{S}_s$ through a 3D `squash` function (Eq. 3) to limit the length of the capsule tensor as it represents the transportation demand, i.e., a longer tensor indicates huger demand.

E. 3D Decoder

The input of decoder is the tensors $\Phi^{l+1} \in \mathbb{R}^{(N_{g_1}, N_{g_2}, p, n^{l+1})}$ from future capsules. It is reshaped to $\hat{\Phi}^{l+1} \in \mathbb{R}^{(p, N_{g_1}, N_{g_2}, n^{l+1})}$, and fed into the 3D deconvolutional layers that leverage the spatia-temporal correlations residing in downstream demand patterns between the target grid and its neighbors in continuous time slots, i.e., spatial, temporal, and spatial-temporal correlations, for the future bike demand. Finally, we minimize the sum of squared differences between the predicted values and ground truth as the loss function.

IV. Experiments

In this section, we first introduce the datasets, followed by evaluation metrics and baselines. Then we represent the implementation details and experiment settings. At last, we evaluate the performance of `BikeCAP` compared with the latest SoAs, including ablation study and parameters analysis.

A. Datasets

By collaborating with the Shenzhen Transportation Agency, we have accessed two Shenzhen bike and subway datasets for experiments. Those two datasets are one-month-long (from 2018-10-01 to 2018-10-31), covering around $30,000$ bikes and 7 subway lines in Shenzhen, China. Our investigation here focuses on using subway and bike datasets as an example to evaluate the performance of leveraging the demand of upstream transportation systems to assist the downstream transportation demand prediction in multi-modal scenarios. The trip records are collected by the device when there is communication between devices and user terminals. The data collection process is approved by the users, and all the data is anonymous.

The subway dataset (as illustrated in Table I) can be further classified into boarding records and disembarking records. They both contain detailed spatial and temporal information, i.e., check-in or check-out time and corresponding stations.

The bike dataset (as illustrated in Table II) contains two types of records, i.e., pick-up and drop-off. It also contains the corresponding time, location (GPS points), and bike ID.

B. Evaluation Metrics and Baselines

To evaluate the performance of our model and the following baselines, two metrics **Mean Absolute Errors (MAE)** and **Root Mean Squared Errors (RMSE)** are adopted.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6}$$

where n is the number of instances, y_i indicates the ground truth of pick-up demands, and $\hat{y}_i$ denotes the predicted results.

- **XGBoost [20]**: It is a boosting-tree-based method that is widely used in data mining. For single-step prediction, we concatenate the historical records from the timeslot $t-h$ to t for each grid, respectively, to predict the number of

TABLE I: Subway-trip Record Format and Example

#Record	SZT ID	Time	Transportation	Status	Stations
0001	00001	2018-10-01 21:32:12	Subway Line No.1	Boarding	Guomao Station
0002	00001	2018-10-01 22:14:34	Subway Line No.1	Disembarking	Window of the World

TABLE II: Bike-trip Record Format and Example

#Record	User ID	Time	Location	Status	Bike ID
0011	00011	2018-10-01 01:24:38	GPS Point	Pick-up	00000
0012	00111	2018-10-02 11:13:43	GPS Point	Drop-off	00011

vehicles at timeslot $t+1$. For multi-step prediction such as two-step prediction, we first predict the outcome at timeslot $t+1$. Then we use it to construct the input data from $t-h+1$ to $t+1$ to predict the outcome at $t+2$. This process is conducted recursively for three or more steps prediction.

- **LSTM [21]**: Long Short-Term Memory Network is for time series prediction. The input of LSTM is the same as that of XGBoost, which comprises a single series of demands in historical time steps, and LSTM is required to learn from the series of historical observations to predict the next value in the sequence. Similar to that for XGBoost, we recursively conduct the process of single-step prediction for two or more steps prediction.
- **convLSTM [22]**: Before LSTM, it augments with convolutional layers for spatial correlations exploration. The size of the kernel in CNN decides the spatial range. A larger size contains more spatial information but induces higher computation costs. In the experiments, the size is 5 considering the balance between performance and cost.
- **PredRNN [23]**: A Predictive Recurrent Neural Network that memorizes both spatial appearances and temporal variations in a unified spatiotemporal memory pool. The input of PredRNN is the same as that of convLSTM.
- **PredRNN++ [24]**: It improves PredRNN with cascaded dual memories for the exploration of single-step dynamics.
- **STGCN [25]**: A Spatial-Temporal Graph Convolutional Network uses ChebNet and 2D convolutional networks to capture spatial and temporal correlations. We transfer each grid as a node, and use h-hop neighbor grids to construct the relation matrix, i.e., adjacency matrix. The grids within h-hop are considered as connected nodes. Then the grid partition is converted into a graph representation.
- **STSGCN [10]**: A Spatial-Temporal Synchronous Graph Convolutional Network combines graph convolutional layers with aggregating and cropping operations for the localized spatial-temporal correlations.

C. Implementation Details

We implement `BikeCAP` with Keras 2.4 [26] and test it on a server with NVIDIA A4000 GPU with Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz, 256GB memory. For the hyper-parameters, we set the batch size as 32, and the learning rate is set to 0.001. We set the number of capsules as 10 for historical inputs and the number of capsules as 1 for future output in a one-step prediction. It varies from 2 to 8 in multi-step prediction. The pyramid size is set to 5, and the dimension of the capsule is set to 4. Besides, we choose L1 Loss as the loss function. We optimize it with the Adam optimizer for 100 epochs and do not apply any non-mentioned optimization techniques, e.g., learning rate decay or weights decay.

D. Experiment Settings

For the data pre-processing, we follow the method in [27] to aggregate 15-minute traffic data (e.g., the number of bike rentals or returns and the number of passengers entering or exiting each subway station) into one time slot, i.e., four time slots for each hour. Because the contribution of one input heavily relies on its relativity to other inputs, re-scaling is required to fix such a problem. In this paper, we adopt the min-max normalization strategy [28], and the values of all features are mapped to the range between 0 and 1. After prediction, we denormalize the prediction value for evaluation.

For the experiments, we split all datasets into training, validation, and testing sets with a $6:2:2$ ratio. We utilize two-hour, i.e., 8 time slots, continuous historical data to predict the future bike demand. In order to verify `BikeCAP` for a multi-step prediction, the length of future time slots ranges from 2 to 8, i.e., from 30 minutes to 2 hours. All the experiments are repeated 5 times, and the performances are presented using the "mean$\pm$standard deviation" format.

E. Experiment Results

1) Comparison to the State-of-the-art Methods: In Table III, we compare the performance of `BikeCAP` with different approaches. We found that `BikeCAP` outperforms other models in most cases, especially when the length of predicted time slots is larger than 5 where `BikeCAP` achieves better performance compared to others whose MAE or RMSE increases dramatically. This suggests that `BikeCAP` is applicable in multi-step prediction.

Generally, spatial-temporal models (e.g., convLSTM, PredRNN++, and STGCN) that take advantage of spatial-temporal correlations achieve better performance compared to those only for time series prediction, e.g., XGBoost and LSTM. Moreover, the graph-based models provide outstanding results compared with grid-based models (i.e., CNN-based models) that include some grids without useful data. This explains that the performance of STGCN and STSGCN are slightly better compared to `BikeCAP` at the beginning.

In particular, convLSTM and STGCN use two modules to model the spatial-temporal correlations, respectively. STGCN

TABLE III: Performance Comparison of the State-of-the-art Approaches

Baselines		XGBoost	LSTM	convLSTM	predrnn	predrnn++	STGCN	STSGCN	**BikeCAP**
MAE	PTS=2	8.27	5.92±1.75	5.77±1.61	5.37±1.30	4.60±1.59	1.48±0.14	**1.37±0.12**	1.66±0.31
	PTS=3	10.56	7.18±1.94	6.93±2.15	6.96±2.64	5.35±2.08	1.60±0.16	**1.43±0.11**	1.79±0.34
	PTS=4	11.34	10.06±2.37	9.28±2.30	8.33±1.38	7.78±1.81	1.82±0.14	**1.81±0.09**	1.82±0.33
	PTS=5	14.08	11.59±2.08	10.37±2.51	9.64±1.84	8.02±1.53	2.13±0.15	2.06±0.17	**1.86±0.41**
	PTS=6	17.96	13.44±2.48	12.60±2.16	10.84±2.23	10.81±1.72	2.70±0.10	2.61±0.12	**1.89±0.37**
	PTS=7	21.20	16.13±3.53	14.65±3.18	12.41±3.17	12.74±2.87	3.13±0.14	3.10±0.10	**1.98±0.32**
	PTS=8	28.35	18.69±3.20	17.34±3.24	15.96±3.30	15.22±2.52	3.32±0.17	3.32±0.17	**2.04±0.33**
RMSE	PTS=2	14.91	13.48±2.19	12.93±2.43	10.35±2.11	9.79±1.47	2.55±0.65	**2.38±0.58**	3.41±0.67
	PTS=3	17.38	12.93±2.76	11.64±2.16	11.07±2.02	10.94±1.73	3.06±0.81	**2.73±0.76**	3.56±0.89
	PTS=4	21.46	14.79±2.41	14.36±2.52	13.75±2.27	13.38±2.19	4.83±0.96	3.77±1.12	**3.70±1.31**
	PTS=5	28.77	18.14±2.87	16.81±2.63	15.71±2.97	14.67±3.45	5.44±1.27	4.01±1.18	**3.79±1.56**
	PTS=6	35.16	19.73±2.69	18.27±3.06	16.42±3.29	15.46±3.67	6.69±1.83	5.23±1.46	**4.19±1.42**
	PTS=7	39.72	21.88±3.48	20.16±2.97	17.38±2.83	16.77±2.71	7.96±1.57	6.81±1.39	**4.54±1.81**
	PTS=8	50.64	24.45±3.04	23.61±3.51	19.65±3.72	17.34±3.95	8.18±1.16	7.55±1.20	**4.94±1.75**

PTS: Predicted Time Slots

shares one module for all different periods, which can hardly capture the dynamics that existed in the multi-step prediction; whereas convLSTM predicts the future time slots in a recursive manner, i.e., utilizes consecutive previous $t-1$ time slots for the prediction of the t^{th} time slot, which leads to accumulated prediction errors. Because of these reasons, their models suffer a decreasing performance as the length of future time slots increases. In contrast, PredRNN and PredRNN++ intend to model the spatial-temporal correlations simultaneously. The limitation is that they concatenate the features of the nodes over neighboring periods directly instead of distinguishing their characteristic features at different time slots. To address this issue, STSGCN differentiates the individual nodes at different time slots and explores the complex spatial-temporal correlations at the same time, but they focus on single-step dynamics and localized areas.

Compared with the aforementioned works, our `BikeCAP` reconstructs the bike demand in each future time slot using the characteristic features of each grid from all historical time slots. The results show we accurately predict the bike demand for at least 8 time slots, i.e., two hours, which is early enough for resultant applications such as a large-scale bike re-rebalancing.

2) Ablation Study for Component Importance: To demonstrate the effectiveness of each component of our `BikeCAP`, we compare four variants with `BikeCAP` as follows.

- **`BikeCap-Sub`**: To justify the effectiveness of the consolidation with subway data, we train a variation of `BikeCAP` called `BikeCap-Sub` only with bike data.
- **`BikeCap-Pyra`**: To justify our consideration for spatial-temporal correlation along traffic propagation directions, we train a variation of `BikeCAP` called `BikeCap-Pyra` that replaced pyramid-based convolutional layers with the traditional ones in historical capsules.
- **`BikeCap-3D`**: To justify our consideration of spatial-temporal correlations in traffic patterns, we train a variation of `BikeCAP` called `BikeCap-3D` that implemented a reshape-based decoder rather than a 3D one.
- **`BikeCap-3D-Pyra`**: It can be regarded as a simple version of the 3D capsule network architecture proposed in Deepcaps [19], which applies a 2D convolutional layer, a 3D routing, and a reshape-based decoder.

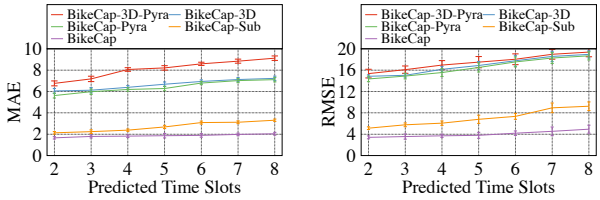


Fig. 7: Performance compared with Different Components.

TABLE IV: The Performance with Varying Size.

Size of Pyramid	RMSE	MAE
2	1.58±0.43	3.61±1.03
4	1.38±0.32	3.15±1.11
6	1.37±0.31	3.18±1.24
8	1.44±0.28	3.27±1.15

Fig. 7 provides the comparison in terms of MAE and RMSE. We notice that `BikeCap-Pyra` outperforms `BikeCap-3D-Pyra` by a large margin, which shows the necessity of exploring spatial-temporal correlations along the propagation direction of traffic flow. Besides, `BikeCap-3D` also peforms better compared to `BikeCap-3D-Pyra`. This is because the 3D deconvolutional layer can leverage the multi-dimensional correlated traffic patterns existing in neighboring grids better, compared with a reshape-based decoder that considers individual isolated grids. Moreover, the difference between `BikeCap-Sub` and `BikeCAP` verifies the effectiveness of leveraging upstream traffic sources, i.e., subways. The results suggest that it is feasible to utilize the subway data to benefit bike demand prediction.

F. Parameters Analysis

The Size of Pyramid Kernel. One key parameter in `BikeCAP` is the size of the pyramid kernel, which influences the range of spatial-temporal correlations. Fig. IV shows the performance with varying sizes. We can observe the performance increases as the size increases from 3 to 7. It is because a larger size considers more spatial-temporal information. However, once the size exceeds the threshold, i.e., 7 in the experiments, it leads to low performance, including some irrelevant information. Intuitively, larger size also causes

TABLE V: The Performance with Varying Dimension.

Dimension of Capsule	MAE	RMSE
2	1.98±0.37	4.23±1.13
4	1.83±0.34	4.11±1.23
8	1.38±0.32	3.15±1.15
16	1.63±0.35	3.94±1.24
32	1.81±0.28	4.37±1.22

a higher computation cost. Therefore, we set the size it to 5, considering the balance between the performance and the cost.

The Dimension of Capsule. Another critical parameter in `BikeCAP` is the dimensions of each capsule, which decides the information diversity (n^l in the model). Table. V shows the performance affected by the capsule's dimension. We observe that, on the one hand, the capsule with a larger dimension contains more information and is beneficial to more accurate spatial-temporal correlations; on the other hand, a too-large dimension significantly increases parameter numbers, leading to over-fitting and decreasing performance.

V. Discussion

A. Lessons Learned

Learned Lesson 1: The capsule network benefits the multi-step prediction. Our experiments confirm the feasibility of deploying the capsule network in a temporal domain, significantly increasing the performance of state-of-the-art methods in multi-step settings (supported by Table III). It is because traditional models such as autoregressive models [23], [24] or one-dimension convolutional layers [10] utilize previously predicted results to predict the following time step in multi-step prediction, while `BikeCAP` constructs the transportation demand at each future time step independently and then reduces the accumulated errors effectively.

Learned Lesson 2: The upstream transportation demand improves the performance of downstream transportation demand prediction. Our experiments prove the effectiveness of the consolidation designs that leverage the upstream transportation demand to assist in predicting the downstream transportation demand (supported by Fig. 7). The performance of `BikeCAP` is better than that of **`BikeCap-Sub`** to some extent.

Learned Lesson 3: Uncorrelated grids harm the performance. Our experiments prove that a larger range of considered neighbors benefit the accurate transportation demand prediction (supported by Table. IV). When the size of the pyramid increases from 3 to 7, the MSE and RMSE decrease gradually. However, when the size exceeds 7, the MSE and RMSE increase suddenly, because a larger pyramid size includes some unrelated grids, which harms the performance of the `BikeCAP`.

B. Limitations

Although the overall performance shows the effectiveness of `BikeCAP`, there are some obstacles to its real-world adoption.

- Stability. From Table III, we can observe that the overall performance of model `BikeCAP` is better than other SoAs, but the variances are relatively bigger than other models, i.e., STSGCN [10]. The high variance is mainly because capsule networks learn the representation of a specific time slot based on the inputs from all the nearby time slots, which may lead to a biased or inaccurate representation if the data in nearby time slots have a large variance. In our recent study, we found we can actually reduce this effect by introducing separated capsules for different time slots. We will do more research in our future work.
- Computation Cost. The vector representation and spatial-temporal routing mechanism cause many parameters, i.e., $646,395$, and training time, i.e., $90.40s$ for each epoch, for the synchronous spatial-temporal correlations with upstream and downstream transportation correlation. Considering the significant performance improvement shown in Table. III, the computation cost is moderate. And we can adopt parallel computing for acceleration.
- Scalability. In this project, we focus on predicting the downstream transportation demand with the aid of the upstream transportation demand. Thus for some positions far away from the upstream transportation stations, the performance is limited when considering the upstream transportation demand. In order to solve this limitation, we will add more transportation systems to realize scalability to improve the performance further.

C. Ethics and Privacy

During the data analysis and data mining of the transportation records, we took careful steps to address the privacy issues. First, all the users who use smart transportation cards are required to digest the Terms of Services, where consent the services companies can collect their trips records for studies. Second, during the data collection process, all the raw data has been preprocessed into aggregated anonymous statistics based on the privacy protection requirements. All the user identifiers are removed, and all the auxiliary information is strictly limited to public available station information (station name, station location, time, etc.). Even though we learn individual behavior in our aggregated analyses, we just analyze the encrypted ID and reduce the concern of privacy leakage of personal data. Third, only the authorized members of the research team who are assigned the strict non-disclosure agreements can access the shared data stored in a well-protected offline server.

D. Future Work

In this work, we focus on leveraging the upstream transportation demand to provide a more accurate prediction about the downstream transportation demand in multi-modal transportation modes. More important future works are worth exploring. First, we are limited on the upstream and downstream transportation datasets collected in Shenzhen. Although it is a good example of a metropolitan city in China with 17.56 million citizens, evaluations on other cities can help generate a thorough understanding of demand prediction in multi-modal transportation modes in large, medium, and small cities.

Second, although we propose an analytical framework for the station-level demand prediction, models that can provide more accurate transfer time prediction results at a station level are still lacking. For example, in future work, we can design

a self-supervised online framework that leverages passengers check-ins in upstream transportation modes to estimate average transfer time to different downstream transportation modes.

Third, leveraging previously mentioned station-level transfer time analyses, we can propose scheduling models to generate guidance to reschedule the transportation operation timetables to improve passengers' satisfaction. For example, we observe that there are a large number of passenger check-ins in upstream transportation and then predict the transfer time at different downstream transportation stations. If the transfer time at downstream transportation stations exceeds a predefined threshold, the operators can reschedule the downstream transportation timetables to reduce the transfer time.

VI. Related Work

Prediction for Bike-Sharing Systems. Many researchers [2], [5], [9] study the traffic prediction problem as a time series prediction problem augmented with multi-source and heterogeneous data. For example, Li et al. [2] designed a hierarchical prediction model to predict the number of rented and returned bikes to each station. Liu et al. [5] utilized the multi-source data, e.g., trip and station status records and weather reports, to predict the bike usage phenomena and then re-balance them accordingly. Hulot et al. [9] focused on predicting the hourly rental and return phenomena at each station augmented with weather reports. However, since the traffic flows along both the spatial and temporal dimensions, the improvement is limited if we only focus on the temporal representation.

To address this issue, some works added the geographical features into account to build up a spatial-temporal framework [6], [7], [11], [12], [29], such as the neighborhood traffic or the functions of regions. Typically, Liu et al. [11] developed a hierarchical bike demand predictor for expanding bike systems. Li et al. [12] learned the representation from heterogeneous spatial-temporal graphs together with multi-source information. However, when dealing with the unexpected surging bike demand caused by some special situations, e.g., festivals or events, these prediction methods may not reflect the increasing demand in time due to the data scarcity, which is inefficient when a large number of bikes are required to be re-balanced.

Spatial-Temporal Architectures. Because we model the bike and subway traffic as spatial-temporal network series, we describe it as the spatial-temporal network data prediction problem with many existing works [30]. The simplest way is to use separate modules to model the spatial and temporal correlations separately [31], [32]. For example, convLSTM [22] integrated the convolutional layers with an LSTM to process the spatiotemporal sequences; STGCN [25] formulated the traffic forecast problem on graphs and built the model with complete convolutional structures; TrajGRU [13] actively learned a location-variant structure for recurrent connections; SA-ConvLSTM [33] introduced a self-attention mechanism (SAM) into convLSTM to memorize features with long-range dependencies in terms of spatial and temporal domains.

To fully capture the simultaneous spatial-temporal correlations, some integrated spatial-temporal models have also been proposed [34]–[36]. For instance, Wang et al. [23] designed the PredRNN that contains a unified memory pool to memorize both spatial appearances and temporal variations simultaneously. Further, Wang improved it to PredRNN++ [24] for exploring single-step dynamics. MIM [37] turned time-variant polynomials into a constant for making the deterministic component predictable in order to learn complicated variations in space and time domains. E3D-LSTM [38] integrated 3D convolutions into RNNs for video prediction tasks. CubicLSTM [39] consists of three branches, i.e., spatial, temporal, and output, for capturing objects and predicting future motion. However, the limitation of these approaches is that they merely concatenate the nodes' features over neighboring periods rather than distinguishing their characteristic influences to individual future time slots. In contrast, STSGCN [10] effectively differentiated the individual nodes at different time slots and then captured the complex localized spatial-temporal correlations. However, this cannot differentiate characteristic influences from each historical time slot to each future time slot.

Capsule Network. Furthermore, with the advance in the capsule network, the idea of grouping the neurons to present more features has gained great attention. For instance, Sabour et al. [14] proposed a dynamic routing algorithm in capsule networks to estimate features of objects such as pose. Rajasegaran et al. [19] designed a deep capsule network architecture using a novel 3D convolution-based dynamic routing algorithm. However, the capsule network made a great process on Computer Vision and has not been applied in traffic prediction broadly. Therefore, the **key novelty** of `BikeCAP` is its first attempt to study the capsule network in the temporal domain, which benefits a multi-step bike demand prediction.

VII. Conclusion

In this work, we designed a deep spatial-temporal capsule network for multi-step bike demand prediction by assisting the correlations between upstream traffic and downstream traffic. The key novelty is its first attempt to study the capsule network in the temporal domain. Technically, we first introduced a pyramid convolutional layer in the historical layers to learn the spatial-temporal feature representation for both the upstream and downstream transportation demand. Then, we adopted a spatial-temporal routing mechanism to capture the time-specific spatial correlations from upstream transportation in historical capsules and downstream transportation in future capsules. Finally, we leveraged a 3D deconvolutional decoder to construct the future bike demand considering the similar bike demand patterns in neighboring grids and adjacent time slots. We evaluated `BikeCAP` based on real-world data collected in Shenzhen, China, to show that `BikeCAP` outperforms state-of-the-art methods in most of the cases.

Acknowledgement

This work is partially supported by NSF 1849238, 1932223, 1951890, 1952096, 2003874, 2047822, and Lehigh Sloan Research Grants. We thank all the reviewers for their insightful feedback to improve this paper.

REFERENCES

[1] D. Singhvi, S. Singhvi, P. I. Frazier, S. G. Henderson, E. O'Mahony, D. B. Shmoys, and D. B. Woodard, "Predicting bike usage for new york city's bike sharing system," in *Workshops at the twenty-ninth AAAI conference on artificial intelligence*, 2015.

[2] Y. Li, Y. Zheng, H. Zhang, and L. Chen, "Traffic prediction in a bike-sharing system," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2015, pp. 1–10.

[3] H. Li, Y. Zhang, H. Ding, and G. Ren, "Effects of dockless bike-sharing systems on the usage of the london cycle hire," *Transportation Research Part A: Policy and Practice*, vol. 130, pp. 398–411, 2019.

[4] Y. Tang, H. Pan, and Q. Shen, "Bike-sharing systems in beijing, shanghai, and hangzhou and their impact on travel behavior," in *Transportation Research Board 90th Annual Meeting*, vol. 11, 2011, p. 3862.

[5] J. Liu, L. Sun, W. Chen, and H. Xiong, "Rebalancing bike sharing systems: A multi-source data smart optimization," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1005–1014.

[6] S. He and K. G. Shin, "Towards fine-grained flow forecasting: A graph attention approach for bike sharing systems," in *Proceedings of The Web Conference 2020*, 2020, pp. 88–98.

[7] X. Yang and S. He, "Towards dynamic urban bike usage prediction for station network reconfiguration," *arXiv e-prints*, pp. arXiv–2008, 2020.

[8] Z. Pan, W. Zhang, Y. Liang, W. Zhang, Y. Yu, J. Zhang, and Y. Zheng, "Spatio-temporal meta learning for urban traffic prediction," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[9] P. Hulot, D. Aloise, and S. D. Jena, "Towards station-level demand prediction for effective rebalancing in bike-sharing systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 378–386.

[10] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 914–921.

[11] J. Liu, L. Sun, Q. Li, J. Ming, Y. Liu, and H. Xiong, "Functional zone based hierarchical demand prediction for bike system expansion," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 957–966.

[12] Y. Li, Z. Zhu, D. Kong, M. Xu, and Y. Zhao, "Learning heterogeneous spatial-temporal representation for bike-sharing demand prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1004–1011.

[13] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Advances in neural information processing systems* 2017, pp. 5617–5627.

[14] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, 2017, pp. 3856–3866.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[16] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *2019 AAAI Conference on Artificial Intelligence (AAAI'19)*, 2019.

[17] P. Schörner, C. Hubschneider, J. Härtl, R. Polley, and J. M. Zöllner, "Grid-based micro traffic prediction using fully convolutional networks," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* IEEE, 2019, pp. 4540–4547.

[18] X. Zhang, C. Huang, Y. Xu, L. Xia, P. Dai, L. Bo, J. Zhang, and Y. Zheng, "Traffic flow forecasting with spatial-temporal graph diffusion network," 2020.

[19] J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo, "Deepcaps: Going deeper with capsule networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 725–10 733.

[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, pp. 802–810, 2015.

[23] Y. Wang, M. Long, J. Wang, Z. Gao, and S. Y. Philip, "Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms," in *Advances in Neural Information Processing Systems*, 2017, pp. 879–888.

[24] Y. Wang, Z. Gao, M. Long, J. Wang, and S. Y. Philip, "Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning," in *International Conference on Machine Learning*, 2018, pp. 5123–5132.

[25] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.

[26] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[27] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1720–1730.

[28] Y. K. Jain and S. K. Bhandare, "Min max normalization based data perturbation method for privacy protection," *International Journal of Computer & Communication Technology*, vol. 2, no. 8, pp. 45–50, 2011.

[29] Y. Yang, F. Zhang, and D. Zhang, "Sharededge: Gps-free fine-grained travel time estimation in state-level highway systems," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–26, 2018.

[30] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[31] Y. Yang, X. Xie, Z. Fang, F. Zhang, Y. Wang, and D. Zhang, "Vemo: Enabling transparent vehicular mobility modeling at individual levels with full penetration," *IEEE Transactions on Mobile Computing*, 2020.

[32] Y. Yang, Z. Fang, X. Xie, F. Zhang, Y. Liu, and D. Zhang, "Extending coverage of stationary sensing systems with mobile sensing systems for human mobility modeling," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, pp. 1–21, 2020.

[33] Z. Lin, M. Li, Z. Zheng, Y. Cheng, and C. Yuan, "Self-attention convlstm for spatiotemporal prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 531–11 538.

[34] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.

[35] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *arXiv preprint arXiv:2007.02842*, 2020.

[36] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[37] Y. Wang, J. Zhang, H. Zhu, M. Long, J. Wang, and P. S. Yu, "Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9154–9162.

[38] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei, "Eidetic 3d lstm: A model for video prediction and beyond," in *International Conference on Learning Representations*, 2018.

[39] H. Fan, L. Zhu, and Y. Yang, "Cubic lstms for video prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8263–8270.