



# Mobility Prediction via Rule-enhanced Knowledge Graph

QIAOHONG YU, HUANDONG WANG, YU LIU, DEPENG JIN, and YONG LI, Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing, China

LIN ZHU and JUNLAN FENG, Tsinghua University-China Mobile Communications Group Co., Ltd. Joint Institute, China Mobile Research, Beijing, China

With the rapid development of location acquisition technologies, massive mobile trajectories have been collected and made available to us, which support a fantastic way of understanding and modeling individuals' mobility. However, existing data-driven methods either fail to capture the long-range dependency or suffer from a high computational cost. To overcome these issues, we propose a knowledge-driven framework for mobility prediction, which leverages knowledge graphs (KG) to formulate the mobility prediction task into the KG completion problem through integrating the structured "knowledge" from the mobility data. However, most related mobility prediction works only focus on the structured information encoded in existing triples, which ignores the rich semantic information of relation paths composed of multiple relation triples. In this article, we apply a dedicated module to extract the supplementary semantic structure of paths in KG, which contributes to the interpretability and accuracy of our model. Specifically, the extracted rules are applied to capture the dependencies between relational facts. Moreover, by incorporating user information in the entity-relation space with the corresponding hyperplane, our method could capture diverse user mobility patterns and model the personal characteristics of users to improve the accuracy of mobility prediction. Extensive evaluations illustrate that our proposed model beats state-of-the-art mobility prediction algorithms, which verifies the superiority of utilizing logical rules and user hyperplanes. Our implementation code is available at <https://github.com/tsinghua-fib-lab/RuleKG-MobiPre.git>

CCS Concepts: • **Human-centered computing** → **User models; Mobile devices**; • **Information systems** → *Spatial-temporal systems*; • **Applied computing** → *Sociology*; • **Networks** → *Network services*;

Additional Key Words and Phrases: Mobility prediction, knowledge graph, logic rule, user plane

This article was supported by the Natural Science Foundation of China under 62171260, 62272260, and U21B2036.

Authors' Contact Information: Qiaohong Yu, Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing, China; e-mail: yuqh20@mails.tsinghua.edu.cn; Huandong Wang (Corresponding author), Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing, China; e-mail: wanghuandong@tsinghua.edu.cn; Yu Liu, Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing, China; e-mail: liuyu18@mails.tsinghua.edu.cn; Depeng Jin, Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing, China; e-mail: jindp@tsinghua.edu.cn; Yong Li, Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing, China; e-mail: liyong07@tsinghua.edu.cn; Lin Zhu, Tsinghua University-China Mobile Communications Group Co., Ltd. Joint Institute, China Mobile Research, Beijing, China; e-mail: zhulinyj@chinamobile.com; Junlan Feng, Tsinghua University-China Mobile Communications Group Co., Ltd. Joint Institute, China Mobile Research, Beijing, China; e-mail: fengjunlan@chinamobile.com.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2024 Copyright held by the owner/author(s).

ACM 1556-472X/2024/10-ART215

<https://doi.org/10.1145/3677019>

**ACM Reference format:**

Qiaohong Yu, Huandong Wang, Yu Liu, Depeng Jin, Yong Li, Lin Zhu, and Junlan Feng. 2024. Mobility Prediction via Rule-enhanced Knowledge Graph. *ACM Trans. Knowl. Discov. Data.* 18, 9, Article 215 (October 2024), 21 pages.  
<https://doi.org/10.1145/3677019>

## 1 Introduction

Rapidly evolving mobile localization technology has enabled us to access enormous spatio-temporal user mobility data, which helps us better analyze and model people's mobility [30, 36, 43]. Currently, most existing approaches adopt data-driven methods to predict users' future movement. As shown in Figure 1(a), these methods utilize powerful machine learning models to implement future mobility prediction based on information propagated along empirical physical paths, e.g., along the historical trajectories [14], or directly from all historical mobility records in a fully connected manner [7]. However, these methods either fail to capture the long-range dependency of trajectories [14] or suffer from a large computational cost [7], whose reason is that the utilized information propagation paths in these data-driven methods are either ineffective or inefficient.

Meanwhile, the emerging techniques of **knowledge graphs (KG)** offer a promising approach for extracting structured "knowledge" from the massive spatio-temporal mobility data [16]. A variety of applications based on KG methods, including user profiling and recommendation systems, have achieved considerable success [18, 27, 33, 36]. Thus, one intriguing research question is whether we can solve the mobility prediction problem in the knowledge-driven paradigm. As shown in Figure 1(b), based on KG, we are able to describe the mobility behavior of users in the form of relational facts in the KG and utilize powerful **knowledge graph embedding (KGE)** techniques to extract features from massive trajectory data. Then, logical rules between relational facts can be further extracted to serve as the information propagation paths. For example, just as the second rule example, assuming that one user visited location  $p_1$  at time  $t_1$  and location  $p_2$  at time  $t_2$ , there must be a potential transfer relationship between these two locations. By injecting this logical rule into our constructed knowledge graph, we can predict future user mobility behavior accurately, e.g.,  $p_5$  in Figure 1(b). In this way, the structured "knowledge" in terms of KGEs and logic paths can predict future user mobility behavior by examining the plausibility of potential facts, which exhibits a promising way to achieve effective and efficient mobility prediction.

However, utilizing KG to model the spatio-temporal mobility behavior of users is also difficult due to a number of key challenges. First, there exists significant diversity between the mobility patterns of different users. For example, the home place of one user might be the workplace of another user, leading to their different mobility patterns across these two places. How to model the diverse spatio-temporal mobility patterns of different users in a uniform manner based on KG techniques is the first challenge. Second, the mobility patterns of users can be represented by multiple types of relational facts describing the different aspects of interactions between the location and time. For example, the fact of visiting some location at some time characterizes the interaction between location entities and time entities, while the fact of moving from one location to another location characterizes the interactions between different locations. Overall, both these complicated relational facts and the dependency between them reflect important aspects of human mobility, while the latter can be well captured via logical rules, which describe the logical relationship between relations in KG. How to extract and utilize useful underlying logical rules to model these multiple types of relational facts is another challenge.

In this article, we focus on utilizing the knowledge graph technique to model the spatio-temporal mobility behavior of users. Specifically, the spatio-temporal mobility patterns of users are modeled

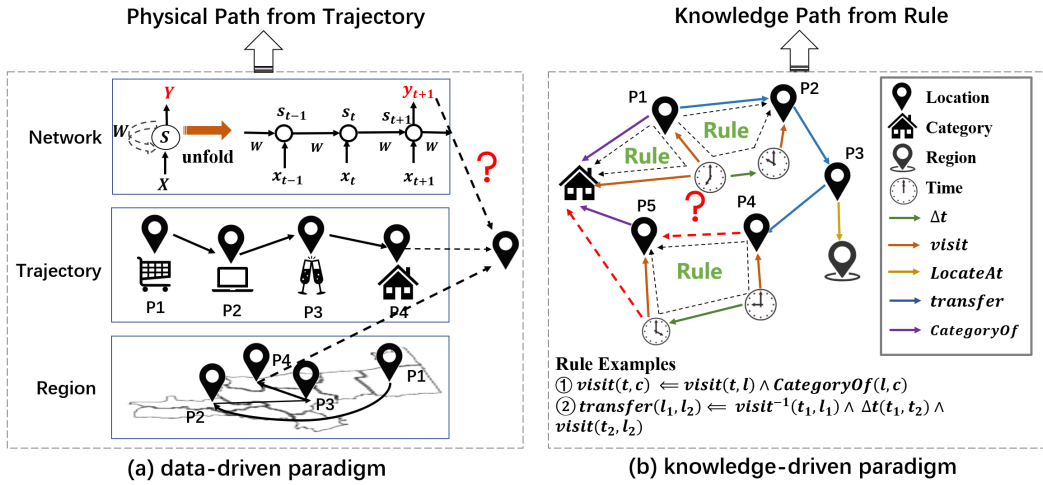


Fig. 1. Illustration of data-driven paradigm and knowledge-driven paradigm for mobility prediction.

by the facts belonging to elaborately designed relation types reflecting different aspects of human mobility. Further, in order to characterize the diversity of mobility patterns across different users, we introduce the concept of “user hyperplane.” The plausibility of fact is evaluated based on the projected embedding vectors of entities on different “user hyperplane,” which has a strong ability to characterize their difference. What’s more, in order to collaboratively learn different types of relational facts, we utilize the structure of logical rules and paths, which capture interpretable relationships and dependencies between differential relational facts. Through dedicated design on the logical rules extraction, more efficient logical rules are captured to characterize users’ complex mobility patterns. Besides, by using the logical rules to regularize the embedding vector of a fact and the path that can derive the fact based on the logic rules, different types of relational facts can be modeled in a cohesive manner by extracting the rich semantic structure between them.

In summary, our article makes the following contributions:

- We adopt a novel framework to model the user’s spatio-temporal mobility patterns as multi-type relational facts in the knowledge graph. Specifically, the mobility patterns are characterized by the relation between temporal entities and location entities under different “user hyperplanes.” Then, the mobility prediction task is solved by estimating the plausibility of the corresponding facts.
- We apply a dedicated logical rule extraction method to capture the valid logical rules, after that, creatively inject these rules and corresponding paths into the knowledge graph to collaboratively learn different types of relational facts in a cohesive manner, which is able to extract the rich semantic structure between them and better characterize the different aspects of mobility patterns.
- Extensive evaluations show that our proposed model beats state-of-the-art mobility prediction algorithms. Specifically, the **mean reciprocal rank (MRR)** is relatively improved by 7.78% on average. What’s more, logical rules have been verified to be beneficial, i.e., the MRR is relatively improved by 6.31% on average by introducing logical rules, demonstrating the effectiveness of our proposed model.

The structure of this article is as follows. We begin by briefly reviewing the related works systematically and underlining the difference of our work. Further, we introduce a mathematical model and formulate the problem. We then present details of our methodology and propose

the mobility prediction. Following our methodology, we evaluate and validate the prediction performance of our proposed approach. Finally, we conclude with a summary and possible future work.

## 2 Related Work

### 2.1 Mobility Prediction

In the past decade, mobility prediction has been widely investigated in spatio-temporal data mining [34], which aims to predict users' future mobility based on historical trajectories. Especially, the early methods focus on Markov model and its variations, representing locations as states while leveraging the transition matrix for prediction. For example, based on the clustered historical trajectories, **hidden Markov models (HMMs)** are firstly utilized to predict the user's next location [21]. Further, GMove [42] characterizes group-level mobility regularity via the ensemble of HMMs. Besides, Multi-Chain [32] builds several interconnected Markov chains to model various mobility patterns for prediction. On the other hand, several recent works explore deep learning for mobility prediction. Long- and Short-Term Preference Modeling [26] focuses on the temporal and spatial correlations between historical and current trajectories, and modifies RNN for robust prediction. DeepMove [7] combines **recurrent neural network (RNN)** and attention mechanism for interpretable mobility prediction. Following the similar design, DeepJMT [3] further performs joint mobility and time prediction. GETNext [40] incorporates users' preference, spatio-temporal context, and other relevant features together into a transformer model to better capture the mobility characteristics. SSDL [8] introduces a novel disentangled representation learning architecture to understand human time-independent and time-dependent mobility patterns. GaGASAN [35] utilizes a multi-scale time encoding technique and a self-attention mechanism to model different temporal patterns and capture long and short range contexts of sequence transitions, respectively. However, most existing mobility prediction models focus on sequence modeling, while ignore the semantic relationships among locations, time points as well as user behaviors, which is exactly what our proposed model achieves.

### 2.2 Knowledge Graph

Due to the limited observations, KG is incomplete and thus needs to be completed, which motivates several important related works [23]. For instance, TransE [1] completes missing links by measuring the distance between head and tail entities via relation translation. DistMult [37] and ComplEx [28] extend Canonical Polyadic decomposition for completion with symmetric and complex-valued embedding adopted, respectively. Recently, more works have begun to develop dynamic knowledge graph models instead of static KG. HyTE [5] achieves the temporal KG completion through projecting entities and relations to the timestamp space. DE-Simple [9] put forward diachronic entity embeddings to characterize the features timely with high expressive ability. TNTComplEx extends ComplEx to temporal KG, and achieves the state-of-the-art performance. Such works above only focus on the direct link in KG, while recent works further consider the logic rule in KG. Specifically, PTransE [13] extracts rule-based relation paths, which are combined with TransE to infer missing links between entities. RJPE [22] uses Horn rules to associate relations at the semantic level, and provides an interpretable KG completion framework. Moreover, both Neural LP [39] and DRUM [24] combine first-order logic with RNN to learn logic rules in KG for completion. Hence, such methods motivate our rule-enhanced KG for mobility prediction. Besides, there is also some research applying KG for user behavior modeling. For example, KGPMF [44] learns an urban movement KG via a multi-view learning framework. IMUP [33] constructs a spatial KG for mobile user profiling in a reinforcement learning framework, while KG-MUP [18] combines both KG and feature engineering for mobile user profiling. However, such constructed KG largely ignores the



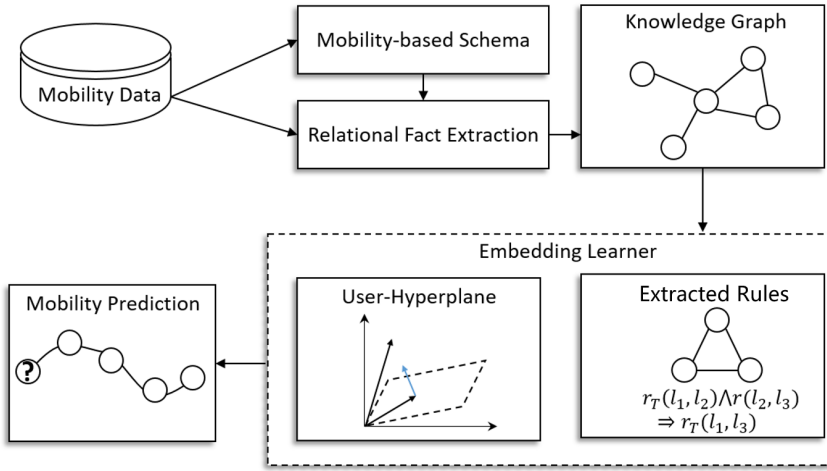


Fig. 2. The workflow of our proposed system.

temporal information, thus not applicable in our scenario. Other approaches incorporate temporal information in user behavior modeling. Specifically, STKG [31] constructs a spatio-temporal urban KG for user mobility prediction. STKGRec [2] builds a spatial-temporal KG from check-in sequences to implement the next **point-of-interest (POI)** recommendation. Although temporal information is considered in these approaches, they describe user behavior based on only a single type of relations, and thus are limited in describing the multiple aspects of user behavior. Different from them, we utilize multiple types of relations to describe the different aspects of user behavior, and further extract logical rules between relations to model their dependency.

### 3 System Overview

#### 3.1 Knowledge Graph

The knowledge graph is a directed graph structure defined on the nodes of entities. The edges connecting different entities correspond to the relational facts, of which the types are defined to the relations. Specifically, each relational fact in the knowledge graph is represented by a three-tuple  $(e_h, r, e_t)$  consisting of the head entity  $e_h$ , tail entity  $e_t$ , and relation  $r$ . The set of possible entities and relation types are organized to be the schema of the knowledge graph. Due to the strong ability of the knowledge graph model in terms of extracting structured knowledge from data and modeling multiple types of relations in a uniform manner, it has shown great success in numerous applications including language understanding and recommendation systems [15, 27]. In this article, we aim to model the human mobility behavior based on the knowledge graph model.

#### 3.2 System Overview

We show the workflow of our proposed system in Figure 2. As we can observe, our system first defines the mobility-based schema to point out the formats and types of entities and relations, which is then used to extract the corresponding relational facts in the constructed knowledge graph owing to the characteristics of multivariate relational facts in the spatio-temporal mobility data, we propose a KGE model based on the concept of user hyperplane. Besides, we extract and inject the logical rules into the knowledge graph to learn semantic information between paths and relations, which helps to capture efficient high-order information and diverse user. Finally, according to the learned embeddings of entities and relations in our knowledge graph, the obtained embedding

model is used to predict user future movement by evaluating the plausibility of the corresponding relational facts.

#### 4 Knowledge Graph Construction

In order to characterize the spatio-temporal mobility behavior based on the knowledge graph model, we must first design the mobility-based schema to define the entities and relations in the knowledge graph explicitly. Specifically, four different types of entities are considered, including the spatial venues, the regions, the categories, and the time units. The spatial venues represent the specific PoI, and we denote the set of spatial venues as  $\mathcal{L}$ . Moreover, the regions and categories denote the pre-divided areas and pre-defined PoI categories, respectively, which are denoted as  $\mathcal{R}$  and  $\mathcal{C}$ . As for the time units, following the extensively used timestamp preprocessing mechanism used in existing methods [36, 44], we divide one day into half-hour time intervals, and we further distinguish time bins belonging to working day and non-working day, i.e., totally 96 time units. We further represent the set of time units as  $\mathcal{T}$ . In addition, we define the set of all entities as  $\mathcal{E}$ . Then, between these entities, we define six different types of relations, including the spatio-temporal visitation relation, spatial transition relation, temporal closeness relation, spatial affiliation relation, spatial category relation, and similar visitation relation, to characterize the mobility behavior of users in the form of relational facts. The definitions of these relations are introduced in detail in the following.

The spatio-temporal visitation relation is defined to describe the visiting behavior to different locations at different times, which characterizes the interaction between locations and time units, where the locations can be PoIs, regions, or categories. Specifically, it can be formally defined as follows.

*Definition 1 (Spatio-Temporal Visitation Relation).* The spatio-temporal visitation relation  $r_V$  exists between the location  $l$  and time  $t$  if and only if a user visits the location  $l$  at time  $t$ , of which the corresponding fact is represented by a three-tuple  $(t, r_V, l)$ .

The spatial transition relation is defined to describe the transiting behavior of users between different locations, which characterizes the interaction between locations. Specifically, it can be formally defined as follows.

*Definition 2 (Spatial Transition Relation).* The spatial transition  $r_T$  exists between two different location  $l_1$  and  $l_2$  if and only if a user moves from the location  $l_1$  and  $l_2$  within a predefined temporal threshold  $T_{max}$ , of which the corresponding fact is represented by a three-tuple  $(l_1, r_T, l_2)$ .

In addition, we also define the temporal closeness relation to characterize the fine-grained structure between different time units. For example, two close time units should have similar embedding vectors. What's more, the temporal closeness relation can be used as a bridge to build path structure based on which the rich semantic information can be transmitted and extracted. The details will be introduced in the methodology section. Overall, the temporal closeness relation is formally defined as follows.

*Definition 3 (Temporal Closeness Relation).* The temporal closeness relation  $r_W(\Delta t)$  exists between two time units  $t_1$  and  $t_2$  if and only if the time gap between  $t_1$  and  $t_2$  is less than  $T_{max}$  and  $\Delta t = |t_1 - t_2|$ . Without loss of generality, we assume  $t_1 \leq t_2$ , then the corresponding relational fact is denoted by a three-tuple  $(t_1, r_W(t_2 - t_1), t_2)$ .

The spatial affiliation relation is defined to describe the affiliation property to different regions at different locations, which characterizes the interaction between spatial venues and regions. Specifically, it can be formally defined as follows.

*Definition 4 (Spatial Affiliation Relation).* The spatial affiliation relation  $r_L$  exists between the location  $l_1$  and the region  $r_1$  if and only if an arbitrary location  $l_1$  is located at the region  $r_1$ , of which the corresponding fact is represented by a three-tuple  $(l_1, r_L, r_1)$ .

The spatial category relation is defined as describing the category property to different categories at different locations, which characterizes the interaction between spatial venues and categories. Specifically, it can be formally defined as follows.

*Definition 5 (Spatial Category Relation).* The spatial category relation  $r_C$  exists between the location  $l_1$  and the category  $c_1$  if and only if the category of an arbitrary location  $l_1$  is the category  $c_1$ , of which the corresponding fact is represented by a three-tuple  $(l_1, r_C, c_1)$ .

The similar visitation relation is defined to describe the similar visitation property for two users who visit the same PoIs or have a higher cosine similarity of PoI category sequence, if and only if one user owns much historical data and the other user has scarce data. Generally speaking, two users sharing the same PoI visitation record or similar PoI category distribution should have similar mobility characteristics. Additionally, the defined relationship plays a significant role in assisting in embedding learning of users with scarce data. Overall, it can be formally defined as follows.

*Definition 6 (Similar Visitation Relation).* The similar visitation relation  $r_S$  exists between the user  $u_1$  and the other  $u_2$  if and only if the user  $u_1$  and user  $u_2$  have visited more than  $T$  identical PoIs or the cosine similarity of the visited PoI category sequence is greater than  $\theta$  and the number of one user's records is less than  $N$ , of which the corresponding fact is represented by a three-tuple  $(u_1, r_S, u_2)$ .

Based on the above-defined relations, we can extract relational facts from the observed spatio-temporal user mobility data. Further, users' future movements can also be represented by unobserved relational facts. Then, predicting users' future mobility can be converted to evaluating the plausibility of possible relational facts.

## 5 Methodology

### 5.1 Embedding Model Based on User Hyperplane

In order to predict the plausibility of unobserved relational facts, we follow the common practices of knowledge graph completion methods and utilize an embedding model to map each entity or relation type into a low-dimensional vector. Then, a scoring function of relational facts is defined based on the embedding vectors of entities and relations to measure the plausibility of relational facts. By training the embedding model on the observed facts, the scoring function is able to evaluate the plausibility of arbitrary unobserved relational facts.

Further, in order to characterize the diverse mobility patterns of different users in the framework of the knowledge graph, we model different users as different hyperplanes [5]. Then, the knowledge subgraph composed of relational facts belonging to each user is projected to the hyperplane to learn the personalized mobility patterns of the corresponding user.

For an arbitrary relational fact  $(e_h, r, e_t)$  belonging to the subgraph of user  $u$ , its plausibility is also evaluated on the hyperplane of user  $u$ . Specifically, we denote their embedding vectors as  $\mathbf{e}_h$ ,  $\mathbf{e}_t$ , and  $\mathbf{r}$ , respectively, and the subgraph of user  $u$  is denoted by  $\mathcal{G}^u$ . The user hyperplane is modeled by the normal vector denoted by  $\mathbf{u}$ . Based on the above definitions, we first compute their projected embedding vectors on the user hyperplane  $\mathbf{u}$  as Equation (1)

$$\begin{cases} \mathbf{e}_h^{(u)} = \mathbf{e}_h - (\mathbf{u}^T \mathbf{e}_h) \mathbf{u}, \\ \mathbf{e}_t^{(u)} = \mathbf{e}_t - (\mathbf{u}^T \mathbf{e}_t) \mathbf{u}, \\ \mathbf{r}^{(u)} = \mathbf{r} - (\mathbf{u}^T \mathbf{r}) \mathbf{u}. \end{cases} \quad (1)$$

Further, we adopt the complex-domain scoring function to evaluate the plausibility of relational facts, which have been proven to have a strong ability in capturing antisymmetric relations [12, 28]. Specifically, we convert each real-valued embedding vector  $e_h$  into the complex-valued vector by splitting it into two equal-sized parts, i.e.,  $e_h^{(u)} = [\hat{e}_h^{(u)}, \check{e}_h^{(u)}] = \hat{e}_h^{(u)} + i\check{e}_h^{(u)}$ . Further,  $\hat{e}_h^{(u)}$  is used as the real component, and  $\check{e}_h^{(u)}$  is used as the imaginary component to jointly form a complex vector in calculating the scoring function. Similar operations are also implemented on the relation  $r$  and entity  $e_t$ . Then, the scoring function of each tuple  $(e_h, r, e_t)$  under the user hyperplane  $u$  is computed by Equation (2)

$$\begin{aligned} f^{(u)}(e_h, r, e_t) &= \text{Re}(\langle \hat{e}_h^{(u)} + i\check{e}_h^{(u)}, \hat{r}^{(u)} + i\check{r}^{(u)}, \overline{\hat{e}_t^{(u)} + i\check{e}_t^{(u)}} \rangle) \\ &= \langle \hat{e}_h^{(u)}, \hat{r}^{(u)}, \hat{e}_t^{(u)} \rangle + \langle \hat{e}_h^{(u)}, \check{r}^{(u)}, \check{e}_t^{(u)} \rangle \\ &\quad + \langle \check{e}_h^{(u)}, \hat{r}^{(u)}, \check{e}_t^{(u)} \rangle - \langle \check{e}_h^{(u)}, \check{r}^{(u)}, \hat{e}_t^{(u)} \rangle, \end{aligned} \quad (2)$$

where  $\langle \cdot \rangle$  is the multi-linear dot product. For an arbitrary complex number  $z$ ,  $\bar{z}$  is its complex conjugate, and  $\text{Re}(z)$  is its real component. Furthermore, the plausibility of whether the fact  $(e_h, r, e_t)$  exists is modeled to be in proportion to its score  $f^{(u)}(e_h, r, e_t)$ , i.e., Equation (2). The cross-entropy loss function can evaluate the difference between the probability distribution output by the model and the true probability distribution. The larger the difference, the more punishment. By minimizing the gap between model predictions and labels, the prediction results are more accurate. Thus, the cross-entropy loss function is widely used in multi-classification problems [12, 29]. Then, the embedding model is trained by using the cross entropy of the tail entity as its loss, which can be calculated as Equation (3)

$$\ell^{(u)}(e_h, r, e_t) = -f^{(u)}(e_h, r, e_t) + \log \left( \sum_{e'_t \in \mathcal{E}} e^{f^{(u)}(e_h, r, e'_t)} \right). \quad (3)$$

Then, based on Equation (3) to calculate the relational fact loss, the total loss for all relational facts can be expressed by:

$$\mathcal{L}_C = \sum_{u \in \mathcal{U}} \sum_{(e_h, r, e_t) \in \mathcal{G}^u} \ell(e_h, r, e_t). \quad (4)$$

## 5.2 Logical Rules Extraction

Learning the embedding model based on the loss function Equation (4) cannot well capture the dependence between relational facts. For example, the combination of several relational facts is possible to derive another relational fact, which is ignored by the optimization only based on Equation (4). On the other hand, the dependence between relational facts can serve as the efficient information propagation path. For example, we can infer whether the relationship exists in the KG by integrating high-order information from other relational facts. Thus, the dependence between relational facts should be learned to extract rich semantics and high-order information. In order to achieve this goal, we utilize the structure of logical rules and paths [13, 22].

For example, the structure of two entities connected by a certain multi-step path can be used to derive another relational fact between these two entities. Specifically, the multi-step paths in the knowledge graph contain rich semantic information. Thus, by injecting logic rules in the knowledge graph, interpretable relationships and dependencies between differential relational facts can be captured [22].

To obtain the explainable and implicit rules from the knowledge graph, we utilize Neural LP [39] to extract logical rules and corresponding confidence automatically, which overcomes the

disadvantages of utilizing fixed rules in previous papers [17]. Specifically, the relation path extraction is to retrieve a ranked list of the relational facts by maximizing the score of the query. For example, as shown in Equation (5), suppose given the query  $(l_1, r_T, l_2)$ , i.e., a user visits two different spatial venues  $l_1$  and  $l_2$  at time units  $t_1$  and  $t_2$  within the temporal threshold  $T_{max}$ , we can obtain a weighted relation path  $(l_1, r_V^{-1}, t_1) \wedge (t_1, r_W, t_2) \wedge (t_2, r_V, l_2)$ , where  $r_V^{-1}$  is the inverse relation of  $r_V$  and the weight denotes the confidence correlated with this path

$$query(l_1, r_T, l_2) \leftarrow (l_1, r_V^{-1}, t_1) \wedge (t_1, r_W, t_2) \wedge (t_2, r_V, l_2). \quad (5)$$

Thus, our purpose is to obtain the rules and corresponding confidences  $\alpha$ . Specifically, followed the operator  $M_R$  representing each relation  $R$  in TensorLog [4], what we want to learn is shown as follows,

$$\sum_l \alpha_l \prod_{k \in \beta_l} M_{R_k}, \quad (6)$$

where  $l$  contains all possible rules,  $\alpha_l$  is the confidence of the rule  $l$  and  $\beta_l$  is an ordered list of all relations in rule  $l$ . Then, the score candidate entity  $e_t$  given an arbitrary entity  $e_h$  equals the inner products between the logical rules and head entity embedding  $e_h$ , just as Equation (7)

$$score(e_t|e_h) = e_t^T \left( \sum_l \left( \alpha_l \left( \prod_{k \in \beta_l} M_{R_k} e_h \right) \right) \right), \quad (7)$$

where  $e_h$  and  $e_t$  denote the embeddings of  $e_h$  and  $e_t$ , respectively. Owing to the prediction task, we only focus on the two types of queries, i.e., relational facts including spatio-temporal visitation relation  $r_T$  and spatial transition relation  $r_V$ . Thus, we can optimize the model by maximizing the score of the query in our constructed knowledge graph, as shown in Equation (8)

$$\max_{(\alpha_l, \beta_l)} \sum_{\{(e_h, e_t)\}} score(e_t|e_h) = \max_{(\alpha_l, \beta_l)} \sum_{\{(e_h, e_t)\}} e_t^T \left( \sum_l \left( \alpha_l \left( \prod_{k \in \beta_l} M_{R_k} e_h \right) \right) \right). \quad (8)$$

Finally, we can recover the logical rules and confidences according to the optimized parameters  $\alpha_l$  and  $\beta_l$ .

### 5.3 Injecting Logic Rules

In order to utilize the rule structure, we first learn the representation of paths corresponding to different rules. Different from the common path representation method adopted in [13, 22], which only utilizes the embedding of the relations of facts along the path, we argue that the entities along the path are also important. For example, in the second rule shown in Figure 1(b), if  $l_1$  and  $l_2$  are the home and workplace of a user, respectively, the spatial transition relational facts between them occur with larger probability with certain time gaps, e.g., 8 hours. The intuition here is that the entities  $t_1$  and  $t_2$  in this path play a critical role in determining the plausibility of the corresponding facts. Thus, the entities along the paths should also be considered in calculating the path representations. Thus, we utilize separate neural networks to extract the embedding of paths with different lengths. By defining  $p$  as the concatenated vector of all entities and relations along the path  $p$  except the head entity of the first fact and the tail entity of the last fact, the path representation vector  $C(p)$  can be calculated as follows:

$$C(p) = \text{MLP}(p). \quad (9)$$

**Algorithm 1:** Training Algorithm

---

**Input:** Constructed knowledge graph  $\mathcal{G}$  as the training set.  
**Output:** embeddings  $\mathcal{U}^E, \mathcal{T}^E, \mathcal{L}^E, \mathcal{R}^E$ .  
**Initialize:** initialise embeddings  $\mathcal{U}^E, \mathcal{T}^E, \mathcal{L}^E, \mathcal{R}^E$ .  
**for**  $i \in \{1, \dots, N_{\text{epoch}}\}$  **do**  
     $\mathcal{S} \leftarrow \mathcal{G}$ ;  
    **while**  $\mathcal{S} \neq \emptyset$  **do**  
        Sample a mini-batch  $\mathcal{S}_{\text{batch}} \subset \mathcal{S}$   
         $\mathcal{S} \leftarrow \mathcal{S} / \mathcal{S}_{\text{batch}}$ ;  
         $\mathcal{L} \leftarrow 0$ ;  
        **for**  $(e_h, r, e_t) \in \mathcal{S}_{\text{batch}}$  **do**  
             $f(e_h, r, e_t) \leftarrow$  compute the score using Equation (2);  
             $\ell(e_h, r, e_t) \leftarrow$  compute the loss using Equation (3);  
             $\mathcal{L} \leftarrow \mathcal{L} + \ell(e_h, r, e_t)$ ;  
            **if**  $P(e_h, e_t) \neq \emptyset$  **then**  
                **for**  $p \in P(e_h, e_t)$  **do**  
                     $\mathcal{L} \leftarrow \mathcal{L} + E(p, r)$ ;  
                **end**  
            **end**  
        **end**  
        Update parameters of embeddings w.r.t the gradients using  $\nabla \mathcal{L}$ ;  
    **end**  
**end**

---

Then, based on the obtained path representation, the similarity between paths and relations is evaluated by the following energy function [22]

$$E(p, r) = \|C^{(u)}(p) - \mathbf{r}^{(u)}\|. \quad (10)$$

We further denote the set of paths that satisfy the form listed in Table 4 and link entities  $e_h$  and  $e_t$  as  $P(e_h, e_t)$ . Then, the loss function derived from the rule structure can be represented as follows

$$\mathcal{L}_R = \sum_{(e_h, r, e_t) \in \mathcal{G} \wedge p \in P(e_h, e_t)} \alpha * E(p, r), \quad (11)$$

where  $\alpha$  represents the confidence of the corresponding relation path.

Due to the above equation of the loss regarding reconstructing the knowledge graph and modeling the dependence between relational facts based on rules, the total loss can be presented as follows:

$$\mathcal{L} = \mathcal{L}_C + \lambda \mathcal{L}_R, \quad (12)$$

where  $\lambda > 0$  is used to balance the loss function of reconstruction loss and logical rule loss. Overall, we can train our model by minimizing the total loss Equation (12), and obtain the embedding vectors of all entities in the knowledge graph, which will be further used in the mobility prediction task.

We present the pseudocode describing the process of learning embedding vectors of all entities and relations in the constructed knowledge graph in Algorithm 1. As we can observe, in each epoch, this algorithm samples a mini-batch of facts in the training set  $\mathcal{S}$ . For each fact in the mini-batch, its loss is calculated based on reconstruction loss and logical rule loss. First, the proposed model evaluates the plausibility of the existence of the relational fact and calculates the reconstruction



**Algorithm 2:** Mobility Prediction Based on KG

**Input:** User  $u$ , the target time  $t$  or the previously visited spatial venue  $l_0$  and the corresponding time  $t_0$  (unknown variables with the value of  $\emptyset$ ).

**Output:** The mobility prediction result  $l \in \mathcal{L}$ .

```

if  $t \neq \emptyset \wedge l_0 = \emptyset$  then
   $l = \arg \max_l f^{(u)}(t, r_V, l);$ 
end
if  $t = \emptyset \wedge l_0 \neq \emptyset$ ; then
   $l = \arg \max_l f^{(u)}(l_0, r_T, l);$ 
end
if  $t \neq \emptyset \wedge l_0 \neq \emptyset$  then
   $l = \arg \max_l f^{(u)}(t, r_V, l) + f^{(u)}(l_0, r_T, l);$ 
end

```

loss based on Equations (2) and (3), respectively. Then, according to the given head entity  $e_h$  and tail entity  $e_t$ , judge whether its relational path set  $P(e_h, e_t)$  is empty. If not empty, traverse all the paths to calculate the logical rule loss derived from the rule structure according to Equation (11). Finally, this algorithm updates embeddings of entities and relations based on the gradient of the total loss.

## 5.4 Mobility Prediction

Similarly, the pseudocode describing the process of implementing mobility prediction based on the knowledge graph model is shown in Algorithm 2. As we can observe, given user  $u$  and the corresponding target time  $t$  or the previously visited location  $l_0$ , this algorithm will implement mobility prediction based on different types of relational facts. Specifically, if only the target time is given, the future movement is predicted by evaluating the plausibility of the corresponding spatio-temporal visitation relational fact. The approach is similar when only the previously visited spatial venue is given. Differently, if both the target time and the previously visited spatial venue are given, the mobility prediction result is given by maximizing the plausibility of the two different relational facts simultaneously, which is derived from the total loss function Equation (12). Note the loss in terms of the utilized rules is not calculated in the mobility prediction task, since it is only influenced by the embedding vectors of the relations and the passing entities of the paths, which is not the function of the tail entities of the paths. Thus, it can be ignored in Algorithm 2.

## 6 Experiments

### 6.1 Experiment Settings

**6.1.1 Datasets.** In our experiment, two representative real-world mobility datasets are processed to evaluate the performance of our proposed model. The first one is a real-world mobile application dataset collected when users request location service through their applications in Beijing. The mobile application dataset contains users' trajectories between 17 September and 31 October in 2016. Each record consists of the user ID, timestamp, and PoI, i.e., the user requests location service of WeChat like search, check-in, etc. Table 1 illustrates two example mobility records of user  $u_1$  and user  $u_2$ , where  $l_1, l_2, l_3$  are different spatial venues and time is defined in like manner. We can observe that the different users hold distinguishing granularity and length and various mobility patterns.

Table 1. Two Simple Example Trajectories of Users

User	Trajectories				
user $u_1$	$(t_1, l_1)$	$(t_2, l_3)$	$(t_3, l_2)$	-	-
user $u_2$	$(t_1, l_3)$	$(t_2, l_1)$	$(t_3, l_2)$	$(t_4, l_1)$	$(t_5, l_3)$

Table 2. Statistics of Datasets Used in the Experiments

Dataset	#Users	#PoIs	#Regions	#categories	#Records
Beijing	1,027	8,168	256	14	206,551
Foursquare	1,082	16,945	1,611	11	46,041

The other is the public Foursquare check-in data from New York, collected from Foursquare API from 3 April 2012, to 16 February 2013. This dataset spans almost one year and contains various mobility patterns, collected from a previous work [38]. In Foursquare, the location information is collected when users actively share their corresponding locations with their friends and the public, resulting in the sparsity of the Foursquare dataset and few records for some users.

Finally, the experiments are conducted in terms of the 7:1:2 division mode. Specifically, the recorded trajectories of each user are ranked according to the time series first, then, the first 70% of each user's trajectory data are trained on our proposed model, and the last 20% are leveraged to evaluate the performance of our method, i.e., predicting the users' future location based on their historical trajectories.

The statistics of datasets are shown in Table 2.

**6.1.2 Evaluation Protocols.** Four principle evaluation metrics are introduced to assess the performance on mobility prediction of our proposed model: the MRR and Accuracy@k ( $k = 1, 5, 10$ ). The mathematical formula is shown as follows

$$MRR = \frac{1}{U \cdot N} \sum_{u=1}^U \sum_{i=1}^N \frac{1}{rank_{u,i}}, \quad (13)$$

$$Accuracy@k = \frac{1}{U \cdot N} \sum_{u=1}^U \sum_{i=1}^N rank_{u,i} \leq k, \quad (14)$$

where  $U$  represents the number of users, and  $N$  denotes the length of user  $u$ ' mobile records,  $rank_{u,i}$  represents the rank of  $i_{th}$  ground truth in the whole candidate locations of user  $u$ ' trajectories. We evaluate the performance by computing the mean results of all the users in terms of all metrics. MRR is the average reciprocal of ranks of predicting results, ranging from 0 to 1. Specifically, achieving high MRR means the effectiveness of our proposed method. Accuracy@k demonstrates the average probability of the ground truth appearing in the top-k candidate locations. Similarly, higher Accuracy@k can verify the superiority of the proposed method. In general, MRR and Accuracy@k metrics are utilized widely in prediction problems like link prediction [22], destination prediction [26], etc. Furthermore, the accurate prediction is meaningful significantly not only for the location recommendations for business [41] and the government's regulation on transportation [20].

**6.1.3 Baselines for Comparison.** To testify the superiority of our proposed approach, we compare it with the following two types of baselines, where the first three baselines are widely used mobility prediction algorithms, and the others are representative knowledge graph completion models.

Table 3. Comparison on Methodology

Models	Structured “Knowledge”	Long-Range Dependency	User Diversity	Data Imbalance
APHMP	✗	✓	✗	✗
DeepMove	✗	✓	✓	✗
STAN	✗	✓	✗	✗
SACN	✓	✗	✓	✗
CompGCN	✓	✗	✗	✗
De-Distmult	✓	✗	✗	✗
xERTE	✓	✓	✓	✗
TNTComplEx	✓	✗	✗	✗
Our	✓	✓	✓	✓

*RNN-Based Models.* *DeepMove* [7] considers the attention module into the RNN to capture well the long-term dependencies between the future mobility and the historical trajectories. *APHMP* [6] applies an attention module to capture useful information in historical mobility records and integrates it into the RNN to model users’ mobility patterns.

*Self Attention-Based Model.* *STAN* [19] aggregates all relevant visits in human mobility records and recalls the most plausible location candidates through weighted representations to achieve location prediction.

*GNN-Based Models.* *SACN* [25] utilizes a weighted graph convolutional network as an encoder to learn the graph connectivity structure, and learned weights to represent the information propagation from neighbors. *CompGCN* [29] jointly embeds nodes and relations in a graph to learn embeddings by leveraging composition operations and demonstrates superiority on the link prediction task.

*KG-Based Models.* *TNTComplEx* [12] proposes a ComplEx [28] decomposition of four-order tensor characterizing the temporal scopes of each triple and incorporates a non-temporal component to represent the static component, which improves accuracy greatly on the temporal knowledge graph completion task. *DE-Distmult* [9] leverages a diachronic entity embedding function to characterize the entities in the corresponding time for temporal knowledge graph completion. *xERTE* [11] iteratively samples relevant edges of entities in the subgraph and applies an attention mechanism on the sampled edges to achieve temporal knowledge graph completion.

**6.1.4 Qualitative Analysis.** Before conducting the experiments, we first analyze the differences between our proposed model and the baselines from a qualitative perspective, containing complexity and methodology.

Considering the complexity, the models of APHMP, DeepMove, and STAN all have about a quadratic growth of the time complexity with the length of the historical trajectories due to the quadratic computational complexity of the attention mechanism, while GNN-based models and KG-based models do not have this issue. These models only require to iterate over each record without the computational time influenced by the historical trajectories, indicating linear complexity with the problem size.

As for methodology, Table 3 illustrates the ability of our proposed model and baselines in terms of four aspects, containing Structured “Knowledge,” Long-Range Dependency, User Diversity, and Data Imbalance. Structured “Knowledge” means incorporating “knowledge” from multiple sources in a cohesive manner to better model user mobility behaviors. Long-range dependency and User Diversity mean capturing the long-range dependency and user-personalized mobility patterns of trajectories, respectively. The algorithm with a check mark in the column Data Imbalance indicates that it can alleviate the data imbalance issue and better model users with scarce data.

Table 4. The Extracted Top-5 Rules on Two Datasets

Beijing Confidence	Foursquare Confidence	Paths	Derived Relations
0.974	0.561	$(l_1, r_V^{-1}, t_1) \wedge (t_1, r_W(\Delta t), t_2) \wedge (t_2, r_V, l_2)$	$(l_1, r_T, l_2)$
0.887	0.926	$(t_1, r_V, l_1) \wedge (l_1, r_L, r_1)$	$(t_1, r_V, r_1)$
0.396	0.474	$(t_1, r_V, l_1) \wedge (l_1, r_C, c_1)$	$(t_1, r_V, c_1)$
0.087	0.278	$(l_1, r_T, l_2) \wedge (l_2, r_L, r_2)$	$(l_1, r_T, r_2)$
0.670	0.456	$(l_1, r_T, l_2) \wedge (l_2, r_C, c_2)$	$(l_1, r_T, c_2)$

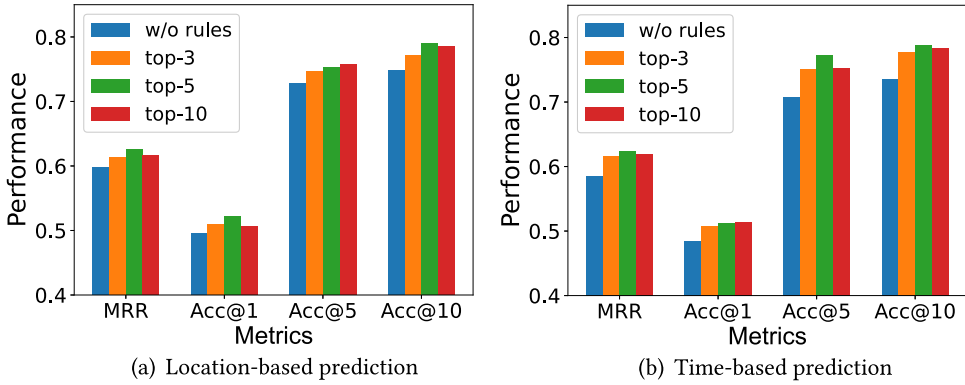


Fig. 3. The performance with top-k rules on two scenarios in terms of four metrics.

**6.1.5 Parameter Setting.** In our experiments, the default embedding dimension of users, locations, and relations are all set as 40. In these two datasets, the embedding regularization is 0.01, and the learning rate ranges from 0.0005 to 0.01. Besides, the predefined temporal threshold  $T_{max}$  is set as 24. In addition, we apply grid search to select the hyper-parameter  $\lambda$ . We manually finetune the hyper-parameter  $\lambda$  in 0, 0.5, 1, 2 to balance the loss in terms of reconstructing the knowledge graph and modeling the dependence between relational facts based on rules. The hyper-parameter  $\lambda$  is assigned as 0.5, where our proposed algorithm achieves the optimal results on validation sets.

## 6.2 Experiment Results

In this section, we demonstrate the extracted rules and evaluate the performance of our proposed approach and baselines on the two datasets in terms of four metrics and two scenarios, including the comparative experiments, ablation study, case study, etc.

**6.2.1 Extracted Rules.** Table 4 in Appendix demonstrates the extracted five rules in our experiments, where the first and second rows present the confidences of the corresponding rules on the two datasets, respectively. From the extracted rules, we can observe that there exist complicated and implicit interactions between times and locations as well as locations and locations, which are stored in the relation paths. Besides, we select the top-k ( $k = 0, 3, 5, 10$ ) rules from the extracted rules for mobility prediction to analyze the performance variation with the number of rules. From the evaluation results displayed in Figure 3, we can observe two interesting insights. (1) our model based on rules acquires much semantic information from the extracted rules and achieves better performance than without rules in terms of four metrics; (2) the extracted rules with higher confidence are more efficient on mobility prediction than lower confidence. For example, our model

Table 5. Performance on the Beijing Dataset

		Location-based Prediction				Time-based Prediction			
		Accuracy@k			MRR	Accuracy@k			MRR
		<i>k</i> = 1	<i>k</i> = 5	<i>k</i> = 10		<i>k</i> = 1	<i>k</i> = 5	<i>k</i> = 10	
RNN-based	APHMP	47.20%	67.69%	70.48%	56.45%	-	-	-	-
	DeepMove	46.83%	68.38%	70.89%	56.48%	-	-	-	-
Self attention-based	STAN	48.33%	68.67%	72.24%	58.21%	-	-	-	-
GNN-based	SACN	49.91%	71.2%	74.9%	59.38%	48.40%	70.20%	74.47%	58.30%
	CompGCN	51.73%	68.86%	72.61%	59.44%	49.65%	67.63%	71.0%	57.56%
KG-based	De-Distmult	48.4%	69.18%	71.74%	57.69%	48.71%	69.01%	71.16%	58.47%
	xERTE	49.79%	70.66%	72.78%	59.14%	50.19%	71.63%	73.88%	59.78%
	TNTComplEx	49.86%	73.67%	76.53%	60.15%	49.93%	72.22%	74.89%	59.87%
	Our	<b>52.15%</b>	<b>77.32%</b>	<b>80.96%</b>	<b>63.09%</b>	<b>51.41%</b>	<b>77.24%</b>	<b>79.41%</b>	<b>62.87%</b>

Bold denotes best (highest) results, and italic indicates different metrics.

with top-5 rules obtains better performance than with top-10 rules. The reason may be that the lower-confidence rules have side effects on embedding learning and then influence prediction performance. Thus, the following experiments are all conducted on the situation using top-5 rules to evaluate the performance of mobility prediction.

**6.2.2 Overall Performance.** Considering different prediction methods, the Location-based prediction represents predicting users' future location given a previously visited location, and the Time-based prediction means predicting users' location at the given time. From the comparative results in Table 5, we can draw several significant conclusions. First, compared with the RNN-based model, the KG embedding models most achieve better performance on the Beijing dataset. Compared with the RNN-based and Self attention-based baselines, our proposed method obtains an improvement of 6.11% and 4.38% in terms of MRR on the Location-based prediction. This demonstrates that KG embedding models could better learn user embeddings, time, and locations by integrating the structured "knowledge" to achieve better mobility prediction. In addition, our proposed model consistently outperforms other KG embedding models in terms of all metrics. In particular, our model achieves a 4.9% performance gain on the Location-based Prediction compared with De-Distmult. It demonstrates that the relation paths indeed supply an excellent supplement for representation learning and capture the rich semantic information of mobile trajectories. Besides, owing to the restrictions on the symmetric relations of Distmult [37], TNTComplEx achieves better performance than De-Distmult, which demonstrates the effectiveness of the complex vectors. GNN-based models apply GCNs as the encoder to characterize the graph structure and KGE models to capture the interactions between entities and relations. CompGCN jointly generates expressive entity and relation embeddings, contributing to capturing the interactions between entities and relations and achieving higher accuracy than SACN, but still less effective than our model, which verifies the superiority of simultaneously capturing the semantic information of paths and entities and relations properties of our proposed model. Generally, the results illustrate that our proposed model outperforms all the baselines in terms of all metrics, which verifies the superiority of the KGE methods. Besides, the results also illustrate the rules indeed extract rich semantic information from the paths and construct the semantic association between relations, assisting learning relation embeddings and achieving better performance.

Furthermore, we implement the experiments on the Foursquare dataset, where the mobility records are sparser. As shown in Table 6, the superiority of KG embedding is not apparent compared with the Beijing dataset. The reason may be that RNN-based and Self attention-based models capture users' mobility patterns more effectively on the long-term user behavior modeling problem. Even so,

Table 6. Performance on the Foursquare Dataset

		Location-based Prediction				Time-based Prediction			
		Accuracy@k			MRR	Accuracy@k			MRR
		k = 1	k = 5	k = 10		k = 1	k = 5	k = 10	
RNN-based	APHMP	6.11%	13.14%	16.21%	9.49%	-	-	-	-
	DeepMove	8.36%	14.99%	16.76%	11.28%	-	-	-	-
Self attention-based	STAN	7.89%	15.65%	19.83%	12.72%	-	-	-	-
GNN-based	SACN	8.13%	16.84%	19.59%	12.22%	22.48%	26.85%	15.39%	
	CompGCN	8.66%	14.15%	16.71%	11.45%	11.57%	17.72%	19.45%	14.43%
KG-based	De-Distmult	8.83%	15.0%	16.81%	11.29%	9.35%	17.11%	22.61%	14.07%
	xERTE	8.18%	18.62%	20.88%	12.76%	10.94%	19.67%	22.85%	15.28%
	TNTCompLex	8.31%	17.97%	21.69%	12.85%	12.76%	19.81%	22.73%	16.23%
	Our	<b>9.43%</b>	<b>21.43%</b>	<b>25.12%</b>	<b>15.09%</b>	<b>13.87%</b>	<b>25.28%</b>	<b>29.37%</b>	<b>19.24%</b>

Bold denotes best (highest) results, and italic indicates different metrics.

Table 7. Prediction Performance of Ablation Study

Models	Location-based Prediction				Time-based Prediction			
	Accuracy@k			MRR	Accuracy@k			MRR
	k = 1	k = 5	k = 10		k = 1	k = 5	k = 10	
Our	<b>52.15%</b>	<b>77.32%</b>	<b>80.96%</b>	<b>63.09%</b>	<b>51.41%</b>	<b>77.24%</b>	<b>79.41%</b>	<b>62.87%</b>
-User hyperplane	49.17%	68.46%	71.55%	57.89%	48.32%	66.35%	68.72%	56.57%
-MLP	50.65%	75.67%	78.49%	61.67%	50.34%	75.5%	78.33%	61.36%
-Rule	49.49%	72.8%	74.76%	58.80%	48.42%	70.73%	73.56%	58.50%

Bold denotes best (highest) results, and italic indicates different metrics. MLP, multilayer perceptron.

our model achieves 4.07% and 4.03% performance gain in terms of Accuracy@5 and Accuracy@10 compared with STAN. Generally, as can be expected, our proposed method could learn better embeddings and achieve more accurate prediction than baselines without the limitation of the dataset's characteristics, which further demonstrates the effectiveness of employing rules for integrating paths to solve the mobility prediction problem.

**6.2.3 Ablation Study.** To verify the effectiveness of different components of our proposed model, we implement the ablation study on the Beijing dataset. Specifically, -User hyperplane means removing the module of the user hyperplane, i.e., without user information. In the experiment, we define  $\mathbf{p}$  as the concatenated vector of all the entities and relations along the path  $p$  to obtain the path representation vector except the head entity and tail entity of two sides. **-Multilayer Perceptron (MLP)** means characterizing the path representation with only the relations along the path  $p$ , abandoning the middle entities within the path. Finally, -Rule means removing the extracted rules and corresponding relation paths and modeling the defined relationships without injecting logical rules.

As shown in Table 7, we can observe that removing the user hyperplane will lead to significant performance degradation with 4.7% and 5.78% in terms of MRR on the Location-based Prediction and Time-based Prediction. The user hyperplane module is proposed to characterize the diverse mobility patterns of different users. From the results, we can conclude that our proposed method captures the characteristics of personalized mobility patterns and consistently achieves better performance in terms of four metrics by constructing the subgraph of the corresponding user.

Furthermore, the performance variation of the model discarding the MLP operator demonstrates the entities of the middle facts in the path contain important semantic information of mobility behaviors. Last, we attempt to delete the extracted rules and corresponding relation paths to



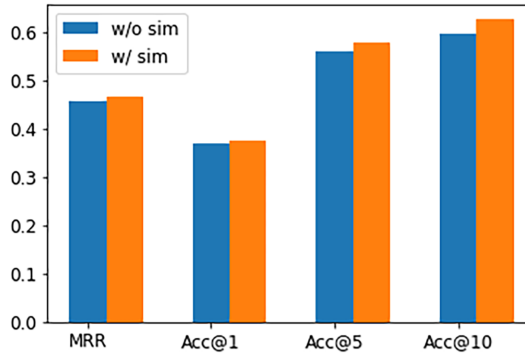


Fig. 4. Performance variation with or without similar visitation relation defined on the Beijing dataset in terms of four metrics.

evaluate the performance. As we can find, our proposed model obtains an improvement of 3.79% and 3.85% in terms of MRR on Location-based Prediction and Time-based Prediction, respectively, compared to the models without rules. This demonstrates that employing rules for extracting more semantic information and integrating paths facilitates relation embeddings and mobility prediction.

**6.2.4 Imbalance Study.** Considering the situation that some users have abundant data in the mobile records, while some have scarce records, directly training the embedding model easily leads to the data imbalance issue. Our proposed model defines the similar visitation relation in the constructed knowledge graph to alleviate the above problem. To verify the efficiency of our model, we filter the users whose records are less than 30 and observe the model performance variation with or without the similar visitation relation. Figure 4 shows the comparison results of the prediction performance in terms of four evaluation metrics, where w/ sim denotes to include the similar visitation relation when constructing the knowledge graph, and w/o sim means not to include. It can be seen that after including this relation, users with scarce data have obvious performance gains in the four metrics, and achieve 2.03% and 3.07% in terms of Acc@5 and Acc@10, respectively. This experiment demonstrates that our model indeed captures the dependencies between users efficiently and assists the embedding learning of users with scarce data. Further, it also verifies that our model has indeed alleviated the problem of data imbalance to a certain extent by utilizing the similar visitation relation to capture the dependencies between users with similar visitation properties.

### 6.3 Case Study

To explore and analyze the performance variation of different users and PoIs, we utilize two rules to select parts of the mobility records and conduct case study experiments. Specifically, the users are clustered into different groups by record number and the radius of gyration [10]. Besides, each user constructs their own knowledge graph independently from one another to demonstrate the capability of capturing user mobility characteristics; similarly, the PoIs are divided by PoI visit frequency and  $\Delta t$  between the adjacent mobility records. The number of records indicates the length of the corresponding user's mobility records. Specifically, a larger number of records means a denser trajectory during the same period, reflecting the sparsity level of the user's mobility on the temporal dimension. The radius of gyration reflects the spatial range of user mobility, representing the entropy of locations in the spatial dimension. For the group of PoIs, PoI visit frequency denotes the visit frequency of certain PoI, representing the visitation popularity of users, while the  $\Delta t$

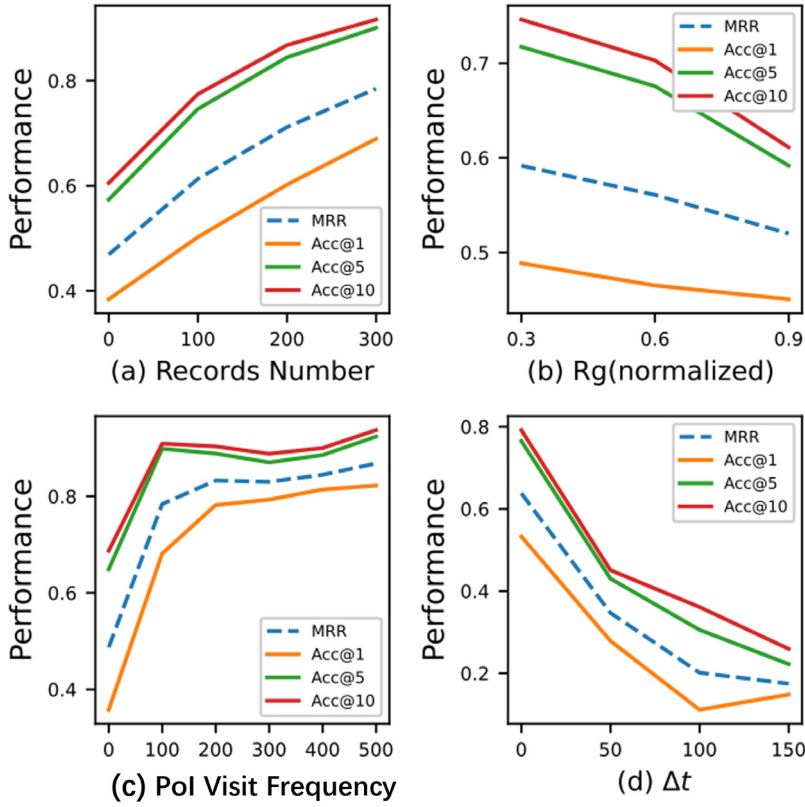


Fig. 5. Performance varies with the number of mobility records/radius of gyration (rg)/PoI visit frequency/ $\Delta t$  on the Beijing dataset in terms of four metrics.

records the time intervals between adjacent mobility records, characterizing the temporal closeness level. The units of  $\Delta t$  and the radius of gyration are hours and kilometers, respectively.

From the evaluation performance in terms of four metrics in Figure 5, we can observe that our model achieves better performance as the number of records and PoI visit frequency increase. Conversely, the prediction accuracy decreases gradually as the radius of gyration and  $\Delta t$  increase. Specifically, from Figure 5(a) and (c), we can infer that entity and relations in the constructed knowledge graph with denser trajectories and higher PoI visit frequency can learn better embeddings and more mobility characteristics, which helps to achieve higher mobility prediction accuracy and also verifies the necessity of defining the similar mobility relation introduced before. Figure 5(b) demonstrates that the expansion of the spatial range of mobility brings more noises and bigger entropy, which influences the performance of mobility prediction, that is, the user tending to visit distant spatial venues has a bigger radius of gyration, making their mobility hard to predict. From Figure 5(d), we can draw the conclusion that the performance variation of  $\Delta t$  reflects that the spatial transition relation does not work with longer intervals between mobility records. Overall, the current mobility is mostly not affected by long-ago trajectories, but only related to the needs of the corresponding time, which is also consistent with two patterns about mobility prediction. Generally speaking, the experiments analyze the prediction performance through clustering the mobility records based on users and PoIs, which is the basis of model design.

## 7 Conclusions and Future Work

In this article, we focus on modeling human mobility behavior based on the knowledge graph technique. By proposing six different types of relations characterizing various aspects of human mobility in terms of temporal and spatial dimensions, we further propose an embedding model based on user hyperplane to model the diversity of mobility patterns of different users. Additionally, the logic rules are injected into the knowledge graph to model the semantic information between multi-step paths and relations. Then, the obtained embedding vectors of entities and relations are used to predict users' future movement by evaluating the plausibility of corresponding facts. Extensive experimental results show that our proposed model beats the state-of-the-art mobility prediction algorithm by improving the prediction accuracy for around 7.78% relatively on average in terms of MRR, demonstrating its effectiveness. In future work, we will consider combining much more information about the locations, such as the latitude and longitude. On the other hand, considering more fine-grained user features to derive a better user hyperplane model is one research direction.

## References

- [1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS '13)*, Vol. 26, 2787–2795.
- [2] Wei Chen, Huaiyu Wan, Shengnan Guo, Haoyu Huang, Shaojie Zheng, Jiamu Li, Shuohao Lin, and Youfang Lin. 2022. Building and exploiting spatial-temporal knowledge graph for next POI recommendation. *Knowledge-Based Systems* 258 (2022), 109951.
- [3] 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 (WSDM '20)*, 106–114.
- [4] William W. Cohen. 2016. Tensorlog: A differentiable deductive database. arXiv:1605.06523. Retrieved from <https://doi.org/10.48550/arXiv.1605.06523>
- [5] Shib S. Dasgupta, Swayambhu N. Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2001–2011.
- [6] Zipei Fan, Xuan Song, Renhe Jiang, Qunjun Chen, and Ryosuke Shibasaki. 2019. Decentralized attention-based personalized human mobility prediction. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (2019).
- [7] Jie Feng, Yong Li, Zeyu Yang, Qiang Qiu, and Depeng Jin. 2020. Predicting human mobility with semantic motivation via multi-task attentional recurrent networks. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* PP, 99 (2020), 1–1.
- [8] Qiang Gao, Jinyu Hong, Xovee Xu, Ping Kuang, Fan Zhou, and Goce Trajcevski. 2023. Predicting human mobility via self-supervised disentanglement learning. *IEEE Transactions on Knowledge and Data Engineering* 36, 5 (2023), 2126–2141. DOI: <https://doi.org/10.1109/TKDE.2023.3317175>
- [9] Rishab Goel, Seyed M. Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, 3988–3995.
- [10] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. 2008. Understanding individual human mobility patterns. *Nature* 453, 7196 (2008), 779–782.
- [11] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2021. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *Proceedings of the International Conference on Learning Representations*.
- [12] Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. arXiv:2004.04926. Retrieved from <https://doi.org/10.48550/arXiv.2004.04926>
- [13] Yankai Lin, Zhiyuan Liu, Huan-Bo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '15)*, 705–714.
- [14] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI '16)*, 194–200.
- [15] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI '20)*, Vol. 34, 2901–2908.

- [16] Yu Liu, Jingtao Ding, Yanjie Fu, and Yong Li. 2023. Urbankg: An urban knowledge graph system. *ACM Transactions on Intelligent Systems and Technology* 14, 4 (2023), 1–25.
- [17] Yu Liu, Jingtao Ding, and Yong Li. 2023. KnowSite: Leveraging urban knowledge graph for site selection. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, 1–12.
- [18] Yu Liu, Zhilun Zhou, Yong Li, and Depeng Jin. 2023. Urban knowledge graph aided mobile user profiling. *ACM Transactions on Knowledge Discovery from Data* 18, 1 (2023), 1–30.
- [19] 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.
- [20] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. 2014. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems* 16, 2 (2014), 865–873.
- [21] Wesley Mathew, Ruben Raposo, and Bruno Martins. 2012. Predicting future locations with hidden Markov models. In *Proceedings of the ACM Conference on Ubiquitous Computing (Ubicomp '12)*, 911–918.
- [22] Guanglin Niu, Yongfei Zhang, Bo Li, Peng Cui, Si Liu, Jingyang Li, and Xiaowei Zhang. 2020. Rule-guided compositional representation learning on knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, 2950–2958.
- [23] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 2 (2021), 1–49.
- [24] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Z. Wang. 2019. Drum: End-to-end differentiable rule mining on knowledge graphs. arXiv:1911.00055. Retrieved from <https://doi.org/10.48550/arXiv.1911.00055>
- [25] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 3060–3067.
- [26] Ke Sun, Tiejun Qian, Tong Chen, Yile Liang, Quoc V. H. Nguyen, and Hongzhi Yin. 2020. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI '20)*, Vol. 34, 214–221.
- [27] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of the ACM Conference on Recommender Systems (RecSys '18)*, 297–305.
- [28] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the International Conference on Machine Learning (ICML '16)*, 2071–2080.
- [29] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*.
- [30] Huandong Wang, Yong Li, Junjie Lin, Hancheng Cao, and Depeng Jin. 2021. Context-aware semantic annotation of mobility records. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16, 3 (2021), 1–20.
- [31] Huandong Wang, Qiaohong Yu, Yu Liu, Depeng Jin, and Yong Li. 2021. Spatio-temporal urban knowledge graph enabled mobility prediction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 5, 4 (2021), 1–24.
- [32] Huandong Wang, Sihan Zeng, Yong Li, and Depeng Jin. 2020. Predictability and prediction of human mobility based on application-collected location data. *IEEE Transactions on Mobile Computing* 20, 7 (2020), 2457–2472.
- [33] Pengyang Wang, Kunpeng Liu, Lu Jiang, Xiaolin Li, and Yanjie Fu. 2020. Incremental mobile user profiling: Reinforcement learning with spatial knowledge graph for modeling event streams. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20)*, 853–861.
- [34] Senzhang Wang, Jiannong Cao, and Philip Yu. 2020. Deep learning for spatio-temporal data mining: A survey. *IEEE Transactions on Knowledge and Data Engineering* 34, 8 (2020), 3681–3700.
- [35] Yahui Wang, Hongchang Chen, Shuxin Liu, Kai Wang, and Yuxiang Hu. 2024. Geo-aware graph-augmented self-attention network for individual mobility prediction. *Future Generation Computer Systems* 151 (2024), 1–11.
- [36] Fengli Xu, Tong Xia, Hancheng Cao, Yong Li, Funing Sun, and Fanchao Meng. 2018. Detecting popular temporal modes in population-scale unlabelled trajectory data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 1–25.
- [37] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. arXiv:1412.6575. Retrieved from <https://doi.org/10.48550/arXiv.1412.6575>
- [38] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2014), 129–142.
- [39] Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. In *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 30.

- [40] Song Yang, Jiamou Liu, and Kaiqi Zhao. 2022. GETNext: trajectory flow map enhanced transformer for next POI recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1144–1153.
- [41] Quan Yuan, Wei Zhang, Chao Zhang, Xinhe Geng, Gao Cong, and Jiawei Han. 2017. PRED: Periodic region detection for mobility modeling of social media users. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 263–272.
- [42] Chao Zhang, Keyang Zhang, Quan Yuan, Luming Zhang, Tim Hanratty, and Jiawei Han. 2016. Gmove: Group-level mobility modeling using geo-tagged social media. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, 1305–1314.
- [43] Yu Zheng. 2015. Trajectory data mining: An overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 3 (2015), 1–41.
- [44] Chenyi Zhuang, Nicholas J. Yuan, Ruihua Song, Xing Xie, and Qiang Ma. 2017. Understanding People Lifestyles: Construction of Urban Movement Knowledge Graph from GPS Trajectory. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '17)*, 3616–3623.

Received 16 November 2022; revised 30 May 2024; accepted 1 June 2024