Learning Semantic Behavior for Human Mobility Trajectory Recovery

Wangchen Long, Zhu Xiao, *Senior Member, IEEE*, Hongbo Jiang, *Senior Member, IEEE*, Yong Xiong, Zheng Qin, You Li, and Schahram Dustdar, *Fellow, IEEE*

***Abstract*— Trajectory recovery aims to restore missing data for reconstructing high-quality human mobility trajectory, which benefits a wide range of intelligent transportation system applications ranging from urban planning to travel recommendation. Inspired by the inherent regularity of human mobility, existing approaches capture spatial-temporal transition regularities in historical trajectory for data recovery. Although promising, existing solutions suffer from two limitations. *i)* These methods fail to recover occasionally-visited points (OVP) due to the lack of semantic information when learning spatial-temporal transition regularities. *ii)* The information before and after missing data is not be fully utilized for trajectory recovery. To overcome the limitations, we propose a novel semantic-aware trajectory recovery framework. First, we leverage heterogeneous information network (HIN) to encode various semantic correlations for obtaining rich semantic embeddings, which are fused with temporal information to form spatial-temporal semantic context. Then, we develop a behavior attention mechanism to capture semantic behavior transition regularities for trajectory recovery based on the bidirectional spatial-temporal semantic context before and after missing data. Extensive experiments on four real-world datasets show that our proposed method outperforms the state-of-the-arts by 7%-11% in term of recall, F1-score and mean average precision.**

***Index Terms*—**
Human mobility, trajectory recovery, heterogeneous information network, attention mechanism.

Manuscript received 25 June 2022; revised 2 September 2023; accepted 2 January 2024. This work was supported in part by NSFC under Grant 62272152, in part by the National Key Research and Development Program of China under Grant 2022YFE0137700, in part by the Humanities and Social Sciences Foundation of Ministry of Education under Grant 21YJCZH183, in part by the Key Research and Development Program of Hunan Province under Grant 2022GK2020, in part by the Hunan Natural Science Foundation of China under Grant 2022JJ30171, in part by the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy [Shenzhen (SZ)] under Grant GML-KF-22-22 and Grant GML-KF-22-23, in part by the Shenzhen Science and Technology Program under Grant JCYJ20220530160408019, in part by the Chinese Association for Artificial Intelligence (CAAI)-Huawei MindSpore Open Fund, and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515011915. The Associate Editor for this article was Q. Kang. *(Corresponding authors: Zhu Xiao; Hongbo Jiang.)*

Wangchen Long is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China, and also with the College of Artificial Intelligence, Zhuhai City Polytechnic, Zhuhai 519090, China (e-mail: mcdragon@hnu.edu.cn).

Zhu Xiao, Hongbo Jiang, and Zheng Qin are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China, and also with the Shenzhen Research Institute, Hunan University, Shenzhen 518055, China (e-mail: zhxiao@hnu.edu.cn; hongbojiang2004@gmail.com; zqin@hnu.edu.cn).

Yong Xiong is with the College of Computer Science and Electronics Engineering, Hunan University, Changsha 410082, China, and also with Hunan Lianzhi Technology Company Ltd., Changsha 410200, China (e-mail: xiongyong@hnu.edu.cn).

You Li is with the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518060, China (e-mail: liyougis@gmail.com).

Schahram Dustdar is with the Distributed Systems Group, TU Wien, 1040 Vienna, Austria (e-mail: dustdar@infosys.tuwien.ac.at).

Digital Object Identifier 10.1109/TITS.2024.3350234

I. Introduction

MINING knowledge from human mobility trajectory facilitate Intelligent Transportation System (ITS) and urban planning applications [1]. For instance, people's travel patterns and preferences can be retrieved from human mobility trajectory, which provide essential understanding of the interrelation between human behavior and the physical environment. It hence facilitates various applications in urban planning [2], [3], [4], traffic management [5], [6], [7], [8], pollution diagnosis [9], [10], travel recommendation [11] and public safety management [12].

The high-value human mobility trajectory relies on the completeness of the data, which is unfortunately not often the case in many ITS applications [13]. In practice, missing data is a common issue in human mobility trajectory. Using trajectory with missing data for analysis and modeling will lead to incorrect results and unreasonable inference [14], thereby degrading the performance of downstream ITS applications. In this line, how to obtain high quality human mobility trajectory without missing data becomes a vital problem for ITS applications. Intuitively, the problem can be solved in the process of the trajectory collection. However, it is with high cost or even impossible to collect high quality human mobility trajectory due to the constraints of device and environment, resulting in missing or even sparse trajectory data [13]. Alternatively, researchers turn to develop trajectory recovery techniques to restore missing data and rebuild high-quality human mobility trajectory.

A prevalent solution for trajectory recovery is to treat human mobility trajectory as time series location data and leverage spatial-temporal correlations among locations to interpolate missing data. Earlier studies lie in the assumption that missing locations can be approximated by a simple function (e.g., Gaussian function [15], linear function [16] or nearest-neighbor function [17]) based on the distances as well as time spans between each missing point and its contextual points. The shortcoming of those methods is that their performance degrades significantly in sparse scenario, as they cannot capture the high uncertainty between two consecutive with long

time span. An optional thought is to explore individual mobility pattern from historical trajectories for trajectory recovery with recent advanced models including topic models [18], [19], gradient boosting decision tree [20], ensemble transfer learning [21] and recurrent neural networks [22]. Nevertheless, these methods mainly focus on adjacent transitions and local regional correlations, which may lead to unsatisfactory performances in highly spare scenarios due to the absence of long-term dependencies and global correlations among locations. Recently, model-based location prediction methods are developed [23], [24], [25], [26], [27], which leverage attentive deep neural networks to capture complex mobility regularity and long short-term spatial-temporal dependencies among different locations with sparse trajectories.

Despite the inspiring results, existing methods fail to properly handle the recovery of occasionally-visited points (OVPs) in the mobility trajectory. The OVPs are locations that visited infrequently by user. Note that OVPs may occupy a large proportion in trajectory, thereby inevitably leading to degraded performance for trajectory recovery. Fig. 1 shows the ratio of OVPs that are visited only once in four widely used human mobility datasets: NYC,[1] TKY,[1] PriCar[2] and Gowalla.[3] As shown in Fig. 1, there is a considerable number (20% to 50%) of OVPs in four datasets when trajectory length is less than 200. In particular, the ratio of OVPs is as high as 40% to 50% in datasets PriCar and Gowalla, even when considering very long trajectory sequences (length $= 400$). Specifically, one can observe that, on the one side, many OVPs in trajectory are not in line with the inherent regularity of human mobility; on the other side, these numerous OVPs are not isolated, they share same properties with other locations in trajectory, which forms semantic behavior patterns and exhibits relative strong regularities. For instance, Alice likes to taste different flavors of food at different restaurants (e.g., Chinese restaurant, Japanese restaurant, Italian restaurant, etc.). From a geographical perspective, she visited lots of OVPs, and these OVPs share the same semantic behavior (i.e., have meals). Motivated by this observarion, state-of-the-art approaches [24], [25], [26], [27] attempt to recover the missing locations via learning spatial-temporal transition regularities. On the downside, OVPs can hardly be covered by the learned spatial-temporal transition regularities due to their low visit frequencies and the absence of semantic information such as location attribute. As such. it remains a a challenging issue to effectively recover the OVPs in human mobility trajectory.

To address the key issue of restoring OVPs, we develop a novel Semantic-aware Trajectory Recovery method, namely STR, to learn semantic behavior for trajectory recovery. The main idea of STR is to utilize heterogeneous information network [28] to obtain semantically rich embeddings. These embeddings contain both geographic and semantic information, so that can accurately describe the similarity between location points in trajectory. Following that, we leverage attention mechanism to learn semantic behavior transition regularities, which use semantic information for similarity retrieval and prediction. In other words, as a crucial insight for recovering VOPs, our proposed STR method selects location points as candidate prediction based on semantics, even if their geographical locations are completely different. The main contributions of this work are summarized as follows.

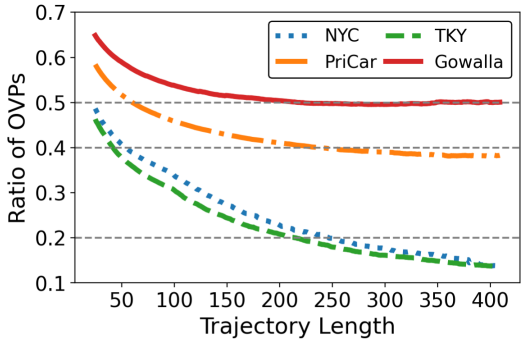


Fig. 1. Analysis on travel behaviors: the ratio of OVPs with respect to trajectory length in four datasets: NYC, TKY, PriCar and Gowalla. High ratio of OVPs implies there are many infrequently visited locations in trajectory, which may be caused by user's similar semantic behaviors at different places.

- We leverage heterogeneous information network to represent rich semantic information by invoking multiple users' trajectory that share similar geographical and semantic locations attributes. In doing so, it can encode various semantic correlations (e.g., user-location and location-location) in heterogeneous information network to obtain semantically rich embeddings, which are essential for identifying user behaviors and conducive to recovery missing OVPs.
- We develop a semantic behavior attention mechanism with three major characteristics to ensure a satisfactory performance of capturing semantic behavior transition regularities. First, we consider not only geographical and semantic location attributes, but also the spatial and temporal intervals between locations to construct the rich semantic context for representing semantic behaviors. Second, only relevant time slots are chosen for behavior attention to avoid huge computing and storage overhead. Third, bidirectional information before and after missing data are utilized to avoid bias when capturing semantic behavior transition regularities.
- We conduct extensive experiments to evaluate the proposed method on four widely-used human mobility trajectory datasets. The results demonstrate the proposed method outperforms state-of-the-arts by 7%-11% in term of *recall*, *F1-score* and *mean average precision*. In addition, we perform statistical analysis and the results further verify the superiority of the proposed method.

The remaining parts of this paper are organized as follows. The literature on related work is reviewed in Section II. Section III presents the preliminaries. After that, the motivations and details of STR are discussed in Section IV. Extensive experiments on real-world datasets are given in Section V. Finally, we conclude this paper in Section VI.

[1]https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[2]https://github.com/HunanUniversityZhuXiao/PrivateCarTrajectoryData
[3]https://snap.stanford.edu/data/loc-gowalla.html

II. Related Work

In this section, we first introduce the related methods of trajectory recovery, and then discuss the recent related works on heterogeneous network representation learning.

A. Trajectory Recovery

Trajectory recovery has been drawn attentions for years and a number of techniques have been proposed in literature. A straightforward solution that adopted by early trajectory recovery methods is to treat human mobility trajectory as spatial-temporal series and leverage spatial-temporal correlations among locations to interpolate missing locations. Early studies assume that missing locations can be approximated by Gaussian function [15], linear function [16], or nearest-neighbor function [17]. However, the mobility of individuals is dynamic, especially in sparse scenarios, where and how far do individuals travel is not available, which makes these interpolation-based methods ineffective.

Song et al. [29] revealed that human mobility exhibits certain transition patterns, in this regard, pattern-based trajectory recovery methods are studied to explore individual behavioral regularities and periodicity. The first line is to leverage machine learning approaches to fill the missing values. Fan et al. [18] leveraged topic model to infer individual movement patterns via collaborative filtering and restored the missing locations with hidden Markov model using the topic distributions. Li et al. [20] incorporated similarity among individuals to investigate the individual mobility patterns, and used four classic machine learning models to reconstruct individual trajectories. Xiao et al. [21] leveraged ensemble transfer regression framework to learn spatial-temporal correlations from historical trajectory for interpolation. The second line focuses on utilizing deep learning approaches (e.g., recurrent neural networks) for trajectory recovery due to their strong power of learning complex transition patterns. Wang et al. [13] integrated a LSTM-based subseq2seq model with Kalman Filter to capture complex transition patterns between locations for trajectory recovery. Xi et al. [30] leveraged LSTM to model the transition patterns as well as personal preference for trajectory recovery. Similar studies can also be found in [22], [23], [31]. However, these methods mainly focus on adjacent transitions and local regional correlations, which may lead to unsatisfactory performances in highly spare scenarios due to the absence of long-term correlations and global correlations among location points in trajectory.

More recent developments turn to model-based methods for trajectory recovery. They leverage attentive deep neural networks to capture complex mobility regularity and long short-term spatial-temporal dependency among different locations in spare scenarios. Ren et al. [32] utilized GRU-based multi-task seq2seq model to recover missing points and map matched them onto the road network simultaneously, where the GRU-based multi-task seq2seq model learned the mobility regularity and the attention mechanism captured global correlations among locations. Shi et al. [25] developed a GAN-based framework to generate missing POI check-ins based on the distribution of locations. Xia et al. [26] tried to solve the problem of trajectory recovery by proposing an attentional neural network. First, an intra-trajectory attention mechanism was proposed to initially fill in the blanks of the current trajectory. Second, inter-trajectory attention mechanism was designed to learn spatial-temporal constraints from observed locations to better rebuild the trajectory. Sun et al. [27] leveraged graph neural network to capture location transition patterns and generate location embedding correspondingly, then attention mechanism was employed to select locations from historical trajectory for the recovery. However, one limitation of recent model-based methods is that the context has been typically modeled based on single individual trajectory without semantic information. If the missing locations are visited infrequently and little correlation is contained in historical trajectory, then they are difficult to be restored with existing methods.

B. Heterogeneous Network Representation Learning

Heterogeneous information network (a.k.a. heterogeneous graph) has achieved success in recommender systems [28] and network security [33] due to its strong power of representing different kinds of nodes and different types of relations. Heterogeneous network representation learning (a.k.a. heterogeneous graph embedding), as a critical research issue in heterogeneous information network, aims to learn representations in a lower-dimension space while preserving the heterogeneous structures and semantics for downstream tasks [34].

Recently, a number of approaches have been proposed to utilize heterogeneous graph embedding to better model user behaviors in recommendation [35], user profiling [36], urban flow pattern mining [37], mobility prediction [38] and so on. These approaches cannot be applied to trajectory recovery directly because they either ignore the spatial-temporal correlations or lack the use of bidirectional information. For instance, approaches for recommendation or link prediction are more concerned about whether there is a connection between users and locations in the future, while the spatial-temporal transition patterns and regularities cannot be captured effectively.

III. Preliminaries

In this section, we first introduce the basic definition and notations used in this work, and then elaborate on the problem statement. TABLE I presents the main notations in the work.

Definition 1 (Trajectory): A trajectory $\mathcal{T}_u$ of user u can be expressed as $\mathcal{T}_u = \{l_u^1, l_u^2, \ldots, l_u^n\}$, where n is the number of time slots, l_u^i denotes the visited location of i-th time slot for a given time interval (e.g., every 30 minutes). All the visited locations form a time-ordered location sequence. Keep in mind that l_u^i is marked by *null* if the location of time slot i is unobserved.

Problem Statement: In the context of trajectory recovery, we have a set of M users $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$, a set of N locations $\mathcal{L} = \{l_1, l_2, \ldots, l_N\}$, a universe set of K location attributes $\mathcal{A} = \{a_1, a_2, \ldots, a_K\}$ and a set of user trajectory $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}$. Given the target user u's trajectory $\mathcal{T}_u = \{l_u^1, l_u^2, \ldots, l_u^n\} \in \mathcal{T}$, recover the missing locations, i.e., $\forall null$ in $\mathcal{T}_u$ to rebuild the complete trajectory.

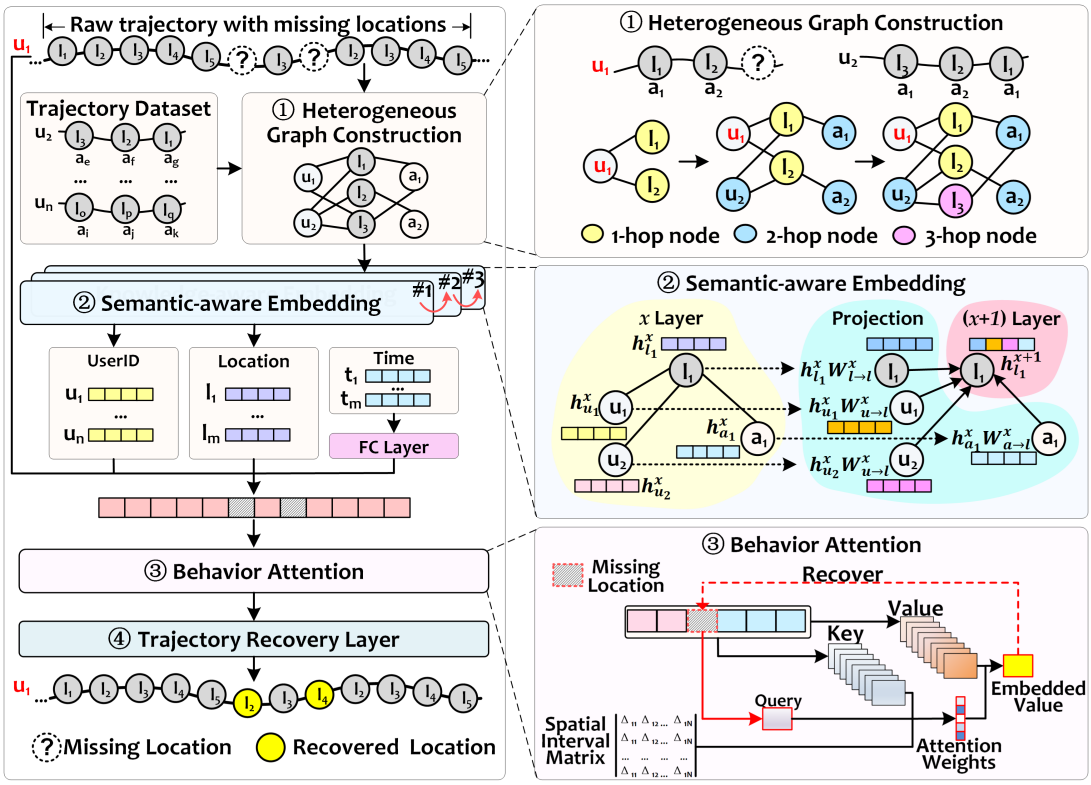


Fig. 2. Overview of the proposed STR.

TABLE I
NOTATIONS

Notation	Meaning
$u, \mathcal{U}$	A user and the set of users
$l, \mathcal{L}$	A location and the set of all locations
$a, \mathcal{A}$	A location attribute and the set of location attributes
M	The total number of users in user set
N	The total number of locations set
K	The total number of location attributes set
$\mathcal{T}_u$	The trajectory of user u
$\mathcal{G}, V, E$	A heterogeneous graph, the set of nodes and edges
$\mathcal{A}, \mathcal{R}$	The set of node types and relation types between nodes
ϕ, ψ	The mapping function from (V to $\mathcal{A}$) and (E to $\mathcal{R}$)
e_u, e_l, e_t	The embedded value of user, location and time
E_u	The spatial-temporal embedded value of user u
d	The latent vector dimension
Δ	The spatial interval matrix
θ	General notation for model parameters

IV. Methodology

In this section, we propose a semantic-aware trajectory recovery method (STR) to recover the missing data in human mobility trajectory. First, we present the overview of STR. Then, we discuss the details of each component in STR.

A. Overview

Fig. 2 presents an overview of STR, which consists of four main components, i.e., *i*) heterogeneous graphs construction; *ii*) semantic-aware embedding; *ii*) behavior attention; *iv*) trajectory recovery layer. First, the heterogeneous graphs are constructed based on the target user's raw trajectory and other relevant users' raw trajectories to express different semantic correlations. Second, a semantic-aware embedding module is designed to encode the semantic correlations contained in the heterogeneous graph. Afterwards, the output of semantic-aware embedding is fused with temporal embeddings, which are encoded by a full-connected layer, to form the spatial-temporal context. Third, a behavior attention layer is developed to capture semantic behavior transitions based on the spatial-temporal context. Finally, a trajectory recovery layer is devised to generate the missing locations for recovery according to knowledge learned by the behavior attention.

B. Heterogeneous Graph Construction

A heterogeneous graph can be defined as $\mathcal{G} = (V, E, \phi, \psi)$, where V is the set of nodes, E represents the set of edges, ϕ is a node type mapping function and ψ is a link type mapping function. Each node $v \in V$ has a node type $\phi(v) \in \mathcal{A}$ and each edge $e \in E$ has an edge type $\psi(e) \in \mathcal{R}$, where $\mathcal{A}$ and $\mathcal{R}$ denote the sets of node types and edge types, $|\mathcal{A}| + |\mathcal{R}| > 2$. There are two edge types in $\mathcal{G}$. The first type is the user-location relation, which describes the fact that a user visits a certain location at a certain time. The second type is the location-attribute relation, indicating that a locations belongs to a PoI category. Note that there are no correlations between same type nodes, and no relationships between users and attributes.

To obtain sufficient representations, we construct heterogeneous graph to gather different interactions among user, location and attributes. The purpose of heterogeneous graph construction is two-fold. First, more collaborative information

between location and attribute can be retrieved. With the attribute information, the user's semantic behavior patterns can be captured as different locations, which share same attribute and can be considered as similar activity locations. Furthermore, locations with same attribute are included in the heterogeneous graph even if the locations are not covered by the target user's trajectory. Doing so is an essential step for restoring missing OVPs. Second, more collaborative information between location and user can be captured, which is important for trajectory recovery as users with similar tastes may have similar mobility patterns. This high-order semantics contains all potential locations for the target user, thereby providing effective information support for trajectory recovery.

Algorithm 1 Heterogeneous Graph Construction

Input: the hop number h, the set of users $\mathcal{U}$, the set of attributes $\mathcal{A}$, the user u and her trajectories $\mathcal{T}_u$

Output: the h-hop extracted heterogeneous graph $\mathcal{G}_u$

1: $U = U_f = \{u\}$, $L = L_f = \emptyset$, $A = A_f = \emptyset$
2: **for** $i = 1, 2, \ldots, h$ **do**
3: $U'_f = \{u_i : u_i \sim L_f\} \backslash U$
4: $L'_f = \{l_i : l_i \sim (U_f \cup \mathcal{A})\} \backslash L$
5: $A'_f = \{a_i : a_i \sim L_f\} \backslash A$
6: $U_f = U'_f$, $L_f = L'_f$, $A_f = A'_f$
7: $U = U \cup U_f$, $L = L \cup L_f$, $A = A \cup A_f$
8: **end for**
9: Let $\mathcal{G}_u$ be the vertex-induced heterogeneous graph whose nodes $\mathcal{E}$ is consisted of (U, L, A)
10: **return** $\mathcal{G}_u$

Note: In line 3 to 5, the symbol $\sim$ means relation. For example, $\{u_i : u_i \sim L_f\}$ is the set of user nodes which have relations with L_f.

Algorithm 1 presents the process of heterogeneous graph construction, which is the extension of the enclosing subgraph extraction methods in [39]. The main idea is to build a heterogeneous graph by adding nodes (i.e., user, location or attribute) related to user u in the trajectory sequence $\mathcal{T}_u$ within h-hops range. Besides, Algorithm 1 describes the details how we extract h-hop heterogeneous graphs, the entire construction process is divided into h rounds. The heterogeneous graph is initialized with only user u. A toy example of the construction is given in upper right of Fig. 2.

C. Semantic-Aware Embedding

To achieve expressive representations of each node in heterogeneous graph, we propose a semantic-aware embedding approach to encode different semantic correlations between nodes and edges. The key idea is to treat all nodes in heterogeneous graph as homogeneous nodes and assign different weights to each node's neighbor nodes. After that, we aggregate all neighbors hidden stats recursively to form the final node embedding by message passing. Follow the common strategies that are used in heterogeneous graph attention networks [40], [41], [42], we stack S layers to obtain the node representations of the whole graph, where each layer contains two operations: *correlation attention* and *neighborhood aggregation*.

1) Correlation Attention: Note that we cannot use existing homogeneous graph embedding methods to capture different semantic correlations in heterogeneous graph, since different types of nodes contained in heterogeneous graph are naturally embedded in distinct spaces. To tackle this issue, a transformation operation is employed to project all nodes from different node spaces to the same low-dimensional target node space. Before the projection, we utilize the straightforward one-hot encoding method to obtain the initial hidden state. In doing so, for each node $v_i \in V$ of heterogeneous graph $\mathcal{G} = (V, E, \phi, \psi)$, we have the initial hidden state $h^0_{v_i}$. Then, a transformation operation is employed to project all nodes from different node spaces to the same low-dimensional target node space. Let $h^x_{v_j}$ denote the embedding hidden state of node v_j in x-th layer. Then, the projection of $h^x_{v_j}$ to target space v_i in x-th layer can be expressed as follows.

$$p^x_{v_j(v_j \to v_i)} = h^x_{v_j} W^x_{v_j \to v_i} \tag{1}$$

where $v_i \in V$, $v_j \in V$ and $W^x_{v_j \to v_i} \in \mathbb{R}^{1 \times d}$ is matrices of learning parameter.

2) Neighborhood Aggregation: After the projection, it is applicable to aggregate all the relevant neighbor nodes for preserving the semantic correlations between nodes. Let $\mathcal{N}_i$ denote the set of direct neighbors of node v_i via edge $e \in E$ in graph $\mathcal{G}$. With the embedding value $h^x_{v_i}$ of node v_i and the projections of its neighbors in x-th layer, we obtain the embedding value $h^{x+1}_{v_i}$ of node v_i in $(x+1)$-th layer.

$$h^{x+1}_{v_i} = W^x_s h^x_{v_i} + \sum_{v_j \in \mathcal{N}_i} \alpha^x_{i,j} W^x_n p^x_{v_j(v_j \to v_i)} \tag{2}$$

where $W^x_s \in \mathbb{R}^{1 \times d}$ and $W^x_n \in \mathbb{R}^{1 \times d}$ are two learnable parameters. $\alpha^x_{i,j}$ is the attention weight between node v_i and its neighbor node v_j, which can be obtained as follows.

$$\alpha^x_{i,j} = softmax(\sigma(\mathbf{a}^\intercal [W^x_s h^x_{v_i} || W^x_n p^x_{v_j(v_j \to v_i)}])) \tag{3}$$

σ is an activation function implemented by LeakyReLU function [43], $||$ denotes the concatenation operator and $\mathbf{a} \in \mathbb{R}^{1 \times d}$ is the trainable attention parameter. The key process of semantic-aware embedding is illustrated in the middle right of Fig. 2. The weight matrices W^x_s and W^x_n are designed to distinguish the updating nodes from different neighbors. Furthermore, they are layer-dependent so as to generate different attention weight $\alpha^x_{i,j}$ for learning contributions of different hops away from the updating nodes.

Finally, all the embedding vectors of nodes in the heterogeneous graph can be obtained. Intuitively, the trajectory recovery problem can be transformed into a heterogeneous graph completion problem, namely, predict whether there exists edges between user and location. However, the heterogeneous graph completion problem are less dependent on spatial-temporal information than trajectory recovery problem, as the former focuses on whether the target user visited the location, while the latter is more concerned on when and where the user had visit. As such, it is not suitable to treat trajectory recovery problem as graph completion problem.

Accordingly, the key to solve trajectory recovery problem is to exploit spatial-temporal semantic transition regularities of human mobility. To that end, we try to extract the spatial-temporal context. Recall that we have already obtained the node embeddings, next, we will discuss how to obtain the expressive temporal embedding.

Previous studies reveal that human mobility shows strong regularity with various granularity such as day and week [5], [19], [44]. To reflect this, we fuse two different time scales (i.e., time of day and day of week) into the temporal features. Both time scales are embedded with one-hot encoding method. Specifically, day of week is mapped to seven slots (from Monday to Sunday) and time of day is mapped to 24 slots. Then, those two parts are concatenated into the representation of time $e_t \in \mathbb{R}^{(7+24)}$. To facilitate the later concatenation, we apply a two-layer fully-connected neural network to produce d-dimension embedded temporal data, which have the same dimension with user and location embedding. Then, we can obtain the i-th time slot spatial-temporal embedding E_u^i by concatenating the embedding of user, time and location.

$$E_u^i = e_u + e_{t_i} + e_{l_i} \tag{4}$$

where e_u is the embedding of user, e_{l_i} denotes the embedding of location and e_{t_i} represents the embedding of time. Note that, we set up a trainable embedding vector $e_{null} \in \mathbb{R}^d$ to represent missing locations. Then, the trajectory $\mathcal{T}_u$ of target user u can be converted to embedded vectors.

D. Behavior Attention

As mentioned above, The key to solve trajectory recovery problem is to exploit spatial-temporal semantic transition regularities of human mobility. Motivated by the inherent regularity of human mobility, we strive to capture the long-term and short-term correlations based on spatial-temporal similarity. To achieve this goal, we propose the behavior attention based on self-attention. The core idea is to infer the missing location based on other locations that share similar spatial-temporal context. Naturally, the spatial-temporal embedding defined in (4) is an essential part of spatial-temporal context. In practice, the spatial and temporal intervals between two visits are important parts of spatial-temporal context. We only consider the spatial intervals, since the temporal interval is a fixed time slot as defined in 1. Specifically, the precise spatial interval between l_i and l_j can be calculated as $Haversine(l_i, l_j)$. To avoid sparsity, we use scaled spatial distance instead of using precise spatial distance to represent spatial interval. Note that the precise spatial distance difference between locations may be very large and become very sparse after discretization. Given the visited location sequence $\mathcal{T}_u = \{l_1, l_2, \ldots l_N\}$ of user u, the minimal spatial interval (other than 0) is Δ^{min}, the maximum spatial interval between two locations is clipped to k. Then, the scaled spatial interval Δ_{ij} can be calculated as follows.

$$\Delta_{ij} = \min(k, \lfloor \frac{Haversine(l_i, l_j)}{\Delta^{min}} \rfloor) \tag{5}$$

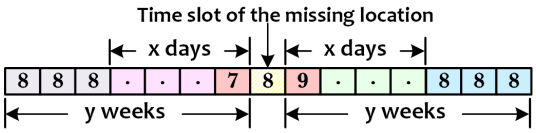


Fig. 3. Attention array.

The trajectory spatial interval matrix $\Delta \in \mathbb{R}^{N \times N}$ can be represented as follow.

$$\Delta = \begin{bmatrix} \Delta_{11} & \Delta_{12} & \ldots & \Delta_{1N} \\ \Delta_{21} & \Delta_{22} & \ldots & \Delta_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_{N1} & \Delta_{N2} & \ldots & \Delta_{NN} \end{bmatrix} \tag{6}$$

Note that, there are missing locations denoted by *null* in $\mathcal{T}_u$ and the spatial intervals between missing locations and existing locations are set to k. With the spatial-temporal embeddings and the spatial interval matrix, we can restore the missing location with behavior attention.

Human behavior is influenced by both short-term and long-term spatial-temporal correlations among different locations. To capture such long short-term spatial-temporal correlations between locations, we perform behavior attention operation on the user's whole trajectory. However, if we apply behavior attention on the whole trajectory directly, the computation and memory consumption will not be affordable as they increase exponentially with the increasing number of locations. To cope with the issue, we design an *attention array* to choose the trajectory of most relevant time slots. Inspired by the multi-level regularity (e.g., day and week) of human mobility, we choose the trajectory within p days and share same time slots within q weeks before and after missing data. Fig. 3 shows an example of attention array, each block represents a time slot and the number on the block denotes the time (e.g., 8 denotes the time is 8:00). In Fig. 3, the time slot of missing location is 8:00, all the time slots before and after the missing location within p days are selected. In addition, all the 8:00 time slots before and after the missing location within q weeks are selected. Hence, the attention array needs to be predefined in advance and may be different according to the corresponding trajectory recovery task.

Afterwards, we define $\mathcal{L}_u$ to denote all the locations whose corresponding time slots are contained in the attention array. We incorporate the spatial interval to the spatial-temporal context and extend the scaled dot-product approach [45] to compute the relevance between time slot i and j. To stabilize the learning process, we adopt multi-head attention [45] and the attention weights under head h can be expressed as follows.

$$\alpha_{i,j}^{(h)} = \frac{\exp(f^{(h)}(E_u^i, E_u^j))}{\sum_{k \in \mathcal{L}_u} \exp(f^{(h)}(E_u^i, E_u^k))} \tag{7}$$

$$f^{(h)}(E_u^i, E_u^j) = \frac{\langle W_Q^{(h)} E_u^i, W_K^{(h)} E_u^j \rangle + \Delta_{ij}}{\sqrt{d_s}} \tag{8}$$

where h is the attention header number, $\langle , \rangle$ denotes the inner product function, $W_Q^{(h)}, W_K^{(h)} \in \mathbb{R}^{d \times d}$ are the transformation matrices and $\alpha_{i,j}^{(h)}$ is the attention weights. The scale factor $\sqrt{d_s}$

is used to avoid large values of the inner product, especially when the dimension is high.

Then, we obtain $\tilde{E}_u^{i(h)}$ by aggregating the information from other locations with corresponding attention weights.

$$\tilde{E}_u^{i(h)} = \sum_{j=1}^{n} \alpha_{i,j}^{(h)} (W_V^{(h)} E_u^j) \tag{9}$$

where n is the total time slot in the trajectory $\mathcal{L}_u$, $W_V^{(h)}$ is the transformation matrix under head h.

After that, we concatenate all the output of different heads to obtain the final embedding value of the time slot i.

$$\tilde{E}_u^i = \tilde{E}_u^{i(1)} || \tilde{E}_u^{i(2)} || \dots || \tilde{E}_u^{i(H)} \tag{10}$$

where $||$ is the concatenation operator, H is the number of attention headers.

E. Trajectory Recovery

Given the final representation denoted by $\tilde{E}_u^i$, we calculate the probability that user u visits location l at time slot i as follows.

$$P_u^i(l) = \frac{\left\langle \tilde{E}_u^i, e_l \right\rangle}{\sum_{k \in \mathcal{L}_u} \left\langle \tilde{E}_u^i, e_k \right\rangle} \tag{11}$$

where $\langle , \rangle$ is the inner product function, $P_u^i(l) \in \mathbb{R}^{|\mathcal{L}_u|}$ denotes the normalized probabilities of all locations visited at time slot i. The location with the maximum probability is identified as the missing location.

Algorithm 2 Training Algorithm for STR

Input: The set of users $\mathcal{U}$, the set of locations $\mathcal{L}$, the set of attributes $\mathcal{A}$, the set of trajectory $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_M\}$, the target user u and her trajectories $\mathcal{T}_u$

Output: Trained Model θ

```
Construct training instances D
Initialize the model parameters θ
for i = 1,2,...,Epoch do
    Shuffle the training instance D into mini-batches
    for j = 1,2,...,Batch number do
        Construct heterogeneous graph according to Algorithm 1 in section IV-B
        Update θ by minimizing the objective loss L
    end for
    Stop training when criteria is met
end for
return Output trained Model θ
```

F. Model Training

The overall loss function consist of two parts. One is the loss L_e for semantic-aware embedding, and the other loss L_r is for the final trajectory recovery. In semantic-aware embedding, it is optimal to restrict those embeddings of users and locations share similar neighbors to be similar with each other. For the final trajectory recovery, we choose the cross-entropy loss function as it is a inherently good and straightforward choice. Therefore, the overall loss function is defined as follows:

$$\begin{aligned} L &= L_r + \lambda L_e + \eta \|\theta\|^2 \\ &= -\sum_{u \in \mathcal{U}} \sum_{t \in \mathcal{T}^m} y_u^t \log(P_u^t) \\ &\quad + \lambda \Big(\sum_{r \in \mathcal{R}} \sum_{x=1}^{S} \|W_n^{x+1} - W_n^x\|^2\Big) + \eta \|\theta\|^2 \end{aligned} \tag{12}$$

where y_u^t is the one-hot representation of the location at t-th time slot in user u's trajectory, $P_u^t \in \mathbb{R}^{|\mathcal{L}_u|}$ denotes the probability of all locations visited at t-th time slot, $\mathcal{T}^m$ denotes the set of missing time slots, S is the number of semantic-aware embedding layers, W_n^x is the weight matrix to distinguish neighbors with different types in (2), θ contains all learnable parameters in the neural network, λ is the weighting hyperparameter, η is a parameter to control the power of regularization. Algorithm 2 outlines the training process of STR and the process is done through stochastic gradient descent across Adam optimizer [46].

According to Algorithm 1 and Algorithm 2, we analyze that the time complexity of STR is $\mathcal{O}(N_{Epoch} N_{Batch} h_{hop} |\mathcal{U}|)$, where N_{Epoch} and N_{Batch} are the number of iterations and batch numbers, h_{hop} is the longest number of hops between nodes in heterogeneous graphs. The larger its value, the richer the node semantics that the heterogeneous graph can express, but the overall computational complexity also increases accordingly. $|\mathcal{U}|$ is the number of users in dataset. Note here N_{Epoch}, N_{Batch}, and h_{hop} are fixed and have moderate values in real scenarios. Therefore, the time complexity of STR grows linearly with $|\mathcal{U}|$, indicating that the time complexity grows linearly with the problem size.

TABLE II
SUMMARY OF DATASETS

Dataset	Time span	#Users	#Locations	#Records
NYC	Apr.2012 - Feb.2013	1,083	38,333	227,428
TKY	Apr.2012 - Feb.2013	2,293	61,858	573,703
PriCar	Feb.2018 - Feb.2019	15,504	143,287	16,976,880
Gowalla	Feb.2009 - Oct.2010	107,092	1,208,969	6,442,892

V. Experiments and Discussions

We evaluate the effectiveness of STR on four real-life mobility trajectory datasets (NYC, TKY, PriCar and Gowalla). The spatial-temporal transition characteristics are different in these four datasets, which can verify the generalizations of STR and baselines over various scenarios. The experiments are conducted based on the MindSpore framework platform. The details of datasets are given in Sec. V-A.

A. Datasets

We perform experiments on four location based on four widely used human mobility datasets. Table II presents the summary of datasets.

- **NYC**[1]: A popular location based social network dataset. This dataset is collected from Foursquare and released

in [47], containing check-ins in New York, the time span is about 10 month (from 12 April 2012 to 16 February 2013).

- **TKY**[1]: This dataset is collected from Foursquare [47], it is similar to NYC except that check-ins are collected in Tokyo.
- **PriCar**[2]: This dataset is obtained from [5], [48], which contains the stop-and-wait locations trajectories of 15,504 private cars from February 1, 2018, to February 1, 2019, in Shenzhen, China.
- **Gowalla**[3]: Another widely used location based social network dataset. It is collected from [49], which contains 6,442,890 check-ins from Gowalla over the period from Feb. 2009 to Oct. 2010.

We treat PoI category as the attribute of location. Note that the PoI categories are contained in datasets NYC, TKY and PriCar. We manually collect the PoI categories for Gowalla from website.[4] Following [26], [27], we set time interval as 30 minutes. We remove the incomplete data, filter out the trajectories with less than 34 time slots (i.e., 70% of one day) and the users with less than 5 day's trajectories.

B. Baselines

- TOP: A simple counting-based method, which chooses the most popular visited locations of each user in the training set as recovery.
- LSTM [50]: A classical RNN model, which contains a memory cell and three multiplicative gates to learn long-term dependence.
- GRU [51]: Similar to LSTM, but has fewer parameters than LSTM, as it lacks an output gate.
- Bi-STDDP [24]: A missing PoI check-in recovery method, which incorporates the bi-directional global spatial and local temporal information to capture complex dependence relationships for recovery.
- Bi-G^{2}AN [25]: A Missing POI Check-in recovery method, which fuses generative adversarial network and gated recurrent unit to recovery the trajectory.
- AttnMove [26]: A state-of-the-art trajectory recovery method, which leverages various attention mechanisms to model the regularity and periodical patterns of user's mobility.
- PeriodicMove [27]: A state-of-the-art trajectory recovery method, which introduces graph neural network and attention network to capture location transition patterns and shifting behavior of human mobility periodicity.

To evaluate the effectiveness of trajectory recovery models, we employ three popular user next location prediction methods as downstream applications. Given the historical trajectory with length n of user u, $T_n^u = p_1^u p_2^u \cdots p_n^u$, the objective of user next location prediction is to infer the next location of user u.

- ST-RNN: It incorporates spatial-temporal contextual information in recurrent neural networks [52].
- HST-LSTM: It extends LSTM model for location prediction by fusing spatial and temporal factors into internal gate [53].
- DeepMove: It utilize recurrent neural networks and attention mechanism to capture individual regularity and personal preference for location prediction [44].

C. Environmental Settings and Hyperparameters

Following the existing works such as [24], [25], [26], [27], three widely-used metrics: Recall (Recall@K), F1-score (F1-score@K) and Mean Average Precision (MAP), are employed to measure the performance. Recall@K is 1 if the ground truth location appears in the top-K ranked list; otherwise is 0. The final Recall@K is the average value over all testing instances. F1-score@K is a comprehensive index reflecting both precision and recall. MAP is a global evaluation for ranking tasks, which is used to evaluate the quality of the whole ranked lists including all locations. Empirically, we set K to 1, 5 and 10. We mask time slots as ground truth and randomly select 60% datasets for training, 20% datasets for validation and 20% datasets for testing.

The classical trajectory recovery models (TOP, LSTM and GRU) and STR are implemented with pytorch, the baseline models are obtained from the author's Github pages. All the models are trained on a machine with NVIDIA GeForce RTX 2080 Ti GPU with 11GB memory, 124 GB RAM and Intel (R) Xeon (R) CPU E5-2678. For LSTM and GRU, we only use a single direction data to recover the missing locations. The hyperparameters of Bi-STDDP, Bi-G^{2}AN, AttnMove and PeriodicMove are default values as they are studied for trajectory recovery using similar datasets and their original hyperparameters achieve the best performance. The hyperparameters of three location prediction methods are set by the default values. For STR, we set the number of semantic-aware layer $S = 3$, the day length in attention array $p = 5$, the week length in attention array $q = 2$, the dimension of output vector $d = 128$ and the scale factor in behavior attention $\sqrt{d_s} = 8$, the training epochs $e = 100$, the learning rate $lr = 0.001$ and the batch size $bs = 32$. We repeat our experiments ten times with different random seed, and the result is presented as "mean±standard deviation".

D. Performance Comparison

In this subsection, we evaluate the performance of STR and baselines through inspecting the prediction results. Besides, we investigate the impact of using only forward (STR-F) and backward (STR-B) trajectory information for trajectory recovery. For each model, we repeat the experiments ten times and report the average result as well as the standard deviation. TABLE III shows the overall performance of STR and baselines. Based on the results, we have the following observations.

For the simple method TOP, it fails to achieve acceptable performance because it restores the missing locations only with the most popular visited locations, which inevitably ignores the spatial-temporal transition regularities in human mobility.

[4]https://www.openstreetmap.org/

TABLE III
PERFORMANCE COMPARISON (MEAN±STD) OF DIFFERENT MODELS IN FOUR DATASETS

Dataset	Model	Recall@1	Recall@5	Recall@10	F1-score@1	F1-score@5	F1-score@10	MAP
NYC	TOP	0.1085±0.0000	0.2867±0.0000	0.3761±0.0000	0.1085±0.0000	0.0963±0.0000	0.0682±0.0000	0.1963±0.0000
	LSTM	0.1223±0.0023	0.3139±0.0034	0.3962±0.0041	0.1223±0.0023	0.1043±0.0019	0.0729±0.0018	0.2113±0.0031
	GRU	0.1268±0.0021	0.3175±0.0028	0.4025±0.0038	0.1268±0.0021	0.1055±0.0016	0.0735±0.0018	0.2154±0.0024
	Bi-STDDP	0.1738±0.0079	0.3477±0.0163	0.4208±0.0165	0.1738±0.0079	0.1163±0.0042	0.0764±0.0033	0.2439±0.0052
	Bi-G^{2}AN	0.1952±0.0091	0.3723±0.0159	0.4345±0.0201	0.1952±0.0091	0.1218±0.0085	0.0796±0.0073	0.2686±0.0085
	AttnMove	0.2218±0.0086	0.4231±0.0109	0.4916±0.0182	0.2218±0.0086	0.1356±0.0064	0.0896±0.0057	0.3025±0.0067
	PeriodicMove	0.2431±0.0098	0.4618±0.0133	0.5379±0.0203	0.2431±0.0098	0.1474±0.0083	0.0968±0.0072	0.3241±0.0085
	STR-F	0.2518±0.0042	0.4822±0.0055	0.5681±0.0059	0.2518±0.0042	0.1506±0.0033	0.1011±0.0026	0.3401±0.0031
	STR-B	0.2521±0.0046	0.4819±0.0061	0.5679±0.0065	0.2521±0.0046	0.1505±0.0025	0.1009±0.0018	0.3400±0.0033
	STR	**0.2607±0.0038**	**0.4986±0.0052**	**0.5762±0.0061**	**0.2607±0.0038**	**0.1585±0.0024**	**0.1038±0.0021**	**0.3486±0.0025**
TKY	TOP	0.1369±0.0000	0.3164±0.0000	0.3955±0.0000	0.1369±0.0000	0.1082±0.0000	0.0733±0.0000	0.2261±0.0000
	LSTM	0.1465±0.0027	0.3554±0.0037	0.4461±0.0052	0.1465±0.0027	0.1185±0.0032	0.0801±0.0049	0.2428±0.0035
	GRU	0.1523±0.0023	0.3563±0.0025	0.4493±0.0029	0.1523±0.0023	0.1125±0.0022	0.0812±0.0024	0.2491±0.0023
	Bi-STDDP	0.2013±0.0081	0.4089±0.0171	0.4784±0.0191	0.2013±0.0081	0.1372±0.0072	0.0873±0.0063	0.2851±0.0062
	Bi-G^{2}AN	0.2141±0.0102	0.4361±0.0153	0.4901±0.0181	0.2141±0.0102	0.1452±0.0091	0.0902±0.0081	0.3118±0.0088
	AttnMove	0.2367±0.0092	0.4893±0.0123	0.5472±0.0193	0.2367±0.0092	0.1613±0.0077	0.1003±0.0059	0.3485±0.0075
	PeriodicMove	0.2561±0.0104	0.5329±0.0132	0.5943±0.0197	0.2561±0.0104	0.1747±0.0095	0.1089±0.0082	0.3794±0.0093
	STR-F	0.2641±0.0051	0.5513±0.0063	0.6228±0.0089	0.2641±0.0051	0.1771±0.0041	0.1129±0.0033	0.3937±0.0040
	STR-B	0.2643±0.0059	0.5511±0.0070	0.6226±0.0084	0.2643±0.0059	0.1772±0.0048	0.1127±0.0041	0.3935±0.0044
	STR	**0.2733±0.0052**	**0.5701±0.0086**	**0.6317±0.0095**	**0.2733±0.0052**	**0.1864±0.0043**	**0.1159±0.0035**	**0.4035±0.0037**
PriCar	TOP	0.1007±0.0000	0.2056±0.0000	0.2766±0.0000	0.1007±0.0000	0.0689±0.0000	0.0443±0.0000	0.1426±0.0000
	LSTM	0.1176±0.0033	0.2468±0.0036	0.2971±0.0063	0.1176±0.0033	0.0757±0.0055	0.0448±0.0072	0.1761±0.0053
	GRU	0.1228±0.0028	0.2606±0.0032	0.2970±0.0053	0.1228±0.0028	0.0776±0.0046	0.0449±0.0053	0.1845±0.0046
	Bi-STDDP	0.1576±0.0112	0.2931±0.0159	0.3061±0.0212	0.1576±0.0112	0.0861±0.0093	0.0574±0.0078	0.2093±0.0088
	Bi-G^{2}AN	0.1681±0.0151	0.3117±0.0193	0.3246±0.0258	0.1681±0.0151	0.0916±0.0112	0.0613±0.0097	0.2234±0.0103
	AttnMove	0.1847±0.0121	0.3425±0.0179	0.3546±0.0233	0.1847±0.0121	0.1005±0.0108	0.0671±0.0092	0.2446±0.0095
	PeriodicMove	0.1956±0.0133	0.3622±0.0189	0.3733±0.0249	0.1956±0.0133	0.1063±0.0118	0.0709±0.0103	0.2592±0.0106
	STR-F	0.2068±0.0063	0.3781±0.0081	0.3975±0.0094	0.2068±0.0063	0.1096±0.0052	0.0736±0.0042	0.2696±0.0045
	STR-B	0.2070±0.0071	0.383±0.0079	0.3974±0.0098	0.2070±0.0071	0.1093±0.0062	0.0734±0.0050	0.2696±0.0048
	STR	**0.2133±0.0059**	**0.3961±0.0087**	**0.4096±0.0089**	**0.2133±0.0059**	**0.1156±0.0051**	**0.0779±0.0042**	**0.2853±0.0044**
Gowalla	TOP	0.0573±0.0000	0.1341±0.0000	0.1779±0.0000	0.0573±0.0000	0.0439±0.0000	0.0326±0.0000	0.1123±0.0000
	LSTM	0.0611±0.0035	0.1526±0.0054	0.2013±0.0062	0.0611±0.0035	0.0493±0.0052	0.0373±0.0069	0.1255±0.0058
	GRU	0.0628±0.0029	0.1593±0.0038	0.2117±0.0049	0.0628±0.0029	0.0516±0.0048	0.0394±0.0062	0.1367±0.0047
	Bi-STDDP	0.1021±0.0126	0.2317±0.0215	0.2867±0.0263	0.1021±0.0126	0.0755±0.0111	0.0531±0.0105	0.1634±0.0119
	Bi-G^{2}AN	0.1131±0.0149	0.2505±0.0223	0.3117±0.0251	0.1131±0.0149	0.0815±0.0120	0.0567±0.0103	0.1762±0.0127
	AttnMove	0.1237±0.0137	0.2762±0.0209	0.3379±0.0243	0.1237±0.0137	0.0896±0.0113	0.0619±0.0101	0.1917±0.0105
	PeriodicMove	0.1311±0.0133	0.2935±0.0198	0.3591±0.0251	0.1311±0.0133	0.0948±0.0118	0.0654±0.0106	0.2028±0.0113
	STR-F	0.1403±0.0095	0.3121±0.0147	0.3827±0.0187	0.1403±0.0095	0.0995±0.0077	0.0677±0.0068	0.2151±0.0063
	STR-B	0.1395±0.0093	0.3119±0.0153	0.3816±0.0192	0.1395±0.0093	0.0998±0.0073	0.0676±0.0063	0.2138±0.0070
	STR	**0.1462±0.0088**	**0.3281±0.0134**	**0.3961±0.00157**	**0.1462±0.0088**	**0.1073±0.0059**	**0.0735±0.0044**	**0.2263±0.0046**

LSTM and GRU perform better than TOP as they can model simple spatial-temporal transition regularities. However, according to the experiment results, they show shortcomings in capturing long-term spatial-temporal correlations given the following reasons. First, LSTM and GRU methods restore the trajectory from a single forward direction perspective, which ignores another direction information. Second, they fail to capture long-term regularity in sparse scenarios due to vanishing sequential dependency.

Compared with LSTM and GRU, Bi-STDDP and Bi-G^{2}AN demonstrate superiority thanks to their sophisticated architectures and utilization of bidirectional information around missing locations. However, the weakness of these models in capturing long-term transition regularities prevents them from achieving excellent performance, which is verified by the experimental results. The reason behind is that, for Bi-STDDP, it only uses a simple feed-forward neural network by concatenating spatial-temporal context directly. This, unfortunately, is not an effective way to capture omplex (e.g., day and week regularity) spatial-temporal transition regularities. For Bi-G^{2}AN, it is a RNNs-based method and has the inherit flaw of capturing long-term regularity in sparse scenarios.

AttnMove and PeriodicMove perform better than Bi-STDDP and Bi-G^{2}AN. This stems from the superiority of attention mechanism in capturing both short-term and long-term mobility transition regularities. When comparing with the proposed STR, the experimental results show that these two method have room for improvement. On the one hand, they lack the ability to perceive the semantic information, so as not to effectively recover the missing OVPs. On the other hand, AttnMove and PeriodicMove only use the information in a short period of time (e.g., one day) instead of long-term bidirectional information around the missing data. This leads to semantic bias in capturing mobility transition regularities.

STR outperforms AttnMove and PeriodicMove with 7% and 15% improvement in terms of Recall, F1-score and MAP in NYC and TKY datasets. The improvements increase to 11% and 18% in PriCar and Gowalla. It is explained that there are more missing OVPs in these two datasets and STR can recover more OVPs compared with AttnMove and PeriodicMove. These results verify that the heterogeneous information network can effectively extract global semantic information. Besides, the behavior attention mechanism helps

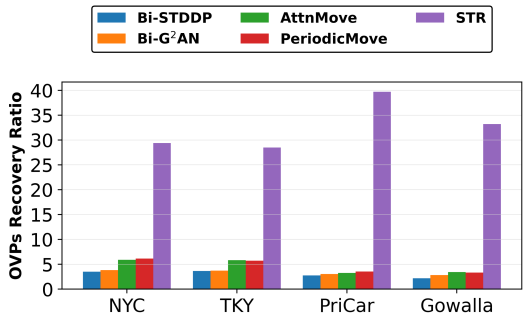


Fig. 4. Comparison of different models' ability to recover OVPs.

to boost the ability of capturing semantic behavior transition regularities.

STR-F and STR-B denote the variants of STR that only use forward and backward trajectory, respectively. As shown in TABLE III, the performance of both STR-F and STR-B is reduced compared with STR, which is in line with intuitive expectations. The reason is that, using only one-direction trajectory cannot fully capture the user's semantic behavior transition regularities, thereby leading to bias in the learned transition regularities.

We conduct experiments on the recovery ratio of OVPs in four datasets for each recovery model. The results are shown in Figure 4, from which we can observe that the STR model outperforms the baselines in terms of the recovery performance of OVPs. The reason is that the heterogeneous graph in the STR model contains the POI semantic information of OVPs and the collaborative information between different users. As such, the proposed method can distinguish OVPs with similar semantics and recover OVPs based on the mined semantic patterns.

Furthermore, we employ three popular location prediction methods (ST-RNN [52], HST-LSTM [53], and DeepMove [44]) as downstream applications to evaluate the effectiveness of trajectory recovery models. We use three metrics to evaluate the performance of location prediction: (1) Hitting ratio (hit), which shows that the ground-truth location appears in the next prediction result; (2) F1-score, which calculates the mean of precision and recall; (3) Mean reciprocal rank (MRR), which is the multiplicative inverse of the rank of the first correct next answer.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}, \tag{13}$$

where $|Q|$ is the total number of testing instances, and $rank_i$ is the predicting rank of instance i.

The next location prediction results are shown in TABLE IV. It is shown that all location prediction models have improved prediction performance by more than 14% in datasets that are recovered by STR compared with the stat-of-the-art recovery model AttnMove and PeriodicMove. It validates the idea that STR can more accurately recover missing trajectory information and provide higher quality trajectory data for downstream applications.

E. Ablation Study

To investigate how does the semantic-aware embedding and behavior attention take effect, we consider two important components in STR: semantic-aware embedding layer and behavior attention layer. To study each component contributing to the performance of STR, we evaluate the following variants:

- STR-SE: STR without semantic-aware embedding layer. In this variant, the semantic-aware embedding layer is replaced with one-hot embedding.
- STR-BA: STR without behavior attention layer. In this variant, the behavior attention layer is replace with LSTM.

To analyze the efficiency of different components in different scenarios, we measure the performances of all variants on four datasets. TABLE V shows the performance of STR and its variants in terms of Recall@K, F1-score@K and MAP. To reflect the data more intuitively, we visualize the result in Fig. 5. According to the results, we can obtain that:

First, both semantic-aware embedding and behavior attention are essential as both STR-SE and STR-BA have evident performance drop on Recall, F1-score and MAP. Without semantic-aware embedding, the performance of STR was reduced by about 18%. Without behavior attention, the performance of STR was reduced by about 32%. The results illustrate that the effectiveness of both components.

Second, The performance of STR-BA is worse than STR-SE in terms of all metrics, which indicates that the behavior attention module makes a greater contribution than semantic-aware embedding.

F. Hyper-Parameter Sensitivity

STR involves a number of parameters and we study how different choices of parameters affect the performance of STR. There are two core parameters in STR: (1) the number of semantic-aware embedding layers S, we tried the number of S over {1, 2, 3, 4, 5}; (2) the output dimensionality d of latent vectors, which affects the ability of representation, we tried the output dimensionality d over {16, 32, 64, 128, 256}; We only report the results on NYC dataset because we observe similar trends on other three dataset. We summarize the results and have the following observations:

Fig. 6 shows that increasing the number of semantic-aware embedding layers is capable of boosting the performance substantially. Because more layers can learn high-order semantic information better. We also observer that the performance can only achieve marginal improvement when the number of layer is larger than 3, it suggests that a three-layer stacked semantic-aware embedding could be sufficient to capture the semantic information in all these four datasets.

Fig. 7 shows the performance of STR with different dimension of output hidden state, the result indicates that the optimal value of dimension of hidden state d is around 128. When d is larger than 128, the model may be tuned to over-fitting and the performance starts to decline. The result is also in line with previous studies [26], [27]. The possible reason is that embedding value with 128 dimensions is enough for representing spatial-temporal context in these four datasets,

TABLE IV
PERFORMANCE COMPARISON (MEAN±STD) OF LOCATION PREDICTION MODELS WITH DIFFERENT RECOVERY MODELS

Prediction Model		**ST-RNN**			**HST-LSTM**			**DeepMove**		
Dataset	**Recovery Model**	Hit(%)	F1-score(%)	MRR(%)	Hit(%)	F1-score(%)	MRR(%)	Hit(%)	F1-score(%)	MRR(%)
NYC	None	16.13±0.08	15.71±0.06	22.13±0.09	16.68±0.09	16.12±0.10	22.46±0.011	19.80±0.14	18.57±0.12	28.18±0.18
	TOP	16.33±0.08	15.78±0.07	22.55±0.09	16.89±0.11	16.13±0.10	23.56±0.12	19.84±0.14	18.68±0.13	28.35±0.17
	LSTM	17.64±0.05	17.31±0.05	24.28±0.07	18.88±0.09	18.46±0.10	25.58±0.13	22.15±0.12	21.63±0.15	31.58±0.18
	GRU	18.14±0.09	17.74±0.08	24.88±0.09	19.68±0.10	19.14±0.11	25.84±0.14	23.14±0.15	22.54±0.13	32.61±0.13
	Bi-STDDP	19.12±0.08	18.67±0.08	25.06±0.10	20.43±0.13	19.35±0.13	26.20±0.14	24.54±0.15	23.93±0.16	33.77±0.18
	Bi-G^{2}AN	18.94±0.09	18.56±0.09	25.16±0.11	20.41±0.15	19.33±0.13	26.35±0.15	24.57±0.17	23.92±0.17	33.60±0.22
	AttnMove	20.06±0.10	19.49±0.10	26.52±0.11	22.71±0.14	21.39±0.14	29.76±0.17	25.39±0.19	24.69±0.15	34.85±0.19
	PeriodicMove	20.73±0.09	19.97±0.10	27.79±0.13	23.93±0.17	22.44±0.16	31.27±0.18	25.83±0.18	25.02±0.19	35.28±0.21
	STR	**23.72±0.11**	**22.78±0.10**	**31.64±0.13**	**27.16±0.16**	**25.83±0.15**	**36.32±0.18**	**29.16±0.19**	**28.05±0.18**	**39.68±0.25**
TKY	None	14.26±0.08	12.28±0.05	18.19±0.09	14.91±0.08	12.25±0.06	19.03±0.10	17.45±0.09	14.69±0.08	22.94±0.13
	TOP	14.42±0.09	12.42±0.05	18.22±0.09	14.96±0.09	12.82±0.06	19.13±0.11	17.51±0.09	14.85±0.08	23.09±0.15
	LSTM	15.58±0.08	13.62±0.06	19.64±0.10	16.67±0.10	14.55±0.07	20.72±0.11	19.56±0.12	17.11±0.10	25.64±0.12
	GRU	15.92±0.08	13.74±0.06	19.87±0.10	17.28±0.11	14.88±0.07	20.65±0.13	20.31±0.10	17.58±0.11	26.14±0.16
	Bi-STDDP	16.79±0.09	14.55±0.06	20.07±0.12	17.85±0.11	14.97±0.09	20.99±0.12	21.55±0.17	18.61±0.15	27.05±0.21
	Bi-G^{2}AN	16.55±0.10	14.4±0.08	20.08±0.11	17.93±0.11	15.00±0.09	21.07±0.11	21.56±0.14	18.57±0.13	26.90±0.18
	AttnMove	17.53±0.10	15.07±0.09	21.11±0.12	19.88±0.13	16.57±0.11	23.77±0.13	22.22±0.15	19.13±0.19	27.83±0.22
	PeriodicMove	18.09±0.11	15.48±0.09	22.17±0.12	20.87±0.13	17.32±0.11	24.89±0.12	22.56±0.12	19.39±0.16	28.15±0.21
	STR	**20.66±0.13**	**17.61±0.10**	**25.20±0.13**	**23.73±0.13**	**19.89±0.12**	**28.96±0.14**	**25.47±0.15**	**21.68±0.16**	**31.66±0.19**
PriCar	None	40.03±0.19	38.47±0.15	41.93±0.19	41.16±0.20	39.52±0.15	44.81±0.0.21	48.39±0.27	45.82±0.22	53.16±0.29
	TOP	40.12±0.18	38.66±0.15	42.89±0.20	41.58±0.19	39.69±0.16	45.02±0.22	48.59±0.26	46.16±0.22	54.33±0.31
	LSTM	43.25±0.23	42.36±0.22	46.22±0.26	46.28±0.21	45.07±0.19	48.77±0.29	54.28±0.33	53.19±0.35	60.33±0.39
	GRU	43.35±0.19	42.41±0.27	46.52±0.24	47.12±0.22	45.94±0.23	48.35±0.31	55.39±0.41	54.28±0.31	61.21±0.32
	Bi-STDDP	45.68±0.25	44.79±0.21	47.01±0.29	48.53±0.29	46.25±0.25	49.12±0.29	58.62±0.35	57.48±0.34	63.33±0.42
	Bi-G^{2}AN	44.93±0.21	44.51±0.26	46.89±0.32	48.61±0.25	46.35±0.22	49.22±0.32	58.54±0.44	57.39±0.37	62.84±0.47
	AttnMove	47.62±0.22	46.59±0.29	49.33±0.28	53.84±0.35	51.22±0.27	55.45±0.31	60.33±0.38	59.11±0.41	65.03±0.39
	PeriodicMove	49.13±0.27	47.77±0.31	51.81±0.33	56.69±0.29	53.47±0.38	58.16±0.37	61.28±0.47	59.84±0.43	65.76±0.46
	STR	**56.16±0.23**	**54.34±0.28**	**58.81±0.31**	**64.49±0.27**	**61.18±0.28**	**67.58±0.31**	**69.21±0.35**	**66.91±0.36**	**73.86±0.40**
Gowalla	None	12.04±0.05	10.77±0.03	13.97±0.05	12.48±0.02	10.85±0.04	13.32±0.05	14.59±0.08	13.16±0.06	16.42±0.10
	TOP	12.26±0.06	10.84±0.04	14.07±0.05	12.59±0.03	11.02±0.03	13.41±0.05	14.74±0.08	13.28±0.06	16.53±0.09
	LSTM	13.12±0.05	11.41±0.02	14.53±0.06	13.95±0.05	11.84±0.02	14.03±0.04	16.36±0.09	14.35±0.08	18.43±0.13
	GRU	12.81±0.08	11.31±0.03	14.49±0.05	13.91±0.04	11.91±0.05	14.79±0.06	16.35±0.06	14.39±0.07	18.46±0.11
	Bi-STDDP	13.48±0.10	11.67±0.04	14.66±0.07	14.36±0.05	11.96±0.05	15.25±0.05	17.31±0.09	14.87±0.06	20.86±0.08
	Bi-G^{2}AN	13.34±0.09	11.68±0.04	14.61±0.05	14.51±0.05	12.05±0.04	15.14±0.05	17.37±0.08	14.91±0.07	20.81±0.09
	AttnMove	14.13±0.08	12.01±0.07	15.01±0.04	16.01±0.05	13.71±0.06	18.05±0.07	17.95±0.10	15.26±0.09	22.16±0.13
	PeriodicMove	14.74±0.11	12.27±0.06	15.44±0.06	16.91±0.06	14.22±0.07	18.47±0.05	18.28±0.13	15.87±0.10	22.32±0.15
	STR	**16.76±0.06**	**13.59±0.04**	**18.57±0.05**	**19.32±0.05**	**16.38±0.05**	**22.16±0.07**	**20.66±0.08**	**18.43±0.07**	**25.63±0.10**

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT VARIATIONS, WHERE δ DENOTED THE PERFORMANCE DECLINE

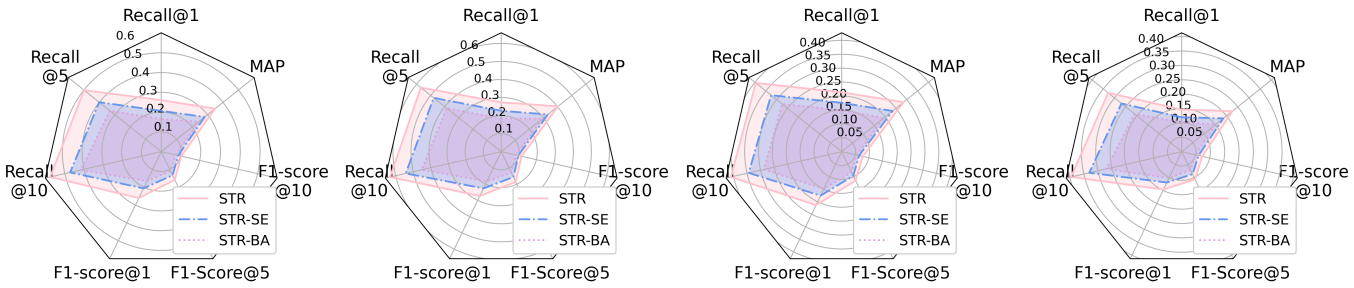
Dataset	**Model**	**Recall@1(δ)**	**Recall@5(δ)**	**Recall@10(δ)**	**F1-score@1(δ)**	**F1-score@5(δ)**	**F1-score@10(δ)**	**MAP(δ)**
NYC	STR-SE	0.2063(-20.9%)	0.4011(-19.6%)	0.4725(-18.0%)	0.2063(-20.9%)	0.1325(-16.4%)	0.0882(-15.0%)	0.2815(-19.2%)
	STR-BA	0.1652(-36.6%)	0.3321(-33.4%)	0.4163(-27.8%)	0.1652(-36.6%)	0.1087(-31.4%)	0.0749(-27.8)	0.2426(-30.4%)
	STR	0.2607	0.4986	0.5762	0.2607	0.1585	0.1038	0.3486
TKY	STR-SE	0.2264(-17.2%)	0.4786(-16.0%)	0.5431(-14.0%)	0.2264(-17.2%)	0.1564(-16.1%)	0.0973(-16.0%)	0.3309(-18.0%)
	STR-BA	0.1835(-32.9%)	0.3865(-32.2%)	0.4511(-28.6%)	0.1835(-32.9%)	0.1293(-30.6%)	0.0844(-27.2%)	0.2931(-27.4%)
	STR	0.2733	0.5701	0.6317	0.2733	0.1864	0.1159	0.4035
PriCar	STR-SE	0.1756(-17.7%)	0.3257(-17.8%)	0.3438(-16.1%)	0.1756(-17.7%)	0.0963(-16.7%)	0.0641(-17.7%)	0.2346(-17.8%)
	STR-BA	0.1488(-30.2%)	0.2747(-30.6%)	0.2877(-29.8%)	0.1488(-30.2%)	0.0816(-29.4%)	0.0553(-29.0%)	0.1938(-32.1%)
	STR	0.2133	0.3961	0.4096	0.2133	0.1156	0.0779	0.2853
Gowalla	STR-SE	0.1193(-18.4%)	0.2689(-18.0%)	0.3308(-16.5%)	0.1193(-18.4%)	0.0871(-18.8%)	0.0603(-18.0%)	0.1855(-18.0%)
	STR-BA	0.1021(-30.2%)	0.2099(-36.0%)	0.2694(-32.0%)	0.1021(-30.2%)	0.0704(-34.4%)	0.0489(-33.5%)	0.1517(-33.0%)
	STR	0.1462	0.3281	0.3961	0.1462	0.1073	0.0735	0.2263

using oversize dimensions may increases the complexity and leads to overfitting during model training.

G. Robustness Analysis

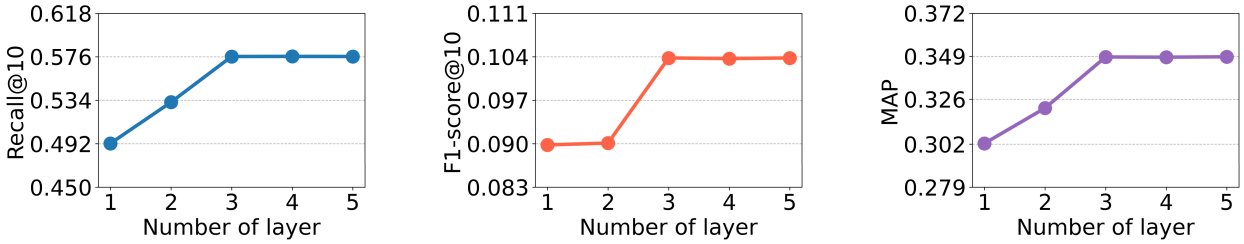
To evaluate the robustness of STR, we conduct an experiment to investigate the performance STR and baselines with respect to dataset with different missing ratios (i.e., the percentage of missing locations). We mask some time slots to simulate the effect of data sparsity and the missing ratio varies from 20% to 80%. For simplicity, we only report the results on PriCar data in Fig. 8 as the results on other datasets are similar. According to the results shown in Fig. 8, we have the following observations:

The performance of STR and other baselines decreases when the missing ratio increases. When the missing ratio increases, the trajectory becomes more sparse with less critical information. Consequently, the human mobility transition regularities become more difficult to be captured, which directly leads to the performance degradation of all algorithms.



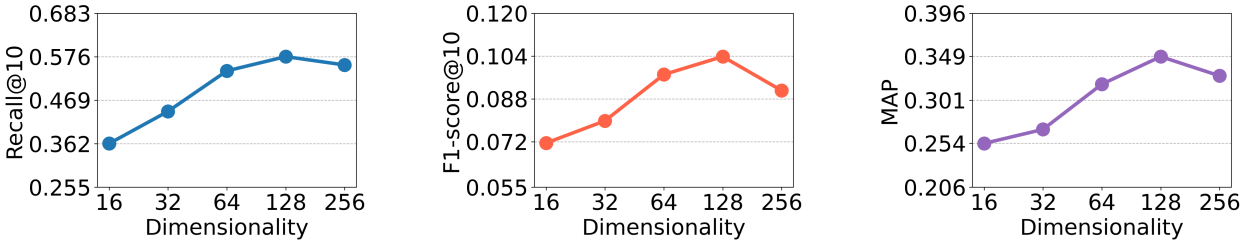
(a) NYC (b) TKY (c) PriCar (d) Gowalla

Fig. 5. Performance comparison of different variations of STR.



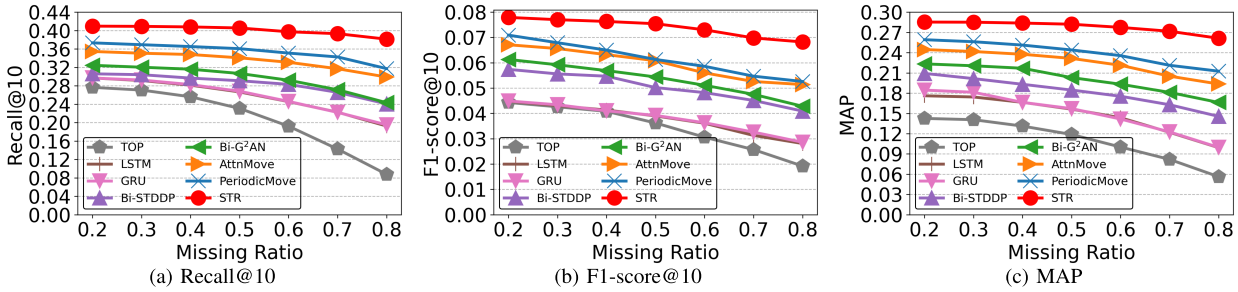
(a) Recall@10 (b) F1-score@10 (c) MAP

Fig. 6. Performance of STR with different number of semantic-aware embedding layer.



(a) Recall@10 (b) F1-score@10 (c) MAP

Fig. 7. Performance of STR with different output dimensionality.



(a) Recall@10 (b) F1-score@10 (c) MAP

Fig. 8. Performance compare w.r.t missing ratios on PriCar dataset.

Compared with baselines, STR shows the best robustness. For one reason, STR can gather more correlations among locations by constructing heterogeneous graph to make up for missing data. This attains strong ability to alleviate performance degeneration in data with high missing ratio. For another reason, the attention mechanism can effectively capture spatial-temporal correlation in sparse data, which can be verified by the results that attention-based models (e.g., AttnMove and PeriodicMove) generally perform better RNN-based models (e.g., LSTM, GRU). In addition, the behavior attention mechanism adopted in STR plays an essential role in keeping the robustness when facing high missing ratio dataset.

H. Statistical Analysis

To further evaluate the performance of the proposed method, we apply the Friedman test [54] to determine whether or not there is a statistically significant difference between STR and other baselines. To accurately and comprehensively reflect the performance of different algorithms, we divide the original

TABLE VI
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON DATASETS WITH RESPECT TO DIFFERENT MISSING RATIO

Dataset(Missing Ratio)	MAP / (Friedman Test Rankings) TOP	LSTM	GRU	Bi-STDDP	Bi-GAN	AttnMove	PeriodicMove	STR
NYC-(20%)	0.1963 (8)	0.2113 (7)	0.2154 (6)	0.2439 (5)	0.2686 (4)	0.3025 (3)	0.3241 (2)	0.3486 (1)
NYC-(40%)	0.1805 (8)	0.1994 (6)	0.1942 (7)	0.2249 (5)	0.2598 (4)	0.2941 (3)	0.3141 (2)	0.3469 (1)
NYC-(60%)	0.1376 (8)	0.1718 (6)	0.1641 (7)	0.2042 (5)	0.2323 (4)	0.2741 (3)	0.2954 (2)	0.3384 (1)
NYC-(80%)	0.0777 (8)	0.1186 (6)	0.1147 (7)	0.1709 (5)	0.1985 (4)	0.2402 (3)	0.2650 (2)	0.3190 (1)
TKY-(20%)	0.2261 (8)	0.2428 (7)	0.2491 (6)	0.2851 (5)	0.3118 (4)	0.3485 (3)	0.3794 (2)	0.4035 (1)
TKY-(40%)	0.2086 (8)	0.2299 (6)	0.2244 (7)	0.2626 (5)	0.3021 (4)	0.3377 (3)	0.3670 (2)	0.4008 (1)
TKY-(60%)	0.1592 (8)	0.1982 (6)	0.1905 (7)	0.2379 (5)	0.2693 (4)	0.3138 (3)	0.3447 (2)	0.3918 (1)
TKY-(80%)	0.0888 (8)	0.1375 (6)	0.1336 (7)	0.1980 (5)	0.2308 (4)	0.2744 (3)	0.3104 (2)	0.3685 (1)
PriCar-(20%)	0.1426 (8)	0.1761 (7)	0.1845 (6)	0.2093 (5)	0.2234 (4)	0.2446 (3)	0.2592 (2)	0.2853 (1)
PriCar-(40%)	0.1312 (8)	0.1664 (6)	0.1663 (7)	0.1932 (5)	0.2169 (4)	0.2376 (3)	0.2512 (2)	0.2838 (1)
PriCar-(60%)	0.1002 (8)	0.1436 (6)	0.1417 (7)	0.1754 (5)	0.1933 (4)	0.2213 (3)	0.2356 (2)	0.2774 (1)
PriCar-(80%)	0.0563 (8)	0.0993 (7)	0.0997 (6)	0.1456 (5)	0.1662 (4)	0.1936 (3)	0.2126 (2)	0.2613 (1)
Gowalla-(20%)	0.1123 (8)	0.1255 (7)	0.1367 (6)	0.1634 (5)	0.1762 (4)	0.1917 (3)	0.2028 (2)	0.2263 (1)
Gowalla-(40%)	0.1033 (8)	0.1192 (7)	0.1239 (6)	0.1503 (5)	0.1707 (4)	0.1853 (3)	0.1972 (2)	0.2253 (1)
Gowalla-(60%)	0.0790 (8)	0.1028 (7)	0.1047 (6)	0.1367 (5)	0.1517 (4)	0.1718 (3)	0.1842 (2)	0.2193 (1)
Gowalla-(80%)	0.0444 (8)	0.0704 (7)	0.0734 (6)	0.1133 (5)	0.1301 (4)	0.1497 (3)	0.1654 (2)	0.2061 (1)
Friedman's Rank	8	6.5	6.5	5	4	3	2	1

4 datasets into 16 datasets with different values of missing ratios. Without loss of generality, we only report the result on MAP as the results on other metrics are similar. Let the null Hypothesis (H_0) of the Friedman test be that the performance of all models are equivalent. First, we calculate the average Friedman test rank of all models with the equation as follows.

$$R_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j \tag{14}$$

where r_i^j denotes the rank of the j-th model on the i-th dataset, N represents the number of datasets.

Then, we obtain the average Friedman's rank of all eight models and the results are shown in Table VI, where R_8(TOP) = 8, R_7(LSTM) = 6.5, R_6(GRU) = 6.5, R_5(Bi-STDDP) = 5, R_4(Bi-G^{2}AN) = 4, R_3(AttnMove) = 3, R_2(PeriodicMove) = 2, R_1(STR) = 1. With the average Friedman's rank, we obtain the Friedman statistics F_F with the following equation.

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1)-\chi_F^2} \tag{15}$$

where K is the number of models, N is the number of datasets, and χ_F^2 is defined as:

$$\chi_F^2 = \frac{12N}{K(K+1)}\left[\sum_j R_j^2 - \frac{K(K+1)^2}{4}\right] \tag{16}$$

which is distributed according to the F-distribution with K-1 and (K-1)(N-1) degrees of freedom.

Finally, we obtain the value of Friedman statistics F_F = 1,245. Since χ_F^2 is distributed according to the F-distribution with K-1 and (K-1)(N-1) degrees of freedom, we have χ_F^2 = F((8-1),(8-1)×(16-1)) = F(7,105) F(7,105). Without loss of generality, let the significance level $\alpha = 0.05$, then, F(7,105) = 2.098, which is less than the value of F_F = 1,245. On this basis, the null Hypothesis H_0 is rejected, which means the performance of all 8 models is different.

To determine the specific differences, we choose the Holm procedure [55] as a *post hoc* test due to its simplicity and efficiency. The test statistics for comparing the i-th and j-th models is defined as follows.

TABLE VII
OUTCOME OF HOLM STATISTICAL TEST WITH SIGNIFICANCE LEVEL α = 0.05

Comparison	*z-value*	*p-value*	Result
STR vs TOP	8.08290	0.00000	H_0 is rejected
STR vs LSTM	6.35085	0.00000	H_0 is rejected
STR vs GRU	6.35085	0.00000	H_0 is rejected
STR vs Bi-STDDP	4.61880	0.00002	H_0 is rejected
STR vs Bi-G^{2}AN	3.46410	0.00160	H_0 is rejected
STR vs AttnMove	2.30940	0.04184	H_0 is rejected
STR vs PeriodicMove	1.15470	0.24821	H_0 is accepted

$$z = \frac{R_i - R_j}{\sqrt{\frac{K(K+1)}{6N}}} \tag{17}$$

where the *z-value* is used to find the corresponding probability from the table of normal distribution, R_i and R_j denote the average rank of the i-th and j-th models, respectively. K and N denote the number of comparison models and the number of datasets, respectively.

We consider the null Hypothesis H_0 as the pair of compared models are equivalent. Then, we obtain the *z-value* based on (17). Accordingly, the probability *p-value* can be computed by using the normal distribution table. In this work, we choose the most commonly-used value, i.e., 0.05, as the significance level α. Finally, we obtain the *z-value* and *p-value* of all models in Table VII. As presented in Table VII, the *p-value* is less than the significance level α in most cases, which implies that the null Hypothesis H_0 is rejected. These results validate that the proposed STR performs better than most baseline models. Additionally, STR outperforms PeriodicMove by 7%-20% as shown in Table VI, while the difference between STR and PeriodicMove is not statistically significant.

VI. Conclusion

In this work, we investigate the problem of trajectory recovery and propose a novel method to learn semantic behavior for restoring sparse human mobility trajectory. Specifically, we introduce the heterogeneous information network to encode semantic knowledge beyond single individual trajectory. Beyond that, we design a behavior attention mechanism to capture semantic transition regularities for trajectory recovery. Compared to existing trajectory recovery models that mine geographic location transition regularities, our proposed model utilizes semantic information to extract semantic transition regularities, which not only includes geographic location transition regularities, but also mines regularities with different geographic information (but with semantically similar locations). It overcomes the problem of geographic location transition regularities being unable to recover OVPs in sparse trajectories. We evaluate our proposed model based on four popular human mobility datasets and the experiments demonstrate the effectiveness of the proposed model.

In the future, we will devote to incorporate side information (such as user's stay time, weather and event information, etc.) into the recovery model so as to more accurately understand the underlying motivation of user mobility behaviors for achieving better recovery performance.

References

[1] D. L. Ferreira, B. A. A. Nunes, and K. Obraczka, "Scale-free properties of human mobility and applications to intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 11, pp. 3736–3748, Nov. 2018.

[2] M. Luca, G. Barlacchi, B. Lepri, and L. Pappalardo, "A survey on deep learning for human mobility," *ACM Comput. Surv.*, vol. 55, no. 1, pp. 1–44, Jan. 2023.

[3] E. Cubukcu, "Does the level of visual detail in virtual environments affect the user's spatial knowledge?" *Environ. Planning B, Planning Design*, vol. 38, no. 4, pp. 741–752, 2011.

[4] S. H. Bostancı and M. Oral, "Experimental approach on the cognitive perception of historical urban skyline," *Iconarp Int. J. Archit. Planning*, vol. 5, pp. 45–59, Dec. 2017.

[5] Y. Huang, Z. Xiao, D. Wang, H. Jiang, and D. Wu, "Exploring individual travel patterns across private car trajectory data," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 12, pp. 5036–5050, Dec. 2020.

[6] Z. Xiao et al., "Understanding private car aggregation effect via spatio-temporal analysis of trajectory data," *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2346–2357, Apr. 2023.

[7] Z. Xiao, H. Fang, H. Jiang, J. Bai, V. Havyarimana, and H. Chen, "Understanding urban area attractiveness based on private car trajectory data using a deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12343–12352, Aug. 2022.

[8] W. Long et al., "Unified spatial–temporal neighbor attention network for dynamic traffic prediction," *IEEE Trans. Veh. Technol.*, vol. 72, no. 2, pp. 1515–1529, Feb. 2023.

[9] Y. Wu et al., "Towards better detection and analysis of massive spatiotemporal co-occurrence patterns," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3387–3402, Jun. 2021.

[10] Z. Xiao et al., "Predicting urban region heat via learning arrive-stay-leave behaviors of private cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 10843–10856, Oct. 2023.

[11] X. Pan, X. Cai, K. Song, T. Baker, T. R. Gadekallu, and X. Yuan, "Location recommendation based on mobility graph with individual and group influences," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8409–8420, Aug. 2023.

[12] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis, and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4927–4943, Jun. 2022.

[13] J. Wang, N. Wu, X. Lu, W. X. Zhao, and K. Feng, "Deep trajectory recovery with fine-grained calibration using Kalman filter," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 921–934, Mar. 2021.

[14] S. Cheng and F. Lu, "A two-step method for missing spatio-temporal data reconstruction," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 7, p. 187, Jun. 2017.

[15] M. Ficek and L. Kencl, "Inter-call mobility model: A spatio-temporal refinement of call data records using a Gaussian mixture model," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 469–477.

[16] S. Hoteit, S. Secci, S. Sobolevsky, G. Pujolle, and C. Ratti, "Estimating real human trajectories through mobile phone data," in *Proc. IEEE 14th Int. Conf. Mobile Data Manage.*, vol. 2, Jun. 2013, pp. 148–153.

[17] S. Hoteit, S. Secci, S. Sobolevsky, C. Ratti, and G. Pujolle, "Estimating human trajectories and hotspots through mobile phone data," *Comput. Netw.*, vol. 64, pp. 296–307, May 2014.

[18] Z. Fan, A. Arai, X. Song, A. Witayangkurn, H. Kanasugi, and R. Shibasaki, "A collaborative filtering approach to citywide human mobility completion from sparse call records," in *Proc. IJCAI*, 2016, pp. 2500–2506.

[19] Z. Xiao et al., "On extracting regular travel behavior of private cars based on trajectory data analysis," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14537–14549, Dec. 2020.

[20] M. Li, S. Gao, F. Lu, and H. Zhang, "Reconstruction of human movement trajectories from large-scale low-frequency mobile phone data," *Comput., Environ. Urban Syst.*, vol. 77, Sep. 2019, Art. no. 101346.

[21] J. Xiao et al., "Vehicle trajectory interpolation based on ensemble transfer regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7680–7691, Jul. 2022, doi: 10.1109/TITS.2021.3071761.

[22] Z. Xiao, X. Qian, H. Jiang, C. Cai, and F. Zeng, "Bidirectional RNN-based private car trajectory reconstruction algorithm," *J. Commun.*, vol. 41, no. 12, p. 171, 2020.

[23] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "BRITS: Bidirectional recurrent imputation for time series," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, 2018, pp. 6775–6785.

[24] D. Xi, F. Zhuang, Y. Liu, J. Gu, H. Xiong, and Q. He, "Modelling of bi-directional spatio-temporal dependence and users' dynamic preferences for missing poi check-in identification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 5458–5465.

[25] M. Shi, D. Shen, Y. Kou, T. Nie, and G. Yu, "Missing POI check-in identification using generative adversarial networks," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Cham, Switzerland: Springer, 2021, pp. 575–590.

[26] T. Xia et al., "AttnMove: History enhanced trajectory recovery via attentional network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, 2021, pp. 4494–4502.

[27] H. Sun, C. Yang, L. Deng, F. Zhou, F. Huang, and K. Zheng, "PeriodicMove: Shift-aware human mobility recovery with graph neural network," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 1734–1743.

[28] C. Shi, Y. Li, J. Zhang, Y. Sun, and P. S. Yu, "A survey of heterogeneous information network analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 17–37, Jan. 2017.

[29] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, Feb. 2010.

[30] D. Xi et al., "Exploiting bi-directional global transition patterns and personal preferences for missing POI category identification," *Neural Netw.*, vol. 132, pp. 75–83, Dec. 2020.

[31] J. Yoon, W. R. Zame, and M. van der Schaar, "Multi-directional recurrent neural networks: A novel method for estimating missing data," in *Proc. Time Ser. Workshop 34th Int. Conf. Mach.*, 2017, pp. 1–5.

[32] H. Ren et al., "MTrajRec: Map-constrained trajectory recovery via seq2seq multi-task learning," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 1410–1419.

[33] S. Hou, Y. Ye, Y. Song, and M. Abdulhayoglu, "HinDroid: An intelligent Android malware detection system based on structured heterogeneous information network," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1507–1515.

[34] X. Wang, D. Bo, C. Shi, S. Fan, Y. Ye, and P. S. Yu, "A survey on heterogeneous graph embedding: Methods, techniques, applications and sources," 2020, *arXiv:2011.14867*.

[35] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.
[36] P. Wang, K. Liu, L. Jiang, X. Li, and Y. Fu, "Incremental mobile user profiling: Reinforcement learning with spatial knowledge graph for modeling event streams," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 853–861.
[37] J. Liu et al., "Urban flow pattern mining based on multi-source heterogeneous data fusion and knowledge graph embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 2133–2146, Feb. 2023, doi: 10.1109/TKDE.2021.3098612.
[38] H. Wang, Q. Yu, Y. Liu, D. Jin, and Y. Li, "Spatio-temporal urban knowledge graph enabled mobility prediction," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 4, pp. 1–24, Dec. 2021.
[39] M. Zhang and Y. Chen, "Inductive matrix completion based on graph neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–12.
[40] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proc. Web Conf.*, Apr. 2020, pp. 2704–2710.
[41] C. Hsu and C.-T. Li, "RetaGNN: Relational temporal attentive graph neural networks for holistic sequential recommendation," in *Proc. Web Conf.*, Apr. 2021, pp. 2968–2979.
[42] H. Hong, H. Guo, Y. Lin, X. Yang, Z. Li, and J. Ye, "An attention-based graph neural network for heterogeneous structural learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 4132–4139.
[43] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, vol. 30, no. 1. CiteSeerX, 2013, pp. 1–6.
[44] J. Feng et al., "DeepMove: Predicting human mobility with attentional recurrent networks," in *Proc. World Wide Web Conf. (WWW)*, 2018, pp. 1459–1468.
[45] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017, pp. 5998–6008.
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[47] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 1, pp. 129–142, Jan. 2015.
[48] D. Wang et al., "Stop-and-wait: Discover aggregation effect based on private car trajectory data," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3623–3633, Oct. 2019.
[49] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2011, pp. 1082–1090.
[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[51] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," 2014, *arXiv:1409.1259*.
[52] 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.*, vol. 30, no. 1, 2016, pp. 194–200.
[53] D. Kong and F. Wu, "HST-LSTM: A hierarchical spatial–temporal long-short term memory network for location prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2341–2347.
[54] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, Dec. 1937.
[55] S. Holm, "A simple sequentially rejective multiple test procedure," *Scand. J. Statist.*, vol. 6, no. 2, pp. 65–70, 1979.

Wangchen Long received the M.S. and Ph.D. degrees from Hunan University, China, in 2012 and 2023, respectively. He is currently an Instructor with the College of Artificial Intelligence, Zhuhai City Polytechnic. His research interests include the Internet of Vehicles and trajectory big data mining.

Zhu Xiao (Senior Member, IEEE) received the M.S. and Ph.D. degrees in communication and information systems from Xidian University, China, in 2007 and 2009, respectively.

From 2010 to 2012, he was a Research Fellow with the Department of Computer Science and Technology, University of Bedfordshire, U.K. He is currently a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University, China. His research interests include wireless localization, the Internet of Vehicles, and intelligent transportation systems. He is currently an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.

Hongbo Jiang (Senior Member, IEEE) received the Ph.D. degree from Case Western Reserve University in 2008. He was a Professor with the Huazhong University of Science and Technology. He is currently a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University. His research interests include computer networking, especially algorithms, and protocols for wireless and mobile networks. He is an elected member of Academia Europaea and a fellow of IET, BCS, and AAIA. He was an Editor of IEEE/ACM TRANSACTIONS ON NETWORKING, an Associate Editor of IEEE TRANSACTIONS ON MOBILE COMPUTING, and an Associate Technical Editor of *IEEE Communications Magazine*.

Yong Xiong received the bachelor's and master's degrees from Hunan Normal University, Changsha, China, in 2008 and 2011, respectively. He is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha. He is also the Vice Director of the Research and Development Center, Hunan Lianzhi Technology Company Ltd., Changsha. His current research interests include Beidou + wireless intelligent perception and data mining algorithms.

Zheng Qin received the Ph.D. degree in computing science from Chongqing University in 2001. He is currently a Professor and the Vice President of the College of Computer Science and Electronic Engineering, Hunan University. He is also the Deputy Director of the Engineering Laboratory of Cyberspace Identity and Data Security in Hunan Province, a Special Expert of the Ministry of National Security, and a member of the CCF Big Data Special Committee. His research interests include big data/cloud computing, network information security and privacy protection, smart cities, and large complex software engineering.

You Li received the Ph.D. degree in geographic information science from Wuhan University, China, in 2017. He is currently an Associate Researcher with the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China. His research interests include LiDAR processing, 3D GIS, and 3D road environment modeling.

Schahram Dustdar (Fellow, IEEE) received the Ph.D. degree in business informatics from the University of Linz, Austria, in 1992. He holds several honorary positions at the University of California (USC) at Los Angeles; Monash University, Melbourne; Shanghai University; Macquarie University, Sydney; and University Pompeu Fabra, Barcelona, Spain. From December 2016 to January 2017, he was a Visiting Professor with the University of Sevilla, Spain. From January to June 2017, he was a Visiting Professor with UC Berkeley. He is currently a Full Professor of computer science with the Research Division of Distributed Systems, TU Wien, Austria. He has an H-index of 78 with more than 36,000 citations. He is an elected member of Academia Europaea: The Academy of Europe, where he is the Chairperson of the Informatics Section. He is an Asia–Pacific Artificial Intelligence Association (AAIA) Fellow in 2021. He was a recipient of multiple awards, such as the IEEE TCSVC Outstanding Leadership Award in 2018, the IEEE TCSC Award for Excellence in Scalable Computing in 2019, the ACM Distinguished Scientist in 2009, the ACM Distinguished Speaker in 2021, and the IBM Faculty Award in 2012. He is the Founding Co-Editor-in-Chief of *ACM Transactions on Internet of Things* (ACM TIoT) and the Editor-in-Chief of *Computing* (Springer). He is an Associate Editor of IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE TRANSACTIONS ON CLOUD COMPUTING, *ACM Computing Surveys*, *ACM Transactions on the Web*, and *ACM Transactions on Internet Technology*. He is on the editorial board of IEEE INTERNET COMPUTING and IEEE TRANSACTIONS ON COMPUTERS.