



Inferring Real Mobility in Presence of Fake Check-ins Data

QIANG GAO and HONGZHU FU, The Complex Laboratory of New Finance and Economics, Southwestern University of Finance and Economics, China

KUNPENG ZHANG, Robert H. Smith School of Business, University of Maryland, USA

GOCE TRAJCEVSKI and XU TENG, Iowa State University, USA

FAN ZHOU, University of Electronic Science and Technology of China, China

Understanding human mobility has become an important aspect of location-based services in tasks such as personalized recommendation and individual moving pattern recognition, enabled by the large volumes of data from geo-tagged social media (GTSM). Prior studies mainly focus on analyzing human historical footprints collected by GTSM and assuming the veracity of the data, which need not hold when some users are not willing to share their real footprints due to privacy concerns—thereby affecting reliability/authenticity. In this study, we address the problem of Inferring Real Mobility (IRMo) of users, from their unreliable historical traces. Tackling IRMo is a non-trivial task due to the: (1) sparsity of check-in data; (2) suspicious counterfeit check-in behaviors; and (3) unobserved dependencies in human trajectories. To address these issues, we develop a novel Graph-enhanced Attention model called **IRMoGA**, which attempts to capture underlying mobility patterns and check-in correlations by exploiting the unreliable spatio-temporal data. Specifically, we incorporate the attention mechanism (rather than solely relying on traditional recursive models) to understand the regularity of human mobility, while employing a graph neural network to understand the mutual interactions from human historical check-ins and leveraging prior knowledge to alleviate the inferring bias. Our experiments conducted on four real-world datasets demonstrate the superior performance of IRMoGA over several state-of-the-art baselines, e.g., up to 39.16% improvement regarding the Recall score on Foursquare.

CCS Concepts: • **Information systems** → **Location-based services**; **Geographic information systems**;

Additional Key Words and Phrases: Fake check-ins, POI graph, human mobility, graph learning, self-attention

ACM Reference format:

Qiang Gao, Hongzhu Fu, Kunpeng Zhang, Goce Trajcevski, Xu Teng, and Fan Zhou. 2024. Inferring Real Mobility in Presence of Fake Check-ins Data. *ACM Trans. Intell. Syst. Technol.* 15, 1, Article 12 (January 2024), 25 pages.

<https://doi.org/10.1145/3604941>

This work was supported by National Natural Science Foundation of China (Grants No. 62102326 and No. 62072077), Natural Science Foundation of Sichuan Province (Grants No. 2023NSFSC1411 and No. 2022NSFSC0505), the Key Research and Development Project of Sichuan Province (Grant No. 2022YFG0314), and National Science Foundation SWIFT (Grant No. 2030249).

Authors' addresses: Q. Gao and H. Fu, The Complex Laboratory of New Finance and Economics, Southwestern University of Finance and Economics, No. 555, Liutai Avenue, Wenjiang District, Chengdu, China, 611130; e-mails: qianggao@swufe.edu.cn, fuhongzhu@mail.swufe.edu.cn; K. Zhang, Robert H. Smith School of Business, University of Maryland, College Park, USA; e-mail: kpzhang@umd.edu; G. Trajcevski and X. Teng, Iowa State University, Iowa, USA; e-mails: {gocet25, xuteng}@iastate.edu; F. Zhou (Corresponding author), University of Electronic Science and Technology of China, No. 4, Section 2, N. Jianshe Road, Chenghua District, Chengdu, China, 610054; e-mail: fan.zhou@uestc.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2157-6904/2024/01-ART12 \$15.00

<https://doi.org/10.1145/3604941>

1 INTRODUCTION

The wide use of **geo-tagged social media (GTSM)** in platforms like Wechat, Facebook, and Twitter, has enabled a generation of huge volumes of location-aware contextual data (e.g., check-ins). This, in turn, provides unprecedented opportunities to understand human mobility and develop various attractive and valuable downstream services, e.g., next **Point of Interest (POI)** recommendation [14, 61], trajectory recovering [48], and social relationship inference [8, 47].

However, while important, understanding human mobility is also a challenging task: not only relying on advances of model design but also being affected by the *reliability* of check-in data. This is due to the fact that users who like to share their daily activities using GTSM may choose not to record their *real* visited POIs, for privacy concerns, or other personal reasons [13, 39, 49]. Thus, some users are either not willing to share these POIs with their friends (and strangers) at all, or they could choose (i.e., “relocate” to) a deceptive but reasonable POI to cover their current true location. For example, Bob who is a fitness enthusiast, may not want to disclose that his actual location is McDonald’s in Wanda Plaza, and will choose another POI in the same plaza when posting a tweet (for which one could even use available apps, such as <https://apps.apple.com/us/app/fake-checkin/id1044569156>). In such and similar scenarios, GTSM could fail to obtain the real daily trajectories of users like Bob, yielding uncertainty-based risks when capturing their mobility patterns or moving intentions, even enabling profiting via fraudulent behaviours [31].

Inspired by the recent studies in Human-Computer Interactions addressing users’ behavioral motivations behind posting fake check-ins [49], in this work, we investigate three types of strategies commonly used by people who attempt to hide their private or purpose-enabled movements, exemplified in Figure 1:

- **(b1)**—choosing a nearby POI (using the nearby POI *A* to substitute for the real visiting place). According to the survey study in Reference [49], there are usually about 16% of check-ins that are close to the real locations, barring the positioning errors. Privacy concerns are the main reason motivating people to choose a nearby POI provided by the GTSMs to replace their actual current location. As illustrated in References [56, 57], these fake locations/POIs are sometimes considered to be the uncertain or fuzzy check-ins in the *collective POIs*.
- **(b2)**—selecting a POI visited before (e.g., POI *B* that Bob has visited a couple times before). GTSMs may have a reward-offering mechanism (e.g., points, badges, and real-world rewards) that encourages users to report their locations with activities from their daily life [44]. For example, a user who uses Foursquare or Weibo is able to earn points after checking in the same venue multiple times [33].
- **(b3)**—picking a popular POI that has never been visited by the user (e.g., *C* is a POI that has been visited by others). To earn more attention from the followers, people may frequently check in some famous or popular places even though they have never been there. That is to say, a user may claim a certain location that is quite far (literature has reported cases even thousands of miles away) from his/her actual location [13, 59].

To tackle these three types of behaviors, we introduce a novel problem, which is **Inferring Real Mobility (IRMo)** of users from their historical traces. Solving the IRMo problem has various merits in multiple application domains—like, for example, offering more personalized and/or precise advertising recommendations by GTSM providers, accurately tracking people who have been infected by COVID-19 (to name a few)—based on a better understanding of human check-in behaviors and promoting the reliability of check-in data. Certainly, other fake check-in behaviors could exist in the real world, but the above three behaviors occur often in our daily life [49].

We note that IRMo is different from anomaly trajectory detection [26], as it aims to predict the real trajectory based on the check-in data containing various types of fake check-ins, whereas

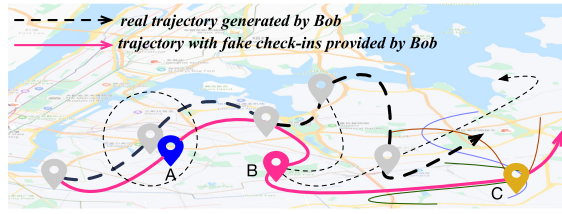


Fig. 1. User mobility behavior with fake check-ins.

the latter only discriminates whether a given trajectory is abnormal. That is to say, solving the IRMo problem not only requires distinguishing whether a POI in a given trajectory is real but also enables inferring the real POI if the given one is fake. The specific additional challenges for IRMo stem from three main sources: (S_1), the fake check-ins lead to deviation and difficulty in modeling spatial and temporal information; (S_2), the uncertainty of human individual trajectories containing fake check-ins hinders us from distilling the real transition regularities and distinguishing which check-in is fake; and (S_3), the unobserved dependencies in real trajectories during inference could make IRMo more complicated than traditional trajectory generation methods [22].

Intuitively, one common solution is to statistically model the mobility patterns from historical check-ins, leveraging Markov-based methods to filter out the real transition regularity [37]. For instance, the transition patterns modeled on fake moves are likely to deviate from the normal ones. To this end, we could leverage Markov chains to predict real check-ins by constructing a transition matrix from a massive amount of trajectories. However, such an approach is only able to handle simple mobility patterns, failing to capture complex mobility regularities. Inspired by the achievements of deep recursive models and end-to-end learning paradigms in various location-specific tasks (e.g., trajectory representation learning and sequential check-in modeling [26, 61]), another solution could be to rely on seq2seq models to capture long-term dependencies in human trajectories [40], which motivates our solution to IRMo. However, the seq2seq-based methods with **recurrent/convolutional neural networks (R/CNNs)** usually concentrate on capturing long-term sequential information (or local interactions) in a given sequence and are unable to simultaneously model both local and global correlations among POIs in a sequence. In addition, it is hard to recognize when the fake check-in will occur in the future, even if we have accumulated their massive historical check-ins—hence, it may not be desirable to infer their real motions by using the existing sequential learning methods directly (e.g., mobility prediction models [50, 61]).

To overcome the drawbacks of the intuitively appealing existing approaches, we judiciously tackle **IRMo with Graph-enhanced Attention (IRMoGA)**, for the sparse and complex GTSM data that possibly contains fake check-ins. More specifically, we first devise an appropriate spatial-temporal embedding module to explore the geographical and temporal information from historical check-ins, primarily seeking to mimic human potential fake check-in behavior, i.e., (**b1**). Next, we attempt to explore users' moving patterns and distill historical check-in behaviors from the personalized historical trajectory. Correspondingly, we propose a fake trajectory encoding component, which contains an attention-based movement extractor for mobility pattern learning and a graph-enhanced behavior interaction module to integrate personalized check-in preferences. Especially, the movement extractor is able to study both local and global correlations among check-ins while the graph-enhanced module aims at alleviating the influence of behavior (**b2**), as such an intention can hardly conceal human real transitional regularities, governed by their long-standing check-in habits. Finally, we use another attention-based procedure to derive unobserved dependencies. All throughout, a novel element for inverse POI popularity is considered to alleviate the inferring bias (i.e., **b3**). For the three types of fake check-in behaviors that we focus upon, our proposed IRMoGA

enables eliminating their impacts and predicts the real human movement corresponding to the true intended POI visits. Our main contributions can be summarized as follows:

- We investigate IRMo, a novel problem tackling real versus fake human mobility from sparse movement data, and we address three types of possible fake check-in behaviors.
- To our knowledge, we are among the first pioneers to introduce a novel model, namely, IRMoGA, to tackle IRMo. It uses an attention mechanism to explore human possible mobility patterns, and we incorporate a graph-based module to enhance the learning of mobility regularities and design a novel inverse POI popularity module to involve the external knowledge for alleviating inferring biases.
- We provide extensive experimental observations conducted on four real-world datasets, demonstrating that IRMoGA is efficient and outperforms several state-of-the-art methods.

In the rest of this article, we review the relevant studies in Section 2, and then we formalize the problem and present the background of self-attention and GNN in Section 3. The details of the proposed IRMoGA framework are discussed in Section 4, and the results of the experimental evaluations quantifying the benefits of our approach are presented in Section 5. Section 6 concludes the article and outlines the directions of future work.

2 RELATED WORK

We now provide a review of two global categories of related works—human mobility mining and the more recent trend of deep representation learning—and we position our results in that context.

2.1 Human Mobility Mining

Understanding human mobility is a primary task for responding to various downstream applications, including exploring the informative correlations/interactions from user historical check-ins, capturing long-term spatial-temporal dependencies and many others. Conventionally, **matrix factorization (MF)**-based methods have been leveraged to mine user-location or location-location correlations by decomposing a “rating” matrix into two low-rank matrices, where some implicit, personalized, and semantic feedback can be integrated, such as user preference, location preference, and geographical influence [3, 23, 24]. Rank-based methods combined with MF have also been developed to discover user behavior patterns and fit user’s preference rankings for POIs [7, 52]. Li et al. group the sequential tweets in user timelines and extend the Bayesian model to estimate the information gains between tweets and locations for the purpose of inferring the location of non-geotagged tweets [21]. Other methods like Bayesian model also can be considered to explore the human check-in behaviors, e.g., identifying the latent visited-POI label of stay points, which are automatically extracted from trajectories [34]. In addition, mining potential dependencies (e.g., spatial-temporal and semantic ones) among check-ins is important for understanding human transition patterns. For example, variants of Markov models have been developed to explore human spatial-temporal transition patterns [38, 45, 64]. Complementary to these, leveraging **hidden Markov models (HMM)** is even capable of capturing semantics-aware mobility [37].

However, the existing methods still fail to elaborate human real check-in behaviors in terms of the IRMo problem—i.e., they cannot cater to various fake check-ins in human trajectories that could lead to deviation in understanding human mobility—which is what we address.

2.2 Deep Representation Learning for POIs

Deep representation learning is one of the inspiring but intricate schemes, which aims at unifying associated information and prior knowledge into low-dimensional vectors to represent various entities. In particular, POI (or check-in) representation embedding is a preliminary but crucial

prerequisite for understanding human mobility. Most of the recent works either employ random initialization methods [4, 48] or word2vec-based techniques [10, 63] to prepare a set of dense representations for alleviating the “Curse of Dimensionality” issue. However, the first type is unable to incorporate any semantic information while the latter usually faces the extreme sparsity issue of human check-in data, which would bring the poor generalization capabilities of check-ins.

By contrast, we present a graph-enhanced spatial embedding method regarding POIs and introduce a time embedding method to explore the spatial-temporal information, which will not be constrained by the scale of collected trajectory data.

2.3 Deep Representation Learning for Trajectories

Trajectory representation embedding is also a fundamental topic studied in recent works. Usually, it should be able to model the long-dependencies in human historical movements. For instance, Yu et al. developed an LSTM-based model called CatDM that dynamically captures human long-short visiting interests behind historical check-ins as latent vectors, which also takes into account geographic influences [53]. In this spirit, we can easily consider applying R/CNN-based models to obtain a latent code for a provided trajectory [30, 63]. Complementary to this, recent attention mechanisms (e.g., multi-head self-attention) have become popular due to their success in modeling dependencies without regard to their distance in the input or output sequences, automatically extracting the most pertinent information for various tasks [2, 20, 42]. For instance, Zhao et al. propose a self-attention-based framework to couple sequential information for next item recommendation [60]. Also, there exist several successful cases to tackle trajectory-based problems such as POI prediction [30], trajectory recovery [48], and bike station-level flow forecasting [12]. All these motivate us to develop an attention-based model to tackle our IRMo problem. Furthermore, the advances of **graph neural networks (GNNs)**, e.g., graph convolutional networks [18] and graph attention networks [43], provide new opportunities for trajectory mining, by enabling the capture of potential interactions on graphs, e.g., leveraging GNN to capture and differentiate station-to-station correlations, modeling dynamic situational context and users’ sequential behaviors [55], and expressing the high-order connectivity among POIs [62].

Unlike various trajectory mining tasks that use either attention-based or graph-based methods, our proposed IRMoGA is among the first attempts for tackling the IRMo problem involving an attention mechanism to couple human sequential mobility while developing a graph-enhanced module to explore mutual interactions from historical check-ins.

3 PRELIMINARIES

We start this section by introducing the terminology used throughout the article and formalizing the IRMo problem. Subsequently, we briefly describe the paradigms of Attentive and Graph Neural Networks. Besides, we summarize the frequently used symbols and explain their meaning in Table 1.

3.1 Basic Definitions

We observe that in the previous works, often times the terms “POI” and “check-in” are used synonymously. In our work, we separate the two by assigning to each check-in not only the spatial attribute as POIs have but also the temporal attribute.

Definition 1. Let $\mathcal{P} = \{p_1, p_2, \dots, p_{|\mathcal{P}|}\}$ be a set of POIs. Each $p \in \mathcal{P}$ is associated with a spatial attribute $p.loc$, which is a pair $(p.lo, p.la)$ (i.e., (longitude, latitude)), or $(p.x, p.y)$ (i.e., (x, y) values in a suitable coordinate system).

Table 1. Frequently Used Notations

Notation	Description
p	a POI
$\mathcal{P}$	a set of POIs
u	user identity
$\mathcal{U}$	a set of users
c_u	a check-in left by user u
t	check-in time
T_u^n	n th trajectory of user u
RT_u^n	user u 's real trajectory
$\mathcal{T}_u$	collection of user u 's trajectories
$\mathbf{v}_s$	spatial embeddings
$\mathbf{v}_t$	temporal embeddings
$\mathbf{T}_u^n$	the dense tensor of T_u^n
$\mathbf{h}$	hidden state
$\mathcal{P}_u$	a set of POIs visited by user u
G_p	POI graph
G_u	user u 's mobility graph
$\overleftarrow{\mathbf{A}}/\overrightarrow{\mathbf{A}}$	outgoing/incoming adjacency matrix
$\overleftarrow{\mathbf{H}}, \overrightarrow{\mathbf{H}}$	hidden layer of GNN
d, d', d''	embedding size
Ψ	POI popularity list
$\mathbf{W}, \mathbf{b}$	trainable parameters
$\mathcal{L}_\theta$	loss function

A *check-in* c_u of a given user u is characterized by a triplet $c_u = \langle u, p, t \rangle$ indicating that a user u has visited the POI $p \in \mathcal{P}$ at time t .

When there is no ambiguity, in the sequel we will omit the “.” symbol and the respective prefixes (i.e., we will use la instead of $p.la$). We also note that if a POI p has a spatial extent, then we will assume that the coordinates of $p.loc$ will correspond to the centroid of that extent.

Definition 2. For a given user u , the *reported check-in trajectory* $T_u = [c_u^1, c_u^2, \dots, c_u^{|T_u|}]$ is a sequence of check-ins.

3.2 Problem Definition (IRMo)

Given the n th trajectory $T_u^n = [c_u^{n,1}, c_u^{n,2}, \dots, c_u^{n,m}]$ of a user u , IRMo aims to infer the real trajectory RT_u^n relative to the reported one T_u^n —since T_u^n may contain fake check-ins.

Let $T_u = \{c_u^1, c_u^2, c_u^3, \dots\}$ denote the collection of user u 's historical trajectories. Since we will investigate the impact of mobility periodicity from human historical traces, we will consider that T_u is segmented into κ sub-trajectories $T_u = \{T_u^1, T_u^2, \dots, T_u^n, \dots, T_u^\kappa\}$ with a time span t_γ (note that the time span may vary—e.g., 1 day or 6 h). When there is no ambiguity, we will simply use T_u^h to denote user u 's historical trajectory.

3.3 Attentive Networks

Attentive networks (e.g., self-attention) have been widely adopted in **natural language processing (NLP)** for sequential knowledge modeling, yielding a higher achievement than recently popular benchmarks in various text/document tasks [5, 11, 19]. For instance, self-attention as the core part of Transformer framework attempts to measure the similarity scores between a given query and a key, whereafter formulating the attentive weight for a value. The canonical self-attention [42]

performs the scaled dot-product as follows:

$$\mathcal{A}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\mathbf{Q}\mathbf{K}^\top / \sqrt{d})\mathbf{V}, \quad (1)$$

where query $\mathbf{Q} \in \mathbb{R}^{L_Q \times d}$, key $\mathbf{K} \in \mathbb{R}^{L_K \times d}$, value $\mathbf{V} \in \mathbb{R}^{L_V \times d}$, and d is the input dimension.

3.4 Graph Neural Network

Nowadays, there exists a popular trend to study the potential knowledge interactions from various graph-based data, such as social networks and knowledge bases. Accordingly, graph-based models (e.g., DeepWalk [36], LINE [41], etc.) have been widely used to learn the dense representation of graph nodes, where the local correlations among them are well extracted. The more complex deep neural networks such as **convolutional neural networks (CNNs)** also enable capturing the local features from graph-based data—however, they fail to handle the non-Euclidean correlation with simple convolutional operations [15]. Towards that end, the Graph Neural Networks have emerged—e.g., **graph convolutional networks (GCNs)** [18]—capable of addressing the said limitation.

Without loss of generality, let $G(\mathcal{V}, \mathcal{E}, \mathbf{X})$ denote a graph where $\mathcal{V}$ refers to a set of nodes (e.g., $|\mathcal{V}| = \eta$ graph nodes), $\mathcal{E}(\subseteq \mathcal{V} \times \mathcal{V})$ represents a set of edges, and $\mathbf{X}$ denotes the associated node feature. As an illustrative example, the popular GCN aims at learning a function f on graph G with a message-passing process. Hence, any k th GCN layer can be written as follows:

$$\mathbf{H}^{(k+1)} = f(\mathbf{H}^{(k)}, \mathbf{A} | \mathbf{W}^{(k)}), \quad (2)$$

where $\mathbf{A}$ refers to the adjacency matrix extracted from $\mathcal{E}$, and $\mathbf{H}^{(k+1)}$ means the hidden feature matrix of $(k + 1)$ th layer where $\mathbf{H}^{(0)} = \mathbf{X}$. Specifically, GCN defines a spectral formulation in Fourier domain for the inherent convolution operator on graph learning, whereas Equation (2) can be rewritten as

$$f(\mathbf{H}^{(k)}, \mathbf{A} | \mathbf{W}^{(k)}) = \varphi(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(k)} \mathbf{W}^{(k)}), \quad (3)$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\mathbf{I}$ is the identity matrix of $\mathbf{A}$; $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$; $\mathbf{W}^{(k)}$ is a layer-specific trainable matrix; φ denotes an activation function such as Sigmoid or ReLu. Note that we usually use a one-hot embedding to represent its feature space if each node has no additional feature and thus this work uses $\mathbf{I}$ to replace $\mathbf{X}$, i.e., $\mathbf{X} = \mathbf{I} \in \mathbb{R}^{\eta \times \eta}$.

4 METHODOLOGY

We now describe the specifics of IRMoGA. We first provide an overview on the proposed framework and then follow with a detailed discussion of each component. Subsequently, we present the training algorithm along with the complexity analysis.

4.1 Overview

Our proposed framework consists of three major components: *spatial-temporal embedding*, *fake trajectory encoding* and *real trajectory inference*. As shown in Figure 2, we first use the *spatial-temporal embedding* to, respectively, obtain the spatial embedding and the temporal embedding based on the collected check-in data, where the transition relation is not used for the purpose of excluding the effect of fake check-ins. Then, our *fake trajectory encoding* attempts to exploit the self-attention mechanism to capture the dynamic dependencies among the check-ins, where an Information Fusion module is devised to smoothly yield the spatial-temporal semantics behind the check-ins. We then use a meticulously designed novel attention-based graph neural network, introduced to distill the individual check-in preferences from their historical trajectories for the purpose of learning the mutual interaction between historical and recent traces. In particular, we

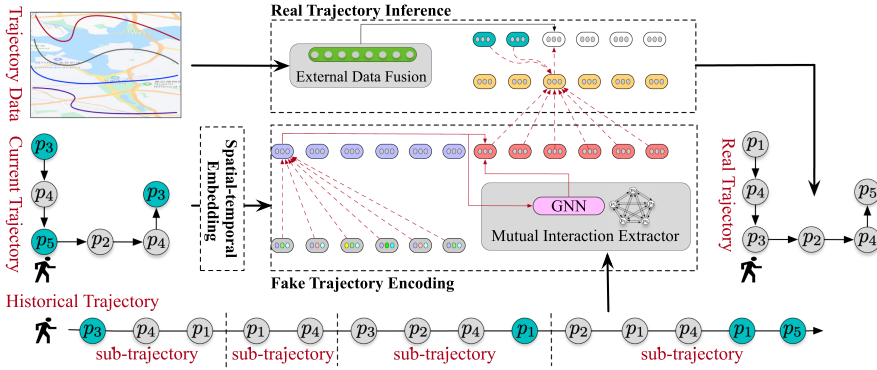


Fig. 2. Overall framework of IRMoGA.

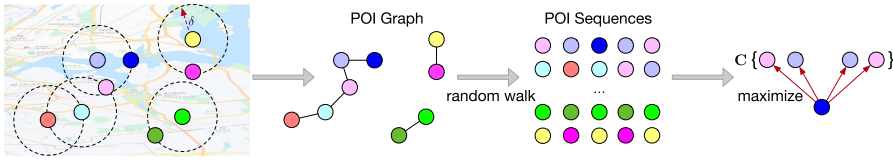


Fig. 3. Illustration of Graph-based Spatial Embedding.

fuse such mutual interactions as the input of our third module, the *real trajectory inference*. It also has an attention-based component in which we bridge an inverse POI popularity to extract other users' check-in preferences to mitigate the issue of inference bias. In the sequel, we describe these components in detail.

4.2 Spatial-temporal Embedding

Obtaining informative representations of check-ins is an initial (and crucial) prerequisite for understanding human mobility, which could potentially boost the performance of trajectory representation. Toward that, we introduce a graph-based spatial embedding method regarding POIs extracted from check-ins, which intends to incorporate geographical proximity among POIs for the purpose of addressing behavior (**b1**). And we also present a temporal embedding method to cater to our proposed IRMoGA. Note that we do not use any transition information from the given check-in data—i.e., fake behaviors will not affect the geographical proximity distillation.

4.2.1 Graph-based Spatial Embedding. In an effort to explore the geographical information among POIs into dense representations and to address the sparsity concern of check-ins, we construct a POI graph. Specifically, we first set each POI p ($p \in \mathcal{P}$) as a node in our graph $G_p(\mathcal{P}, \mathcal{E}_p)$. For the edges, we collect each POI's geographical neighbors that are within a given distance threshold δ (in this study, we adopt great circle distance to measure the distance between two POIs, and following existing works [9], the threshold δ is set to 1 km for all datasets). If the distance between POI p_i and POI p_j is $\leq \delta$, then we create an undirected edge $e \in \mathcal{E}_p$ connecting them. Inspired by previous works [36, 46, 58, 63], we obtain the dense representation of each POI by generating massive POI sequences based on random walks over G_p , which is capable of exploring spatial proximity among POIs. As shown in Figure 3, we first use the random walk strategy to generate massive POI sequences with a fixed length (e.g., 40). Next, we regard these sequences as the POI

corpus and embed each POI p by maximizing the probability of its occurrence in a surrounding POI context. For instance, given a POI sequence: $p_1, p_2, \dots, p_\tau, \dots$, we have a surrounding context $C(p_\tau) = p_{\tau-w} : p_{\tau+w}$ where w is the fixed size of the sliding window with respect to p_τ . Since such a surrounding context $C(p_\tau)$ is a set of geographical neighbors of p_τ , thereby we can preserve the information of geographical proximity among them by maximizing the following probability:

$$\begin{aligned} P(C(p_\tau)|p_\tau) &= \prod_{\substack{p' \in C(p_\tau) \\ p' \neq p_\tau}} P(p'|p_\tau) \\ &= \prod_{\substack{p' \in C(p_\tau) \\ p' \neq p_\tau}} \frac{\exp(\mathbf{v}_s(p') \cdot \mathbf{v}_s(p_\tau))}{\sum_{p'' \in \mathcal{P}} \exp(\mathbf{v}_s(p'') \cdot \mathbf{v}_s(p_\tau))}, \end{aligned} \quad (4)$$

where $\mathbf{v}_s$ denotes a set of learnable parameters. Since enumerating each p'' in $\mathcal{P}$ is computationally inefficient, we adopt the popular Negative Sampling technique [32] in the above optimization, which is similar to the popular word embedding toolkit word2vec [32]. Finally, each POI p is embedded into a low dimensional vector $\mathbf{v}_s(p) \in \mathbb{R}^d$, where d is the dimensionality in the lower space, and the geographical similarities among POIs are well incorporated.

4.2.2 Temporal Embedding. To capture temporal dependencies in user mobility, we leverage the positional embedding method [42] and encode the temporal information of each check-in into a low-dimensional representation. In practice, for a check-in c_u^τ , we generate its corresponding temporal embedding as

$$\begin{cases} \mathbf{v}_t(\tau_{2i}) = \sin(\tau/10,000^{2i/d'}), \\ \mathbf{v}_t(\tau_{2i+1}) = \cos(\tau/10,000^{2i/d'}), \end{cases} \quad (5)$$

where i denotes the i th dimension. And d' refers to the dimensionality of temporal embedding.

4.3 Fake Trajectory Encoding

Given the existence of fake check-ins in human trajectories, an attention mechanism becomes a natural choice to deal with correlations among check-ins by weighing distinct attention to real versus fake check-ins.

4.3.1 Information Fusion. Given $T_u^n = [c_u^{n,1} c_u^{n,2} \dots c_u^{n,m}]$, we fuse the spatial and temporal embedding of each check-in into a single one with a linear transformation. For instance, given a check-in $c_u^{n,\tau}$ in T_u^n , the fused embedding becomes

$$\mathbf{v}_u^{n,\tau} = (\mathbf{v}_s(c_u^{n,\tau})\mathbf{W}_v + \mathbf{b}_v) + \mathbf{v}_t(c_u^{n,\tau}), \quad (6)$$

where $\mathbf{W}_v \in \mathbb{R}^{d \times d'}$ and $\mathbf{b}_v \in \mathbb{R}^{d'}$ are the learnable parameters. Finally, we can obtain

$$\mathbf{T}_u^n = [\mathbf{v}_u^{n,1}, \mathbf{v}_u^{n,2}, \dots, \mathbf{v}_u^{n,m}]^\top, \quad (7)$$

where $\mathbf{T}_u^n \in \mathbb{R}^{m \times d'}$.

4.3.2 Movement Extractor. We now use a multi-head attention mechanism to encode the dynamic dependencies among check-ins. For $\mathbf{v}_u^{n,\tau}$ in $\mathbf{T}_u^n$, we use a self-attention to model its mutual interactions with other check-ins in $\mathbf{T}_u^n$. Assume that there are $\mathcal{H}$ heads of self-attention. We take the head $\pi(\pi \in \mathcal{H})$ as the example to present the workflow of such interactions. According to the principle of self-attention [42], we prepare three learnable matrices $\mathbf{W}_Q^\pi$, $\mathbf{W}_K^\pi$ and $\mathbf{W}_V^\pi$ to formulate

the query $\mathbf{Q}^\pi$, key $\mathbf{K}^\pi$ and value $\mathbf{V}^\pi$ as follows:

$$\mathbf{Q}^\pi(\mathbf{v}_u^{n,\tau}) = \mathbf{v}_u^\tau \mathbf{W}_Q^\pi, \quad (8)$$

$$\mathbf{K}^\pi(\mathbf{T}_u^n) = \mathbf{T}_u^n \mathbf{W}_K^\pi, \quad (9)$$

$$\mathbf{V}^\pi(\mathbf{T}_u^n) = \mathbf{T}_u^n \mathbf{W}_V^\pi. \quad (10)$$

We use $\mathbf{Q}^\pi$, $\mathbf{K}^\pi$ and $\mathbf{V}^\pi$ ($\in \mathbb{R}^{d' \times d'}$) to obtain m scores estimating the contribution of each check-in in $\mathbf{T}_u^n$, whereafter an aggregation operation is leveraged to formulate the latent space of $\mathbf{v}_u^{n,\tau}$, denoted by

$$\mathbf{h}_u^{n,\tau(\pi)} = \text{softmax} \left(\frac{\mathbf{Q}^\pi(\mathbf{v}_u^{n,\tau}) \mathbf{K}^\pi(\mathbf{T}_u^n)^\top}{\sqrt{d'}} \right) \mathbf{V}^\pi(\mathbf{T}_u^n). \quad (11)$$

Herein, $\frac{1}{\sqrt{d'}}$ is the scaling factor that enables preventing excessive inner product values. Since there are $\mathcal{H}$ heads, we use a concatenation operator (denoted as “||”) to aggregate all different views (heads) regarding the interactions by

$$\mathbf{h}_u^{n,\tau} = (\mathbf{h}_u^{n,\tau(1)} || \mathbf{h}_u^{n,\tau(2)} || \dots || \mathbf{h}_u^{n,\tau(\mathcal{H})}) \mathbf{W}_o + \mathbf{b}_o, \quad (12)$$

where $\mathbf{W}_o \in \mathbb{R}^{\mathcal{H}d' \times d'}$ and $\mathbf{b}_o \in \mathbb{R}^{d'}$ are learnable parameters.

In addition, we use a residual network shown below to prune the network connections, which allows us to leverage more promising residual information instead of the original input:

$$\mathbf{h}_u^{n,\tau} = \text{LN}(\mathbf{v}_u^{n,\tau} + \text{DR}(\mathbf{h}_u^{n,\tau})), \quad (13)$$

$$\mathbf{h}_u^{n,\tau} = \text{LN}(\mathbf{h}_u^{n,\tau} + \text{DR}(\text{ReLU}(\mathbf{h}_u^{n,\tau} \mathbf{W}_{r_1} + b_{r_1}) \mathbf{W}_{r_2} + b_{r_2})), \quad (14)$$

where $\mathbf{W}_{r_1} \in \mathbb{R}^{d' \times d''}$, $b_{r_1} \in \mathbb{R}^{d''}$, $\mathbf{W}_{r_2} \in \mathbb{R}^{d'' \times d'}$, and $b_{r_2} \in \mathbb{R}^{d'}$ are learnable parameters. Herein, $\text{ReLU} = \max(0, z)$ is the popular activation function for non-linear fitting, LN refers to the Layer Normalization operation for the fast and stable training [1], and DR denotes the dropout operation for the alleviating overfitting problem. Notably, we can use the above way (Equations (8)–(14)) to obtain each hidden space corresponding to the check-in in $\mathbf{T}_u^n$, i.e., $\{\mathbf{h}_u^{n,1}, \dots, \mathbf{h}_u^{n,\tau}, \dots, \mathbf{h}_u^{n,m}\}$. For simplicity, we describe the parallel process of *Movement Extractor* as

$$\begin{bmatrix} \mathbf{h}_u^{n,1} \\ \mathbf{h}_u^{n,2} \\ \vdots \\ \mathbf{h}_u^{n,m} \end{bmatrix} = \text{ME} \left(\begin{bmatrix} \mathbf{v}_u^{n,1} \\ \mathbf{v}_u^{n,2} \\ \vdots \\ \mathbf{v}_u^{n,m} \end{bmatrix}, \begin{bmatrix} \mathbf{v}_u^{n,1} \\ \mathbf{v}_u^{n,2} \\ \vdots \\ \mathbf{v}_u^{n,m} \end{bmatrix}, \begin{bmatrix} \mathbf{v}_u^{n,1} \\ \mathbf{v}_u^{n,2} \\ \vdots \\ \mathbf{v}_u^{n,m} \end{bmatrix} \right), \quad (15)$$

where the three inputs of ME are the same, and we use them to formulate the query, key, and value, respectively.

4.3.3 Mutual Interaction Extractor (MIE). According to the early efforts [10, 30], human historical trajectories T_u^h exhibit periodicity. Thus, we argue that human recent mobility would be affected by their historical check-in habits, which motivates us to incorporate their historical check-ins from up to T_u^n . However, users’ historical trajectories are usually extremely longer sequences compared to short recent trajectories, i.e., $|T_u^n| \ll |T_u^h|$. Besides, users’ movement varies from each other due to their different lifestyles. We can obtain each user’s personalized movement habits from his/her long-term historical check-ins, which can further help us expose real transitional patterns among POIs. Previous works usually leverage an R/CNN-based module to learn sequential behavior of user check-ins while an attention mechanism is used to capture user’s personal

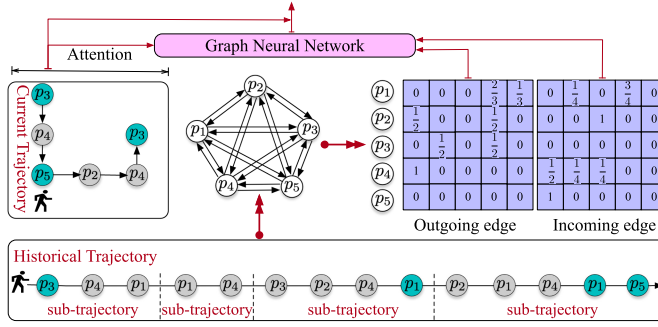


Fig. 4. Mutual interaction extractor (MIE).

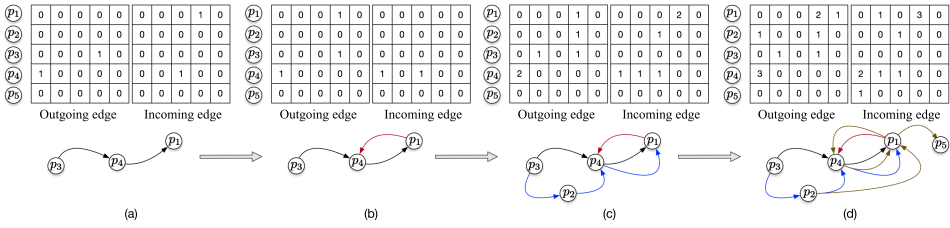


Fig. 5. General illustration of outgoing/incoming adjacency matrix construction.

periodicity from their historical trajectories [30]. However, they are unable to consider the higher-order connections among the check-ins from historical trajectories, possibly leading to the failure of eliminating the influence of fake check-ins such as the behavior (b2). Hence, we design a coupling graph neural network to distill mutual interactions between T_u^n and T_u^h by a simple attention mechanism.

Specifically, we first construct a personalized mobility graph G_u to capture transition regularities from T_u^h , as shown in Figure 4. We regard the POI set P_u extracted from T_u^h as the node set of G_u . Since we focus on distilling the transition patterns in historical trajectories, we define an outgoing adjacency matrix $\vec{A}_u^h = \{\vec{a}_{i,j} | i, j \in \{1, 2, \dots, |P_u|\}\}$ and an incoming adjacency matrix $\hat{A}_u^h = \{\hat{a}_{i,j} | i, j \in \{1, 2, \dots, |P_u|\}\}$, which, respectively, refers to the weighted connections of outgoing edges and incoming edges. Figure 5 presents a simple illustration of the generation of these adjacency matrices. Each value of the respective cell refers to the statistical frequency, and then we calculate the corresponding ratio value of each cell. More specifically, $\vec{a}_{i,j}$ indicates the outgoing ratio from POI p_i to POI p_j and $\hat{a}_{i,j}$ indicates the incoming ratio from POI p_j to POI p_i . As for the mutual interactions between a trajectory T_u^n and the personalized mobility graph G_u —up to now, we have obtained the hidden representations $\{h_u^{n,1}, \dots, h_u^{n,\tau}, \dots, h_u^{n,m}\}$ corresponding to T_u^n . We bridge the interaction between each hidden state (e.g., $h_u^{n,\tau}$) and G_u by a fused GNN as follows:

$$\vec{H}_u^h = \vec{A}_u^h [v_u^{n,1}, \dots, v_u^{n,m}]^\top W_{g_1} + b_{g_1}, \quad (16)$$

$$\hat{H}_u^h = \hat{A}_u^h [v_u^{n,1}, \dots, v_u^{n,m}]^\top W_{g_2} + b_{g_2}, \quad (17)$$

$$\hat{H}_u^h = (ReLU(\vec{H}_u^h) W_{\vec{g}} + b_{\vec{g}}) || (ReLU(\hat{H}_u^h) W_{\hat{g}} + b_{\hat{g}}), \quad (18)$$

where $\mathbf{W}_* \in \mathbb{R}^{d' \times d'}$ and $\mathbf{b}_* \in \mathbb{R}^{d'}$ are learnable parameters. Next, we present a variant of Bahdanau Attention [2] to aggregate the graph information from $\hat{\mathbf{H}}_u^h$ into $\mathbf{h}''^{n,\tau}_u$. Taking $\mathbf{h}''^{n,\tau}_u$ as the query and $\hat{\mathbf{H}}_u^h$ as the value, we can obtain the final state of $c_u^{n,\tau}$ by

$$\alpha = \text{softmax}[(\mathbf{h}''^{n,\tau}_u \mathbf{W}_{\alpha_1} + \mathbf{b}_{\alpha_1})(\hat{\mathbf{H}}_u^h \mathbf{W}_{\alpha_2} + \mathbf{b}_{\alpha_2})^\top], \quad (19)$$

$$\bar{\mathbf{h}}_u^{h,\tau} = \alpha \hat{\mathbf{H}}_u^h, \quad (20)$$

where $\mathbf{W}_*$ and $\mathbf{b}_*$ are learnable parameters and α are the attention weights. Correspondingly, we combine $\mathbf{h}''^{n,\tau}_u$ and $\bar{\mathbf{h}}_u^{h,\tau}$ to formulate the personalized context-aware state. For example, the context-aware state of $c_u^{n,\tau}$ can be denoted as

$$\hat{\mathbf{h}}_u^{n,\tau} = \bar{\mathbf{h}}_u^{h,\tau} \parallel \mathbf{h}''^{n,\tau}_u. \quad (21)$$

Now, we obtain a set of personal context-aware states corresponding to T_u^n , denoted as $\hat{\mathbf{T}}_u^n = [\hat{\mathbf{h}}_u^{n,1}, \dots, \hat{\mathbf{h}}_u^{n,\tau}, \dots, \hat{\mathbf{h}}_u^{n,m}]^\top$, which will be used as the input of the following real trajectory inference.

4.4 Real Trajectory Inference

We now describe how to infer the real transition dependencies among check-ins using another attention mechanism. In practice, we do not have any prior knowledge regarding whether the current check-in of a given trajectory is real or fake. In this study, we attempt to check and infer real check-ins one by one in an auto-regressive manner.

4.4.1 Movement Inference. During inference, which is distinct from the former *Movement Extractor*, we should not only capture the inner interaction in currently predicted results but also integrate the mutual interaction between the personalized context-aware state $\hat{\mathbf{T}}_u^n$ and currently predicted results. That is to say, we assume that we have predicted out the sub-sequence $\widetilde{RT}_u^{n,\tau-1} = \tilde{c}_u^{n,1}, \dots, \tilde{c}_u^{n,\tau-1}$. The goal is to use $\widetilde{RT}_u^{n,\tau-1}$ and the personalized context-aware state $\hat{\mathbf{T}}_u^n$ to forecast the next real check-in $\tilde{c}_u^{n,\tau}$. Thus, we first use Equation (6) to prepare the fused embeddings for $\widetilde{RT}_u^{n,\tau-1} = \tilde{c}_u^{n,1}, \dots, \tilde{c}_u^{n,\tau-1}$, i.e., $\widetilde{RT}_u^{n,\tau-1}$. In accordance with Equations (8)–(12), we design a similar structure to capture the inner interaction in $\widetilde{RT}_u^{n,\tau-1}$. We also apply $\mathcal{H}$ heads in self-attention, and for each head π (omitting n for brevity/clarity), we have

$$\mathbf{Q}_1^\pi(\widetilde{RT}_u^{\tau-1}) = \widetilde{RT}_u^{\tau-1} \mathbf{W}_{Q_1}^\pi, \quad (22)$$

$$\mathbf{K}_1^\pi(\widetilde{RT}_u^{\tau-1}) = \widetilde{RT}_u^{\tau-1} \mathbf{W}_{K_1}^\pi, \quad (23)$$

$$\mathbf{V}_1^\pi(\widetilde{RT}_u^{\tau-1}) = \widetilde{RT}_u^{\tau-1} \mathbf{W}_{V_1}^\pi, \quad (24)$$

$$\widetilde{\mathbf{Z}}_u^{\tau-1(\pi)} = \text{softmax}\left(\frac{\mathbf{Q}_1^\pi(\widetilde{RT}_u^{\tau-1}) \mathbf{K}_1^\pi(\widetilde{RT}_u^{\tau-1})^\top}{\sqrt{d'}}\right) \mathbf{V}_1^\pi(\widetilde{RT}_u^{\tau-1}), \quad (25)$$

where $\mathbf{W}_{Q_1}$, $\mathbf{W}_{K_1}$, and $\mathbf{W}_{V_1}$ denote the trainable matrices. Similar to the above *Movement Extractor*, we use a concatenation operator (i.e., Equation (12)) to aggregate each π head of $\widetilde{\mathbf{Z}}_u^{\tau-1(\pi)}$, and obtain $\widetilde{\mathbf{Z}}_u^{\tau-1}$. Now, we use the last state $\widetilde{\mathbf{Z}}_u^{\tau-1} \in \widetilde{\mathbf{Z}}_u^{\tau-1}$ as the query for next real check-in reference.

To incorporate the mutual interaction between $\tilde{\mathbf{z}}_u^{\tau-1}$ and $\hat{\mathbf{T}}_u^n$, we use a variant of self-attention with $\mathcal{H}$ heads for next real check-in reference. For $\pi \in \mathcal{H}$ head, we have

$$\mathbf{Q}_2^\pi(\tilde{\mathbf{z}}_u^{\tau-1}) = \tilde{\mathbf{z}}_u^{\tau-1} \mathbf{W}_{Q_2}^\pi, \quad (26)$$

$$\mathbf{K}_2^\pi(\hat{\mathbf{T}}_u^n) = \hat{\mathbf{T}}_u^n \mathbf{W}_{K_2}^\pi, \quad (27)$$

$$\mathbf{V}_2^\pi(\hat{\mathbf{T}}_u^n) = \hat{\mathbf{T}}_u^n \mathbf{W}_{V_2}^\pi, \quad (28)$$

$$\tilde{\mathbf{z}}_u^{\tau(\pi)} = \text{softmax}\left(\frac{\mathbf{Q}_2^\pi(\tilde{\mathbf{z}}_u^{\tau-1})\mathbf{K}_2^\pi(\hat{\mathbf{T}}_u^n)^\top}{\sqrt{d'}}\right)\mathbf{V}_2^\pi(\hat{\mathbf{T}}_u^n), \quad (29)$$

where $\mathbf{W}_{Q_2}$, $\mathbf{W}_{K_2}$, and $\mathbf{W}_{V_2}$ denote the learnable matrices. Herein, we also adopt Equation (12) to make head aggregation with concatenation operator, and obtain $\tilde{\mathbf{z}}_u^\tau$. Following Equations (13) and (14), we also apply the residual connection and layer normalization here to obtain the final state $\tilde{\mathbf{z}}_u^{\tau'}$ regarding $\tilde{\mathbf{z}}_u^\tau$, and use a dense layer to obtain the final result—i.e., $\tilde{c}_u^{n,\tau}$ —formulated by

$$\tilde{c}_u^{n,\tau} = \arg \max (\mathbf{W}_c \tilde{\mathbf{z}}_u^{\tau'} + \mathbf{b}_c), \quad (30)$$

where $\mathbf{W}_c$ and $\mathbf{b}_c$ are projection parameters for the next check-in prediction.

4.4.2 External Data Fusion. Recall the behavior (b3) (cf. Section 1). We argue that the user's intention of generating a fake check-in may be influenced by other users. That is to say, they could have been located at a POI that the user u has never visited. However, when that POI is frequently visited by other users, it enables us to integrate prior knowledge of other users' check-ins to assist in the discovery of fake check-ins for a given user u . Thus, we design an inverse POI popularity bias to incorporate other users' interests. Specifically, we first obtain a POI popularity list $\Psi = \{\psi_i | i \in \{1, 2, \dots, |\mathcal{P}|\}\}$ by calculating the frequency of each POI visited by other users. Since IRMo aims at eliminating fake check-ins, we further use the inverse of Ψ , denoted by $\check{\Psi} = \{\frac{1}{\psi_i} | i \in \{1, 2, \dots, |\mathcal{P}|\}\}$, to mitigate the bias from extremely popular POIs. In other words, we regard $\check{\Psi}$ as the inference bias to deal with the (b3). To this end, we replace Equation (30) with

$$\tilde{c}_u^{n,\tau} = \text{softmax}((\mathbf{W}_c \tilde{\mathbf{z}}_u^{\tau'} + \mathbf{b}_c + \text{softmax}(\check{\Psi}))), \quad (31)$$

$$\tilde{c}_u^{n,\tau} = \arg \max (\tilde{c}_u^{n,\tau}), \quad (32)$$

where $\mathbf{W}_c \in \mathbb{R}^{d' \times |\mathcal{P}|}$ and $\mathbf{b}_c \in \mathbb{R}^{|\mathcal{P}|}$ refer to the learnable parameters. $\tilde{c}_u^{n,\tau}$ is the probability distribution of the predicted result. Finally, the predicted trajectory is denoted as $\widetilde{RT}_u^n = \tilde{c}_u^{n,1}, \dots, \tilde{c}_u^{n,m}$.

4.5 Training

Herein, we conclude our training objective, where the loss function of IRMoGA is defined as the cross-entropy of the prediction and the ground truth. It is denoted as

$$\mathcal{L}_\theta = \sum_{u \in \mathcal{U}} \sum_{n=1}^{|T_u|} \sum_{\tau=1}^{|T_u^n|} -c_{u,r}^{n,\tau} \log (\tilde{c}_u^{n,\tau}), \quad (33)$$

where $|\mathcal{U}|$ is the number of users, $|T_u|$ means that user u has $|T_u|$ trajectories to infer, θ denotes the learnable parameters in IRMoGA, and $c_{u,r}^{n,\tau}$ is the one-hot representation of a ground-truth check-in. For a given trajectory T_u^n , it contains both real and fake check-ins. In fact, we do not know which check-in in trajectory T_u^n is fake, i.e., uncertainty. Hence, the optimization of Equation (33) actually contains two parts. For a real check-in, Equation (33) can be considered as reconstruction loss. For a fake check-in, Equation (33) can be treated as the correction loss.

We now summarize the algorithmic details for the purpose of formalizing the workflow of the proposed IRMoGA.

ALGORITHM 1: IRMoGA Training

Input: user trajectories $\mathcal{T} = \{T_1, \dots, T_u, \dots, T_{|\mathcal{U}|}\}$, ground truth $\mathcal{RT} = \{RT_1, \dots, RT_u, \dots, RT_{|\mathcal{U}|}\}$.

- 1 extract POI set $\mathcal{P}$ from $\mathcal{T}$;
- 2 generate an inverse POI popularity $\check{\Psi}$ from $\mathcal{T}$;
- 3 obtain POI embeddings $\mathbf{v}_s$ based on $\mathcal{P}$;
- 4 initialize the train model θ ;
- 5 **for** *epoch* $i=0; i=i+1; i < N$ **do**
- 6 **foreach** $u \in \mathcal{U}$ **do**
- 7 **foreach** $T_u^n \in T_u$ **do**
- 8 obtain the T_u^m and T_u^h ;
- 9 represent each check-in in T_u^n based on $\mathbf{v}_s$;
- 10 make temporal embedding for each check-in in T_u^n based on $\mathbf{v}_t$;
- 11 use Equation (6) to make information fusion;
- 12 obtain $\{\mathbf{h}_u^{n,1}, \dots, \mathbf{h}_u^{n,\tau}, \dots, \mathbf{h}_u^{n,m}\}$ based on Equations (8)–(14);
- 13 use Equations (16)–(21) to tackle T_u^h and obtain $\hat{T}_u^n = \{\hat{\mathbf{h}}_u^{n,1}, \dots, \hat{\mathbf{h}}_u^{n,\tau}, \dots, \hat{\mathbf{h}}_u^{n,m}\}$;
- 14 generate a predicted trajectory $\widehat{RT}_u^m$ based on Equations (22)–(31);
- 15 compute the training loss according to Equation (33);
- 16 update the learnable parameters;
- 17 **end**
- 18 **end**
- 19 **end**

Output: the optimal Model θ^*

In a nutshell, IRMoGA first extracts the POI set $\mathcal{P}$ from training data $\mathcal{T}$ and uses this set to obtain the POI embeddings based on the POI graph, whereafter an inverse POI popularity $\check{\Psi}$ is prepared. For each user's sub-trajectory (e.g., T_u^n), we leverage Fake Trajectory Encoding to formulate a set of hidden states regarding it. Next, we use the corresponding historical trajectory to construct the user's personalized mobility graph G_u , and then apply our Mutual Interaction Extractor to obtain a set of personal context-aware states corresponding to the given sub-trajectory. Finally, we leverage our Real Trajectory Inference to predict human real trajectory, where the inverse POI popularity is fused. In the practical implementation, we use the mini-batch for model training.

When it comes to the complexity of Algorithm 1, we note that there mainly exist three attention structures in IRMoGA, where attentions in Movement Extractor and Movement Inference are similar. Recall that (cf. Equation (11)), we use the dot-product computation for m check-ins in T_u^n , resulting in the time complexity of $O(|T_u^n|^2 d')$. Fortunately, users' recent trajectories are extremely shorter than their respective historical trajectories—i.e., $|T_u^n| \ll |T_u^h|$. For the last attention in Mutual Interaction Extractor, the time complexity is $O(|T_u^n| \times |P_u| \times d')$ according to Equation (19). Due to human periodic regularity, we also have $|P_u| \ll |T_u^h|$, which demonstrates that the graph-based model enables reducing the time cost. This leads to the drawback of memory usage—however, we use the individual graph for each user, which is a smaller structure. As such, it is capable of decreasing the time cost and space usage.

5 EVALUATION

We now present the experimental observations regarding the performance of IRMoGA. First, we introduce four popular GTSM datasets and the preprocessing procedure to cater to IRMo settings. Next, we provide the most relevant and popular baselines, whereafter the evaluation metrics are selected. Finally, we discuss the comparative experimental results, followed by ablation, sensitivity analysis, and visualization.

Table 2. Descriptive Statistics of Datasets

Dataset	$ \mathcal{U} $	$ \mathcal{P} $	#Check-ins	#Sub_T	avg_T	POI sequences
OSM	280	4,335	35,934	3,371	78	42,930
Yelp	2,270	7,531	133,937	44,706	45	75,340
Foursquare	1,083	9,815	120,007	36,182	90	98,180
Gowalla	3,987	21,354	239,493	66,612	47	213,570

5.1 Datasets

We conducted our experiments on four publicly available datasets: OSM traces (<https://www.openstreetmap.org/traces>), Yelp [25], Foursquare [51], and Gowalla [54]. For OSM traces, we crawled the GPS points from open-source traces located in Boston ($42^{\circ}19.8'N$ – $42^{\circ}24'N$, $71^{\circ}8.4'W$ – $71^{\circ}1.2'W$). Since the Yelp dataset contains massive geo-tagged businesses and reviews, we only use the check-in data from them. In addition, we selected the check-in data from New York City in Foursquare and California City in Gowalla, where each check-in contains a user ID, timestamp, POI ID, and geographical coordinates. Table 2 presents the statistics of datasets after preprocessing.

Notably, #Sub_T is the number of sub-trajectories after splitting, and avg_T denotes the average length of each user's historical trajectory. In addition, we use the random walk strategy to generate massive POI sequences for spatial embedding. Thus, we randomly generate 42,930 sequences containing 1,717,200 synthetic check-ins for OSM, 75,340 sequences containing 3,013,600 synthetic check-ins for Yelp, 98,180 sequences containing 3,927,200 synthetic check-ins for Foursquare, and 213,570 sequences containing 8,542,000 synthetic check-ins for Gowalla. To help other researchers, our datasets are also publicly available at <https://github.com/gcooq/IRMoGA>.

5.2 Data Preprocessing

In practice, the behavioral phenomenon (b1) occurs most frequently in daily life (cf. Reference [49]). Hence, we first choose the OSM dataset to respond to this scenario. In OSM, the publicly available GPS traces uploaded by anonymous users usually contain noise check-ins that deviate from the road network due to privacy concerns, inaccuracy of GPS devices, or man-made interference. Thus, we leverage a widely employed map matching method [28] to obtain the trajectories that are aligned with the road network and set them as the ground truth with respect to the original GPS traces. In particular, all GPS points are assigned an ID number according to the road network structure provided by OSM. In addition, for the Yelp dataset, we adopt the data preprocessing method in References [56, 57] to replace the POI in the real check-in with one of its associated collective POIs, where we regard the POI in the real check-in as the ground truth and the replaced POI as the fake one. Notably, the collective POI is a neighbor of a given POI within 500 meters. We follow [25] and remove those users with fewer than 25 check-ins as well as those POIs visited by fewer than 10 users. After preprocessing, the Yelp dataset contains about 15% fake check-ins.

To enable investigating more diverse settings (e.g., (b1), (b2), and (b3)) for the IRMo task, we use another two datasets—Foursquare and Gowalla—to generate the complex scenarios. We first remove those POIs that have been visited by fewer than 5 users. For each user, we concatenate all check-ins to form a single trajectory. Subsequently, we divide each trajectory into sub-trajectories with the time interval of 6 hours each, as was done in prior studies [6, 9, 16]. Then, we randomly replace 20% of check-ins in each sub-trajectory with generated ones, where three types of fake check-in behaviors are used, with equal probability. For (b1), we randomly select a nearby POI that is less than 1km away from the real POI. For (b2), we randomly choose from POIs that the user has visited before. As for (b3), we randomly pick a POI from among the top 50 frequently visited POIs by other users.

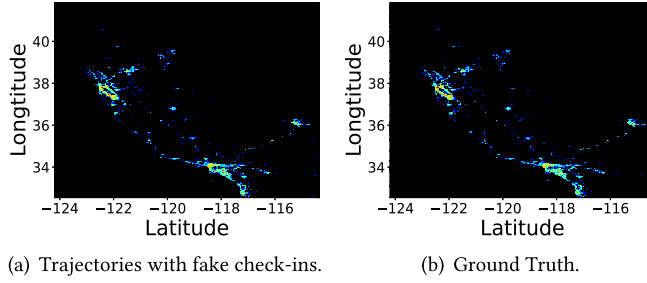


Fig. 6. Check-in Distribution of Gowalla.

For all datasets, we choose each user's first 80% sub-trajectories as the training set and the remaining 20% as testing data. As shown in Figure 6, it is hard to observe the changes between them—illustrating that our data preprocessing will not destroy the ground-truth distribution of original check-in data.

5.3 Baselines

We compare our IRMoGA with seven state-of-the-art methods, covering both the traditional trajectory mining and more recent deep learning-based approaches:

- **Popularity**: A simple and intuitive method that selects the most frequently visited POI from the user's history to replace the low-frequency POI in his/her trajectory.
- **MF** [24] explores interactions among POIs. We use it to decompose the POI co-occurrence matrix and leverage the similarity function to distill most unrelated POIs in a given trajectory, while recommending similar POIs.
- **Markov** [38] estimates check-in regularities using the constructed transitional matrix. We use Markov to distinguish the low-frequency transition and choose POIs with the highest transitional probabilities to replace each of them.
- **NeuNext** [61] adopts a recurrent neural network for transitional pattern learning, we leverage it to infer human real mobility.
- **GM-VSAE** [26] leverages two recurrent neural networks to model human sequential mobility in an end-to-end manner, and discriminates the abnormal trajectories. We use it to predict human real trajectories, where the GRU cell is employed.
- **MHA-LSTM** [29] is an end-to-end method by integrating an attention mechanism to explore high-order sequential patterns.
- **STAN** [27] attempts to exploit relative spatio-temporal information of historical check-ins with self-attention layers. We extend it to tackle IRMo problem.
- **AttnMove** [48] leverages multiple self-attention mechanisms to generate fine-grained trajectories. We adapt it to IRMo by following a similar self-attention module to learn human mobility for real trajectory inference.

5.4 Metrics

We select three common metrics for evaluations: *Recall*, *F1*, and *Accuracy* [10, 48]. *Recall* denotes the percentage of the same part between the predicted trajectory and the ground truth—e.g., *Recall* is 1 if the predicted trajectory is the same as the ground truth. *F1* is the harmonic mean of the precision and recall. *Accuracy* is the ratio that the predicted trajectories and the ground-truth trajectories are the same in the entire test set (in a sense, *Accuracy* is a more strict metric).

Table 3. Performance Comparison on four Datasets

Method	OSM			Yelp			Foursquare			Gowalla		
	Recall	F1	Accuracy	Recall	F1	Accuracy	Recall	F1	Accuracy	Recall	F1	Accuracy
Popularity	31.40%	30.44%	1.22%	33.67%	33.66%	6.19%	24.78%	24.74%	3.72%	17.44%	17.45%	0.48%
MF	9.87%	9.88%	1.68%	19.43%	19.39%	0.50%	29.00%	28.37%	3.47%	30.72%	30.07%	4.43%
Markov	9.44%	9.37%	0.46%	20.85%	20.84%	0.04%	24.57%	24.56%	4.44%	19.76%	19.74%	0.44%
NeuNext	36.48%	28.85%	5.26%	59.75%	53.10%	39.24%	33.40%	27.34%	15.39%	27.92%	19.66%	6.05%
GA-VSAE	42.52%	33.78%	5.11%	60.15%	53.31%	38.80%	36.56%	30.76%	19.10%	27.28%	19.83%	7.50%
MHA-LSTM	36.14%	28.84%	4.12%	48.89%	46.99%	39.35%	36.83%	30.98%	19.28%	27.53%	20.11%	7.73%
STAN	39.29%	33.01%	5.14%	74.98%	71.09%	59.46%	39.89%	35.36%	27.98%	29.99%	22.95%	9.03%
AttnMove	38.69%	32.30%	6.53%	86.23%	83.91%	75.79%	40.51%	37.31%	30.34%	51.53%	43.14%	24.46%
IRMoGA	57.03%	50.27%	11.22%	94.12%	91.15%	78.76%	79.67%	73.13%	53.67%	77.83%	69.92%	44.03%

We implemented all the baselines and our IRMoGA in Python, while the scikit-learn and Tensorflow are, respectively, leveraged for traditional and deep learning-based methods, accelerated by the NVIDIA RTX 3090 GPU 24G. The batch size is 64, and the dimensionality of POI is 64. The initial learning rate is 0.001, the hidden size is set to 128 for OSM, Yelp, and Foursquare, and it is set to 256 for Gowalla. The number of hidden layers is 1, the number of attention heads is 2, and the optimizer is Adam [17].

5.5 Empirical Results

Table 3 summarizes the performance of IRMoGA as well as the baselines on the four location-based datasets with respect to *Recall*, *F₁*, and *Accuracy*. We observe that IRMoGA significantly outperforms all the baselines on all the datasets. The three conventional methods (Popularity, MF, and Markov) perform worse than the deep learning-based methods, which indicates that the deep learning-based approaches enable better incorporation of the underlying correlation among check-ins, as well as highly complex transitional regularities from massive trajectory data containing numerous fake check-ins. In particular, AttnMove and STAN have higher gains than NeuNext, GA-VSAE, and MHA-LSTM. The plausible reason could be that AttnMove and STAN enable integrating multiple interactions among check-ins. In contrast, our IRMoGA outperforms AttnMove and STAN due to two possible reasons. First, IRMoGA considers the geographical similarity during check-in embedding, primarily seeking to mimic human potential fake check-in behavior—i.e., they possibly select a nearby POI to hide their real moving intention. Although STAN explicitly considers the spatial distance between POIs, it could fail to capture the geographical similarity between different POIs. Second, IRMoGA adopts the graph learning scheme to distill people’s personalized check-in preferences in terms of transitional regularities. Meanwhile, we also consider the impact of other users’ check-in behaviors, since the daily activities of an individual could be affected by others. We note that the performance of IRMoGA, while still superior to the baselines, is worse on the OSM dataset (which contains only **(b1)** fake check-in behavior). It appears that the main reason for this is the insufficient amount of regularities in the human traces. Conversely, when tackling richer and more diverse human behavioral phenomena, IRMoGA performs in the other datasets. Table 4 presents our experimental observations regarding the model efficiency of IRMoGA in terms of both training and inference phases (the latter indicated in “()”), in comparison with the deep learning-based approaches. Due to the graph construction in the MIE, IRMoGA is slower—however, we observe that without MIE, IRMoGA achieves similar efficiency to AttnMove while retaining significant effectiveness improvements.

5.6 Ablation Analysis

We now report our evaluation of the effectiveness of each module in IRMoGA, and the study of its performance by considering more diverse fake check-in behaviors, which enables a more thorough understanding of various human check-in decisions.

Table 4. Model Efficiency

Method	Runtime (s/epoch)			
	OSM	Yelp	Foursquare	Gowalla
NeuNext	7 (1)	40 (7)	52 (8)	144 (47)
MHA-LSTM	42 (3)	184 (21)	57 (9)	213 (23)
AttnMove	7 (3)	72 (28)	61 (17)	91 (42)
IRMoGA	13 (5)	132 (42)	73 (21)	177 (58)

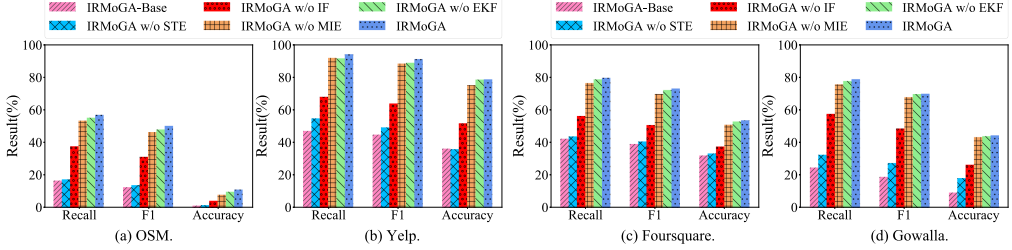


Fig. 7. Ablation Analysis.

5.6.1 Impact of Individual Modules. We designed four variants of IRMoGA to estimate the contribution of each proposed module:

- *IRMoGA-Base* is a basic model that only uses the Movement Extractor and Movement Inference, where we make check-in embedding with a random initialization method.
- *IRMoGA w/o STE* removes the Spatial-temporal Embedding module and uses a learnable matrix for check-in embeddings.
- *IRMoGA w/o IF* removes the Information Fusion and uses the sum operation to fuse the spatial-temporal embedding directly.
- *IRMoGA w/o MIE* removes the Mutual Interaction Extractor in IRMoGA.
- *IRMoGA w/o EKF* does not incorporate the External Knowledge Fusion part when inferring the real trajectory.

Figure 7 shows the impact of different modules in “%” relative to the complete IRMoGA. We observe that each module contributes differently to tackling the IRMo problem. More specifically, IRMoGA-Base does not have any domain knowledge such as geographical information, individual check-in behaviors, and external trajectory information, resulting in the worst performance. The results of IRMoGA w/o STE suggest that incorporating geographical similarity can significantly improve performance. The possible reason is that people usually tend to move in a small area, which is also in line with the social studies on human daily activities [35]. The results of IRMoGA w/o IF show that the learnable information fusion can adapt to spatial-temporal information extraction and effectively improve the model accuracy. The results of IRMoGA w/o MIE and IRMoGA w/o EKF indicate that mutual interaction extractor and external knowledge fusion do help learn better human check-in behaviors. However, the external knowledge fusion part has only a slight improvement, which encourages future work to develop a more efficient module to involve external knowledge.

In addition, we explain the motivation for using the geo-spatial information when constructing the POI graph. First, each check-in left by users is often associated with geo-spatial data, which are highly correlated to users’ visiting preferences. Second, users may prefer to visit nearby places. For quantitative comparison, we present the transitional distance between all the adjacent

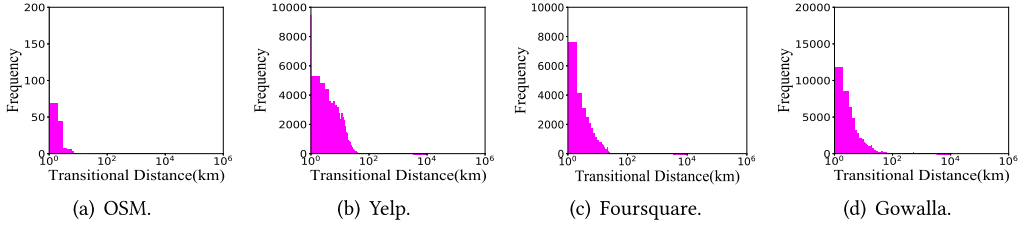


Fig. 8. Distribution of transitional distance.

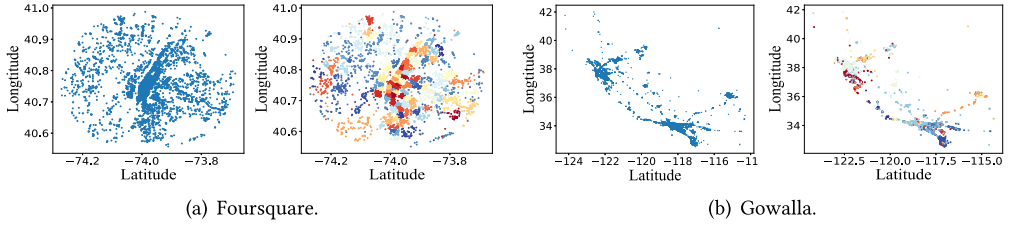


Fig. 9. Visualization of POIs.

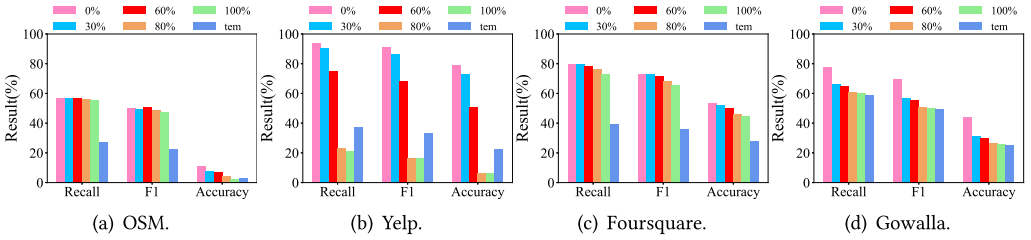


Fig. 10. Impact of Periodic Regularity.

check-ins for all users in Figure 8. As shown, users often prefer logging into a location within a very short distance even if they have plenty of opportunities to choose others. In fact, this observation motivated our choice of distance threshold (e.g., 1 km) to construct the POI graph. To show the performance of our graph-based POI embedding, we use the K-means method to cluster POI embeddings (e.g., Foursquare and Gowalla) and mark different colors for distinct clusters in Figure 9. We observe that our embedding method is capable of incorporating geographical similarity.

5.6.2 Impact of Periodic Regularity. As we mentioned in Section 1, exploiting human periodic regularity is the core work to achieve the discovery of fake check-ins. Therefore, we turn to an experimental perspective to reveal the impact of human periodic regularity on solving the IRMo problem. Specifically, we formulate a new training set by randomly shuffling the order of the POIs in each trajectory from the training data set. And we use it to train the IRMoGA while the testing data is not changed. In addition, we remove the temporal embedding in our IRMoGA, tagged as “tem.” As shown in Figure 10, we can find that the performance (e.g., Accuracy) drops significantly as more trajectories in the training set are shuffled (human periodic regularities are corrupted). Besides, IRMoGA without the temporal embedding performs clearly worse. Therefore, we conclude that considering the human periodic regularity behind human check-ins does help us detect fake check-ins and further infer real check-ins.

Table 5. Comparison on Different Behaviors

Behavior	Foursquare			Gowalla		
	Recall	F1	Accuracy	Recall	F1	Accuracy
b1	82.43%	76.54%	58.48%	77.94%	70.22%	44.96%
b2	77.17%	70.69%	51.93%	77.19%	69.20%	43.10%
b3	76.81%	70.07%	51.86%	76.62%	68.92%	44.15%
b1+b2	80.83%	74.27%	54.78%	77.19%	69.24%	43.42%
b1+b3	77.70%	70.14%	52.40%	76.48%	68.80%	43.88%
b2+b3	78.19%	71.73%	53.39%	75.56%	67.64%	42.17%
b1+b2+b3	79.67%	73.13%	53.67%	77.83%	69.92%	44.03%

Table 6. Sensitivity Test

Rate	Foursquare			Gowalla		
	Recall	F1	Accuracy	Recall	F1	Accuracy
10%	97.07%	96.04%	92.24%	89.18%	85.44%	72.80%
15%	95.70%	94.06%	87.46%	87.19%	82.83%	68.72%
20%	79.67%	73.13%	53.67%	77.83%	69.92%	44.03%
30%	76.46%	68.37%	47.10%	74.20%	67.11%	47.99%
40%	61.79%	52.10%	28.28%	51.07%	39.67%	12.98%
Random	83.29%	72.24%	58.06%	71.70%	62.82%	37.16%

5.6.3 Impact of Fake Check-in Behaviors. Recall the three kinds of fake check-in behaviors mentioned in Section 1 (**b1**), (**b2**), and (**b3**). We evaluated the effectiveness of IRMoGA in the context of diversity of fake check-ins and, as shown in Table 5, IRMoGA performs the best in the context of (**b1**). It performs worse in the context of (**b2**); and worst in the context of (**b3**) (note that the gray-highlighted part, which takes these three types of fake check-in behavior into account, is the same as the last row in Table 3). This indicates that the deceitful behavior of people choosing a POI that they have never visited is difficult to detect. Undoubtedly, there may be other fake check-in behaviors in real scenarios, which motivates further explorations, involving social sciences and HCI domain expertise.

In addition, we provide a sensitivity test on various amounts of fake check-in data. According to Reference [49], people usually have about 57% real check-ins that are corresponding to their real physical locations. Thus, we scale the rate of fake check-ins from 10% to 40%. In addition, to demonstrate the selection biases of different types of fake check-in, we provide a random setting that makes each trajectory randomly generates 10%–30% fake check-ins of each behavior. According to the results in Table 6, IRMoGA performs well on both datasets when the fake check-in rate is within 30%. Nevertheless, the results of IRMoGA on *Accuracy* drop faster as the scale of fake check-ins increases. The main reason is that *Accuracy* is a strict metric that reports the prediction trajectory exactly the same as the ground truth. Moreover, we find that IRMoGA achieves promising results in the context of **Random**. In reality, people have diverse preferences when deciding to make fake check-ins. For instance, some people prefer to choose a nearby POI to replace their real location while others may tend to choose a POI that is far away from their real location. Therefore, the results of **Random** demonstrate that our proposed IRMoGA can well address the diverse and intricate fake check-in behaviors in real-world scenarios.

5.7 Sensitivity Analysis on Hyper-parameters

We now discuss the impacts of some key hyper-parameters that could affect the model performance. Figures 11 and 12 report the results, and we have the following observations.

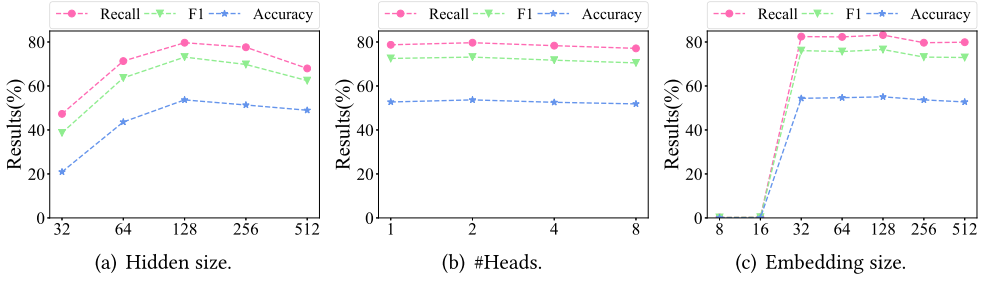


Fig. 11. Sensitivity Analysis on Foursquare.

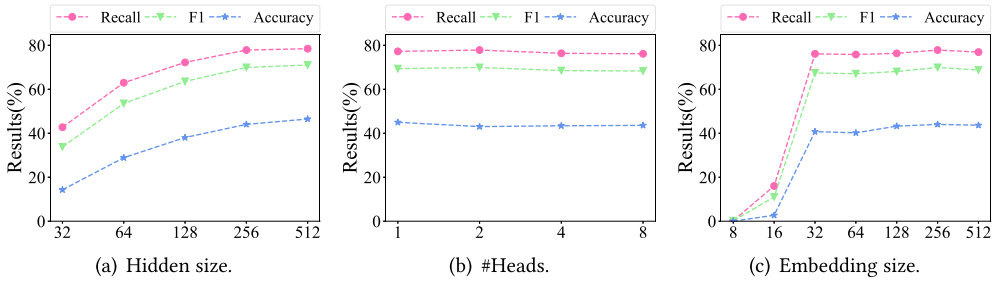


Fig. 12. Sensitivity Analysis on Gowalla.

- *Influence of hidden size.* We investigate the impact of hidden size controlling the scale of the attention neural network. As shown in Figures 11(a) and 12(a), we clearly observe that it does affect the model performance, however, the larger hidden size will not bring the model improvement.
- *Influence of head number.* Figures 11(b) and 12(b) show the robustness of IRMoGA performance regarding the head number. It also suggests that it is not sensitive to the model, thus we use a few heads in our IRMoGA (e.g., two heads).
- *Influence of embedding size.* Finally, we study the dimensional influence of our spatial-temporal embedding ranging from 8 to 512 on two datasets. As Figures 11(c) and 12(c) show, we find that the larger dimension will bring a higher improvement. This phenomenon is similar to other check-in embedding in various mobility-based tasks. Also, using too large dimension will not bring significant improvements, possibly because the model may cause the over-fitting problem.

5.8 Visualization Analysis

We now discuss the properties of IRMoGA from the perspective of visualization. We take the Foursquare dataset as an illustrative example. As Figure 13 shows, we first visualize check-in distribution of testing data. Notably, the darker color indicates that particular regions are visited more frequently. We can observe that it is hard to discriminate among Figures 13(a), 13(b), and 13(c) from a macro perspective, indicating that tackling IRMo problem is indeed an extremely challenging task. We are incapable of being familiar with users' correct check-ins, and we also have no prior information about their fake check-in behaviors. It also demonstrates that the appearance of fake check-ins could not seriously affect long-established human visiting habits, which inspires us to regard the proposed Mutual Interaction Extractor as the auxiliary learner to refine human transitional regularities from their historical trajectories.

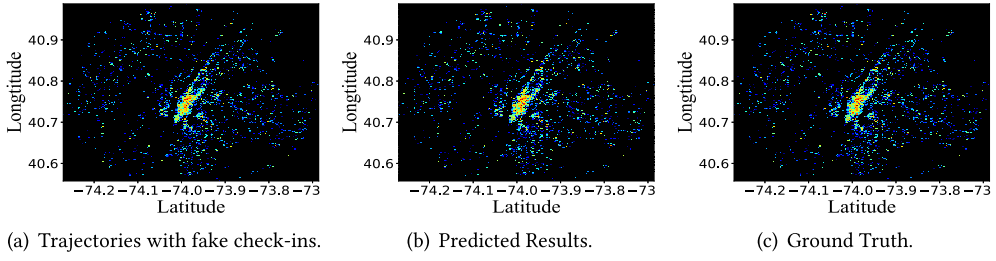


Fig. 13. Comparison: Complete check-in data distribution.

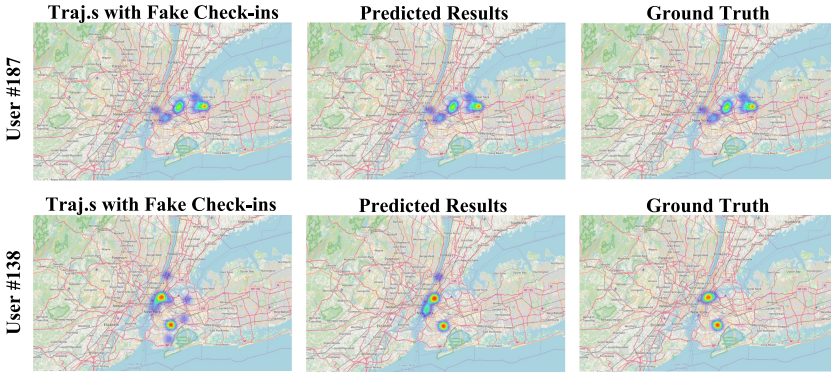


Fig. 14. Comparison: Actual users' trajectories.

Last, we test the results of two users' trajectories on Openstreet Map (www.openstreetmap.org), and show the performance of IRMoGA. As the top portion of Figure 14 shows, for User #187, these three sub-figures are similar, indicating that our IRMoGA enables refining human real visiting preferences in terms of check-in distillation. Complementary, as shown for User #138 (bottom row of Figure 14) the user's check-in distribution is significantly different from the ground truth. Interestingly, the results of the middle part show that our IRMoGA is capable of discovering their real moving intentions. In sum, IRMoGA enables tackling diverse check-in behaviors with respect to inferring their real mobility.

6 CONCLUSION

We formulated IRMo, a novel problem targeting the inference of human real mobility from massive trajectory data containing potential fake check-ins. To the best of our knowledge, this challenging task has been under-explored. We developed a graph-enhanced attention model (coupling check-in behaviors), called IRMoGA, to capture complex spatial-temporal dependencies and mutual interactions for the purpose of inferring human real mobility regularity. We presented the results of extensive experiments on real-world datasets, demonstrating that IRMoGA achieves significant gains against several relevant baselines. We note that IRMoGA, while being first, is also an extensible solution for detecting human real mobility, and it could be easily enhanced by incorporating other prior knowledge such as diverse social information. In the future, we plan to study the above issues in an incremental learning context that is more similar to the manner of human learning and knowledge accumulation. In addition, we plan to consider how to tackle the human traces that have no rich mobility patterns or regularities.

REFERENCES

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. Retrieved from <https://arxiv.org/abs/1607.06450>
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*. 1–15.
- [3] Ling Cai, Jun Xu, Ju Liu, and Tao Pei. 2018. Integrating spatial and temporal contexts into a factorization model for POI recommendation. *Int. J. Geogr. Info. Sci.* 32, 3 (2018), 524–546.
- [4] Yile Chen, Cheng Long, Gao Cong, and Chenliang Li. 2020. Context-aware deep model for joint mobility and time prediction. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 106–114.
- [5] Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be more with less: Hypergraph attention networks for inductive text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. 4927–4936.
- [6] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. 2017. POI2Vec: Geographical latent representation for predicting future visitors. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 102–108.
- [7] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new POI recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 2069–2075.
- [8] Huiji Gao, Jiliang Tang, and Huan Liu. 2012. Exploring social-historical ties on location-based social networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 6. 114–121.
- [9] Qiang Gao, Fan Zhou, Goce Trajcevski, Fengli Zhang, and Xucheng Luo. 2020. Adversity-based social circles inference via context-aware mobility. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'20)*. IEEE, 1–6.
- [10] Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. 2017. Identifying human mobility via trajectory embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 1689–1695.
- [11] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Xiangyang Xue, and Zheng Zhang. 2020. Multi-scale self-attention for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7847–7854.
- [12] Suining He and Kang G. Shin. 2020. Towards fine-grained flow forecasting: A graph attention approach for bike sharing systems. In *Proceedings of the Web Conference 2020*. 88–98.
- [13] Wenbo He, Xue Liu, and Mai Ren. 2011. Location cheating: A security challenge to location-based social network services. In *Proceedings of the 31st International Conference on Distributed Computing Systems*. IEEE, 740–749.
- [14] Zheng Huang, Jing Ma, Yushun Dong, Natasha Zhang Foutz, and Jundong Li. 2022. Empowering next POI recommendation with multi-relational modeling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1–5.
- [15] Di Jin, Yingli Gong, Zhiqiang Wang, Zhizhi Yu, Dongxiao He, Yuxiao Huang, and Wenjun Wang. 2022. Graph neural network for higher-order dependency networks. In *Proceedings of the ACM Web Conference 2022*. 1622–1630.
- [16] Pavlos Kefalas, Panagiotis Symeonidis, and Yannis Manolopoulos. 2018. Recommendations based on a heterogeneous spatio-temporal social network. *World Wide Web* 21, 2 (2018), 345–371.
- [17] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Retrieved from <https://arxiv.org/abs/1412.6980>
- [18] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*. 1–14.
- [19] Gaël Letarte, Frédéric Paradis, Philippe Giguère, and François Laviolette. 2018. Importance of self-attention for sentiment analysis. In *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 267–275.
- [20] Fuxian Li, Jie Feng, Huan Yan, Depeng Jin, and Yong Li. 2022. Crowd flow prediction for irregular regions with semantic graph attention network. *ACM Trans. Intell. Syst. Technol.* 13, 5 (2022), 14 pages. <https://doi.org/10.1145/3501805>
- [21] Pengfei Li, Hua Lu, Nattiya Kanhabua, Sha Zhao, and Gang Pan. 2018. Location inference for non-geotagged tweets in user timelines. *IEEE Trans. Knowl. Data Eng.* 31, 6 (2018), 1150–1165.
- [22] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S. Jensen, and Wei Wei. 2018. Deep representation learning for trajectory similarity computation. In *Proceedings of the IEEE 34th International Conference on Data Engineering (ICDE'18)*. IEEE, 617–628.
- [23] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. 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*. 831–840.
- [24] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. 2013. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1043–1051.

- [25] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. 2017. An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proc. VLDB Endow.* 10, 10 (2017), 1010–1021.
- [26] Yiding Liu, Kaiqi Zhao, Gao Cong, and Zhifeng Bao. 2020. Online anomalous trajectory detection with deep generative sequence modeling. In *Proceedings of the IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 949–960.
- [27] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the Web Conference 2021*. 2177–2185.
- [28] Dennis Luxen and Christian Vetter. 2011. Real-time routing with OpenStreetMap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 513–516.
- [29] Kaouther Messaoud, Itheri Yahiaoui, Anne Verroust-Blondet, and Fawzi Nashashibi. 2020. Attention based vehicle trajectory prediction. *IEEE Trans. Intell. Vehic.* 6, 1 (2020), 175–185.
- [30] Congcong Miao, Ziyang Luo, Fengzhu Zeng, and Jilong Wang. 2020. Predicting human mobility via attentive convolutional network. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 438–446.
- [31] Sara Migliorini, Mauro Gambini, and Alberto Belussi. 2019. A blockchain-based solution to fake check-ins in location-based social networks. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Analytics for Local Events and News*, Haiquan Chen, Federico Iuricich, and Amr Magdy (Eds.). 1–4.
- [32] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.
- [33] Dina Abbas Abd El Moniem and Hoda M. O. Mokhtar. 2019. Check 2: A framework for fake check-in detection. In *Intelligent Computing—Proceedings of the Computing Conference*. Springer, 1–12.
- [34] Kyosuke Nishida, Hiroyuki Toda, Takeshi Kurashima, and Yoshihiko Suhara. 2014. Probabilistic identification of visited point-of-interest for personalized automatic check-in. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 631–642.
- [35] Atsuyuki Okabe. 2016. *GIS-based Studies in the Humanities and Social Sciences*. (CRC) Taylor & Francis.
- [36] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 701–710.
- [37] Hongzhi Shi, Yong Li, Hancheng Cao, Xiangxin Zhou, Chao Zhang, and Vassilis Kostakos. 2019. Semantics-aware hidden Markov model for human mobility. *IEEE Trans. Knowl. Data Eng.* 33, 3 (2019), 1183–1194.
- [38] Xuan Song, Quanshi Zhang, Yoshihide Sekimoto, Ryosuke Shibasaki, Nicholas Jing Yuan, and Xing Xie. 2016. Prediction and simulation of human mobility following natural disasters. *ACM Trans. Intell. Syst. Technol.* 8, 2 (2016), 1–23.
- [39] Zhu Sun, Chen Li, Yu Lei, Lu Zhang, Jie Zhang, and Shunpan Liang. 2021. Point-of-interest recommendation for users-businesses with uncertain check-ins. *IEEE Trans. Knowl. Data Eng.* 34, 12 (2021), 5925–5938.
- [40] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*. 3104–3112.
- [41] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. 1067–1077.
- [42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.
- [43] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the International Conference on Learning Representations*. 1–12.
- [44] Gang Wang, Sarita Y. Schoenebeck, Haitao Zheng, and Ben Y. Zhao. 2016. “Will check-in for badges”: Understanding bias and misbehavior on location-based social networks. In *Proceedings of the 10th International AAAI Conference on Web and Social Media*. 417–426.
- [45] Huandong Wang, Sihan Zeng, Yong Li, and Depeng Jin. 2020. Predictability and prediction of human mobility based on application-collected location data. *IEEE Trans. Mobile Comput.* 20, 7 (2020), 2457–2472.
- [46] Suhang Wang, Charu Aggarwal, and Huan Liu. 2019. Beyond word2vec: Distance-graph tensor factorization for word and document embeddings. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1041–1050.
- [47] Yongji Wu, Defu Lian, Shuowei Jin, and Enhong Chen. 2019. Graph convolutional networks on user mobility heterogeneous graphs for social relationship inference. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 3898–3904.
- [48] Tong Xia, Yunhan Qi, Jie Feng, Fengli Xu, Funing Sun, Diansheng Guo, and Yong Li. 2021. AttnMove: History enhanced trajectory recovery via attentional network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4494–4502.
- [49] Fengli Xu, Guozhen Zhang, Zhilong Chen, Jiaxin Huang, Yong Li, Diyi Yang, Ben Y. Zhao, and Fanchao Meng. 2018. Understanding motivations behind inaccurate check-ins. *Proceed. ACM Hum.-Comput. Interact.* 2, CSCW (2018), 1–22.

- [50] Hao Xue, Flora Salim, Yongli Ren, and Nuria Oliver. 2021. MobTCast: Leveraging auxiliary trajectory forecasting for human mobility prediction. *Adv. Neural Info. Process. Syst.* 34 (2021), 30380–30391.
- [51] Dingqi Yang, Daqing Zhang, Vincent W. Zheng, and Zhiyong Yu. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Trans. Syst., Man, Cybernet.: Syst.* 45, 1 (2014), 129–142.
- [52] Haochao Ying, Liang Chen, Yuwen Xiong, and Jian Wu. 2016. PGRank: Personalized geographical ranking for point-of-interest recommendation. In *Proceedings of the 25th International Conference Companion on World Wide Web*. 137–138.
- [53] Fuqiang Yu, Lizhen Cui, Wei Guo, Xudong Lu, Qingzhong Li, and Hua Lu. 2020. A category-aware deep model for successive POI recommendation on sparse check-in data. In *Proceedings of the Web Conference*. ACM, New York, NY, 1264–1274.
- [54] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 363–372.
- [55] Zixuan Yuan, Hao Liu, Yanchi Liu, Denghui Zhang, Fei Yi, Nengjun Zhu, and Hui Xiong. 2020. Spatio-temporal dual graph attention network for query-poi matching. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 629–638.
- [56] Lu Zhang, Zhu Sun, Jie Zhang, Horst Kloeden, and Felix Klanner. 2020. Modeling hierarchical category transition for next POI recommendation with uncertain check-ins. *Info. Sci.* 515 (2020), 169–190.
- [57] Lu Zhang, Zhu Sun, Jie Zhang, Yu Lei, Chen Li, Ziqing Wu, Horst Kloeden, and Felix Klanner. 2021. An interactive multi-task learning framework for next POI recommendation with uncertain check-ins. In *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence*. 3551–3557.
- [58] Zhiqian Zhang, Chenliang Li, Zhiyong Wu, Aixin Sun, Dengpan Ye, and Xiangyang Luo. 2020. Next: A neural network framework for next poi recommendation. *Front. Comput. Sci.* 14, 2 (2020), 314–333.
- [59] Zengbin Zhang, Lin Zhou, Xiaohan Zhao, Gang Wang, Yu Su, Miriam Metzger, Haitao Zheng, and Ben Y. Zhao. 2013. On the validity of geosocial mobility traces. In *Proceedings of the 12th ACM Workshop on Hot Topics in Networks*. 1–7.
- [60] Jing Zhao, Pengpeng Zhao, Lei Zhao, Yanchi Liu, Victor S Sheng, and Xiaofang Zhou. 2021. Variational self-attention network for sequential recommendation. In *Proceedings of the IEEE 37th International Conference on Data Engineering (ICDE’21)*. IEEE, 1559–1570.
- [61] Pengpeng Zhao, Anjing Luo, Yanchi Liu, Fuzhen Zhuang, Jiajie Xu, Zhixu Li, Victor S. Sheng, and Xiaofang Zhou. 2022. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Trans. Knowl. Data Eng.* 34, 5 (2022), 2512–2524.
- [62] Ting Zhong, Shengming Zhang, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, and Jin Wu. 2020. Hybrid graph convolutional networks with multi-head attention for location recommendation. *World Wide Web* 23, 6 (2020), 3125–3151.
- [63] Fan Zhou, Yurou Dai, Qiang Gao, Pengyu Wang, and Ting Zhong. 2021. Self-supervised human mobility learning for next location prediction and trajectory classification. *Knowl.-Based Syst.* 228 (2021), 107214.
- [64] Wanzheng Zhu, Chao Zhang, Shuochao Yao, Xiaobin Gao, and Jiawei Han. 2018. A spherical hidden markov model for semantics-rich human mobility modeling. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 4613–4620.

Received 31 May 2022; revised 22 December 2022; accepted 20 May 2023