8-2015

Reinforced Imitative Graph Learning for Mobile User Profiling

Dongjie Wang
University of Central Florida

Pengyang Wang
University of Macau

Yanjie Fu
University of Central Florida

Kunpeng Liu
Portland State University, kunpeng@pdx.edu

Hui Xiong
Rutgers University - Newark

See next page for additional authors

Citation Details

Authors

Dongjie Wang, Pengyang Wang, Yanjie Fu, Kunpeng Liu, Hui Xiong, and Charles Hughes

Reinforced Imitative Graph Learning for Mobile User Profiling

Dongjie Wang, Pengyang Wang, Yanjie Fu, Kunpeng Liu, Hui Xiong *Fellow, IEEE* and Charles E. Hughes

Abstract—Mobile user profiling refers to the efforts of extracting users' characteristics from mobile activities. In order to capture the dynamic varying of user characteristics for generating effective user profiling, we propose an imitation-based mobile user profiling framework. Considering the objective of teaching an autonomous agent to imitate user mobility based on the user's profile, the user profile is the most accurate when the agent can perfectly mimic the user behavior patterns. The profiling framework is formulated into a reinforcement learning task, where an agent is a next-visit planner, an action is a POI that a user will visit next, and the state of the environment is a fused representation of a user and spatial entities. An event in which a user visits a POI will construct a new state, which helps the agent predict users' mobility more accurately. In the framework, we introduce a spatial Knowledge Graph (KG) to characterize the semantics of user visits over connected spatial entities. Additionally, we develop a mutual-updating strategy to quantify the state that evolves over time. Along these lines, we develop a reinforcement imitative graph learning framework for mobile user profiling. Finally, we conduct extensive experiments to demonstrate the superiority of our approach.

Index Terms—Mobile User Profiling; Incremental Learning; Reinforcement Learning; Spatial Knowledge Graph.

✦

1 INTRODUCTION

MOBILE user profiling refers to the efforts of extracting user interests and behavioral patterns from mobile activities. Consider the existence of many mobile users in a city, each user is equipped with mobile sensing equipment moving from one location to another location and generates a mobility event stream in real time. Classical mobile user profiling collects large-scale spatio-temporal event data, and then, learns profile representations to characterize user patterns and preferences using the collected data.

Prior literature in mobile user profiling includes: (1) explicit profile extraction [4], (2) factorization-based approaches [5], [10], and (3) deep learning-based approaches [27], [29]. Other studies exploit adversarial deep learning to emphasize substructure patterns in mobile user profiling [21]. All of these approaches can be regarded as exploiting user activities' prediction bias or reconstruction loss of the users' profile structure as the learning criteria to model users' profiles. The key limitation of these approaches is the modeling procedure is solely based on the individual user, which lacks a global perception of the dynamic varying environment. However, the behavioral data generated by the mobile users are usually a mixed-user, spatially and temporally discrete, event stream. Users inject their impacts in the environment mixed and chronologically, leading to the dynamic changes in the environment that inversely affect a users' decision.

- *Dongjie Wang and Pengyang Wang are with equal contributions.*
- *Dongjie Wang, Yanjie Fu, and Charles E. Hughes are with the University of Central Florida (UCF). Email: wangdongjie@knights.ucf.edu, yanjie.fu@ucf.edu, charles.hughes@ucf.edu.*
- *Pengyang Wang is with the State Key Laboratory of Internet of Things for Smart City, University of Macau. Email: pywang@um.edu.mo.*
- *Kunpeng Liu is with Portland State University. Email: kunpeng@pdx.edu.*
- *Hui Xiong is with Rutgers University. Email: hxiong@rutgers.edu.*

After exploring many profiling methods, we found that minimizing user activities' prediction bias or reconstruction loss may not be the best criteria to evaluate profiling accuracy. Unlike traditional loss minimization, we identify a better criteria, which we call the *imitation based criteria*: considering the objective of teaching an autonomous agent to imitate a mobile user to plan where a user will visit next, based on the profile of the user. The user profile is the most accurate when the agent can perfectly copy the activity patterns of the user.

The emerging reinforcement learning can train an agent to plan for its next actions in order to function in its environment. Such an ability provides great potential to implement the imitation based criteria in order to achieve more accurate user profiling.

As a result, we propose to formulate the problem into a reinforcement learning framework. In this framework, an agent is a next-visit planner that tries to perfectly imitate a set of mobile users. The state of environment is a fused representation of a given user and spatial entities (*e.g.*, POIs, activity types, functional zones). An action is a POI that a given mobile user will visit, which is estimated based on the state of the environment by the agent. An event where the user takes the action to visit the POI, will change the environment, resulting into a new state of the user and the spatial entities, which helps the agent to better estimate the next visit. The reward of an action is the reduction of the gap between the agent's activity patterns and the user's activity patterns. After the reformulation, our new objective is to exploit the reinforcement learning framework to extract dynamic profile representations of various users in the state of the environment by incremental learning from an user-visit event stream.

To further improve the profiling accuracy of the framework, we analyze how a mobility event connect mobile users with spatial entities (*e.g.*, POIs, functional zones), and

identify *two important structured information*.

Firstly, there is *semantic connectivity among spatial entities*. Specifically, the profile of a user can be reflected by a sequential composition of mobility events. The semantics of an event are about which building the user visits (POI), what type of activity the user conducts during the visit (POI category), and in which region the POI is located (urban functional zone). Therefore, improving the representations of POIs, activity types, and functional zones of events can, in return, improves user profiling. The semantic connectivity among these spatial entities refers to the observation that every time a mobile user visits a POI, a new connection is established or reinforced among a POI, an activity type, and a functional zone, which indeed is a heterogeneous graph with geographic knowledge. Therefore, we propose to use a spatial knowledge graph (*KG*) to describe such semantic connectivity, and a translation based embedding method is employed to learn the embedding of the spatial *KG*.

Secondly, there is *mutual influence between users and spatial entities*. An event that a user visits and interacts with the POI, will change and reinforce the edges (semantic connectivity) of the spatial knowledge graph, resulting into new state representations of POIs, activity types, and functional zones. If the states of spatial entities are updated, once a user visits these newly-updated spatial entities, then the state of the user profile will be updated as well. In other words, the state updates between spatial entities and users are sequentially nested together per mobility event. We propose a sequentially nested updating strategy to update user states by jointly considering the spatial KG and temporal contexts, and update spatial KG states by jointly considering user states and temporal contexts.

In summary, we propose a reinforced imitative graph learning framework for mobile user profiling by integrating a reinforcement learning framework with spatial KG to solve. Our contributions are: (1) We propose a new imitation based criteria for evaluating the accuracy of user profiling: the better an agent imitates a mobile user, the more accurate the user profile is. (2) Motivated by the imitation based criteria, we reformulate the mobile user profiling problem into a reinforcement learning framework, where an agent is a next-visit planner, the state of environment is the fused representation of users and spatial KG, an action is a POI visited by a user, and the reward of an action is how well the agent can imitate mobile users. (3) We develop a new reward function to evaluate the quality of actions. We extend the reward function in [23] to be negative when the predicted POI visits are far away from the real, which would positively encourage the agent on good actions and punish the agent on bad actions. (4) We extend the policy network in [23] based on Double DQN with hierarchical graph pooling, which avoids the overestimation of specific POI visits. (5) We identify and describe the semantic connectivity of spatial entities by a spatial KG, which is integrated into the reinforcement learning framework. (6) We develop a new sequentially nested state update strategy by modeling the long-short term effects of interactions between users and spatial KG, by paying more attention to recent ones. (7) We present extensive experimental results with real-world mobile check-in data to demonstrate improved performances. (8) Our framework can be adapted and generalized to the

TABLE 1: Summary of Notations.

Symbol	Definition
$u, \mathbf{u}^{(\cdot)}$	User, user state (representations).
$c_{(\cdot)}$	POI category.
$a_{(\cdot)}$	Action.
$P_{(\cdot)}, \mathbf{h}_{P_{(\cdot)}}$	POI, POI representations.
$\mathbf{rel}_{(\cdot)}$	Relation representations.
$\mathbf{t}_{(\cdot)}$	Tail (categories, locations) representations.
$\mathbf{g}^{(\cdot)} = <\mathbf{h}_{P_{(\cdot)}}, \mathbf{r}_{(\cdot)}, \mathbf{t}_{(\cdot)}>$	Spatial *KG* state (representations).
$s^{(\cdot)} = (\mathbf{u}^{(\cdot)}, \mathbf{g}^{(\cdot)})$	State.
$r_{(\cdot)}$	Reward.
Q	Policy.
$x_{(\cdot)}$	Priority score.
$\mathbf{T}^{(\cdot)}, \tilde{\mathbf{T}}^{(\cdot)}$	Temporal context, transformed.
$\mathbf{W}_{(\cdot)}, \mathbf{b}_{(\cdot)}$	Weights, biases of model.
$\lambda_{(\cdot)}$	Weights of rewards.

tasks of incremental learning with mixture event streams to support mobile user profiling and other applications.

2 PRELIMINARIES

2.1 Definitions and Problem Statement

We first introduce the key definitions and the problem statement. Then, we present an overview of the proposed framework, followed by a discussion of how our approach differs from those presented in the current literature. All the notations are summarized in Table 1.

2.1.1 Spatial Knowledge Graph (*KG*).

We construct spatial *KG* to demonstrate semantic connectivities between spatial entities. Specifically, in the spatial *KG*, there are three types of spatial entities: POIs, POI categories and locations (*i.e.* functional zones); and two types of relations: "belong to" which is to demonstrate the affiliation relations between POIs and POI categories, and "locate at" which is to demonstrate the geospatial relation between POIs and the functional zones. The spatial *KG* is defined as the following groups of triplet facts: (1) <POI, "belong to", POI category>, and (2)<POI, "locate at", functional zones>.

2.1.2 Temporal Context.

In this paper, temporal context quantifies the temporal factors when users make the decision to visit a POI. Following the approach in previous studies [13], we utilize the snapshot of transportation traffic in a small time window for quantifying temporal context. Specifically, we first segment the entire area into m grids. For a given grid, we calculate the inner traffic, in-flow traffic and out-flow traffic. Then, we obtain a traffic matrix $\mathbf{T} \in \mathbb{R}^{m \times 3}$, where each row denotes a grid, each column denotes the inner traffic, in-flow traffic and out-flow traffic of the given grid respectively. We use the traffic matrix $\mathbf{T}$ to represent the temporal context.

2.1.3 Key Components of Reinforcement Learning.

In our problem setting, we define the key components of our reinforcement learning framework as follows:

1) **Agent.** We consider the next-visit planner as the agent. The agent provides the personalized POI prediction based on the current environment status.
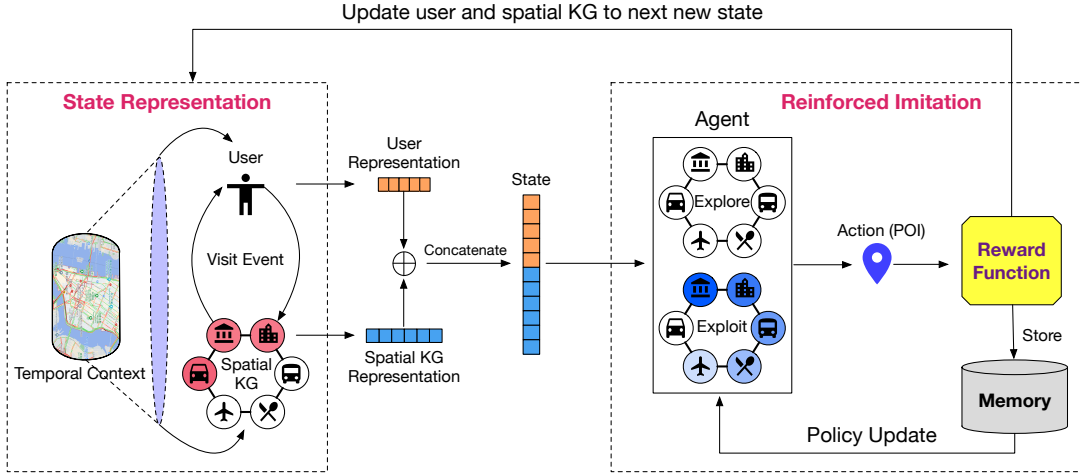
Fig. 1: Our framework includes: 1)state representation module, which is to learn the representation of users and spatial KG. 2) reinforced imitation module, which is to imitate the user behavior based on the output of the previous module.

2) **Actions.** Actions are defined as the visit event such that at each step, the user will visit which POI. Formally, let a_j denote the action that visits the POI P_j. The action space is the number of POIs. Suppose the user visits the POI P_j at the step l, then the policy would take the action $a^l = a_j$.

3) **Environment.** The environment is defined as the combination of all users and the spatial *KG*. Within the environment, users would interact with spatial *KG* by visiting any POIs in the spatial *KG*. Thus, on one hand, the visit behavior of users would affect the representation of the spatial *KG*; on the other hand, the spatial *KG* inversely would affects each user's representation as well.

4) **State.** The state s is to describe the environment composed by users and the spatial *KG*. At the step l, the state s^l is defined as a pair of $(\mathbf{u}^l, \mathbf{g}^l)$. Specifically, $\mathbf{u}^l = \{\mathbf{u}_i^l | u_i \in \mathcal{U}\}$, where $\mathbf{u}_i^l$ denotes the representation of user u_i at step l; $\mathbf{g}^l = < \mathbf{h}^l, \mathbf{rel}, \mathbf{t}^l >$, where $\mathbf{h}^l$ denotes the heads (*i.e.*, POIs) and $\mathbf{t}^l$ denotes the tails (*i.e.*, categories and functional zones).

5) **Reward.** In the conference version [23], we propose the reward function as the weighted summation of the reciprocal of the distance r_d, the category similarity r_c, and binary accuracy between the real and predicted POI visit r_p. The range of such reward function is above zero, which means the agent would be always encouraged positively no matter whether the generated action is good or bad. However, intuitively, the agent should be encouraged positively for the good actions, but be punished for the bad actions. Based on such intuition, we propose a new formulation for the reward function, which would assign positive values for good actions as the encouragement, and negative values for bad actions as the penalty. Formally, the new reward function can be denoted as

$$r = \lambda_d \times (r_d - b_d) + \lambda_c \times (r_c - b_c) + \lambda_p \times (r_p - b_p), \quad (1)$$

where b_d, b_c and b_p are the reward baselines of r_d, r_c and r_p respectively. We set the reward baselines as follows: (1) we first run the experiment for 100 rounds,

and obtain the empirical range of the reward values; (2) we then set the value of b_d, b_c, and b_p as the first quartile of the corresponding empirical reward range. So if the agent behaves better than the baselines, the reward is positive, otherwise, it is negative. This reward function urges the agent to find the next POI that is close to the real visited POI from the POI category and location.

Specifically, we calculate the euclidean distance between the real and predicted POI according to their longitude and latitude coordinates, and regard the reciprocal value of the distance as r_d. We exploit *GloVe*[1] pre-trained word vectors of the category names to calculate the cosine similarity between the predicted and real POI categories as the similarity r_c.

2.1.4 Problem Statement.

In this paper, we study the problem of learning to profile users from user mobile activity data. Due to the large-scale, nested, sequential and semantic nature of user mobile activities, we reformulate the mobile user profiling problem as an incremental user modeling with the integration of reinforcement learning and spatial *KG*.

Formally, given a mobile activity sequence of mixed-user and spatial *KG*, we aim to find a mapping function $f : (\mathbf{u}^l, \mathbf{g}^l) \rightarrow (\mathbf{u}^{l+1}, \mathbf{g}^{l+1})$, that takes as input the state of the environment (representations of users and spatial *KG*) at the step l, and outputs the state at the next step $l + 1$, while simultaneously following the imitation-based criteria to provide accurate personalized prediction, based on the incrementally updated user representations from the mutual interactions with spatial *KG*.

2.2 Framework Overview

Figure 1 shows an overview of our framework that includes two key components: (1) state representation, and (2) reinforced imitation. For the state representation, we consider the mutual interactions between users and spatial *KG* given the temporal context. Specifically, at each step, the user representations are updated based on the influence from

1. https://nlp.stanford.edu/projects/glove/

spatial *KG* given the temporal context; inversely, the spatial *KG* representations are updated based on the influence from the user given the temporal context. For the reinforced imitation, we employ a reinforced agent to imitate the user mobility patterns via exploration and exploitation. The prediction of the agent is evaluated by a reward function. Following that, the user visit trajectories are stored in memory to offline train the agent for updating learned policies. In addition, the user and spatial KG are changed to the next state for starting a new visit event. When the reinforced imitation mimics user mobility perfectly, our framework outputs the most effective user profiles.

Comparison with literature. Despite the promising results from prior studies on mobile user profiling, one major concern arises that most of these methods are trained on offline data without the ability of self-updating, which is essential for quantifying user' dynamic mobility behaviors changing over time. Therefore, in our work, we embrace reinforcement learning for mobile user modeling. When exploiting reinforcement learning for user modeling, online recommender systems have achieved great performance by regarding the online behavior as a sequential decision-making process. Different from the online user modeling, mobile user behavior (*e.g.*, POI visiting) is offline behavior which is constrained by many offline factors (*e.g.*, time, traffic, location). Therefore, we incorporate temporal contexts and simultaneously update user and spatial *KG* representations based on the mutual interactions. In the meantime, the integrated spatial *KG* provides rich semantics to better understand user patterns and preferences.

3 POLICY DESIGN

In this section, we will introduce how to teach the next-visit planner (agent) to mimic users' patterns and preferences, given state (representations of user and spatial *KG*) and temporal contexts.

3.1 Network Structure

While DQN shows exceptional performance on learning strategies [23], the maximization step of the action value estimation process leads the DQN to overestimate Q-value, resulting in learning local optimal policies. In order to alleviate this issue, we replace the DQN to the Double Deep Q-Network (DDQN) [18].

Figure 3 shows the structure of DDQN. Compared to the model structure (Figure 2) of the conference version [23], we implement two neural networks (Q_e and Q_t) to decouple the maximization operation in DQN into action selection and action evaluation. Specifically, we first concatenate the user representations and spatial KG representations together as the state vector. Then, we input the state into the Q_e network to select the action (POI) with the highest Q-value. Moreover, we input the state and the previously selected action into the Q_t network to estimate the real value of the action in order to update learned policies. During this process, Q_e and Q_t own the same model structure and the same initialization. But we first update the parameters of Q_e for several steps and freeze the parameters of Q_t. After that, we store the parameters
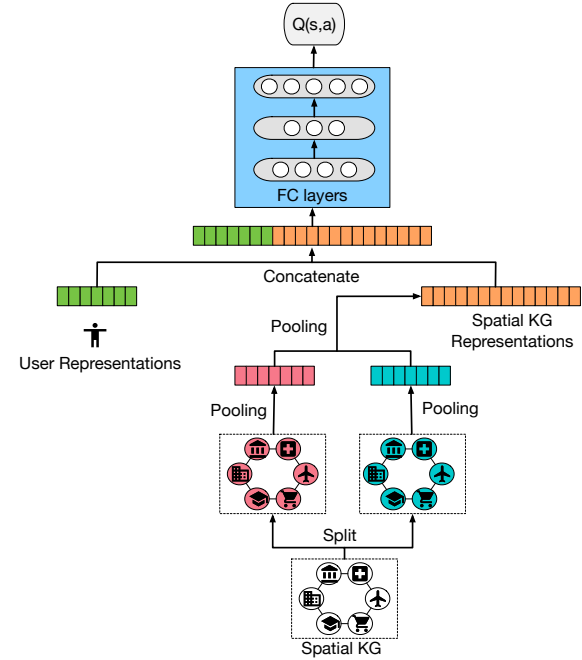

Fig. 2: Imitation Agent (DQN) takes as input user and spatial KG representations and outputs the next-visit POI.
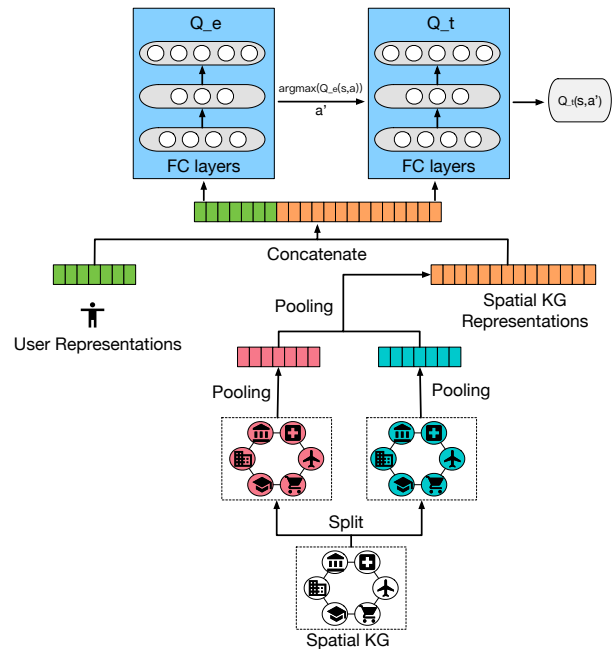

Fig. 3: Advanced Imitation Agent (Double-DQN) mitigates Q-value overestimation by utilizing two neural networks.

of Q_e into Q_t and reiterate the previous learning step. Thus, Q_e and Q_t have different estimation abilities at the same time. The max Q-value estimations in the two networks associated with different actions. So, the Q-value used to update policies is more rational than the Q-value in DQN. Such a learning strategy has been proved effective in avoiding the over-optimistic problem of DQN and learning more effective policies [18].

Since DDQN only takes vectors/matrices as input, graph pooling is desired for graph-structure state **G** transformed into vectors. But current graph pooling operations do not

fit for the heterogeneity of spatial *KG*. Therefore, we design the graph pooling operation in a hierarchical fashion. We first split the spatial *KG* into two graphs based on the two types of relations as (1) "location" graph and (2) "belonging" graph. Then, we leverage graph average pooling for both the "location" and "belonging" graph to generate graph-level vectorized representations respectively. Finally, we employ average pooling over the two graph-level vectors to obtain the single unified vectorized representation $\mathbf{g}^l$ for the spatial *KG* at step l.

After the hierarchical pooling module, we concatenate the user state $\mathbf{u}_i^l$ with the spatial *KG* state $\mathbf{g}^l$ as the input of the fully connected (*FC*) layers. Then, the *FC* layers would map the given state $\mathbf{s}^l$ into a group of $Q(s^l, a)$ for each POI in the spatial *KG*. The policy chooses the POI with the highest $Q(s^l, a)$ as the prediction result.

3.2 Improved Sampling Strategy in Experience Replay

Since the space of POIs visiting events is very large, the proposed reinforcement learning framework is computationally intensive. Therefore, we propose a training strategy to accelerate exploration procedure based on experience replay [11]. The training strategy involves two stages: (1) priority assignment, which assigns the priority score for each data sample (s^l, a^l, r^l, s^{l+1}); and (2) sampling strategy, which selects data samples from memory for training.

3.2.1 Priority

We design two types of priority scores, including reward-based and temporal difference-based.

1) Reward-based. Intuitively, the higher the reward of the POI visit action a^l is, the better the next-visit planner (agent) mimic the users, then the more significant the data sample can contribute to the policy training. Therefore, formally, for each data sample (s^l, a^l, r^l, s^{l+1}), the reward-based priority score x_r is defined as the reward r^l:

$$x_r(s^l, a^l, r^l, s^{l+1}) = r^l. \quad (2)$$

2) Temporal difference (*TD*)-based. The *TD* error is originally set for updating the *DQN*. The larger the *TD* error is, the more valuable and informative is the data sample for the next-visit planner (agent) to learn. Therefore, we define the *TD* error as the *TD*-based priority score that

$$x_{TD}(s^l, a^l, r^l, s^{l+1}) = r^l + \gamma \max_{a^{l+1}} Q(s^{l+1}, a^{l+1}) - Q(s^l, a^l), \quad (3)$$

where γ is the discount factor.

We will evaluate and discuss the performance of these two priority scores in the experiment.

3.2.2 Sampling Strategy

After we obtain the priority score x_*, we need to construct a distribution from the priority score x_* for sampling data. Therefore, we employ *softmax* to convert the priority score into a distribution

$$P(k) = \frac{e^{x^{(k)}}}{\sum\limits_{k'=1}^{K} e^{x^{(k')}}}, \quad (4)$$

where $P(k)$ is the sampling probability of the k-th data sample given the corresponding priority score $x^{(k)}$.

Then, we sample a batch size of data from the memory based on the assigned probability.
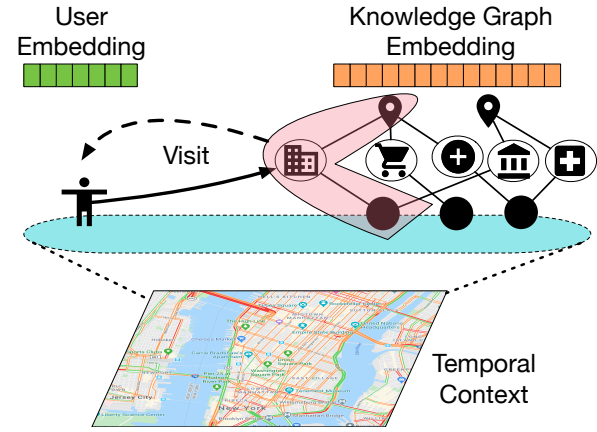

Fig. 4: An example of mutual interactions between users and spatial *KG*.

4 STATE REPRESENTATION LEARNING

In this section, we introduce details about how to initialize and update states based on the mutual interactions between mobile users and spatial *KG*.

4.1 State Initialization

Here, we discuss the initialization of user state (representations), followed by how to initialize spatial *KG* state (representation).

4.1.1 User

We leverage *StructRL* [21] to initialize the user state (representations). *StructRL* is a mobile user profiling framework that models the global and substructure patterns of user behaviors by minimizing the structural loss between the input and reconstructed human mobility graph through adversarial learning. For each user, we extract the very first portion (10%) of user mobility trajectories to construct human mobility graph, where the nodes are POI categories and edges are visit transition frequencies or duration among POI categories. Then, we employ *StructRL* to learn user representations. The learned user representations are regarded as the initialized user state fed to the reinforcement learning framework.

4.1.2 Spatial *KG*

We exploit *TransD* [7] to initialize spatial *KG* state (representations). *TransD* is a translation-based model that utilizes two vectors to learn entity/relation representations and construct mapping matrix dynamically with considering both the diversity of entities and relations, which outperforms *TransE* [1], *TransH* [26], and *TransR* [12] in learning *KG* representations. Specifically, to be more compatible with reinforcement learning framework, we adopt the projected entity embedding as the initialization of spatial *KG*. In this way, the representations of entity and relations will be in the same feature space, which will make the state updating more feasible.

4.2 State Update

In this section, we will introduce how to update states based on the mutual interactions between user and spatial *KG*.

Figure 4 shows an interaction example in which a user visits a POI in spatial *KG* During the interaction, the influence from the user visit is directly injected into the visited POI. Then, the influence is propagated to other spatial entities through the semantics and topology of spatial *KG*. The influence from spatial *KG* then inversely affects the user state (representations). Without the loss of the generality, we consider the scenario of a POI visit event where a user u_i visits the POI P_j at step l. The states will update for the step $l + 1$.

In the conference version [23], we propose a state update strategy that updates the states incrementally by equally treating the impacts of each visit. However, intuitively, the impact of each visit should be different, so the most recent visits should be given more attention than previous ones. To solve the problem, we follow the idea from the Long Short Term Memory (LSTM) networks, in which the model should forget partial old information and combine new changes simultaneously. Based on this rule, we propose the new state update strategy as follows.

(1) **User**

We update the user state $\mathbf{u}_i^{l+1}$ based on old user state $\mathbf{u}_i^l$ and the interaction between $\mathbf{u}_i^l$ and $\mathbf{h}_{P_j}^l$ in a temporal context. The update process can be represented as:

$$\mathbf{u}_i^{l+1} = \alpha_u \times \mathbf{u}_i^l + (1 - \alpha_u) \times (\mathbf{W}_u \cdot (\mathbf{h}_{P_j}^l)^\intercal \cdot \tilde{\mathbf{T}}^l)), \quad (5)$$

where α_u is a scalar that indicates the ratio of information of $\mathbf{u}_i^l$ in $\mathbf{u}_i^{l+1}$. At this point, α_u is calculated by $\mathbf{u}_i^l$ and the calculation process can be represented as:

$$\alpha_u = \sigma(\mathbf{W}_{\alpha_u} \cdot \mathbf{u}_i^l + \mathbf{b}_{\alpha_u}), \quad (6)$$

where $\mathbf{W}_{\alpha_u} \in \mathbb{R}^{1 \times N}$ is weight, N is the dimension of the weight matrix, and $\mathbf{b}_{\alpha_u} \in \mathbb{R}^{1 \times 1}$ is bias.

(2) **Spatial** *KG*

a) Updating visited POI $\mathbf{h}_{P_j}^{l+1}$

Similar to the update process of $\mathbf{u}_i^{l+1}$, the new POI state $\mathbf{h}_{P_j}^{l+1}$ is updated by the old POI state $\mathbf{h}_{P_j}^l$ and the interaction between $\mathbf{u}_i^l$ and $\mathbf{h}_{P_j}^l$ in a temporal context. The update process can be represented as:

$$\mathbf{h}_{P_j}^{l+1} = \alpha_p \times \mathbf{h}_{P_j}^l + (1 - \alpha_p) \times (\mathbf{W}_p \cdot (\mathbf{u}_i^l)^\intercal \cdot \tilde{\mathbf{T}}^l), \quad (7)$$

where α_p is a scalar that represents the proportion of information of $\mathbf{h}_{P_j}^l$ in $\mathbf{h}_{P_j}^{l+1}$. Here, α_p is calculated by $\mathbf{h}_{P_j}^l$ and the process can be represented as:

$$\alpha_p = \sigma(\mathbf{W}_{\alpha_p} \cdot \mathbf{h}_{P_j}^l + \mathbf{b}_{\alpha_p}), \quad (8)$$

where $\mathbf{W}_{\alpha_p} \in \mathbb{R}^{1 \times N}$ is weight , N is the dimension of the weight matrix, $\mathbf{b}_{\alpha_p} \in \mathbb{R}^{1 \times 1}$ is bias.

b) Updating category and functional zones (tail) $\mathbf{t}_*^{l+1}$

We update tail $\mathbf{t}_*^{l+1}$ based on the $\mathbf{t}_*^l$ and the combination between $\mathbf{h}_{P_j}^{l+1}$ and $\mathbf{rel}_{(P_j, \cdot)}$. The update process can be represented as:

$$\mathbf{t}_{(P_j, \cdot)}^{l+1} = \alpha_t \times \mathbf{t}_{(P_j, \cdot)}^l + (1 - \alpha_t) \times (\mathbf{h}_{P_j}^{l+1} + \mathbf{rel}_{(P_j, \cdot)}), \quad (9)$$

where α_t is a scalar that denotes the ratio of information of $\mathbf{t}_{(P_j, \cdot)}^l$ in $\mathbf{t}_{(P_j, \cdot)}^{l+1}$. Here, α_t is calculated by $\mathbf{t}_{(P_j, \cdot)}^l$, the process can be represented as:

$$\alpha_t = \sigma(\mathbf{W}_{\alpha_t} \cdot \mathbf{t}_{(P_j, \cdot)}^l + \mathbf{b}_{\alpha_t}), \quad (10)$$

where $\mathbf{W}_{\alpha_t} \in \mathbb{R}^{1 \times N}$ is weight, N is the dimension of the weight matrix, $\mathbf{b}_{\alpha_t} \in \mathbb{R}^{1 \times 1}$ is bias.

c) Updating same category and location POIs $\mathbf{h}_{P_{j-}}^{l+1}$

We update the $\mathbf{h}_{P_{j-}}^{l+1}$ based on $\mathbf{h}_{P_{j-}}^l$ and the current state of $\mathbf{t}_{(P_{j-}, \cdot)}^l$ and $\mathbf{rel}_{(P_{j-}, \cdot)}$. The update process can be represented as:

$$\begin{cases} \mathbf{h}_{P_{j-}}^{l^n} = \mathbf{t}_{(P_{j-}, \cdot)}^{l+1} - \mathbf{rel}_{(P_{j-}, \cdot)}, \\ \mathbf{h}_{P_{j-}}^{l+1} = \alpha_{p-} \times \mathbf{h}_{P_{j-}}^l + (1 - \alpha_{p-}) \times \mathbf{h}_{P_{j-}}^{l^n}, \end{cases} \quad (11)$$

where $\mathbf{h}_{P_{j-}}^{l^n}$ represents the new state of P_{j-} that is calculated by $\mathbf{t}_{(P_{j-}, \cdot)}^l$ and $\mathbf{rel}_{(P_{j-}, \cdot)}$; α_{p-} indicates the ratio of information of $\mathbf{h}_{P_{j-}}^l$ in $\mathbf{h}_{P_{j-}}^{l+1}$. Here, α_{p-} is calculated by $\mathbf{h}_{P_{j-}}^l$, the process can be represented as:

$$\alpha_{p-} = \sigma(\mathbf{W}_{\alpha_{p-}} \cdot \mathbf{h}_{P_{j-}}^l + \mathbf{b}_{\alpha_{p-}}), \quad (12)$$

where $\mathbf{W}_{\alpha_{p-}} \in \mathbb{R}^{1 \times N}$ is weight, N is the dimension of the weight matrix, $\mathbf{b}_{\alpha_{p-}} \in \mathbb{R}^{1 \times 1}$ is bias. The same process is used to update strategy1, here, we only update the local subgraph, which makes the update process efficiently when the spatial *KG* is large.

5 EXPERIMENT

This section details our empirical evaluation of the proposed method on real-world data.

5.1 Data Description

Table 2 shows the statistics of our two check-in datasets from two cities: New York [28] and Beijing [24]. Each dataset includes User ID, Venue ID, Venue Category ID, Venue Category Name, Latitude, Longitude, and Time. Additionally, we also collected taxi data to represent traffic of the New York and Beijing in Table 3. The taxis traffic data includes ID, Pick-up Latitude, Pick-up Longitude, Drop-off Latitude , Drop-off Longitude, Pick-up time and Drop-off time. We split the taxi data into small segments with one-hour time window. For each time window, we calculate the temporal context $\mathbf{T}$, compatible with time users' visit events.

TABLE 2: Statistics of the checkin data.

City	# Check-ins	# POIs	# POI Categories	Time Period
New York	227, 428	38, 334	251	4/2012-2/2013
Beijing	6, 465	3, 434	9	3/2011-5/2011

TABLE 3: Statistics of the taxi data.

City	# Transactions	Time Period
New York	161, 211, 550	4/2012-2/2013
Beijing	12, 000, 000	3/2011-5/2011

5.2 Evaluation Metrics

We evaluate the model performances over the prediction on the activity types (*i.e.*, POI categories) and locations in terms of the following four metrics:

(1) Precision on Category (Prec_Cat): We regard the prediction on activity type as the multi-class classification task. We
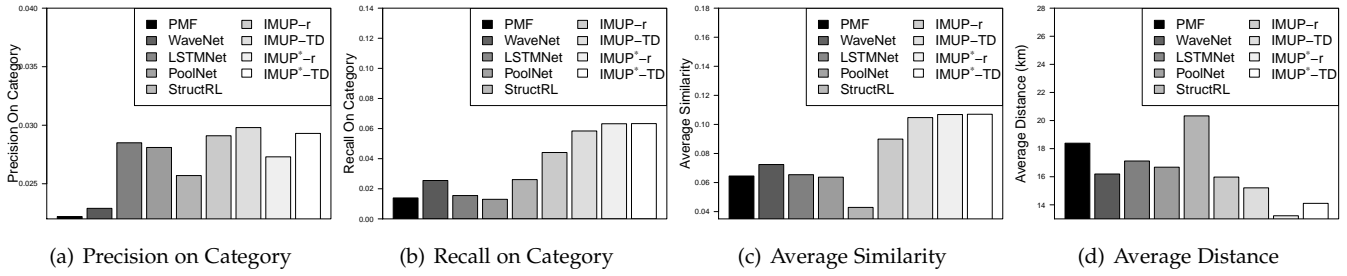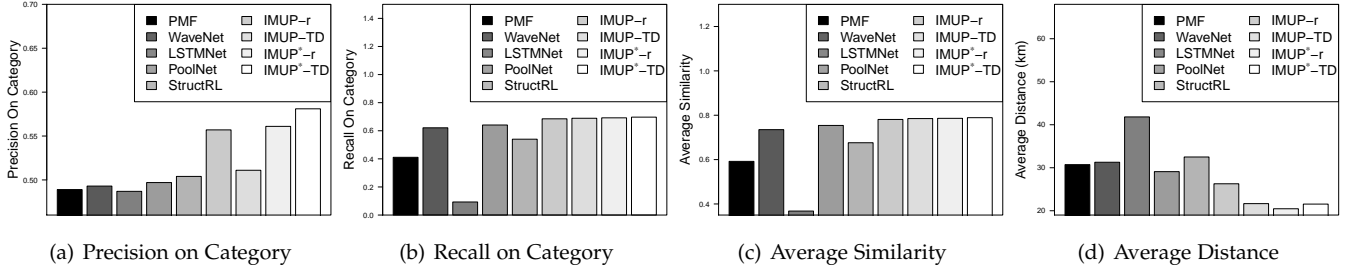
Fig. 5: Overall comparison *w.r.t.* New York dataset.

Fig. 6: Overall comparison *w.r.t.* Beijing dataset.

evaluate the classification prediction via the weighted precision. Let c_k denote the k-th category, $|c_k|$ denote the number of activity types, I_{TP}^k denote the number of true positive predictions, and I_{FP}^k denote the number of false positive predictions, then the weighted precision on categories can be represented as

$$\text{Prec\_Cat} = \frac{|c_k| \cdot I_{TP}^k}{\sum\limits_k |c_k|(I_{TP}^k + I_{FP}^k)} \quad (13)$$

(2) Recall on Category (Rec_Cat): Continuing with the definition of Prec_Cat, we use the weighted recall to evaluate the recall on category prediction. Let I_{FN}^k denote the number of false negative prediction for the category c_k, the weighted recall can be represented as

$$\text{Rec\_Cat} = \frac{|c_k| \cdot I_{TP}^k}{\sum\limits_k |c_k|(I_{TP}^k + I_{FN}^k)} \quad (14)$$

(3) Average Similarity (Avg_Sim): In addition to evaluating the precision and recall of prediction, we also evaluate the average similarity between the real and predicted POI categories. We adopt the pre-trained *GloVe* word vectors to calculate the cosine similarity between the category word vector "word^l" of the real visited POI and the category word vectors "$\hat{\text{word}}^l$" of the predicted POI. Let L denote the total visit number, then the average similarity on categories is

$$\text{Avg\_Sim} = \frac{\sum\limits_l cosine(\text{word}^l, \hat{\text{word}}^l)}{L}. \quad (15)$$

The higher the value of Avg_Sim, the better the prediction.
(4) Average Distance (Avg_Dist): We evaluate the prediction on location with the average distance. Let $Dist(P^l, \hat{P}^l)$ denote the distance between the real location P^l and the predicted location $\hat{P}^l$ for the l-th visit, then the average distance is

$$\text{Avg\_Dist} = \frac{\sum\limits_l Dist(P^l, \hat{P}^l)}{L}. \quad (16)$$

The lower the value of Avg_Dist, the better the prediction.

5.3 Baseline Algorithms

We compare the performances of our method (namely "IMUP") against the following baseline algorithms.
(1) PMF is a classic framework for modeling users through the probabilistic matrix factorization over the user-item interaction matrix [14].
(2) PoolNet is proposed to learn user representations by averaging the representations of items with which they have interacted through a deep neural network model [3].
(3) WaveNet is originally designed for generating raw audio waveforms in a generative way. It can also be applied to learning representations for sequential decision-making through stacked causal atrous convolutions [16].
(4) LSTMNet represents users as the hidden state at each timestep with a recurrent neural network by feeding the visiting sequence in to the model [6].
(5) StructRL models substructures of user mobility graph for quantifying specific activity patterns through the adversarial learning paradigm [21].

By contrast, in the conference version [23], there are two variants for incremental user profiling: **(1) IMUP-r**, where the model utilizes the sampling strategy with the priority x_r; **(2) IMUP-TD**, where the model utilizes the sampling strategy with the priority x_{TD}. In this work, we propose a new reward function, a new policy network, and a new state update strategy enhance the conference version work. To validate the effectiveness of each part, we develop

Here, we develop two variants for this paper: **(1) IMUP*-r, (2) IMUP*-TD**. Compared with **IMUP-r** and **IMUP-TD**, the difference is the two variants utilize all new modifications of this paper during the process of incremental user profiling.

In the experiment, we split the datasets into two non-overlapping sets: for each user, the earliest 90% of check-ins are the training set and the remaining 10% check-ins are testing set. We set the dimension of user representations as 200 for all the baselines and our proposed methods. We adopt the implementation [2] for evaluating PMF. And we
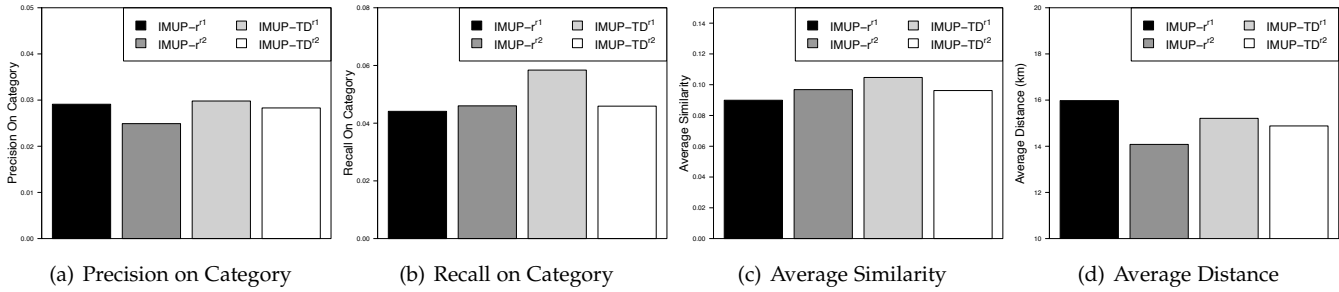
2. https://github.com/fuhailin/Probabilistic-Matrix-Factorization

(a) Precision on Category   (b) Recall on Category   (c) Average Similarity   (d) Average Distance

Fig. 7: The comparison between the original reward ($r1$) and the enhanced reward ($r2$) *w.r.t.* New York dataset.

(a) Precision on Category   (b) Recall on Category   (c) Average Similarity   (d) Average Distance
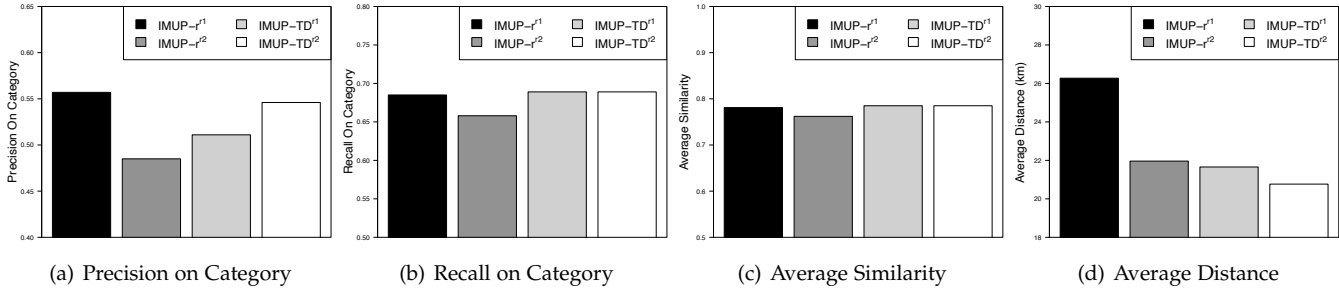
Fig. 8: The comparison between the original reward ($r1$) and the enhanced reward ($r2$) *w.r.t.* Beijing dataset.

adopt "spotlight" [9] for evaluating PoolNet, WaveNet, and LSTMNet, where the learning rate is set as 0.01. We follow the parameter setting of StructRL in [21]. Our code and data are released by Dropbox [3].

All evaluations were conducted using Ubuntu 18.04.3 LTS, on an Intel(R) Core(TM) i9-9920X CPU@ 3.50GHz, with Titan RTX 32 GB and RAM memory size 128G.

5.4 Overall comparison

Figure 5 and Figure 6 show the comparison results of user profile quality in terms of "Precision on Category", "Recall on Category", "Average Distance" and "Average Similarity". We can find that "IMUP*-r" and "IMUP*-TD" outperform other baseline models on both New York and Beijing datasets, and significantly enhance "Average Distance" and "Average Similarity". A potential reason for the observation is that the representation of the spatial *KG* involves the geospatial correlations into state vectors. So, the reinforced agent can capture the user mobility semantics and interests based on such states to mimic user mobility patterns more accurately. Moreover, the new state update strategy is able to automatically discard old user visit interests and incorporate new user visit preferences. Thus, the next visit prediction of our method becomes as close as possible to the real one from the semantic and geographical perspectives.

5.5 Comparison of different sampling strategies

We propose two priority-based sampling strategies to improve the policy learning procedure. Figure 5 and Figure 6 show the comparison results in terms of all evaluation metrics. We can find that the TD-based sampling strategy is slightly better than reward-based sampling strategy. A potential reason is that the reinforced agent covers the obvious user visit interests but makes mistakes for several

3. https://www.dropbox.com/sh/l6syw989teu9b96/AADXfq-6Es6LjMEFfrf4t3Kna?dl=0

implicit user preferences. The TD-based strategy can force the reinforced agent to learn such confused knowledge by sampling related data samples in order to improve the user mobility imitation.

5.6 Comparison of different rewards

We replace the reward function ($r1$) of the conference version [23] with the new reward function ($r2$) in this work. We develop four variants based on different sampling strategies and the two rewards, namely "IMUP$-r^{r1}$","IMUP$-r^{r2}$", "IMUP$-TD^{r1}$","IMUP$-TD^{r2}$". Figure 7 and Figure 8 show the comparison results. We can find that $r2$ outperforms $r1$ significantly in terms of "Average Distance", and keeps stable in terms of "Precision on Category", "Recall on Category", "Average Similarity". The underlying driver is that $r2$ encourages the agent to search for possible visited POIs that are semantically and geographically close to the real visited POI. If the predicted POI is further from the real one than our expectation on geographical distance, the agent will be punished via getting negative rewards. Thus, in order to get more positive rewards, the agent makes more accurate next-visit predictions that are more semantically and geographically close to real visited POIs.

5.7 Comparison of different policy networks

We replace the reinforced module of our framework from DQN with double DQN to mitigate Q-value overestimation to improve model performance. We use "dqn" and "$ddqn$" to denote the DQN and double DQN respectively, and develop four variants of IMUP based on two sampling strategies: "IMUP$-r^{dqn}$","IMUP$-r^{ddqn}$", "IMUP$-TD^{dqn}$" and "IMUP$-TD^{ddqn}$".

Figure 9 and Figure 10 show the comparison results in terms of all evaluation metrics. We can find that $ddqn$ fluctuates slightly compared with dqn in terms of "Precision on Category", "Recall on Category", "Average Similarity"
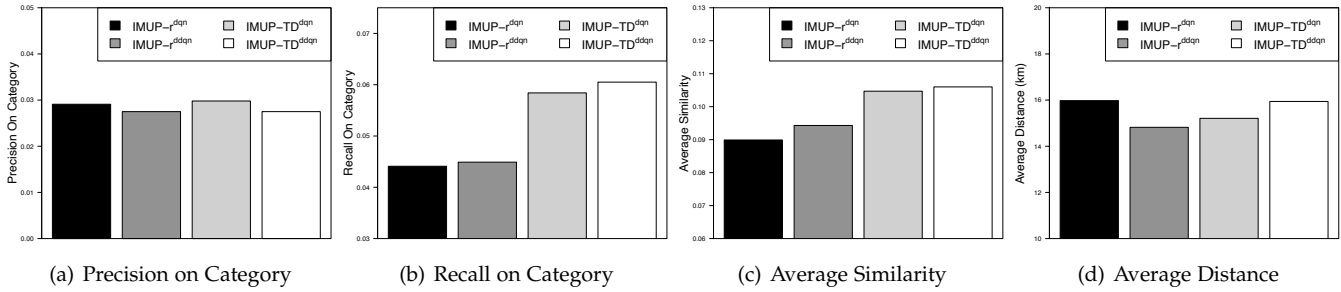
Fig. 9: The comparison between the original agent (dqn) and the enhanced agent ($ddqn$) w.r.t. New York dataset.
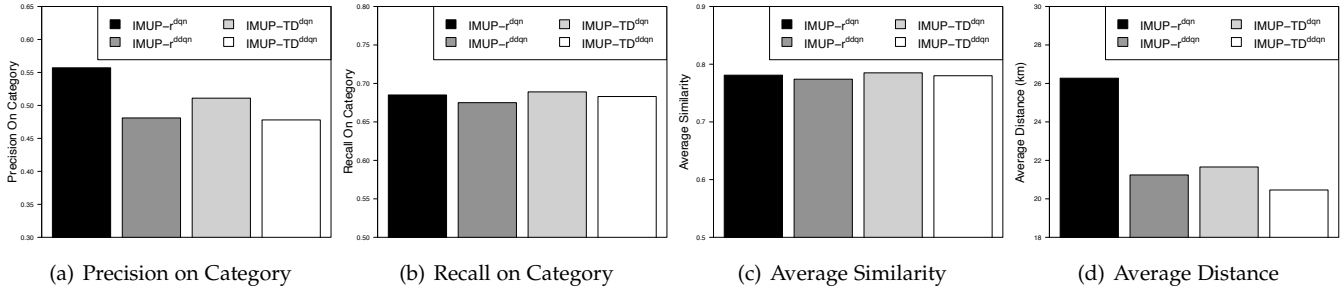

Fig. 10: The comparison between the original agent (dqn) and the enhanced agent ($ddqn$) w.r.t. Beijing dataset.
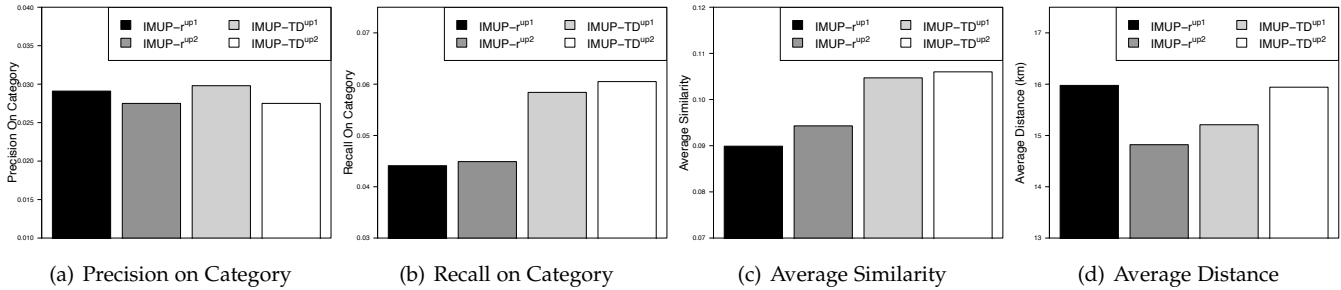

Fig. 11: The comparison between the original ($up1$) and enhanced ($up2$) state update strategies w.r.t. New York dataset.
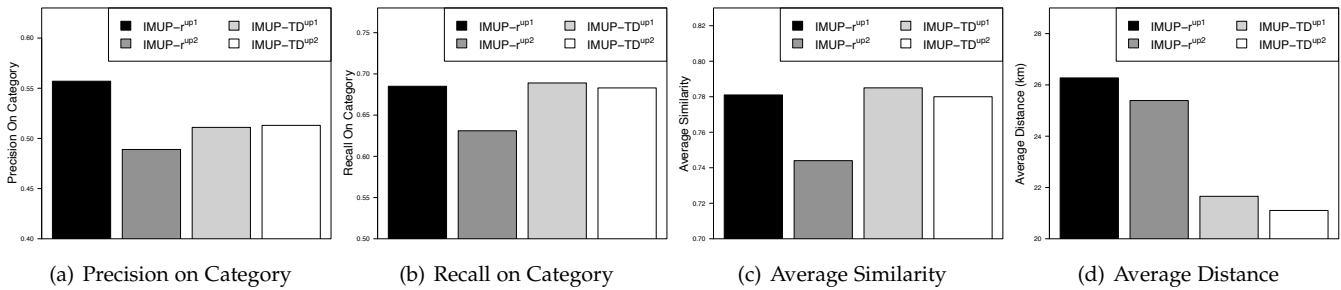

Fig. 12: The comparison between the original ($up1$) and enhanced ($up2$) state update strategies w.r.t. Beijing dataset.

and outperforms dqn in terms of "Average Distance" on both New York and Beijing datasets. A potential reason for this observation is that the double DQN alleviates the Q-value overestimation of DQN. So the double DQN can provide an objective and fair Q-value for each action, resulting in learning more effective action selection and action evaluation policies. Thus, the reinforced agent can imitate the user mobility behavior better than before.

5.8 Comparison of different state update strategies

We compare our newly proposed state update strategy (i.e., long-short term influence of interactions) with the one from the conference version (i.e., incremental update without differentiating the importance of interactions). Specifically, we use $up1$ and $up2$ to denote the state update strategy of the conference version and the

one newly proposed in this work respectively. Based on two sampling strategies, we further develop four variants of IMUP, namely "IMUP$-r^{up1}$","IMUP$-r^{up2}$", "IMUP$-TD^{up1}$","IMUP$-TD^{up2}$".

Figure 11 and Figure 12 show the comparison results in terms of "Precision on Category", "Recall on Category", "Average Similarity", and "Average Distance". We can observe that $up2$ is better than $up1$ in terms of majority evaluation metrics and fluctuates in a few ones on both New York and Beijing datasets. A potential reason is that the long-short term influenced based state update strategy can discard old and ambiguous user visit preferences and capture the newest user interests in time. As a result, the state update strategy of this work can help the reinforced agent imitate user mobility behavior quickly and perfectly.
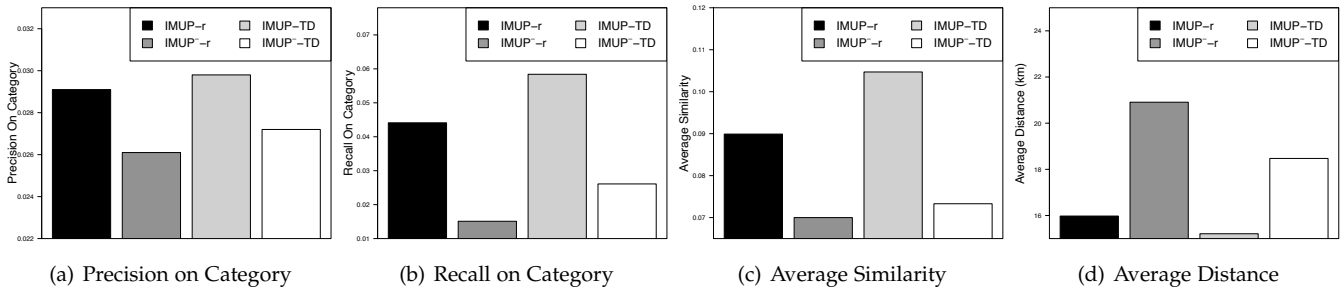
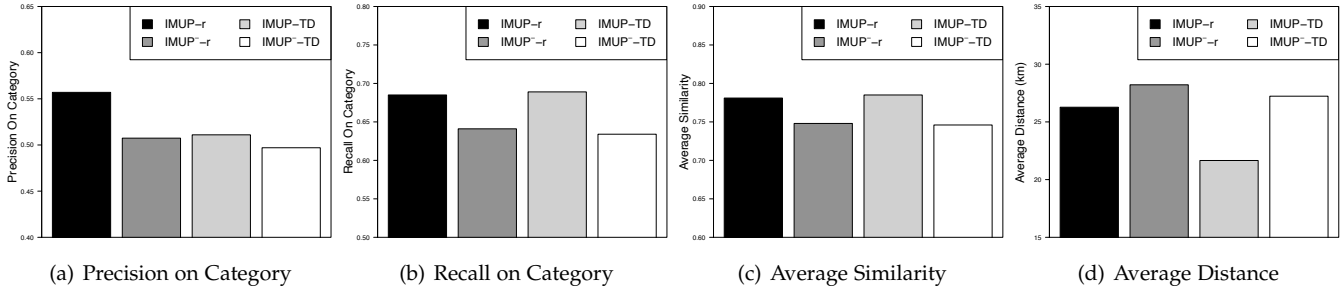Fig. 13: Analysis of spatial *KG w.r.t.* New York dataset.

Fig. 14: Analysis of spatial *KG w.r.t.* Beijing dataset.

TABLE 4: Comparison of different state initialization methods *w.r.t.* New York dataset

	StructRL	GAE	VGAE	ARGA	ARGVA
Prec_CAT	0.02216	0.02859	0.02955	0.02628	0.02902
Rec_CAT	0.04942	0.04773	0.05465	0.05461	0.05364
Avg_Sim	0.09892	0.08328	0.1039	0.1039	0.09131
Avg_Dist	14.8151	13.2210	15.2624	16.7266	14.3054

TABLE 5: Comparison of different state initialization methods *w.r.t.* Beijing dataset

	StructRL	GAE	VGAE	ARGA	ARGVA
Prec_CAT	0.48019	0.47945	0.49185	0.49783	0.47597
Rec_CAT	0.69242	0.69242	0.67542	0.66461	0.66615
Avg_Sim	0.78636	0.78648	0.77479	0.76637	0.76746
Avg_Dist	20.7080	21.2318	29.8552	30.1306	22.9128

5.9 Comparison of different state initialization

We adopt StructRL to convert user mobility graphs to representations, which are used to initialize the user state of our framework. To study the influence of state initialization strategies on user profiling, we conduct the comparative experiment based on the journal version framework (IMUP*) by replacing the StructRL with GAE [8], VGAE [8], ARGA [17], and ARGVA [17] respectively. Table 4 and Table 5 show the comparison results in terms of "Precision on Category", "Recall on Category", "Average Distance", and "Average Similarity". We can find that the next-visit prediction performance fluctuates slightly under different state initialization strategies. This demonstrates the robustness of IMUP* against distinct state initialization methods. A potential reason is that the state update strategy is able to discard the old and useless information and capture the new and significant characteristics of user mobility behaviors in time. As the progress of the learning process, the information in the initial state will be replaced with new user mobility characteristics, resulting in the robustness of IMUP*.

5.10 Analysis of spatial *KG*

We evaluate the spatial *KG*'s contribution on modeling user representations. To set the control group, we develop a vari-

ant of the proposed "IMUP", namely "IMUP$^-$". "IMUP$^-$" takes only the user and POI representations as the environment state, while other components of remains the same. The POI state (representations) is randomly initialized and state update is still based on the mutual interactions, but Equation 9 and Equation 11 are omitted.

Figure 13 and Figure 14 show the comparison results. We can observe that the performance of "IMUP" outperforms "IMUP$^-$" in terms of the four metrics over both two datasets. The results validate that the integration of semantics from spatial *KG* indeed enhances the modeling of user preferences on visit event. In the meantime, the incrementally updated next-visit planner (agent) is forced to bring in semantics from spatial *KG* to mimic the personalized user patterns, paced with the incrementally updated user representations (state).

5.11 Analysis of rewards

There are two types of rewards in the experiment: $r1$ and $r2$. Each reward function includes three parts: (1) distance r_d, (2) category similarity r_c, and (3) POI difference between the predicted and real user visit events r_p, where contributions are controlled by three weights λ_d, λ_c, and λ_p respectively. To analyze the contribution of these three factors, given the learning rate=$1e-5$, we project the results $(\lambda_d, \lambda_c, \lambda_p, \text{metric})$ into the 3D space, where the x, y, z axis corresponds to each factor respectively. We assign the gradient color to each point such that the better the performance over the metric, the darker the color is assigned.

Figure 15 and Figure 16 show the reward analysis of $r1$. An interesting observation is that in the case of "Precision on Category", "Recall on Category" and "Average Similarity", the contribution of category similarity r_c is higher than the other two factors, but in the case of "Average Distance", the contributions of distance r_d and POI r_p surpass the category similarity r_c. The reason is quite intuitive that the category similarity r_c directly determines the direction of policy training towards exploring more similar POI categories, while
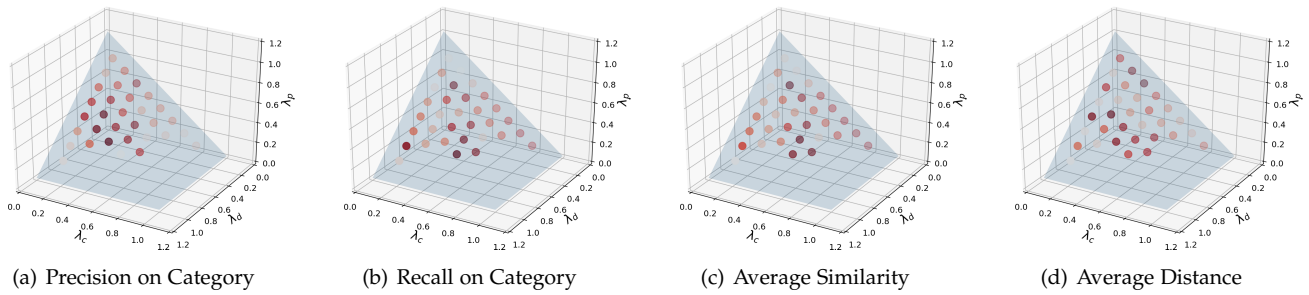
(a) Precision on Category   (b) Recall on Category   (c) Average Similarity   (d) Average Distance

Fig. 15: Reward analysis of $r1$ *w.r.t.* New York dataset.

(a) Precision on Category   (b) Recall on Category   (c) Average Similarity   (d) Average Distance
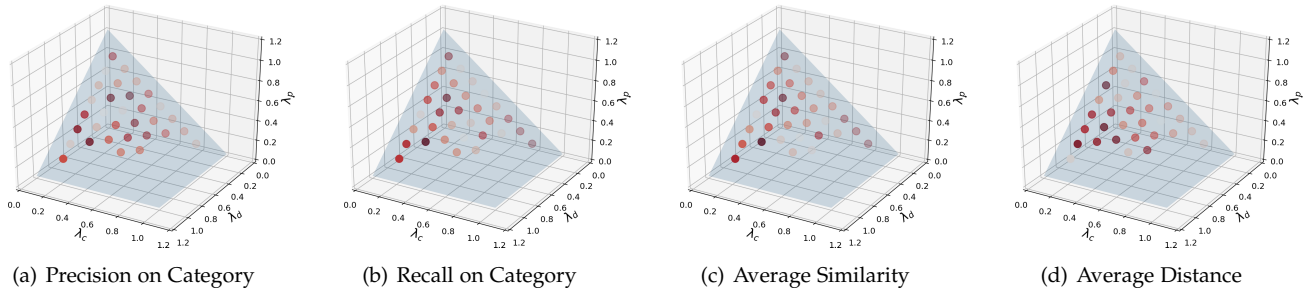
Fig. 16: Reward analysis of $r1$ *w.r.t.* Beijing dataset.

(a) Precision on Category   (b) Recall on Category   (c) Average Similarity   (d) Average Distance
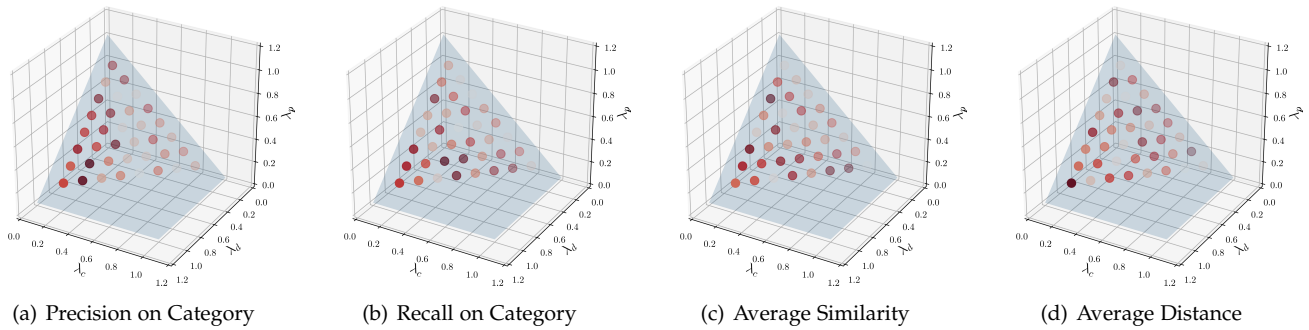
Fig. 17: Reward analysis of $r2$ *w.r.t.* New York dataset.

(a) Precision on Category   (b) Recall on Category   (c) Average Similarity   (d) Average Distance
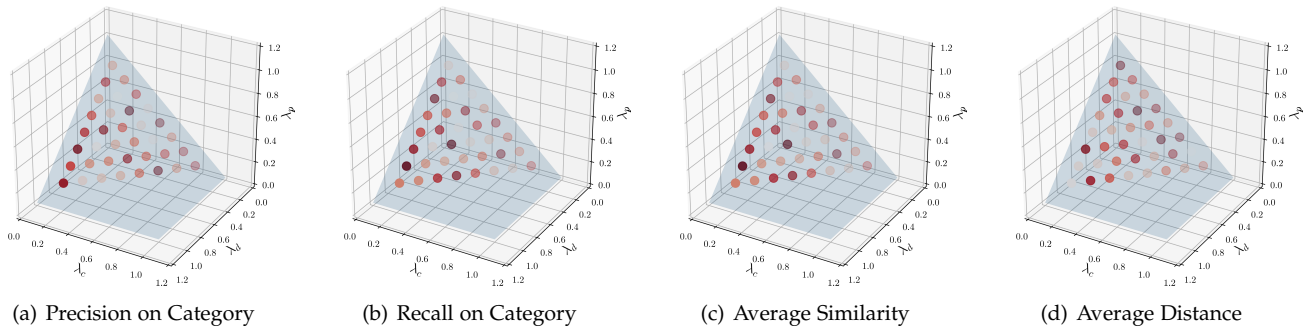
Fig. 18: Reward analysis of $r2$ *w.r.t.* Beijing dataset.

distance r_d and POI r_p guides the policy to find POIs as close as possible to the users' intention.

A careful inspection of Figure 15 and Figure 16 suggests that although the performance over certain metric is highly related to specific factors (*e.g.*, the category similarity r_c is highly related to category-related performance), the best results are not achieved at the extreme case such that some factors are pushed to zeros. On the contrary, the best results are achieved at the balance of these three factors, which may reveal some dependencies among POI locations and categories introduced by the spatial *KG*.

Figure 17 and Figure 18 show the result of reward analysis for the reward function $r2$. An interesting observation is that compared with $r1$, the dark points all shift to the left side for all evaluation metrics. The observation means that in order to obtain better predictions, we need to increase the contribution of r_d by increasing the value of λ_d. A potential reason is after subtracting baseline expectations, the value of r_d is lower than r_c and r_p. As a result, the agent considers fewer geographical factors of user mobility habits, resulting in poor next-visit prediction performance. To accurately simulate user mobility, we must consider the

geographical and semantic characteristics of user mobility completely. Thus, we can improve the contribution of r_d by increasing the value of λ_d.

6 RELATED WORK

Mobile User Profiling. Our work is connected to Mobile user profiling. User profiling refers to quantifying users' characteristics [21]. User profiling methods can be categorized into two groups: (1) explicit extraction, in which user profiles are explicitly predefined on documents, and (2) learning-based approach, which focus on learning user representations from users' historical behavior data [20], [21]. Our work is especially related to the learning-based approach for mobile users. For example, factorization-based approaches are exploited to model the integration of geographical and temporal influences of human mobility behaviors [5], [10]; deep learning-based approaches are then proposed to learn latent representations of users by leveraging the power of deep neural networks [27], [29]; more advanced techniques (*e.g.*, adversarial learning) are further introduced with emphasis on substructures of user mobility patterns [21], [22].

Reinforcement Learning for Online User Modeling. Our work is related to reinforcement learning for online user modeling. Reinforcement learning-based algorithms model online user behavior (*e.g.*, clicks, reviews, purchases) by regarding the online behavior as a sequential decision-making process [19], [25], [30]. For example, Zhao *et al.* propose the model "DEERS" to model both the positive and negative feedback through learning the sequential interactions via the reinforcement learning, with recommending trial-and-error items and receiving reinforcements of these items from users' feedback [31]. Zheng *et al.* propose a reinforcement learning framework to provide personalized new recommendation by considering user return pattern to capture implicit user feedback and provide new news by effective exploring strategies [32]. Chen *et al.* propose to utilize Markov Decision Process to model the sequential interactions between users and online recommender systems, and employing reinforcement learning to force an optimal policy for generating recommendation [2]. Isshu *et al.* exploits the multi-armed bandit approach to solve the cold-start problem based on the rewards from testing each options [15].

7 CONCLUSION REMARKS

In this paper, we propose a new learning criteria, namely reinforced imitative graph learning (RIGL), for mobile user profiling. Different from the minimization of traditional reconstruction loss or prediction loss, the proposed RIGL leverages an agent to mimic users' patterns and preferences. The optimal user profiles will be obtained as the agent can perfectly imitate users' decisions. In such setting, we integrated spatial *KG* to reinforcement learning to incrementally learn user representations and generate the next-visit prediction. Specifically, we formulated the state as the combination of users and spatial *KG*, where the mutual interactions were modeled to incrementally update the representations of users and spatial *KG* based on the temporal context, by

considering long-short term influences of interactions. The newly developed policy network exploits Double DQN to avoid the problem of overestimating user decisions. The policy aimed to mimic users' patterns to generate accurate next-visit prediction with the newly designed hierarchical *DQN* and improved sampling strategy. From the experiment, we can observe that the semantics introduced by the spatial *KG* improve the modeling of user representations for better understanding user patterns and preferences. This demonstrates that the proposed priority-based sampling strategy enhances the learning procedure, where *TD*-based is better than reward-based priority approach.

ACKNOWLEDGMENTS

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[2] Sungwoon Choi, Heonseok Ha, Uiwon Hwang, Chanju Kim, Jung-Woo Ha, and Sungroh Yoon. Reinforcement learning based recommender system using biclustering technique. *arXiv preprint arXiv:1801.05532*, 2018.

[3] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[4] Daniela Godoy and Analia Amandi. User profiling in personal information agents: a survey. *The Knowledge Engineering Review*, 20(4):329–361, 2005.

[5] Jean-Benoit Griesner, Talel Abdessalem, and Hubert Naacke. Poi recommendation: Towards fused matrix factorization with geographical and temporal influences. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 301–304, 2015.

[6] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[7] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, 2015.

[8] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[9] Maciej Kula. Spotlight. https://github.com/maciejkula/spotlight, 2017.

[10] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 831–840, 2014.

[11] Long-Ji Lin. Reinforcement learning for robots using neural networks. Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1993.

[12] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[13] Kunpeng Liu, Pengyang Wang, Jiawei Zhang, Yanjie Fu, and Sajal K Das. Modeling the interaction coupling of multi-view spatiotemporal contexts for destination prediction. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 171–179. SIAM, 2018.

[14] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

[15] Isshu Munemasa, Yuta Tomomatsu, Kunioki Hayashi, and Tomohiro Takagi. Deep reinforcement learning for recommender systems. In *2018 international conference on information and communications technology (icoiact)*, pages 226–233. IEEE, 2018.

[16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[17] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2609–2615, 2018.

[18] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[19] Dongjie Wang, Kunpeng Liu, Hui Xiong, and Yanjie Fu. Online poi recommendation: Learning dynamic geo-human interactions in streams. *arXiv preprint arXiv:2201.10983*, 2022.

[20] Dongjie Wang, Pengyang Wang, Kunpeng Liu, Yuanchun Zhou, Charles E Hughes, and Yanjie Fu. Reinforced imitative graph representation learning for mobile user profiling: An adversarial training perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4410–4417, 2021.

[21] Pengyang Wang, Yanjie Fu, Hui Xiong, and Xiaolin Li. Adversarial substructured representation learning for mobile user profiling. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 130–138, 2019.

[22] Pengyang Wang, Yanjie Fu, Yuanchun Zhou, Kunpeng Liu, Xiaolin Li, and Kien Hua. Exploiting mutual information for substructure-aware graph representation learning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*. AAAI Press, 2020.

[23] Pengyang Wang, Kunpeng Liu, Lu Jiang, Xiaolin Li, and Yanjie Fu. Incremental mobile user profiling: Reinforcement learning with spatial knowledge graph for modeling event streams. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 853–861, 2020.

[24] Pengyang Wang, Jiawei Zhang, Guannan Liu, Yanjie Fu, and Charu Aggarwal. Ensemble-spotting: Ranking urban vibrancy via poi embedding with multi-view spatial graphs. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 351–359. SIAM, 2018.

[25] Xiting Wang, Kunpeng Liu, Dongjie Wang, Le Wu, Yanjie Fu, and Xing Xie. Multi-level recommendation reasoning over knowledge graphs with reinforcement learning. In *The Web Conference*, 2022.

[26] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*, 2014.

[27] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1245–1254, 2017.

[28] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):129–142, 2014.

[29] Hongzhi Yin, Weiqing Wang, Hao Wang, Ling Chen, and Xiaofang Zhou. Spatial-aware hierarchical collaborative deep learning for poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 29(11):2537–2551, 2017.

[30] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38, 2019.

[31] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1040–1048, 2018.

[32] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, pages 167–176, 2018.

Dongjie Wang received the BE degree from the Sichuan University, China, 2016, and the MS degree from Southwest Jiaotong University, China, 2017. He is currently working toward a Ph.D. degree at the University of Central Florida (UCF). His research interests include anomaly detection, generative adversarial network, reinforcement learning, and spatio-temporal data mining.

Yanjie Fu received his Ph.D. degree from Rutgers University in 2016, the B.E. degree from Univ. of Sci. and Tech. of China in 2008, and the M.E. degree from the Chinese Academy of Sciences in 2011. He is currently an Assistant Professor at the University of Central Florida. His general interests are data mining and big data analytics. He has published prolifically in refereed journals and conference proceedings, such as IEEE TKDE, ACM TKDD, IEEE TMC, ACM SIGKDD.

Pengyang Wang received the BE degree from the Xi'dian University, China, in 2014, the MS degree from the Beijing University of Posts and Telecommunications, China, 2017. He is currently a Ph.D. candidate at the University of Central Florida (UCF). His research interests include data mining, business analytics, geoeconomics, and representation learning.
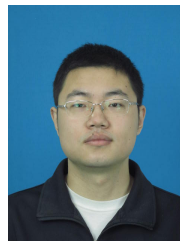
Kunpeng Liu is an Assistant Professor in the Department of Computer Science, at Portland State University. Prior to that, he received a Ph.D. degree from the CS Department of the University of Central Florida. He received both his M.E. degree and B.E. degree from the Department of Automation, University of Science and Technology of China (USTC). He has rich research experience in industry research labs, such as Geisinger Medical Research, Microsoft Research Asia, and Nokia Bell Labs.

Hui Xiong is currently a Full Professor and Vice Chair of the Management Science and Information Systems Department at the Rutgers, the State University of New Jersey. For his outstanding contributions to data mining and mobile computing, he was elected an ACM Distinguished Scientist in 2014 and an IEEE Fellow in 2020. His general area of research is data and knowledge engineering, with a focus on developing effective and efficient data analysis techniques for emerging data-intensive applications.

Charles E. Hughes is a Pegasus Professor at the University of Central Florida (UCF). Meanwhile, he is the co-leader of the Learning Sciences Cluster and the Center for Research in Education Simulation Technology at UCF. In addition, he is the contributing faculty at the Institute for Simulation & Training. His general research interest is virtual learning environments, human-centered computing, affective computing, theory of computation, and tree of life (tree manipulation and reasoning).