











# Forecasting Citywide Crowd Transition Process via Convolutional Recurrent Neural Networks

Zekun Cai , Renhe Jiang , *Member, IEEE*, Xinlei Lian , Chuang Yang , *Graduate Student Member, IEEE*, Zhaonan Wang , Zipei Fan , *Member, IEEE*, Kota Tsubouchi , *Member, IEEE*, Hill Hiroki Kobayashi , Xuan Song , *Member, IEEE*, and Ryosuke Shibasaki , *Member, IEEE*

**Abstract**—Perceiving and modeling urban crowd movements are of great importance to smart city-related fields. Governments and public service operators can benefit from such efforts as they can be applied to crowd management, resource scheduling, and early emergency warning. However, most prior research on urban crowd modeling has failed to describe the dynamics and continuity of human mobility, leading to inconsistent and irrelevant results when they tackle multiple homogeneous forecasting tasks as they can only be modeled independently. To overcome this drawback, we propose to model human mobility from a new perspective, which uses the citywide crowd transition process constituted by a series of transition matrices from low order to high order, to help us understand how the crowd dynamics evolve step-by-step. We further propose a Deep Transition Process Network to process and predict such new high-dimensional data, where novel grid embedding with Graph Convolutional Network, parameter-shared Convolutional LSTM, and High-Dimensional Attention mechanism are designed to learn the complicated dependencies in terms of spatial, temporal, and ordinal features. We conduct experiments on two datasets generated by a large amount of GPS data collected from a real-world smartphone application. The experiment results demonstrate the superior performance of our proposed methodology over existing approaches.

**Index Terms**—Crowd transition process, dynamic crowd flow, urban computing, deep learning.

Manuscript received 8 March 2023; revised 22 June 2023; accepted 24 August 2023. Date of publication 31 August 2023; date of current version 4 April 2024. This work was supported in part by Yahoo! Japan Research, and in part by Japan Science and Technology Agency (JST) SPRING, under Grant JPMJSP2108. Recommended for acceptance by A. Conti. (*Corresponding author: Renhe Jiang.*)

Zekun Cai, Renhe Jiang, Xinlei Lian, Chuang Yang, Zhaonan Wang, Zipei Fan, and Ryosuke Shibasaki are with the Center for Spatial Information Science, The University of Tokyo, Tokyo 113-8654, Japan (e-mail: caizekun@csis.u-tokyo.ac.jp; jiangrh@csis.u-tokyo.ac.jp; vickie\_lxl@csis.u-tokyo.ac.jp; chuang.yang@csis.u-tokyo.ac.jp; znwang@csis.u-tokyo.ac.jp; fanzipei@iis.u-tokyo.ac.jp; shiba@csis.u-tokyo.ac.jp).

Kota Tsubouchi is with Yahoo Japan Corporation, Tokyo 102-8282, Japan (e-mail: ktsubouc@yahoo-corp.jp).

Hill Hiroki Kobayashi is with the Center for Spatial Information Science, The University of Tokyo, Tokyo 113-8654, Japan, and also with Information Technology Center, The University of Tokyo, Tokyo 113-8654, Japan (e-mail: kobayashi@ds.itc.u-tokyo.ac.jp).

Xuan Song is with the Center for Spatial Information Science, The University of Tokyo, Tokyo 113-8654, Japan, and also with SUSTech-UTokyo Joint Research Center on Super Smart City, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, Guangdong 518055, China (e-mail: songxuan@csis.u-tokyo.ac.jp).

Digital Object Identifier 10.1109/TMC.2023.3310789

## I. INTRODUCTION

WITH the rapid popularity of personal mobile devices and Location Based Services (LBS), big human mobility data is continuously being generated through various sources, making it possible to perceive and model the urban crowd movements by these location data. Governments and public service operators can benefit from such efforts as these big human mobility data imply a wealth of urban knowledge [1], which can help tackle challenges such as crowd management [2], [3], resource scheduling [4], and early emergency warning [5]. To better understand human mobility, prior research mostly models the crowd movements by two types of representation. The first type describes *the state of the crowd* at a specific moment in the form of a snapshot, such as population density [3], and travel demand [6]. The other type is to describe *the change of the crowd* over a period of time in the form of a snippet, such as in/outflow [7], and the Origin-Destination (OD) matrix [8], [9].

However, existing crowd representation methods are hardly satisfactory when applied to practical applications. Specifically, snapshot data fail to describe crowd flow dynamics, which can only be used as statistical information in specific scenarios. Snippet data, although overcoming this problem, is limited by the fixed time interval when describing human mobility. Taking the OD matrix in Fig. 1(b) and (c) as an example, these two sub-figures show the distribution of people departing from Shinjuku station in Tokyo (marked in Fig. 1(a)) at 18:00 after 10 and 60 minutes. Different stakeholders in the city have distinct concerns regarding crowd dynamics: for example, event organizers focus on the short-interval matrix (i.e., Fig. 1(b)) as their main objective is to manage high-density gatherings of people in a specific spatio-temporal locality. Conversely, transportation departments may pay more attention to the long-interval matrix (i.e., Fig. 1(c)) to estimate travel demand and monitor abnormal traffic over a longer period. Both the immediate and long-term perspectives are essential as they enable informed decision-making across a range of urban applications. Nevertheless, effectively handling these tasks concurrently is a challenge for existing works as they are typically designed with a single granularity focus. A compromise approach is to train multiple isolated models, each with different OD time interval settings and targeting a specific application. However, since these tasks are deeply interconnected and homogeneous, implementing separate models not only results in inefficient use

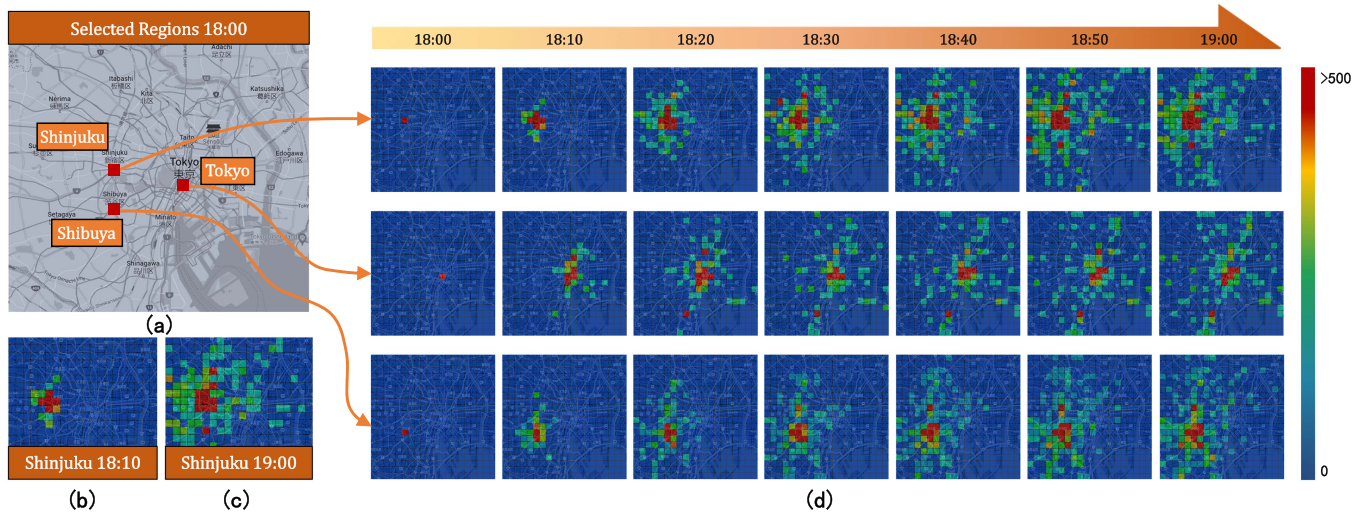


Fig. 1. People at Shinjuku/Tokyo/Shibuya Station on a weekday evening gradually flow to other regions of Tokyo within 1 h, which are three illustrative examples of human mobility process with respect to one specific location. Movements are shown as a mesh-grid layer overlaid on the map, with warm colors indicating larger flows. Compared to the one-time crowd transition in (b) and (c), the crowd transition process (d) can help us understand the diffusion of the crowd.

of computational resources but also disrupts the continuity and consistency inherent in human mobility patterns.

In this study, we investigate the modeling of citywide human mobility from a new perspective, which is defined as citywide **Crowd Transition Process (CTP)**. Our aim is to integrate the previously homogeneous problems by understanding *how the crowd dynamics evolve step-by-step*. The CTP data describes the flow of the crowd over a period of time through a series of transition matrices. Each matrix describes the diffusion of the population at a different timestamp during this period. Such a set of matrices together constitutes an indicator of the current crowd mobility. Three examples of CTP data are shown in Fig. 1(d), where crowds at Tokyo/Shinjuku/Shibuya stations progressively disperse to other city areas from 18:00 to 19:00 on a typical weekday evening. These six matrices at {18:10,..., 19:00} together compose a CTP tensor for one location. By stacking the CTP tensors of all areas, we can form a citywide CTP tensor. Compared to the traditional OD matrix that only uses the matrix at 19:00 to describe human mobility between 18:00 and 19:00, citywide CTP data can provide us with dynamic and continuous information within a single tensor, which allows urban stakeholders to unify previously independent sub-tasks into a single, consistent problem and address them under one comprehensive framework.

Given the historical observed citywide CTP tensor, predicting the next step of the citywide CTP tensor is a highly challenging task that is affected by the following three aspects. **1) Significant and Complicated Dependencies.** The CTP data are characterized by three dependencies - spatial, temporal, and ordinal. Spatial dependency arises due to frequent interactions and movements between different regions. Temporal dependency can be seen as the inter-hour correlation. It refers to the overall coarse-grained dependence that captures long-term trends and patterns of mobility behavior, such as the future CTP tensor can depend on the state of the crowd in the past few

hours. The ordinal dependency, taking Fig. 1(d) as an example, is that the transition matrix of the later timestamp shows a significant progressive divergence from the earlier ones, with each matrix representing one step in a diffusion. This is a more fine-grained dependence that captures the intro-hour fluctuations and variations within one CTP tensor. All three dependencies should be considered in a unified way. **2) High-dimensional Tensor.** The CTP prediction task necessitates dealing with high-dimensional tensors due to the stacking of transition matrices. Current models [3], [6], [7], [8], [10], [11] can struggle to handle the high dimensionality of CTP data as they have been following an analogous way to the image/video (up to 4D tensor) prediction tasks. Adapting these models to the new CTP data requires extra strategies to reduce the dimensionality such as embedding or tensor decomposition. However, traditional embedding methods lead to homogenizing the feature space and failing to provide distinctive representations for CTP data as their indiscriminate inclusion of irrelevant regions in the feature representation. A novel embedding approach should be introduced to deal with the relationship between information retention and compression. **3) Data Sparsity.** Crowd starting from one region rarely covers the entire urban area after a while, resulting in a very sparse CTP tensor. For example, the blue regions in Fig. 1(d) are empty. Such a sparse tensor will make the model difficult to converge and prone to skew.

To tackle these challenges, we propose a graph convolutional recurrent neural network, called **Deep Transition Process Network (DTP-Net)** to solve the prediction task. DTP-Net is built as a novel high-dimensional deep neural network which includes semantic Graph Convolutional Neural network (GCN), shared Convolutional LSTM (ConvLSTM), and High-Dimensional Attention (HD-Attention) modules. The network uses these modules to decouple complex dependencies in high-dimensional data. The semantic GCN will first learn grid embedding to capture the correlation existing between regions. Then

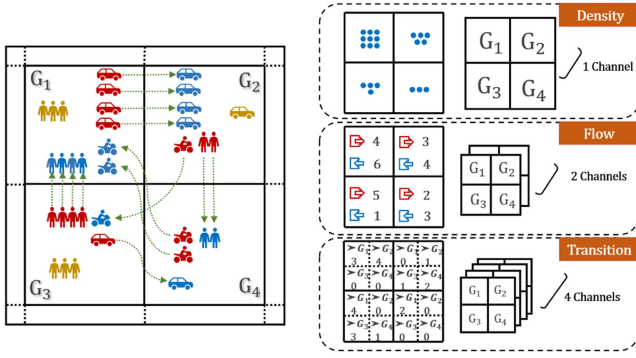


Fig. 2. Crowd-based citywide crowd prediction: density, in/out flow, and transition, analogous to video data prediction on 4D tensor (*Timestep*, *Height*, *Width*, and *Channel*).

the shared ConvLSTM will go through the high-dimensional data to explore the unique characteristics of each individual region. HD-Attention is further employed to take into account the diversity of different ordinal transitions. We conducted experiments on two datasets generated by a large amount of GPS mobility data. Experiment results demonstrate the superior performance of our proposed methodology. In summary, our work has the following contributions:

- We propose to use citywide crowd transition process as a new perspective to model the human mobility, which can significantly help us to understand the dynamics and continuity of the crowd movement.
- To effectively address the prediction of citywide CTP data, we propose a novel graph convolutional recurrent deep learning model, which deconstructs complex dependencies in high-dimensional data and explores them under a unified framework.
- We validate our model on two real-world smartphone GPS datasets. And we publish our data as the first large-scale crowd transition process dataset to help other researchers follow this work<sup>1</sup>.

## II. RELATED WORK

Many recent studies have analyzed human mobility data and they are summarized as a new research field called urban computing [1]. Among them, citywide human mobility prediction has been a representative branch of research. According to the modeling strategy, it could be divided into two categories: trajectory-based prediction and crowd-based prediction. The trajectory-based methods directly model trajectories as typical sequential data, whereas the crowd-based methods map trajectories to urban subregions and then do aggregation and prediction.

### A. Trajectory-Based Mobility Prediction

Many trajectory-based deep learning models were proposed to predict each individual's movement [12], [13], [14]. [12] extended a regular RNN by utilizing time and distance specific

transition matrices to propose an ST-RNN model for predicting the next location. DeepMove [13], considered as a state-of-the-art model for trajectory prediction, designed a historical attention module to capture periodicity and augment prediction accuracy. VANext [14] further enhanced DeepMove by proposing a novel variational attention mechanism. Besides, some studies focus on modeling millions of individuals' mobility for perceiving crowd movements at big events. [15], [16] simulated human emergency mobility following disasters. CityMomentum [17] predicted the rare behavior of each individual in a social crowd. DeepUrbanMomentum [18] further extended CityMomentum to an online deep learning version. In our experiment, we implement DeepMove [13] and DeepUrbanMomentum [18] as the trajectory-based baselines.

### B. Crowd-Based Mobility Prediction

The crowd-based mobility modeling methods divide a city into several regions, then the urban crowd movement can be revealed by aggregating the trajectory information of each region [19]. Depending on the statistical strategy, these methods can be further divided into three categories.

**Crowd Density Prediction:** The number of objects in each region can be regarded as density. Based on regions, predicting citywide crowd density with historical *Timestep* observations can be represented by a tensor of shape (*Timestep*, *Height*, *Width*, *Channel*=1) as demonstrated in Fig. 2. Density prediction is often used for gathering emergency warnings or travel demand prediction. DeepUrbanEvent [3] designed a multitask encoder-decoder to predict multiple-step crowd density. DeepSD [20], DMVST-Net [6], and Periodic-CRN [10] predicted taxi demand using the taxi request dataset collected from car-hailing companies. CoST-Net [21] predicted multiple transportation demands using both taxi and bike data.

**In/Out Flow Prediction:** Forecasting the citywide crowd flow based on mesh-grids has been proposed and addressed by [2]. As illustrated in Fig. 2, they define inflow and outflow to represent how many people will flow into or out of a certain region. The prediction data can be represented by a tensor (*Timestep*, *Height*, *Width*, *Channel*=2), where *Channel* stores the inflow and outflow. [2], [7], [22] have built deep learning models using deep neural network (DNN) and convolutional neural network (CNN) to make citywide predictions. Since then, a series of deep learning models are proposed to improve the performance of ST-ResNet [7], including STDN [23] and DeepSTN+ [11]. Some research, such as DeepCrowd [24] and STRN [25], have been proposed to solve fine-grained flow forecasting problems.

**Crowd Transition Prediction:** Since the in/out flow can't indicate the source and destination of the crowd, researchers further investigated the crowd transition modeling. As illustrated by Fig. 2, citywide crowd transition can depict how a crowd move among the entire mesh-grids. The problem can be represented by a tensor (*Timestep*, *Height*, *Width*, *Channel*=*Height* × *Width*), or (*Timestep*, *N*, *N*) where *N* = *Height* × *Width*. MDL [8] utilized Multitask Learning to simultaneously model and predict crowd in/out flow and crowd transition. [27], [28] conducted transition estimation from aggregated population data

<sup>1</sup><https://github.com/deepkashiwa20/DTP-Net>



TABLE I  
COMPARISON BETWEEN VARIOUS CROWD-BASED DATA REPRESENTATIONS  
(SHAPE AS IN NON-EUCLIDEAN OR EUCLIDEAN SPACES)

Perspective	Snapshot	Snippet		Process
Modeling Tensor	Density	Flow	Transition	Transition Process
Shape	(T,N,1)	(T,N,C)	(T,N,N)	(T,N,N,C)
Shape (Expanded)	(T,H,W,1)	(T,H,W,C)	(T,H,W,H,W)	(T,H,W,H,W,C)
Methods	[6], [10], [21]	[7], [23], [24]	[8], [9], [26]	-

and [29] estimates the transition populations using inflow and outflow.

It is worth mentioning that in addition to the division of urban areas into mesh-grids, recent studies have also explored spatio-temporal modeling in non-euclidean space. Graph convolution-based models have shown great success in such tasks. For example, STGCN [30], ASTGCN [31], MTGNN [32], STJGCN [33], and GraphWaveNet [34] for predicting urban traffic speed data demonstrate the efficiency and effectiveness of such designs [35]. ST-MGCN [36], Stg2seq [37], and Dynamic-GRCNN [38] constructed graphs to model the relationships between irregular regions and utilized GCN for predicting passenger demand and flow. GEML [9] and ODCRN [26] predicted the OD matrix via a novel graph neural network. STTN [39] combined GCN and Transformer to dynamically model long-range spatial-temporal dependencies.

We summarize the representation of the crowd-based mobility data in Table I, where modeling tensor, data shapes, and typical prediction models are explicitly compared. In the next section, we will introduce how to construct the new crowd transition process tensor.

### III. CITYWIDE CROWD TRANSITION PROCESS

To be general, in this study, we use the object to denote different GPS data sources generated by people, vehicles, bicycles, and so on. The trajectory of each object can be retrieved through the object uid ( $u$ ) from trajectory database  $\mathcal{T}$ . Given a spatial city area map divided into  $N = H \times W$  mesh-grids  $\{g_1, g_2, \dots, g_N\}$ , the trajectory of each object  $\mathcal{T}_u$  is linearly interpolated using a constant sampling rate  $\Delta t$  and then mapped onto mesh-grids as follows:

$$\mathcal{T}_u = (t_1, g_1), \dots, (t_n, g_n) \wedge \forall k \in (1, n], |t_k - t_{k-1}| = \Delta t. \quad (1)$$

**Definition 1** (Citywide crowd transition): Citywide crowd transition utilizes a matrix  $\Omega \in \mathbb{R}^{N \times N}$  to store how many objects transit from one grid to the others during the next time period, which is defined as follows:

$$\Omega_{ij}^{t, \Delta t} = |\{u | \mathcal{T}_u.g_t = g_i \wedge \mathcal{T}_u.g_{t+\Delta t} = g_j\}|, \quad (2)$$

where  $|\cdot|$  denotes the cardinality of a set.  $\Omega^{t, \Delta t}$  is 1-order transition matrix that stores the number of transitions from the timestamp  $t$  to timestamp  $t + \Delta t$ ,  $\Omega^{t, 2\Delta t}$  is 2-order transition matrix that stores the number of transitions from the timestamp  $t$  to timestamp  $t + 2\Delta t$ , and so on.

**Definition 2** (Citywide crowd transition process): Citywide crowd transition process  $\mathcal{O}_t$  is defined as a list of consecutive

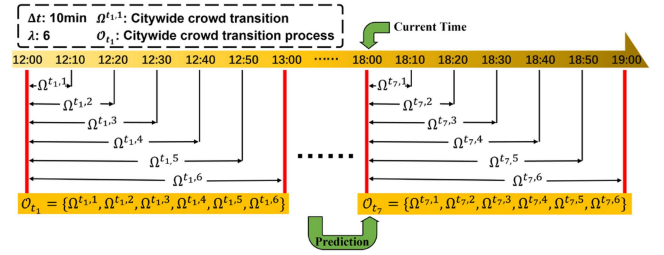


Fig. 3. Illustration of transition process prediction.

$\lambda$ -order transition matrices as follows:

$$\mathcal{O}_t = \{\Omega^{t, \Delta t}, \Omega^{t, 2\Delta t}, \dots, \Omega^{t, \lambda \Delta t}\}. \quad (3)$$

Here, we simplify the citywide crowd transition process as  $\mathcal{O}_t = \{\Omega^{t, 1}, \Omega^{t, 2}, \dots, \Omega^{t, \lambda}\} \in \mathbb{R}^{N \times N \times \lambda}$  by omitting the time interval  $\Delta t$ , through which the crowd transition from the 1st order to  $\lambda$ -th order could be retrieved.

**Problem 1** (Citywide crowd transition process prediction): Given observed  $\alpha$ -steps of citywide crowd transition process tensor  $\mathcal{X}_{in} = \{\mathcal{O}_{t_1}, \dots, \mathcal{O}_{t_{\alpha-1}}, \mathcal{O}_{t_\alpha}\}$  and metadata  $M_{t_\alpha}$  (calendar, weather, etc.), the prediction for the next step of citywide crowd transition process  $\hat{\mathcal{O}}_{t_{\alpha+1}}$  is modeled as follows:

$$\begin{aligned} \hat{\mathcal{O}}_{t_{\alpha+1}} &= \underset{\mathcal{O}_{t_{\alpha+1}}}{\operatorname{argmax}} P(\mathcal{O}_{t_{\alpha+1}} | \mathcal{O}_{t_1}, \dots, \mathcal{O}_{t_{\alpha-1}}, \mathcal{O}_{t_\alpha}, M_{t_\alpha}) \\ \forall i &\in (1, \alpha], t_i - t_{i-1} = \lambda \Delta t. \end{aligned} \quad (4)$$

A description of the crowd transition process tensor can be referred to Table I. We also give an illustration of the crowd transition process prediction task in Fig. 3, where the process in the future (18:00~19:00) is predicted by the last observations (6 hours from 12:00 to 18:00). Regarding our definition, the following points should be clarified:

- **What is the difference between crowd transition and crowd transition process?** Crowd transition is a one-time (single-order) transition matrix, while crowd transition process is a list of transition matrices from 1-order to  $\lambda$ -order.
- **What is the input for prediction of crowd transition and crowd transition process?** Given historical observations and  $N$  mesh-grids, the former is represented by  $(\alpha, N, N)$ , while the latter is represented by  $(\alpha, N, N, \lambda)$ .
- **What is the output for prediction of crowd transition and crowd transition process?** The next step of crowd transition is represented by  $(N, N)$ , while the next step of crowd transition process is represented by  $(N, N, \lambda)$ .

### IV. DEEP TRANSITION PROCESS NETWORK

Given the observed citywide Crowd Transition Process (CTP) tensor  $\mathcal{X}_{in} \in \mathbb{R}^{\alpha \times N \times N \times \lambda}$ , there are three dependencies along the tensor axis are required to be addressed: spatial dependence on the second and third dimensions  $(N, N)$ , temporal dependence on the  $\alpha$  dimension, and ordinal dependence on the  $\lambda$  dimension. We propose **Deep Transition Process Network (DTP-Net)** to decouple the dependencies in high-dimensional

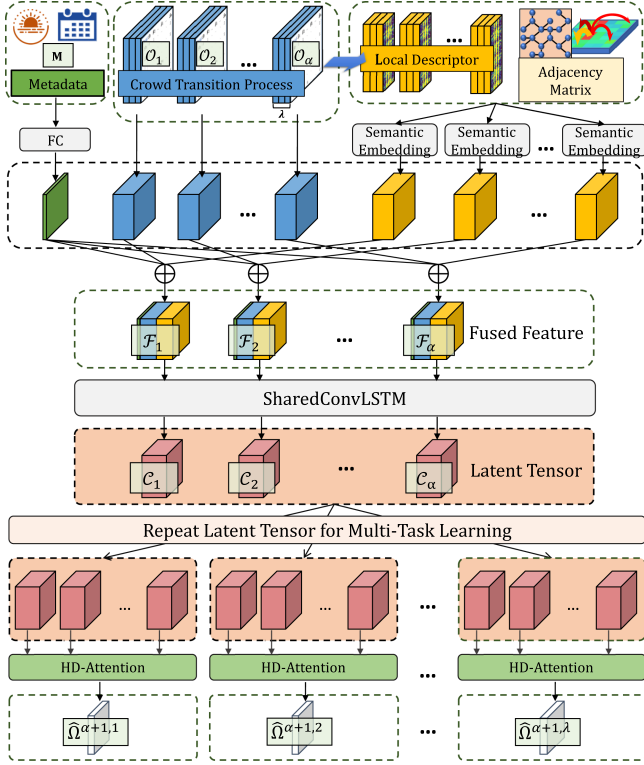


Fig. 4. Illustration of the structure of DTP-Net.

data by three sequential steps: 1) embedding the correlation between neighbor grids; 2) inferring the future transition process of each grid, and 3) considering the diversity among different transition orders, which presents a novel technique to address the great challenge of mining complex dependencies in CTP data. Fig. 4 presents an overview of our model. DTP-Net takes CTP data and metadata as input. The network first learns grid embedding by graph convolution based on the local transition descriptors, and then the embedding data is aggregated with the original CTP data and metadata as a fused tensor. After extracting the latent tensor from the fused tensor by the shared ConvLSTM, the final prediction of 1 to  $\lambda$  order crowd transition matrices are obtained by the high-dimensional attention mechanism. In what follows, we will present the technical details for each part.

#### A. Semantic Grid Embedding Via Graph Convolution

The frequent flow of people between two regions indicates a strong correlation between them, which in turn affects the crowd transition of one grid by the other. Integrating this dependency for each grid will assist in predicting how transitions will change in the future. Therefore, we design a local transition descriptor and perform grid embedding to explore interactions between different grids.

**Local Transition Descriptor:** We do the grid embedding for each order of the CTP data separately. Given the transition matrix of order  $o$ , an embedding captures correlations between grids by placing semantically similar grids close together in the

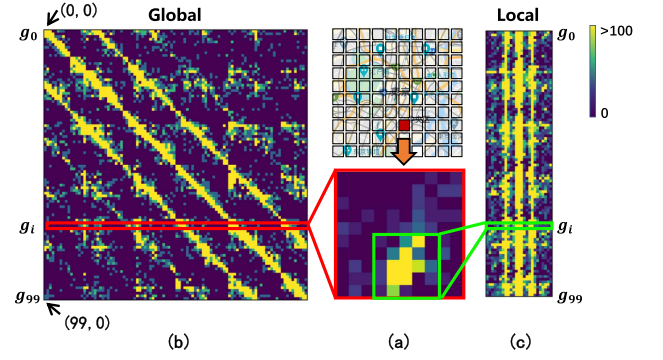


Fig. 5. Illustration of different designs of the grid feature.

embedding space. In practice, current embedding methods [8], [9] consider the destination distribution vector  $\Omega_i^{t,o} \in \mathbb{R}^{1 \times N}$  of grid  $g_i$  as its raw features. All features from the correlated grids of  $g_i$  are summed with learnable weights to produce the embedding result of  $g_i$ . However, this processing method is not suitable for the CTP prediction task. Since the inactive and low-speed people account for the majority of the population most of the time, the destination distribution of one grid roughly follows a two-dimensional normal distribution centered on itself as shown in Fig. 5(a), which means that the crowd destination distribution will surround the neighboring regions of the source grid, while the distant regions are sparsely populated. If we take the destination from this grid to the whole city as its raw feature (encircled by the red rectangle in Fig. 5(a)), after flattening and stacking all the grid features, the feature matrix will exhibit a crisscross structure that large values are concentrated on the diagonal. An example is shown in Fig. 5(b). The element at  $(0, 0)$  represents the number of people transiting from  $g_0$  to its nearest grid so it is a very large value. In contrast, the element at the parallel position  $(99, 0)$  represents crowd from  $g_{99}$  to its farthest grid so it is close to zero. Embedding such a matrix is relatively difficult: the feature space will be homogenized and over-smoothed after the weighted summation, which cannot provide us with a distinctive embedding representation for each grid.

To address this issue, we ignore the sparse distant regions but only focus on the dense neighbors of the source grid. We propose a new descriptor to characterize the grid instead of using the simple raw transition vector. As the green rectangular shown in Fig. 5(a), we utilize a kernel window with a shape of  $S = s \times s$ , which regards the crowd transition from  $g_i$  to its near neighbors as the local transition descriptor of  $g_i$ . The flattened vector of the local transition descriptor serves as the new original feature of the grid. Zero paddings are used for the grid of city borders. The local transition descriptors bring us to avoid the problem of over-smoothing after multi-layer convolution operations, as after stacking the descriptor for each grid over the entire city, the resulting local feature matrix  $\Psi^{t,o} \in \mathbb{R}^{N \times S}$  will appear structurally as Fig. 5(c), which makes it possible to maintain the consistency of the semantic structure while retaining important crowd transition information. The local transition matrix will

subsequently be input to the semantic GCN to gather correlations between grids.

**Semantic Grid Embedding:** As the different grids of the city are connected by roads and railways, the crowd transition in the mesh-grid map is analogous to message propagation in a graph, indicating that non-local relationships exist in the transition data. Since the limitation of CNN on gathering such non-local spatial dependency, we utilize Graph Convolutional Network (GCN) on crowd transition data based on the semantic adjacency matrix to learn grid embedding. The adjacency relation in the city is modeled as a semantic graph as  $\mathcal{G} = (V, E, A)$ , where  $V$  is the set of grids,  $E$  is the set of edges, and  $A$  is the semantic adjacency matrix defined by the transition intensity as follows,

$$A_{i,j} = \frac{\sum_{t=1}^T \sum_{o=1}^{\lambda} \mathcal{O}(t, i, j, o)}{\sum_{t=1}^T \sum_{j=1}^N \sum_{o=1}^{\lambda} \mathcal{O}(t, i, j, o) + \epsilon}, \quad (5)$$

where  $T$  represents all available observation steps.  $\epsilon$  is a small value close to zero preventing the denominator from being zero.

Existing research generalizes the CNN to GCN from two directions. The first one is the spectral-based method that defines graph convolutions in the spectral domain after the graph Fourier transforming [40]. The second one is the spatial-based method that aggregates the features of one node and its neighbors to form a new representation for the node [41], [42]. Since the spatial-based GCN has higher efficiency and flexibility, after getting the semantic adjacency matrix, we define the graph convolution on each local transition matrix in the spatial domain to obtain the embedding matrix as follows:

$$\Psi_{l+1}^o = \sigma(A\Psi_l^o W_l^o), \quad (6)$$

where  $W_l^o$  is the learnable parameters in the  $l$  layer and  $\sigma$  is the activation function. Note that we set the output dimension of the final embedding layer to  $N$  for subsequent operations. 1 to  $\lambda$  order embedding matrices are concatenated to form the embedding tensor  $\mathcal{E}_t \in \mathbb{R}^{N \times N \times \lambda}$ .

Through the semantic GCN, we capture the correlation between grids. Furthermore, due to the embedding tensor only containing partial information of the original CTP data, and external information such as date and weather can have a significant influence on human mobility, we fuse the raw CTP data, embedding data, and metadata together to form the fused tensor as follows:

$$\mathcal{F}_t = \mathcal{O}_t \oplus \mathcal{E}_t \oplus \mathcal{M}_t \quad (7)$$

$$\mathcal{X}_{fu} = \{\mathcal{F}_{t-\alpha+1}, \dots, \mathcal{F}_{t-1}, \mathcal{F}_t\}, \quad (8)$$

where  $\mathcal{M}_t \in \mathbb{R}^{N \times N \times 1}$  is the output of the metadata  $M_t$  go through fully connected layers.  $\oplus$  denotes the concatenation operator.  $\mathcal{X}_{fu} \in \mathbb{R}^{\alpha \times N \times N \times \gamma}$  is the fused tensor series and  $\gamma = 2 \times \lambda + 1$ .

### B. Transition Inferring Via Shared ConvLSTM

The inter-grid correlation and external influences are taken into account in the fusion tensor  $\mathcal{X}_{fu}$  by semantic GCN. Additionally, the possible future distribution of the transition is

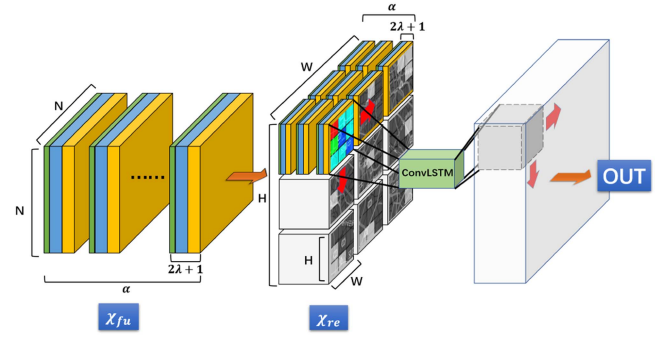


Fig. 6. Illustration of shared ConvLSTM module for transition inferring. The module first expands the input 4D tensor to a 6D tensor, then a parameter-shared ConvLSTM operator moves along the first two axes to extract features. The output tensor converts the expansion back to the same shape as the input.

then required to be inferred based on the temporal dependencies along with the embedding vectors. We propose to use ConvLSTM [43] as our sequential analysis component to capture the global spatio-temporal patterns within the CTP data, as it has been successfully applied in various time series prediction tasks and has proven to be a powerful tool for modeling data sequences [3], [44]. However, the fused tensor  $\mathcal{X}_{fu} \in \mathbb{R}^{\alpha \times N \times N \times \gamma}$  breaks the real spatial proximity, i.e., the neighbors indexed by  $(N, N)$  do not represent the physically adjacent grids, making it meaningless to perform the convolution operation directly on it. Therefore, we expand  $N$  to  $(H, W)$ , and then the unfolded tensor  $\mathcal{X}_{re} \in \mathbb{R}^{\alpha \times H \times W \times \alpha \times H \times W \times \gamma}$  will preserve the spatial adjacency between grids.

However, applying ConvLSTM, which typically deals with video-like 4D data, to such a 6D tensor is a non-trivial task. One major challenge is the potential loss of valuable spatio-temporal information when encoding multiple dimensions into a single channel to make it compatible with the ConvLSTM architecture. This can reduce the ability of the model to capture complex patterns in the data. Additionally, when dealing with high-dimensional data, the computational complexity associated with 6D tensors also increases, requiring the learning of a large number of parameters and posing a risk of over-fitting.

To address this problem, we propose a novel parameter-sharing ConvLSTM mechanism. Since the relationships between the grids have been embedded in the data in Section IV-A, we can focus on each grid individually when dealing with  $\mathcal{X}_{re}$ . The parameter-sharing mechanism is borrowed from convolutional neural networks, where the ConvLSTM cell slides over the 6D tensor as a convolution kernel slides over an image. This means that the feature extractor of one grid in the city is applied to the other grids as well. Specifically, as shown in Fig. 6, we first reshape and swap the axes of  $\mathcal{X}_{fu}$ , so that it becomes  $\mathcal{X}_{re} \in \mathbb{R}^{H \times W \times \alpha \times H \times W \times \gamma}$ . Then the ConvLSTM cell moves along the first two axes of  $\mathcal{X}_{re}$  to extract features for the 6D tensor. The output tensors will be concatenated and reshaped back to the same shape as the input to achieve multi-layer stacking. The parameter-sharing mechanism enhances the generalization ability of the model by reducing the parameter



space, making the computation concise and efficient. In addition, repeated training of a ConvLSTM cell alleviates the problem of divergence due to data sparsity, which provides us with a good latent representation of CTP data. Accordingly, the shared ConvLSTM layer can be formulated as follows:

$$\begin{aligned}
i_t(i, j) &= \sigma(W_{xi} * \mathcal{X}_{re}(i, j, t) + W_{hi} * \mathcal{H}_{t-1}(i, j) \\
&\quad + W_{ci} \odot \mathcal{C}_{t-1}(i, j) + b_i) \\
f_t(i, j) &= \sigma(W_{xf} * \mathcal{X}_{re}(i, j, t) + W_{hf} * \mathcal{H}_{t-1}(i, j) \\
&\quad + W_{cf} \odot \mathcal{C}_{t-1}(i, j) + b_f) \\
\tilde{\mathcal{C}}_t(i, j) &= \tanh(W_{xc} * \mathcal{X}_{re}(i, j, t) + W_{hc} * \mathcal{H}_{t-1}(i, j) + b_c) \\
\mathcal{C}_t(i, j) &= f_t(i, j) \odot \mathcal{C}_{t-1}(i, j) + i_t(i, j) \odot \tilde{\mathcal{C}}_t(i, j) \\
o_t(i, j) &= \sigma(W_{xo} * \mathcal{X}_{re}(i, j, t) + W_{ho} * \mathcal{H}_{t-1}(i, j) \\
&\quad + W_{co} \odot \mathcal{C}_t(i, j) + b_o) \\
\mathcal{H}_t(i, j) &= o_t(i, j) \odot \tanh(\mathcal{C}_t(i, j)), \tag{9}
\end{aligned}$$

where  $W$  is weight,  $b$  is bias,  $*$  denotes the convolution operator and  $\odot$  represents Hadamard product.

### C. Multi-Order Learning Via Attention

As shown in Fig. 1(d), as the period between 18:00 and 19:00 progresses, objects become increasingly dispersed throughout the city, resulting in a varied distribution of transition matrices of different orders. This necessitates taking diversity into account while predicting the transition process. Naturally, the attention mechanism can be used to highlight the most relevant information in the latent representation, which helps capture the unique distribution of each transition order. The attention mechanism operates by assigning different weights to the latent representation based on their relevance to the prediction order, enabling it to focus on the most important elements generated by the shared ConvLSTM while suppressing the less important ones. Originally, attention models accept a two-dimensional tensor (*TimeStep*, *Feature*) as input, and generated a one-dimensional attention feature vector. Nevertheless, the computation of attention involves a matrix multiplication operation between the query, key, and value matrices, which have a size proportional to the length of the sequence and the dimension of the hidden representation. This leads to the computation time increasing quadratically, making it computationally intensive and prohibitively expensive for the high-dimensional CPT tensor.

In this study, we propose an efficient **High-Dimensional Attention (HD-Attention)** mechanism to handle the computation of the transition process. The module extends the original attention module by taking a 6D tensor  $\mathcal{C} \in \mathbb{R}^{H \times W \times \alpha \times H \times W \times \gamma}$  generated by the shared ConvLSTM ((9)) as input and outputs a 4D attention tensor  $\hat{\Omega} \in \mathbb{R}^{H \times W \times H \times W}$ . HD-Attention decomposes the attention calculation into each city region by first computing self-attention scores on the spread tensor along the temporal dimension. It then weights each element with the scores and applies 1D convolution to obtain a compact 4D tensor. Finally, the compact representation of the city as a whole is multiplied

by a learnable parameter matrix to fine-tune the result based on the characteristics of each city region and process order. The formulas for the HD-Attention block are listed as follows:

$$\begin{cases} z_t = \tanh(W_Q \cdot \mathcal{C}_t(i, j) + b_Q) \\ \varphi_t = \frac{e^{z_t}}{\sum_j e^{z_j}} \\ \mathcal{D}(i, j) = \text{Conv}(\sum_{t=1}^{\alpha} \varphi_t \cdot \mathcal{C}_t(i, j)) \\ \hat{\Omega} = W_S \circ \mathcal{D}, \end{cases} \tag{10}$$

Here,  $W_Q$  is the weight and  $b_Q$  is the bias,  $\mathcal{C}_t(i, j)$  is the  $t$ -th hidden state outputted by shared ConvLSTM.  $W_S$  are learnable parameters for adjusting the degree of transition in different city areas at different orders. *Conv* is a convolutional operation with one convolutional filter, and  $\circ$  is Hadamard product (i.e., element-wise multiplication). Note that  $\mathcal{C}_t(i, j)$  is a 3D tensor ( $H, W, \text{Filter}$ ),  $W_Q$  and  $W_S$  have ( $H \times W \times \text{Filter}$ ) and ( $H \times W \times H \times W$ ) learnable parameters, respectively. Through the HD-Attention, the complexity of the computation is reduced to a linear relationship of the input dimensions, which greatly improves the efficiency while ensuring that the model captures the ordinal dependency.

For each transition order prediction within the next hour, we construct an independent HD-Attention branch to consider the diversity among them. Defining the output as  $\{\hat{\Omega}^{\alpha+1,1}, \hat{\Omega}^{\alpha+1,2}, \dots, \hat{\Omega}^{\alpha+1,\lambda}\}$ , then the model can be trained by minimizing the sum prediction error for each branch as follows:

$$L(\theta) = \sum_{o=1}^{\lambda} \beta \left\| \hat{\Omega}^{\alpha+1,o} - \Omega^{\alpha+1,o} \right\|^2 + \left| \frac{\hat{\Omega}^{\alpha+1,o} - \Omega^{\alpha+1,o}}{(\hat{\Omega}^{\alpha+1,o} + \Omega^{\alpha+1,o})/2} \right|, \tag{11}$$

where  $\theta$  are all learnable parameters in the DTP-Net.  $\beta$  is a hyperparameter to adjust the loss weight. The loss function consists of mean square error and mean absolute percentage error to avoid the training being dominated by large values.

We summarize the optimization process of DTP-Net as detailed pseudo-code in Alg. 1. During the training process, we first build the semantic adjacency matrix  $A$  by (5) and the local transition descriptor  $\mathcal{L}_t$  from the historical crowd transition process tensor  $\mathcal{O}_t$  (Lines 2-3). We then combine them with metadata  $M_t$  to construct a training sample (Line 7). We randomly select a batch of samples to feed into DTP-Net, apply the gradient descent approach, and update the model parameters  $\theta$  by (11) (Line 13). The trained model is obtained after a maximum number of epochs.

## V. EXPERIMENT

### A. Setting

**Dataset:** A smartphone application called Yahoo! Bousai is developed by Yahoo Japan Corporation to provide early information and warnings in response to different disasters such as earthquakes, rain, snow, and tsunamis in Japan. To precisely send local disaster alerts to users in relevant areas, the app collects real-time GPS trajectory data anonymously with the consent of users. The GPS logs are being generated from around 6 million users. The file size of each day is about 50 GB, containing

**Algorithm 1:** Training Process of DTP-Net.

---

**Require:** Historical crowd transition process tensor:  
 $\mathcal{O}_1, \dots, \mathcal{O}_T$ ; Metadata:  $M_1, \dots, M_T$ ;  
**Ensure:** Learned DTP-Net Model;

- 1: Initialization;
- 2: Build the semantic adjacency matrix  $A$  using (5);
- 3: Build the local transition descriptor tensor  $\mathcal{L}_1, \dots, \mathcal{L}_T$ ;
- 4: **for**  $\forall t \in [\alpha, T]$  **do**
- 5:    $I_{trs} = [\mathcal{O}_{t-\alpha}, \dots, \mathcal{O}_t]$ ;
- 6:    $I_{loc} = [\mathcal{L}_{t-\alpha}, \dots, \mathcal{L}_t]$ ;
- 7:   Append  $\{(I_{trs}, I_{loc}, M_t, A), \mathcal{O}_{t+1}\}$  to  $\mathcal{D}_{train}$ ;
- 8: **end for**
- 9: Initialize all learnable parameters  $\theta$  in DTP-Net;
- 10: **repeat**
- 11:   Randomly select a batch  $\mathcal{D}_{bt}$  from  $\mathcal{D}_{train}$ ;
- 12:   Calculate gradient  $\nabla g(\theta)$  using (11);
- 13:   Update  $\theta \leftarrow \theta + \alpha \nabla g(\theta)$ ;
- 14: **until** Stopping criteria is met;
- 15: **return** Learned DTP-Net;

---

approximately 800 million GPS records. Each record includes the user ID, timestamp, latitude, and longitude. In this study, data from Tokyo and Osaka cities are selected as the target datasets (named TokyoCTP and OsakaCTP), and 100 consecutive days (i.e., 2017/4/1 to 2017/7/9) are chosen as the target period.

*Preprocessing:* To address the sparseness, noise, and skewing of the data, we perform data cleaning and denoising before using linear interpolation to obtain 10-minute uniformly sampled calibrated human trajectories, (i.e.,  $\Delta t = 10$  minutes). Furthermore, we set  $\lambda$  to 6, then the time intervals of 1 to 6 order transition matrices are 10 min, 20min..... 60 min, and the time interval of two consecutive CTP tensors is 60 min. By setting  $\Delta Lon.=0.0125$  and  $\Delta Lat.=0.0083$  (approximately 1000 m $\times$ 1000 m), each metropolitan area is partitioned into a 20 $\times$ 20 mesh-grid map, then according to the Definition 2, the citywide CTP tensor can be generated from trajectories and represented by two tensors with the shape of (2400, 400, 400, 6). We normalize tensors to the range [0, 1] and rescale predicted tensors back to the normal values. One-hot encoding is used to transform the metadata (i.e., WeekOfYear, DayOfWeek, HourOfDay, and Holidays). The details of the dataset are summarized in Table II.

*Setup:* We apply an 80/20 training/test split on the dataset, and further select 20% of the training set as the validation set to adjust hyperparameters. The observation step  $\alpha$  is set to 6. 1 GCN layer with the kernel window shape of 5 $\times$ 5 local transition descriptor is utilized to get embedding tensor. The kernel size of shared ConvLSTM is set to 3 and the number of filters is 32. The depth of the shared ConvLSTM in transition inferring is set to 1.  $\beta$  is set to 1e9 to balance the loss scale. An early-stop Adam algorithm is used to control the overall training process, where the batch size is set to 1 and the learning rate is 0.001. Experiments are performed on a GPU server with two TESLA P40 graphics cards.

*Metrics:* We evaluate the overall performance based on three metrics: RMSE (Rooted Mean Squared Error), MAE (Mean

TABLE II  
SUMMARY OF EXPERIMENTAL DATASETS

Dataset	TokyoCTP	OsakaCTP
Mesh Size	1km $\times$ 1km	1km $\times$ 1km
H, W	20, 20	20, 20
Time Period	2017/04/01~ 2017/07/9	2017/04/01~ 2017/07/9
Time Interval	1 hour	1 hour
$\Delta t$	10 min	10 min
Maximum Value	6800	3300
Train/Test	8:2	8:2

Absolute Error), and MAPE (Mean Absolute Percentage Error).

$$RMSE = \sqrt{\frac{1}{n} \sum_i^n \|\hat{Y}_i - Y_i\|^2} \quad (12)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{Y}_i - Y_i| \quad (13)$$

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\hat{Y}_i - Y_i}{(\hat{Y}_i + Y_i)/2} \right| \quad (14)$$

where  $n$  is the number of samples,  $Y$  and  $\hat{Y}$  are the ground-truth tensor and predicted tensor.

### B. Baseline Models

- *CopyYesterday:* We use the value of the same time from the previous day as the predicted value.
- *CopyLastFrame:* We use the most recent observation as the predicted value.
- *DeepUrbanMomentum [18]:* DeepUrbanMomentum is proposed to predict individual trajectory. This model designs an RNN architecture to utilize one person's recent observations to predict his next multi-step movements.
- *DeepMove [13]:* DeepMove is an advanced individual trajectory prediction model with an attention mechanism and user embedding. It combines information from both recent observations and historical movements.
- *ConvLSTM [43]:* Convolutional LSTM is first introduced for precipitation nowcasting. It extends convolutional input and recurrent transformations to LSTM to handle video-like spatio-temporal prediction problems.
- *ST-ResNet [7]:* ST-ResNet is proposed to predict crowd flow of each region in a city. This model merges the time and flow dimensions together and uses three branches of CNN network to extract the seasonality of the data.
- *MDL [8]:* Flow prediction based on multi-task learning is an improved version of ST-ResNet, which can predict the crowd flow and transition at the same time.
- *DeepCrowd [24]:* A novel deep learning model for large-scale citywide crowd density and in-out flow prediction, by designing pyramid ConvLSTMs, 4D high-dimensional attention block, and early-fusion mechanism.
- *STGCN [30]:* A spatio-temporal traffic speed data prediction model that combines graph convolution with 1D convolution.



TABLE III  
OVERALL PERFORMANCE EVALUATION

Model	TokyoCTP				OsakaCTP			
	RMSE	MAE	MAPE	$\Delta$ RMSE	RMSE	MAE	MAPE	$\Delta$ RMSE
CopyYesterday	11.969	0.384	3.72%	+164%	3.352	0.131	2.01%	+144%
CopyLastFrame	5.756	0.291	3.78%	+27%	1.792	0.106	2.04%	+31%
DeepUrbanMomentum [18]	9.992	0.548	3.58%	+121%	3.574	0.181	1.80%	+160%
DeepMove [13]	8.445	0.491	3.76%	+87%	3.350	0.178	1.80%	+144%
ConvLSTM [43]	4.527	0.684	24.78%	-	1.373	0.142	2.14%	-
ST-ResNet [7]	3.407	0.574	6.13%	-25%	1.420	0.146	1.63%	-3%
MDL [8]	3.713	0.432	5.00%	-18%	1.241	0.185	1.96%	-10%
DeepCrowd [24]	2.181	0.236	3.31%	-52%	0.716	0.081	1.48%	-48%
STGCN [30]	3.197	0.934	4.14%	-29%	1.542	0.778	3.98%	+12%
MTGNN [32]	3.789	0.866	4.52%	-16%	1.412	0.465	2.17%	+3%
STTN [39]	3.266	0.749	4.15%	-28%	1.232	0.257	2.33%	-10%
GraphWaveNet [34]	3.002	0.456	4.49%	-34%	1.140	0.121	2.36%	-17%
GEML [9]	2.405	0.682	10.58%	-47%	0.982	0.259	2.92%	-28%
<b>DTP-Net (Ours)</b>	<b>1.251</b>	<b>0.167</b>	<b>2.80%</b>	<b>-72%</b>	<b>0.600</b>	<b>0.068</b>	<b>1.38%</b>	<b>-56%</b>

- *MTGNN* [32]: A general graph neural network framework that fuses external knowledge and one-way relationships between variables through a graph learning module.
- *STTN* [39]: The latest graph model proposes Spatial Transformer and Temporal Transformer modules to dynamically model long-range spatial-temporal dependencies in traffic data.
- *GraphWaveNet* [34]: A popular graph multivariate modeling model that uses parametric graph inputs and a WaveNet-like temporal dilated structure.
- *GEML* [9]: The origin-destination matrix prediction model is a state-of-the-art graph-based transition prediction model that utilizes graph embedding and periodic-skip LSTM to predict the OD matrix.

For trajectory-based baselines (i.e., DeepUrbanMomentum and DeepMove), we used these models to predict each person's movements for the next 6 steps, then aggregated the predictions to form the citywide CTP tensor to make it comparable to our model. For crowd-based baselines (i.e., ConvLSTM, ST-ResNet, MDL, DeepCrowd, STGCN, MTGNN, STTN, GraphWaveNet, and GEML), since they can not directly handle the high-dimensional CTP data, for each method, we trained six separate models for the 1 to 6 order crowd transition matrix prediction respectively, then concatenated the six results to form the predicted CTP tensor. Some statistical-based methods such as SARIMA and VAR have difficulty handling such high-dimensional data, so we did not compare with them.

### C. Performance Comparison

1) *Effectiveness Evaluation: Overall Performance:* Table III presents experiment results under three evaluation metrics of our model and baselines. We observe that (1) the state-of-the-art trajectory-based model cannot effectively handle the CTP data prediction problem. Their results are even worse than the simple baseline of CopyLastFrame. (2) Crowd-based models demonstrate the potential to solve the problem of CTP data prediction. The effect of these methods has advantages over other types of methods. (3) DTP-Net performs best among all crowd-based models under all metrics, and there is a significant improvement

compared with other methods: the results of RMSE show that DTP-Net is relatively 42% and 16% better than the second-best model on two datasets.

*Result Analysis:* First, the poor performance of the trajectory-based model is because these models only focus on the accuracy of the current trajectory but ignore the crowd behaviors. The crowd transition at a citywide level follows some periodic distributions and patterns, which are difficult to capture by trajectory-based models. Then, for existing crowd-based models, current processing methods used for high-dimensional spatio-temporal data primarily involve dimensionality reduction, which can lead to colossal information loss and thus compromise the accuracy of the model. Furthermore, for the citywide CTP prediction problem, all the crowd-based baselines require the construction of six separate models for prediction, which fail to capture the ordinal correlation of the CTP data. Finally, our proposed DTP-Net utilizes grid embedding, shared ConvLSTM, and HD-Attention to capture complex dependencies in a simultaneous and unified way, which supports the model achieve optimal performance compared to all baselines.

2) *Efficiency Evaluation:* In addition to comparing predictive accuracy, we also present efficiency comparisons of different methods in terms of computation time and neural network complexity, as they are important when deciding which method to use in real-world applications. The results are presented in Fig. 7. Based on the results, we observe that the proposed DTP-Net obtains a fairly competitive performance compared to all baselines, being above average in critical factors regarding time and storage, presenting an ideal trade-off in practical applications. Among the considered baselines, it achieves the best performance with 7x faster and 25x smaller compared to the second highest performing method DeepCrowd. Moreover, DTP-Net requires almost the same training time as the fully convolutional networks like ST-ResNet and MDL. This is because these baselines require training multiple models separately for transition process prediction, which significantly drags down the computation time. Accordingly, DTP-Net demonstrates a superior characteristic in terms of prediction performance, training efficiency, and storage occupation, making it a recommended solution for real-world CTP data prediction tasks.

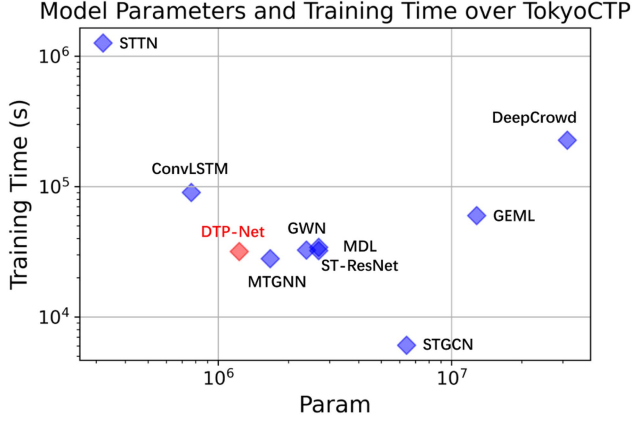


Fig. 7. Comparison of model complexity and training time.

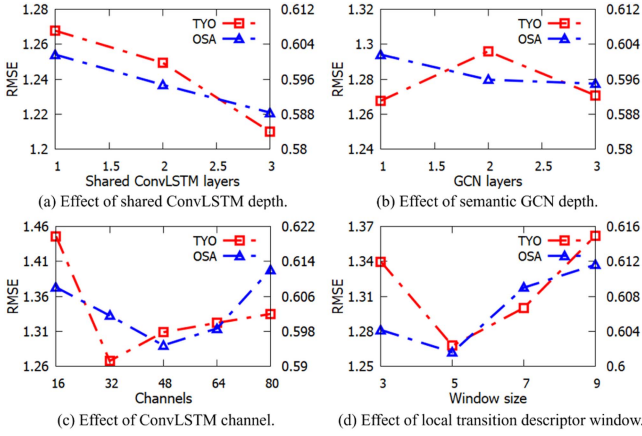
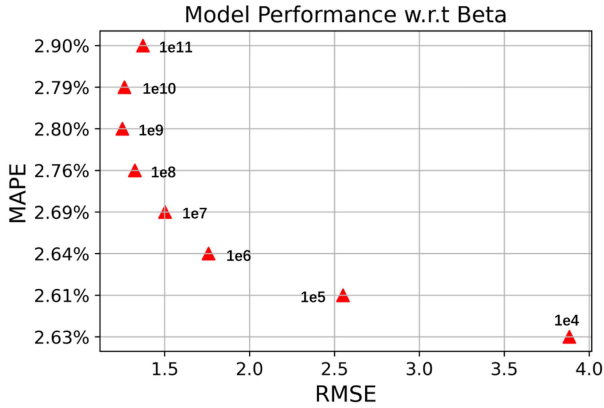


Fig. 8. Illustration of the hyper-parameter study.

Fig. 9. Influence of  $\beta$  on the model performance.

#### D. Ablation Test

To provide a comprehensive understanding of each component in DTP-Net, we conducted a series of experiments to validate the effectiveness of each module. We used GCN, Shared, and HD to represent the semantic GCN, shared ConvLSTM, and HD-Attention module of DTP-Net, respectively. Furthermore,

TABLE IV  
PERFORMANCE EVALUATION OF VARIANT MODELS

Method	TokyoCTP			
	RMSE	MAE	MAPE	$\Delta$ RMSE
Shared	1.427	0.190	2.912%	-
Shared+HD	1.373	0.186	2.868%	-3.74%
Shared+GCN(S/L)	1.294	0.172	2.776%	-9.30%
Shared+GCN(D/L)	1.434	0.182	2.838%	+0.54%
Shared+GCN(S/G)	1.326	0.174	2.842%	-7.03%
Shared+GCN(S/L)+HD	<b>1.251</b>	<b>0.167</b>	<b>2.80%</b>	<b>-11.21%</b>

different variants of the semantic GCN were also evaluated. The abbreviations S and D are used to represent whether the Semantic adjacency matrix or the inverse Distance matrix is used as the adjacency matrix. The abbreviations L and G are used to represent whether the Local transition descriptor (Fig. 5(c)) or the Global transition descriptor (Fig. 5(b)) is used as the raw grid embedding features. All the results are shown in Table IV, we can observe that,

- Even the simplest variant can achieve satisfactory results. It is because the shared ConvLSTM module helps us avoid compressing tensors so that high-dimensional data could be disposed of better.
- Both Shared+HD and Shared+GCN(S/L) achieve better results, which demonstrates the effectiveness of these modules. The improvement is due to the semantic GCN further encoding the knowledge from the correlated grids to the feature matrix, and the HD-Attention module enables each branch to focus on the unique distributions of each transition matrix.
- The low performance of Shared+GCN(D/L) indicates that the geographically adjacent regions may be weakly correlated. Besides, the performance of Shared+GCN(S/G) is worse than Shared+GCN(S/L), suggesting that the global transition descriptor can not provide the best grid embedding result.

#### E. Hyper-Parameter Study

- *The Depth of Shared ConvLSTM*: Fig. 8(a) shows the effect of shared ConvLSTM layer depth on both datasets. The results show that deeper networks yield lower RMSE values, indicating better performance. Due to memory and efficiency limitations, we only tested up to a depth of 3 and ultimately chose a depth of 1 as the experimental setup.
- *The Depth of Semantic GCN*: Fig. 8(b) illustrates the impact of semantic GCN layer depth on two datasets. The results indicate that the depth of the GCN layer has little effect on model accuracy. This is because the semantic adjacent matrix already captures all correlated grids within the 1-hop neighbor, eliminating the need for multi-layer stacking to capture multi-hop relationships.
- *The Filter Number of ConvLSTM*: Fig. 8(c) shows the impact of the ConvLSTM filter on two datasets. The results indicate that wider structures improve model performance in general. However, when the network width exceeds 48, the performance of the model becomes worse in both

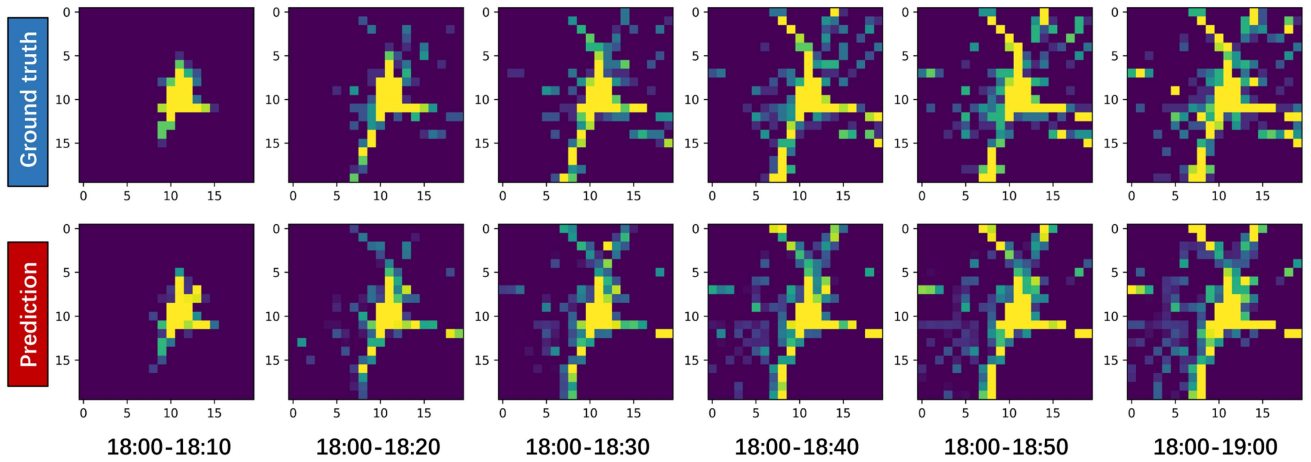


Fig. 10. Visualization of the crowd transition process from 2017-06-23 18:00 to 2017-06-23 19:00 w.r.t Tokyo Station. The top row shows the ground truth value and the bottom row shows the prediction of DTP-Net. Each of the six columns from left to right shows the increasingly higher-order transition within this hour. Warmer colors represent higher transition values.

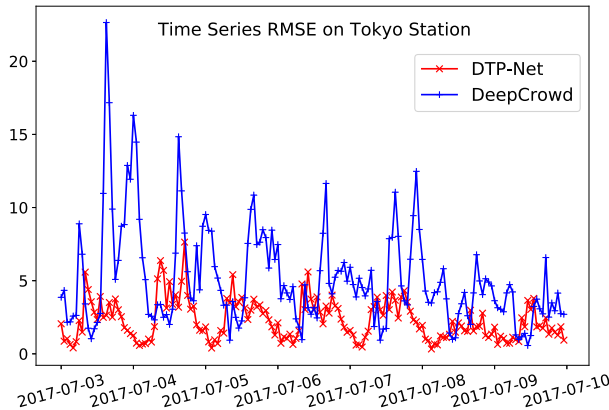


Fig. 11. Time-series RMSE on Tokyo Station from 2017-07-03 to 2017-07-10.

datasets, which is considered a typical vanishing gradient problem where very wide models cannot be fully trained.

- *The Window Size of Local Transition Descriptor:* We evaluated the effect of window size of local transition descriptor as shown in Fig. 8(d). The best results are obtained on the two datasets when the window size is 5, which shows the advantage of making good use of local information compared to using more global information.
- *The Loss Weight  $\beta$ :* The  $\beta$  in the loss function is used to balance the impact of both large and small errors, which can provide a fair view of the model's performance and help it learn to make more accurate predictions. We try to evaluate how  $\beta$  has impacts on the model's performance. Fig. 9 shows the sensitivity analysis results of  $\beta$ , where scatters represent the scores of model RMSE and MAPE under different  $\beta$  settings on TokyoCTP dataset. We find that with the increase of  $\beta$ , the MAPE shows a gradually increasing trend, while RMSE decreases first, reaches the minimum when  $\beta$  is  $1e9$ , and then begins to increase. Based

on this observation, we set  $\beta$  to  $1e9$  to balance the trade-off between the two metrics.

#### F. Case Study

We conducted two case studies on TokyoCTP to verify the prediction performance at a fine-grained spatio-temporal granularity. First, we plot time-series RMSE on Tokyo Station from 2017-07-03 to 2017-07-10 (the last week of test data). Based on the overall performance in Table III, we select the second-best model DeepCrowd [24] as a comparison. Through Fig. 11, we observe that DTP-Net almost outperforms DeepCrowd at each timestamp on Tokyo Station, and our model has a relatively stable effect at all times, while the DeepCrowd exhibits large performance fluctuations at the rush hour.

Second, we present a visualization to further illustrate the efficacy of our model using Fig. 10. The figure shows six columns of heat maps of crowd transition processes in Tokyo, captured during the evening rush hour from Tokyo Station, where each map represents the progression of crowd transition from Tokyo Station at increasing time intervals. The top row showcases the actual values, and the bottom row shows the predicted results from DTP-Net. The side-by-side comparison demonstrates our DTP-Net model effectively captures the primary patterns, trends, and magnitudes of crowd movements. As observed, the model accurately predicts the initial crowd dispersal from Tokyo Station to surround the Tokyo perimeter, further expanding into four prominent streams - two towards the north, one to the east, and one to the south. Importantly, these accurate predictions can provide invaluable insights for various urban applications like city planning, traffic management, and event organization, allowing relevant stakeholders to optimize resources, improve public services, and effectively respond to dynamic urban needs.

#### VI. CONCLUSION

In this study, we model human mobility from a new perspective that uses the citywide crowd transition process to describe



the urban crowd movement dynamics. A graph convolutional recurrent model called DTP-Net is designed to process and predict the high-dimensional crowd transition process data. The model utilizes graph convolution based on the local transition descriptor, parameter-sharing ConvLSTM, and high-dimensional attention modules to simultaneously capture all spatial, temporal, and ordinal dependencies. Experimental results based on two big real-world human trajectory datasets demonstrated the state-of-the-art performance of our model.

In the future, we intend to use a semi-sharing parameter module to take the characteristics of different locations into account and fuse more heterogeneous data to further boost performance. We also prepare to explore the simulation and prediction of new patterns of the crowd transition process in event situations, based on techniques such as lifelong learning, online learning, and concept drift adaptation. Also, based on our prediction model, we will build a real-time citywide crowd management prototype system to serve the general public, which can be used for urban computing critical issues such as traffic forecasting, anomaly analysis, and event or disaster response.

## REFERENCES

- [1] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: Concepts, methodologies, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, 2014, Art. no. 38.
- [2] M. X. Hoang, Y. Zheng, and A. K. Singh, "FCCF: Forecasting citywide crowd flows based on big data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2016, pp. 1–10.
- [3] R. Jiang et al., "DeepUrbanEvent: A system for predicting citywide crowd dynamics at big events," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2114–2122.
- [4] S. Ruan et al., "Dynamic public resource allocation based on human mobility prediction," in *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 1–22, 2020.
- [5] R. Jiang et al., "Predicting citywide crowd dynamics at big events: A deep learning system," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 2, pp. 1–24, 2022.
- [6] H. Yao et al., "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 2588–2595.
- [7] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.
- [8] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, Mar. 2020.
- [9] 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. Discov. Data Mining*, 2019, pp. 1227–1235.
- [10] A. Zonoozi, J.-J. Kim, X. Li, and G. Cong, "Periodic-CRN: A convolutional recurrent model for crowd density prediction with recurring periodic patterns," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3732–3738.
- [11] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, "DeepSTN+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1020–1027.
- [12] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 194–200.
- [13] J. Feng et al., "DeepMove: Predicting human mobility with attentional recurrent networks," in *Proc. World Wide Web Conf.*, 2018, pp. 1459–1468.
- [14] Q. Gao, F. Zhou, G. Trajcevski, K. Zhang, T. Zhong, and F. Zhang, "Predicting human mobility via variational attention," in *Proc. World Wide Web Conf.*, 2019, pp. 2750–2756.
- [15] X. Song, Q. Zhang, Y. Sekimoto, T. Horanont, S. Ueyama, and R. Shibasaki, "Modeling and probabilistic reasoning of population evacuation during large-scale disaster," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 1231–1239.
- [16] X. Song, Q. Zhang, Y. Sekimoto, R. Shibasaki, N. J. Yuan, and X. Xie, "A simulator of human emergency mobility following disasters: Knowledge transfer from big disaster data," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 730–736.
- [17] Z. Fan, X. Song, R. Shibasaki, and R. Adachi, "CityMomentum: An online approach for crowd behavior prediction at a citywide level," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2015, pp. 559–569.
- [18] R. Jiang et al., "DeepUrbanMomentum: An online deep-learning system for short-term urban mobility prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 784–791.
- [19] R. Jiang et al., "Yahoo! Bousai crowd data: A large-scale crowd density and flow dataset in Tokyo and Osaka," in *Proc. IEEE Int. Conf. Big Data*, 2022, pp. 6676–6677.
- [20] D. Wang, W. Cao, J. Li, and J. Ye, "DeepSD: Supply-demand prediction for online car-hailing services using deep neural networks," in *Proc. IEEE 33rd Int. Conf. Data Eng.*, 2017, pp. 243–254.
- [21] J. Ye, L. Sun, B. Du, Y. Fu, X. Tong, and H. Xiong, "Co-prediction of multiple transportation demands based on deep spatio-temporal neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 305–313.
- [22] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2016, Art. no. 92.
- [23] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5668–5675.
- [24] R. Jiang et al., "DeepCrowd: A deep model for large-scale citywide crowd density and flow prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 276–290, Jan. 2023.
- [25] Y. Liang et al., "Fine-grained urban flow prediction," in *Proc. Web Conf.*, 2021, pp. 1833–1845.
- [26] R. Jiang et al., "Countrywide origin-destination matrix prediction and its application for COVID-19," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Springer, 2021, pp. 319–334.
- [27] Y. Akagi, T. Nishimura, T. Kurashima, and H. Toda, "A fast and accurate method for estimating people flow from spatiotemporal population data," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3293–3300.
- [28] A. Sudo et al., "Particle filter for real-time human mobility prediction following unprecedented disaster," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2016, Art. no. 5.
- [29] Y. Tanaka, T. Iwata, T. Kurashima, H. Toda, and N. Ueda, "Estimating latent people flow without tracking individuals," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3556–3563.
- [30] 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.*, 2018, pp. 3634–3640.
- [31] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 922–929.
- [32] J. Sun, J. Zhang, Q. Li, X. Yi, Y. Liang, and Y. Zheng, "Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2348–2359, May 2022.
- [33] C. Zheng et al., "Spatio-temporal joint graph convolutional networks for traffic forecasting," *IEEE Trans. Knowl. Data Eng.*, early access, Jun. 13, 2023, doi: [10.1109/TKDE.2023.3284156](https://doi.org/10.1109/TKDE.2023.3284156).
- [34] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.
- [35] R. Jiang et al., "DL-Traff: Survey and benchmark of deep learning models for urban traffic prediction," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 4515–4525.
- [36] X. Geng et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3656–3663.
- [37] 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.*, 2019, pp. 1981–1987.
- [38] H. Peng et al., "Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting," *Inf. Sci.*, vol. 521, pp. 277–290, 2020.
- [39] M. Xu et al., "Spatial-temporal transformer networks for traffic flow forecasting," 2020, *arXiv: 2001.02908*.

- [40] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *Proc. Int. Conf. Learn. Representations*, 2014.
- [41] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2014–2023.
- [42] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
- [43] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.
- [44] Z. Yuan, X. Zhou, and T. Yang, "Hetero-ConvLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 984–992.



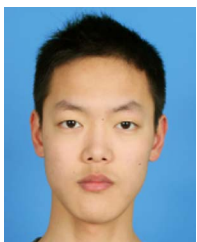
**Zekun Cai** received the BS degree in computer science and technology from the University of Electronic Science and Technology of China, in 2018, the MS degree in socio-cultural environmental studies from the University of Tokyo, Japan, in 2021. Now, he is working toward the PhD degree with the University of Tokyo. His research interests are mainly artificial intelligence, deep learning, and ubiquitous computing.



**Renhe Jiang** (Member, IEEE) received the BS degree in software engineering from the Dalian University of Technology, China, in 2012, the MS degree in information science from Nagoya University, Japan, in 2015, and the PhD degree in civil engineering from The University of Tokyo, Japan, in 2019. From 2019, he became an assistant professor with Information Technology Center, The University of Tokyo. His research interests include ubiquitous computing, deep learning, and spatio-temporal data analysis.



**Xinlei Lian** received the BS degree in urban, community, and regional planning from the Harbin Institute of Technology, China, in 2018, the MS degree in socio-cultural environmental studies from the University of Tokyo, Japan, in 2020. Her research interests include urban computing, neural networks, image change detection, and convex optimization.



**Chuang Yang** (Graduate Student Member, IEEE) received the BS degree in computer science and technology from the Southern University of Science and Technology (SUSTech) in 2019. From 2020, he became a master student with Department of Socio-Cultural Environmental Studies, The University of Tokyo. His current research interests are in the area of spatio-temporal data analysis, data visualization, and ubiquitous computing.



**Zhaonan Wang** received the BS degree in geographical information systems from Peking University, China, in 2014, and the MS degree in city planning from Boston University, USA, in 2017. From 2018, he joined National Institute of Advanced Industrial Science and Technology as a technical staff. His research interests are mainly on ubiquitous computing and spatio-temporal data mining.



**Zipei Fan** (Member, IEEE) received the BS degree in computer science from Beihang University, China, in 2012, the MS and PhD degrees in civil engineering from The University of Tokyo, Japan, in 2014 and 2017 respectively. From 2017, he became a project assistant professor with the Center for Spatial Information Science, The University of Tokyo. His research interests include ubiquitous computing, machine learning, spatio-temporal data mining, and heterogeneous data fusion.



**Kota Tsubouchi** (Member, IEEE) received the PhD degree from the University of Tokyo, Japan, in 2010. Until 2012, he did research about on-demand traffic systems with the University of Tokyo. Since 2012, he became a data scientist and senior researcher with Yahoo JAPAN Research. His research interest is data analysis with a focus on human activity logs (location information, search logs, shopping history, sensor data, etc.).



**Hill Hiroki Kobayashi** received the BS degree in computer science from California State University, USA, in 2005 and the PhD degree in interdisciplinary engineering from the University of Tokyo, Japan, in 2010. Now, he is a professor with Information Technology Center, The University of Tokyo, Japan. His research are mainly nonverbal interaction, sustainable interaction design, and nature conservation interface.



**Xuan Song** (Member, IEEE) received the BS degree in information engineering from Jilin University, China, in 2005 and the PhD degree in signal and information processing from Peking University, China, in 2010. He was promoted to project assistant professor and project associate professor with the Center for Spatial Information Science, The University of Tokyo, in 2012 and 2015, respectively. His research focus on smart city, intelligent system design, multi-target tracking, sensor fusion, and abnormality detection.



**Ryosuke Shibasaki** (Member, IEEE) received the BS, MS, and doctoral degrees in civil engineering from The University of Tokyo, Japan, in 1980, 1982, and 1987, respectively. Currently, he is working as a professor with the Center for Spatial Information Science, The University of Tokyo. His research interests cover three-dimensional data acquisition for GIS, conceptual modeling for spatial objects, and agent-based microsimulation in a GIS environment.