Points-of-Interest Relationship Inference with Spatial-enriched Graph Neural Networks

Yile Chen[1], Xiucheng Li[1], Gao Cong[1], Cheng Long[1], Zhifeng Bao[2], Shang Liu[1],
Wanli Gu[3], Fuzheng Zhang[3]

[1]Nanyang Technological University   [2]RMIT University   [3]Meituan

{yile001@e.,xli055@e.,gaocong@,c.long@,shang006@e.}ntu.edu.sg,
zhifeng.bao@rmit.edu.au,{guwanli,zhangfuzheng}@meituan.com

ABSTRACT

As a fundamental component in location-based services, inferring the relationship between points-of-interests (POIs) is very critical for service providers to offer good user experience to business owners and customers. Most of the existing methods for relationship inference are not targeted at POI, thus failing to capture unique spatial characteristics that have huge effects on POI relationships. In this work we propose PRIM to tackle POI relationship inference for multiple relation types. PRIM features four novel components, including a weighted relational graph neural network, category taxonomy integration, a self-attentive spatial context extractor, and a distance-specific scoring function. Extensive experiments on two real-world datasets show that PRIM achieves the best results compared to state-of-the-art baselines and it is robust against data sparsity and is applicable to unseen cases in practice.

1 INTRODUCTION

With a fast development of urban intelligence, points-of-interest (POI) has been extensively utilized in numerous location-based services (e.g., Yelp, Meituan). Massive POIs enable us to conduct various analysis to facilitate local business ranging from marketing to delivery. POIs are correlated explicitly or implicitly under different relationships, among which competitive and complementary relationships have important applications and have been studied in previous work [14, 17, 23, 33]. In general, it is of great importance for service providers to have a deep understanding of POIs, as well as their latent relationships. Specifically, given POIs and some available relationships among them, such as competitive and complementary relationships, and side information such as category taxonomy, we aim to infer other missing relationships. POI relationship inference can bring significant benefits for different groups of people. For example, business owners can design targeted operation strategies to attract more customers according to

competitive POIs, and customers can be recommended with places of their interests based on complementary POIs. Furthermore, the study of relationships among POIs can help government understand regional functionality in terms of industry, commerce and lifestyle, and hence make sustainable urban plannings. In practice, POI relationship inference model has been integrated into services to provide an automatic and accurate way of enriching Meituan's internal spatial knowledge graph, which as a type of data source that is applied in various business scenarios, such as search, tagging and recommendation. For example, it has been utilized in knowledge-enhanced models [8] to improve the performance of POI recommendation [31].

A number of studies have been conducted to infer the relationships between entities in various domains, especially for companies in business management and products in e-commerce. Earlier approaches focus on modeling text content (e.g., reviews and news) via topic modeling techniques [17, 38] or constructing pairwise neural networks [23]. Recently, graph representation learning methods [1] have been applied to solve this problem by organizing entities as a graph with different model designs, such as adding path constraints [33], multi-task learning objectives [5, 42] and Graph Neural Networks (GNN) [14]. However, these approaches are not primarily designed for POIs and do not explore unique features brought by POIs. Moreover, some of them may not even be applicable to POI relationship inference problem [5, 42]. Meanwhile, there are very few approaches tailored for POI settings. Zhou et al. [45] propose to extract POI pairwise features by sampling a large number of graphlet patterns, and then send these features to neural networks to infer the relationship. The most recent method, DeepR [13], is proposed to adapt GNN to perform neighbor aggregation from different geographical sectors, and incorporate brand and aspect knowledge from external data sources to improve the model performance.

Although these methods, including POI-specific methods and non POI-specific methods that can be applicable to POIs, have made progress in inferring the relationships for POIs, they still suffer from at least one of the following issues. <u>Issue 1</u> – Some methods only focus on a particular type of relationship. For example, recent POI-specific methods [13, 45] consider the competitive relationship only. While they can be extended to more relationships by decomposing relationship graph into sub-graphs with each sub-graph containing only one relation type, these methods fail to model the inherent interactions of POIs under different types of relationships in a unified framework. <u>Issue 2</u> – As a unique property of POI, spatial features have important effects on POI relationships. For example, two Starbucks would exhibit a strong *competitive* relationship if they are close, but have no relationship if far away. However, such spatial

features are not captured in either representation learning modules [45] or predictive functions [13] in previous work, thus leading to suboptimal model performance. Issue 3 – Some methods propose to enhance the model with structural knowledge (e.g., fuzzy logics [33] or graphlet patterns [45]). However, they require either heavy processing operations or efforts on manually designed constraints. There is a lack of simple and direct ways to integrate extra structural knowledge into the model. Issue 4 – All existing methods focus on utilizing the relationships of interest without considering the context information of target POIs. In other words, for a target POI, they only focus on its connected POIs that have relationships with it. Apart from connected POIs, the POIs that are spatially close to the target POI can also provide rich context information. For example, given a POI with shopping centers and entertainment spots nearby, we can infer that it is located at a commercial area, which indicates a different degree of competitive environment as compared to a residential area. Such context information can be useful for relationship inference.

To address these issues, we propose a novel *POI Relationship Inference Model*, named PRIM, which can handle multiple relation types within a unified framework. First, to address Issue 1, we propose to build PRIM upon the weighted relational graph neural network (WRGNN), in which we apply a two-level aggregation process, namely intra-relationship aggregation and inter-relationship aggregation, to update POI representations. We model the influence on the target representation aggregated from different relationships with a relation-specific operator that explicitly models interactions between a neighbor POI and its relationship. Second, we propose a spatial-aware attention mechanism for WRGNN to measure the importance of different neighbors by considering the POI spatial characteristics, which alleviates Issue 2 in representation learning module. Moreover, we design a distance-specific scoring function that is able to model the impact of pairwise distance of a given POI pair. By using this scoring function, we compute the likelihood for different relationship, which further enhances the model capacity in predictive function for the second issue. Third, we address Issue 3 by utilizing category taxonomy as external structural knowledge to enhance reasoning capacity of the model. Specifically, we extract the category path from the category taxonomy and seamlessly integrate it to guide the reasoning of WRGNN via embedding learning method without requiring any extra feature generation procedures or human efforts. Fourth, apart from learning based on POIs with relationships, we propose to enrich the POI representations from spatial neighbors with a self-attentive spatial context extractor which combines semantic and geographical influence together, which helps remedy Issue 4.

The main contributions of our work are summarized as follows:
- To the best of our knowledge, we are the first to propose a unified framework to handle different POI relationships for various business scenarios.
- We propose a full-fledged solution called PRIM, which features several novel designed techniques, including a weighted relational graph neural network, category taxonomy integration, a self-attentive spatial context extractor and a distance-specific scoring function. These techniques are designed to address the aforementioned four issues encountered by the existing relationship inference models.

- We conduct comprehensive experiments on two real-world datasets to evaluate the effectiveness of the proposed model. Experimental results show that PRIM outperforms the state-of-the-art baselines, including relationship inference models as well as graph representation learning models.

2 RELATED WORK

In this section, we present related work on relationship inference, POI mining, and graph neural networks.

Relationship Inference. Relationship inference has attracted much attention in various domains for practical usage, especially in business management and e-commerce. In business management, several studies are proposed to mine competitive relationship based on either text and social network [36, 38], or company relationship graphs [5, 42]. In e-commerce, product relationship inference is crucial to online advertisement and recommendation services. Several studies propose to model product reviews by topic modeling methods [17] or variational autoencoder [23] to predict the complementary and substitutable relationship between two products. Unfortunately, these methods consider each product pair *independently* and ignore rich relationship information from the underlying product graph. To handle this issue, some methods are proposed to apply graph representation learning techniques on the product graph for relationship inference by incorporating heuristic path constraints [33], preserving node proximity [43] or adopting Graph Neural Networks [14]. Some of these methods can be extended for POI relationship inference. However, they suffer from several weaknesses discussed in Section 1. There are very limited studies on POI relationship inference. Zhou et al. [45] and Li et al. [13] propose to perform POI competitive analysis, but they only focus on one particular relationship.

POI Mining. POIs have been utilized to facilitate numerous applications in location-based services. Extensive studies have been conducted on POI recommendation based on user historical check-ins to understand user behavior patterns and preferences for POIs [15, 16]. Furthermore, POI is leveraged in a broader view to enhance urban intelligence [44]. In addition, POIs associated with other data sources (e.g., text) contain richer knowledge about locations and are utilized in urban event detection [41, 46], human mobility modeling [39, 40], etc. On the other hand, to better support the above applications, methods are proposed to improve the quality of data management of POIs, especially when combined with other data sources, including streaming scenarios [2], visualization [9], scalability [35] and efficient query retrieval [3, 4]. These studies are orthogonal to our problem of mining relationships among POIs.

Graph Neural Networks. Graph Neural Networks (GNN) [34] is a generic method on modeling graph-structured data and has achieved great successes in learning effective node representations [47]. Conventional GNN [10, 12, 30] perform message passing and message aggregation from neighbors for each node iteratively to update node representations. However, they are less capable of dealing with heterogeneous graphs consisting of various node-and-edge types which usually exist in real-world applications. To overcome this issue, Heterogeneous Graph Neural Networks (HGNN) have been proposed to adapt GNN to heterogeneous graphs, and they can be roughly divided into two categories. The first is to

convert a heterogeneous graph into multiple graphs based on pre-defined meta-paths, and then aggregate information from multiple graphs to obtain final representations [6, 25, 32]. However, they demand the domain expertise in good meta-paths design, which is both labor-intensive and expensive to acquire. The second is to modify the conventional message passing and aggregation to be node-type or edge-type dependent operations to integrate various node and edge type information [11, 24, 28]. Unfortunately, they either ignore different importance of node neighbors, or fail to capture unique spatial characteristics of POIs. The state-of-the-art relationship inference methods, DeepR [13] and DecGCN [14], fall into conventional GNN and the second type of HGNN, respectively. We follow these methods to adopt GNN as the building block of our model, and propose a weighted relational graph neural network to overcome the aforementioned limitations.

3 PROBLEM FORMULATION

In this section, we formulate the problem of POI relationship inference, and first we introduce some notations to be used. Let $\mathcal{P} = \{p_1, p_2, ..., p_N\}$ denote a set of POIs and $C = \{c_1, c_2, ..., c_M\}$ denote a set of POI categories. Each POI $p \in \mathcal{P}$ is associated with a location l_p (longitude and latitude pair), an attribute $x_p \in \mathcal{X}$ and a category $c_p \in C$. POI location and category information can be used to define spatial neighbors and category taxonomy.

Definition 3.1. (**Spatial Neighbors**). Given a distance threshold d, the spatial neighbors of a POI p is defined to be $S_p := \{p' \in \mathcal{P} \mid \text{dist}(p, p') < d\}$.

Definition 3.2. (**Category Taxonomy**). The category taxonomy $T = (C \cup \mathcal{H}, \mathcal{E}_T)$ is a hierarchical tree, where category set C and hypernym set $\mathcal{H}$ belong to leaf nodes and non-leaf nodes respectively, and each directed edge $e_t \in \mathcal{E}_T$ represents a hypernymy relation between a category (e.g., food) and its sub-category (e.g., burger). A snapshot of category taxonomy is shown in Figure 1.

Multiple types of relationships among POIs can be represented as a heterogeneous graph as defined below.

Definition 3.3. (**Heterogeneous POI Relationship Graph**). A heterogeneous POI relationship graph is defined as $G = (\mathcal{P}, \mathcal{E}, \mathcal{R}, \mathcal{X})$. Each $p \in \mathcal{P}$ is a POI, and each $e \in \mathcal{E}$ is an edge between a POI pair and associated with a relation type defined by a mapping function $\psi(e) : \mathcal{E} \to \mathcal{R}$; $\mathcal{R}$ is a set of pre-defined semantic relationships between a POI pair.

Our target is to infer the relationship of a given POI pair that either belongs to a relation type $r \in \mathcal{R}$ or a non-relation type ϕ. We formally define the problem as below:

Definition 3.4. (**POI Relationship Inference Problem**). Given a heterogeneous POI relationship graph G, a category taxonomy T, a distance threshold d, and candidate relation types $\mathcal{R}^* = \mathcal{R} \cup \{\phi\}$, we aim to learn a predictive function $f(\mathcal{P} \times \mathcal{P}|G, T, d) \to \mathcal{R}^*$ that maps a POI pair (p_i, p_j) to a certain relation type.

In this paper, we focus on two scenarios: (1) *competitive* and *complementary* relationships, as they are of most interests to the business [14, 17, 23, 33]. POI pairs with competitive relationships refer to those that are interchangeable and provide similar services, while POI pairs with complementary relationships refer to those that tend to be both visited by users; (2) finer-grained multiple

relationships that can potentially exist in more complex business situations, such as large-scale spatial knowledge graph completion in the company.
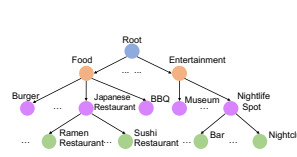
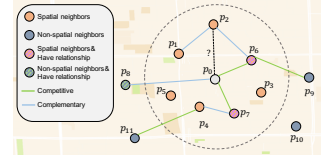**Figure 1: A snapshot of POI category taxonomy**

Figure 2: POI relationship inference example

Example. As shown in Figure 2, p_0 is the target POI, p_1-p_7 which are within the circle are spatial neighbors of p_0, p_6-p_8 have competitive/complementary relationships with p_0 and p_8 is not the spatial neighbor of p_0, and p_9-p_{11} neither are spatial neighbors nor have relationships with p_0. The problem here is to infer the relationship between p_0 and p_2. From the example we can observe that spatial neighbors might not have relationships with the target POI (e.g., p_3,p_5), while POIs that have relationships might not be spatial neighbors of the target POI (e.g., p_8). In this case, it is important to leverage both of POIs with relationships and spatial neighbors to serve as different aspects of knowledge of a target POI.

4 METHODOLOGY

4.1 Model Overview

Our proposed PRIM consists of four components: weighted relational graph neural network, taxonomy integration module, self-attentive spatial context extractor, and distance-specific scoring function. Figure 3 illustrates the proposed framework. Before presenting the details of these components, we discuss the intuitions acquired from the *Beijing* dataset (details in Section 5.1) that guide our module design as well as our technical contributions w.r.t. the design of each component.
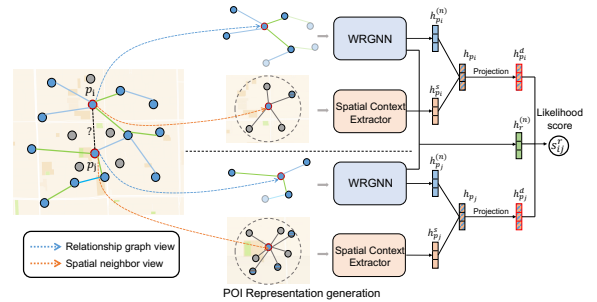
Figure 3: PRIM overview

First, as discussed in Section 1 and 2, the state-of-the-art GNN based relationship inference methods [13, 14] and other GNN models [11, 28, 32] have at least one of the limitations of (1) failing to handle graph heterogeneity, (2) not capturing spatial characteristics of POIs, (3) ignoring different importance of neighbor nodes. This motivates us to propose a novel weighted relational graph neural network (WRGNN). Specifically, to handle graph heterogeneity, we propose a two-level aggregation process to perform intra-relationship and inter-relationship interactions. Moreover, we design a spatial-aware attention mechanism in which we take

into consideration both POI semantic and spatial characteristics to differentiate the importance of neighbor nodes in the aggregation process (Section 4.2).

Then, we propose to integrate external structural knowledge, i.e., category taxonomy, to guide the reasoning of WRGNN. Intuitively, POIs whose categories are close in category taxonomy (e.g., *bar* and *nightclub*) tend to be more semantically similar and show higher degree of competitiveness as compared to distant ones (e.g., *bar* and *sushi restaurant*). We calculate the average path distance (i.e., the number of edges along the shortest path between two nodes) on category taxonomy for the corresponding categories of the POI pairs with relationships in our dataset, and we observe that the average path distance is 1.72 for competitive relationships and 3.53 for complementary relationships. Based on this result, we design a simple and effective approach to leverage the semantic information in the category taxonomy (Section 4.3).

Next, different from all the previous methods that only focus on modeling the relationship graph, we extract another type of POI representations from spatial neighbors, which provide extra context knowledge about target POIs. This is based on the fact supported by our dataset that we observe some cases where two POI pairs with the same category set or even the same brand set show different relationships when they are located at regions with different spatial context. For example, a KFC and a McDonald, when located in a shopping center, would show less competitiveness than when they are located in a residential area due to larger flow of people. Motivated by this, we propose a self-attentive spatial context extractor to obtain the functionality of the locations where the POIs are distributed, and thus it can help enrich POI representations to encode more information of different aspects (Section 4.4).

Finally, we argue that the relationship of a POI pair should vary against their spatial distance. For example, given a target restaurant and two identical source restaurants that are $1km$ and $4km$ away to the target restaurant respectively, a distance-agnostic model would yield the same relationship prediction for the two source restaurants. In reality, however, the distant restaurant is less likely to show competitive relationship since competitiveness usually decays with the distance increase. This can be verified in our dataset that 50.1% of competitive POI pairs are within the distance of $2km$, however the ratio for complementary POI pairs is 21.2%. Therefore, we propose a distance-specific scoring function to project POI representations to different latent spaces induced by the pairwise distance, thus enhancing the model capacity to capture distance effect of relationships (Section 4.5).

4.2 Weighted Relational Graph Neural Network

Due to the lack of capacity to model the heterogeneous POI relationship graph in previous methods as discussed in Section 4.1, we propose a weighted relational graph neural network (WRGNN) to model graph heterogeneity while considering spatial characteristics in measuring the different importance of neighbors in the aggregation process.

Given a POI p_i, a two-level aggregation process is adopted. Specifically, neighbors connected by the same relation type are first aggregated (intra-relationship aggregation), and then results from different types of relationships are further aggregated (inter-relationship aggregation) to generate the final representation for p_i:

$$\mathbf{h}_{p_i}^{(l+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \sum_{p_j \in \mathcal{N}_{p_i}^r} \alpha_{ij}^r \mathbf{W}^{(l)} \gamma(\mathbf{h}_{p_j}^{(l)}, \mathbf{h}_r^{(l)})\right) \quad (1)$$

Here, $\mathbf{h}_{p_i}^{(l)}, \mathbf{h}_r^{(l)}$ are the representations of p_i and relationship of type r in the l-th layer, respectively, $\mathcal{N}_{p_i}^r$ denotes a set of graph neighbors of p_i with relationship of type r, α_{ij}^r is normalized importance score of p_j to p_i, $\mathbf{W}^{(l)}$ is the weight matrix in the l-th layer, $\gamma(.,.)$ is a relation-specific operation, and $\sigma(.)$ is an activation function.

The relation-specific operator $\gamma(.,.)$ takes a POI-relationship pair as input and allows the interaction between a POI and a relationship in the aggregation process. It has several advantages: (1) we jointly learn POI and relationship representations, which can be seamlessly applied to the proposed scoring function (Section 4.5); (2) relation semantics are considered so that a neighbor POI would have different influence on the target POI for each relation type; (3) $\gamma(.,.)$ is quite flexible and can be chosen from various options, such as multiplication [37], circular-correlation [21], and complex neural network based operations [26]. Here we choose element-wise multiplication $\gamma(\mathbf{h}_{p_j}^{(l)}, \mathbf{h}_r^{(l)}) = \mathbf{h}_{p_j}^{(l)} \odot \mathbf{h}_r^{(l)}$ due to its efficiency and comparable results to other options.

Furthermore, the relationship representation of $r \in \mathcal{R}$ is updated as follows:

$$\mathbf{h}_r^{(l+1)} = \mathbf{W}_r^{(l)} \mathbf{h}_r^{(l)} \quad (2)$$

where $\mathbf{W}_r^{(l)}$ is a weight matrix in the l-th layer for all the relationships. As a result of these two updating functions, POI and relationship representations are refined alternatively by stacking multiple such layers. By doing this, these two types of representations are highly coupled, and thus can complement each other to learn rich semantic characteristics from the heterogeneous POI relationship graph. Such a design makes WRGNN more capable of handling graph heterogeneity than those methods that model each relation type separately through decomposition [13, 14].

In previous methods [13, 24, 28], neighbors in each relationship are treated equally and α_{ij}^r is set to be $\frac{1}{|N_{p_i}^r|}$. Some GNN models have shown the capability of attention mechanism on deriving the importance of neighbors by automatically reasoning their features [6, 11]. Inspired by them, we propose a spatial-aware attention mechanism to learn the importance score α_{ij}^r of graph neighbors by combining the POI semantic and spatial characteristics together as follows:

$$e_{ij}^r = \sigma\left(\mathbf{a}_r^\top \cdot \left[\mathbf{W}_a \mathbf{h}_{p_i}^{(l)} \left\| \mathbf{W}_a \mathbf{h}_{p_j}^{(l)} \right\| \mathbf{W}_d \mathbf{d}_{ij}\right]\right), \quad (3)$$

$$\alpha_{ij}^r = \text{softmax}\left(e_{ij}^r\right) = \frac{\exp\left(e_{ij}^r\right)}{\sum_{k \in \mathcal{N}_{p_i}^r} \exp\left(e_{ik}^r\right)}, \quad (4)$$

where $\mathbf{a}_r$ is the attention vector shared for the neighbors with respect to the relationship r, $\mathbf{W}_a$ is the transformation matrix applied to POI representations, $\mathbf{W}_d$ is the transformation matrix applied to spatial distance $\mathbf{d}_{ij}$ between p_i and p_j, $\|$ denotes the vector concatenation operation, $\sigma()$ is an activation function, and e_{ij}^r is normalized by a softmax function to get α_{ij}^r.

As suggested in [29, 30], the above attention mechanism can be extended to multiple heads, which helps stabilize the learning process. Specifically, K independent attention mechanisms are executed and then concatenated to form the output POI representations. We re-write Equation 1 as follows:

$$\mathbf{h}_{p_i}^{(l+1)} = \overset{K}{\underset{k=1}{\big\|}} \sigma \left(\sum_{r \in \mathcal{R}} \sum_{p_j \in \mathcal{N}_{p_i}^r} \alpha_{ij,k}^r \mathbf{W}_k^{(l)} \gamma(\mathbf{h}_{p_j}^{(l)}, \mathbf{h}_r^{(l)}) \right) \quad (5)$$

where $\alpha_{ij,k}^r$ is the normalized importance score of p_j to p_i with respect to relationship r in the k-th attention head, $\|$ denotes the concatenated representation from K attention heads.

4.3 Taxonomy Integration

We propose to integrate category taxonomy as auxiliary structural knowledge to guide the learning of WRGNN. As shown in Figure 1, the categories are organized into different levels to present category concepts at different degrees of granularity, from general to specific. Category taxonomy provides explicit concept similarity for different categories that can be utilized to enrich the learned POI representations from WRGNN.

As discussed in Section 4.1, we adopt a simple yet effective approach to encode category taxonomy. Here we use a POI p_i as an example to illustrate. Specifically, each node $t \in T$ in the category taxonomy tree is embedded into a vector $\mathbf{e}_t$. Then given POI p_i, we backtrack to retrieve all the nodes from its corresponding leaf node to the root node in T, denoted by Q_{p_i}. For example, the category path for *bar* is [*root, entertainment, nightlife spot, bar*]. Then we derive the category representation $\mathbf{q}_{p_i}$ for p_i as follows: $\mathbf{q}_{p_i} = \sum_{t \in Q_{p_i}} \mathbf{e}_t$

In this case, close categories would share more common elements in the category path, thus resulting in more similar representations. After that, we concatenate the original representations generated from WRGNN and the category representations defined as: $\mathbf{h}_{p_i}^{*(l)} = [\mathbf{h}_{p_i}^{(l)} \| \mathbf{q}_{p_i}]$. Afterwards, we integrate the taxonomy by simply replacing the original POI representation $\mathbf{h}_{p_i}^{(l)}$ with $\mathbf{h}_{p_i}^{*(l)}$ in Equation 3 & 5 and adjusting the dimension of parameters correspondingly. Note that we do not apply more complicated techniques, such as Tree-LSTM [27] and hyperbolic embedding [20], to model category taxonomy because they are less efficient and found to show no improvement over the proposed solution in our preliminary experiments.

4.4 Self-attentive Spatial Context Extractor

The modules introduced so far focus on modeling a target POI based on its neighbors in the heterogeneous POI relationship graph. Apart from graph neighbors whose number is usually limited, a POI usually has a large number of spatial neighbors that are overlooked in WRGNN. They can be leveraged to provide extra spatial context that is useful in inferring the relationship between POIs as discussed in Section 4.1.

We propose to generate spatial context as another view of POI representations. Specifically, we apply the self-attention [29] technique where a target POI is treated as a query and its spatial neighbors are treated as a set of key-value pairs. Given the learned POI representations from WRGNN, we calculate the spatial context $\mathbf{h}_{p_i}^s$

of POI p_i as follows:

$$\mathbf{h}_{p_i}^s = \sum_{p_j \in \mathcal{S}_{p_i}} \beta_{ij} (\mathbf{W}_V \mathbf{h}_{p_j}^{(L)})$$

$$\beta_{ij} = \text{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{p_m \in \mathcal{S}_{p_i}} \exp(e_{im})} \quad (6)$$

Here, $\mathbf{h}_{p_j}^{(L)}$ is the output after L layers of WRGNN for p_j, $\mathcal{S}_{p_i}$ is the spatial neighbors of p_i, β_{ij} is the normalized weight derived from e_{ij} via softmax function which indicates the attention score between p_i and p_j, and $\mathbf{W}_V$ is the projection matrix for values.

To derive the attention scores, the target POI is regarded as a query to attend over all the spatial neighbors which are regarded as keys, and the attention score for one spatial neighbor p_j can be expressed as follows:

$$e_{ij}' = \frac{\left(\mathbf{W}_Q \mathbf{h}_{p_i}^{(L)}\right)^\top \cdot \left(\mathbf{W}_K \mathbf{h}_{p_j}^{(L)}\right)}{\sqrt{d_p}} \quad (7)$$

where $\mathbf{W}_Q$ and $\mathbf{W}_K$ are the projection matrices for query and keys respectively, and d_p is the dimension of the POI representations. However, it does not explicitly take the geographical influence into consideration. According to the First Law of Geography [19], everything is related to everything else, but near things are more related than distant things. It indicates that more emphasis should be put on nearby spatial neighbors than on farther ones to extract spatial context. Therefore, we use the radial basis function (RBF) kernel to assign weights for spatial neighbors as follows:

$$D\left(l_{p_i}, l_{p_j}\right) = \exp\left(-\theta \left\| l_{p_i} - l_{p_j} \right\|^2\right) \quad (8)$$

where l_{p_i} and l_{p_j} are location coordinates of p_i and p_j respectively and θ is a scaling parameter. After that, we combine Equation 7 and 8 to derive the final attention scores which consider both semantic information and geographical influence for POIs by multiplication:

$$e_{ij} = e_{ij}' \cdot D\left(l_{p_i}, l_{p_j}\right) \quad (9)$$

Thus far, we have obtained the representations of POI from two views, namely the *heterogeneous POI relationship graph view* and the *spatial context view*. For each $p_i \in \mathcal{P}$, we fuse these two views to get the final POI representation by:

$$\mathbf{h}_{p_i} = \mathbf{h}_{p_i}^{(L)} + \mathbf{h}_{p_i}^s \quad (10)$$

4.5 Distance-specific Scoring Function

We have obtained the relationship representations and the final POI representations. Now we propose a distance-specific scoring function based on the intuition discussed in Section 4.1.

First, we project the learned POI representations to a distance-specific hyperplane according to the involved POI pairs. Specifically, we split the distance into non-overlapping bins (e.g., 0-1km, 1-2km, etc) and assign each bin with a hyperplane specified by a unit normal vector $\mathbf{w}_b$. Then, given the POI pair $(p_i, p_j) \in \mathcal{P} \times \mathcal{P}$ and their final representations, we project them into the hyperplane specified by distance as follows:

$$\mathbf{h}_{p_i}^d = \mathbf{h}_{p_i} - \mathbf{w}_{g(d_{ij})}^\top \mathbf{w}_{g(d_{ij})} \mathbf{h}_{p_i} \quad (11)$$

where $\mathbf{w}_{g(d_{ij})}$ is the unit normal vector of the bin to which the POI pair belongs, $g(d_{ij})$ is a look-up function to map the distance d_{ij} between p_i and p_j to its bin, and $\mathbf{h}_{p_i}^d$ is the projected representation for p_i.

Next, we compute the likelihood of different relationships based on the projected POI representations and relationship representations. Since the symmetric property holds for the relationship of a POI pair, we adopt the scoring function in [37] to capture this property. For a POI pair (p_i, p_j) and a relation type r, it is defined as follows:

$$s_{ij}^r = \mathbf{h}_{p_i}^{d\top} \operatorname{diag}\left(\mathbf{h}_r^{(L)}\right) \mathbf{h}_{p_j}^d \qquad (12)$$

where $\mathbf{h}_r^{(n)}$ is the representation for relation type $r \in \mathcal{R}^*$ after n layers of WRGNN, and s_{ij}^r is the likelihood score of p_i and p_j having relationship r.

4.6 Training and Inference

We train the model via the cross-entropy loss with negative sampling [18]. For each observed positive triplet (p_i, r, p_j) that denotes a relationship r exists between POI pair (p_i, p_j), we sample ω negative pairs by replacing a POI with a randomly sampled one. Then the loss function can be expressed as follows:

$$\mathcal{L} = \sum_{(p_i, r, p_j) \in \mathcal{D}} y_{ij}^r \log \sigma(s_{ij}^r) + (1 - y_{ij}^r) \log \sigma(1 - s_{ij}^r) \qquad (13)$$

where $\mathcal{D}$ is the total set of positive and negative triples, σ is the sigmoid function, and y_{ij}^r is the label indicator which is set to 1 for positive triples and 0 for negative ones.

During inference, given a pair of (p_i, p_j), we calculate the likelihood score w.r.t. each relation type. Then we rank the scores and select the relationship with the highest score as the prediction result: $\hat{r}_{ij} = \arg\max_{r \in \mathcal{R}^*} s_{ij}^r$.

Time complexity. Assuming that the POI relationship graph consists of m edges and n POIs, the dimension of POI representation is d, and the average spatial neighbors for each POI is $\tilde{S}$, the time complexity of WRGNN (Sec 4.2-4.3), spatial context extractor (Sec 4.4) and distance-specific scoring function (Sec 4.5) for all the POIs would be $O(Lnd^2 + Lmd)$, $O(nd^2 + n\tilde{S}d)$ and $O(nd)$, respectively. Since $m \gg n$ and other parameters can be considered as constants, $O(Lmd)$ is the dominating complexity term that grows linearly with the edge number.

5 EXPERIMENTS

In this section, we study the performance of PRIM on two real-world datasets.

5.1 Experimental Settings

5.1.1 Datasets. We evaluate all methods using two real-world city-wide datasets, *Beijing* (BJ) and *Shanghai* (SH), from Meituan, which is one of the largest location-based service providers in China. We construct relationships between POIs from user logs extracted from these two datasets. Note that all the user information has been anonymized in the datasets for privacy protection. Following the previous work [14, 17, 33], we generate the ground truth of *competitive* relationship for a POI pair if the two POIs in the pair are "viewed/clicked together" by users within a query session, and we

generate the ground truth of *complementary* relationships for a POI pair if the two POIs in the pair are "also viewed/clicked" by same users across different query sessions. Details of the two datasets are shown in Table 1.

Table 1: Statistics of the datasets

Dataset	Taxonomy		#POIs	#Relational Edges
	#Non-leaf nodes	#Categories		
Beijing	95	805	13,334	122,462
Shanghai			10,090	112,848

5.1.2 Baselines and Evaluation Metrics. We compare our model with five types of baseline methods, including rule-based methods (**CAT**, **CAT-D**), random walk based graph embedding methods (**Deepwalk** [22], **node2vec** [7]), vanilla GNN (**GCN** [12], **GAT** [30]), heterogeneous GNN (**HAN** [32], **HGT** [11], **R-GCN** [24], **CompGCN** [28]), and the state-of-the-art relationship inference methods (**DecGCN** [14], **DeepR** [13]). To evaluate the performance of different methods, we adopt the widely-used *Micro-F1* and *Macro-F1* in previous studies [13, 33] to jointly consider precision and recall. A higher score indicates a better performance.

5.1.3 Parameter Settings. In our experiments, we randomly sample 10% of the edges as validation data, 20% of the edges as test data, and different fractions of the remaining edges are used as training data. Apart from the relational edges, 16,000 POI pairs which do not have relationships are randomly sampled for the non-relation type for testing. In our model, we set the threshold d for determining spatial neighbors to be $1.15km$, the scaling factor θ in the radial basis function (RBF) kernel to be 2, the number of heads to be 4, the category embedding size to be 128, and the number of negative samples ω to be 5. For random walk based models, we set window size to 5, walk length to 30, and walks per node to 20. For all the GNN based methods, we set the number of layers to be 3. To achieve fair comparison, POI embedding size is set to be 128 for all the methods. We use Adam optimizer with learning rate of 0.001 and batch size of 512 to train our model.

5.2 Performance Comparison

The results of different methods are reported in Table 2, where we use different fractions of datasets from 40% to 70% as training data (denoted by "Train%"). We have several observations: (1) Methods that can model the relation heterogeneity achieve relatively good results as compared to other baselines. Moreover, the methods that decompose a graph into sub-graphs for different relation types (DecGCN, DeepR) perform worse than those that model multiple relation types within a graph in a unified way (HGT, CompGCN). (2) HGT and CompGCN outperform other baselines in most cases, which affirms the effectiveness of attention mechanism in aggregation from neighbor nodes and learning relation representations. (3) Our proposed PRIM outperforms all the baselines in all metrics and in all cases since it is featured with novel model designs.

To further validate the effectiveness of handling multiple relation types, based on the number of times appearing in user logs, we further create finer-grained relationships to differentiate the degree of competitiveness/complementarity between POI pairs. Finally, we end up with 6 relationships and the experimental results for GNN based models are shown in Table 3. It can be observed that the proposed PRIM still performs the best, which further demonstrates the superiority of our model.

Table 2: Results on the two datasets in terms of Macro-F1 and Micro-F1 (with best in bold and second-best underlined)

Dataset	Metric	Train%	CAT	CAT-D	Deepwalk	node2vec	GCN	GAT	HAN	HGT	R-GCN	CompGCN	DecGCN	DeepR	PRIM
BJ	Macro-F1	40%	0.464	0.519	0.638	0.640	0.707	0.724	0.782	0.779	0.789	<u>0.794</u>	0.757	0.783	**0.845**
		50%	0.464	0.519	0.691	0.692	0.737	0.748	0.811	0.814	0.814	<u>0.832</u>	0.801	0.815	**0.870**
		60%	0.464	0.519	0.731	0.734	0.755	0.776	0.839	0.842	0.820	<u>0.860</u>	0.811	0.842	**0.882**
		70%	0.464	0.519	0.757	0.761	0.770	0.795	0.857	0.857	0.828	<u>0.870</u>	0.823	0.861	**0.895**
	Micro-F1	40%	0.559	0.579	0.707	0.710	0.729	0.753	0.817	0.813	0.808	<u>0.827</u>	0.805	0.820	**0.879**
		50%	0.559	0.579	0.758	0.762	0.766	0.776	0.842	0.845	0.827	<u>0.859</u>	0.823	0.847	**0.895**
		60%	0.559	0.579	0.783	0.784	0.780	0.804	0.867	0.869	0.832	<u>0.882</u>	0.826	0.871	**0.907**
		70%	0.559	0.579	0.816	0.817	0.796	0.821	0.882	0.883	0.839	<u>0.892</u>	0.843	0.887	**0.913**
SH	Macro-F1	40%	0.443	0.509	0.652	0.655	0.673	0.692	0.788	<u>0.790</u>	0.777	0.776	0.765	0.786	**0.822**
		50%	0.443	0.509	0.682	0.684	0.704	0.711	0.807	<u>0.811</u>	0.802	0.808	0.797	0.808	**0.844**
		60%	0.443	0.509	0.698	0.702	0.713	0.729	0.834	0.838	0.828	0.837	0.817	<u>0.839</u>	**0.861**
		70%	0.443	0.509	0.707	0.711	0.731	0.735	0.853	<u>0.858</u>	0.839	0.857	0.832	0.852	**0.875**
	Micro-F1	40%	0.551	0.573	0.724	0.727	0.744	0.769	0.843	0.841	0.822	0.836	0.824	<u>0.845</u>	**0.886**
		50%	0.551	0.573	0.766	0.768	0.775	0.788	0.861	<u>0.864</u>	0.837	0.860	0.846	0.863	**0.896**
		60%	0.551	0.573	0.783	0.786	0.793	0.806	0.886	0.888	0.864	<u>0.891</u>	0.856	0.890	**0.909**
		70%	0.551	0.573	0.796	0.798	0.804	0.811	0.897	0.901	0.872	<u>0.903</u>	0.869	0.898	**0.920**

Table 3: Results on multiple relationships

Dataset	Metric	Train%	Deepwalk	node2vec	GCN	GAT	HAN	HGT	R-GCN	CompGCN	DeepR	PRIM
BJ	Micro-F1	40%	0.638	0.637	0.673	0.675	0.678	<u>0.716</u>	0.680	0.699	0.714	**0.759**
		50%	0.667	0.669	0.696	0.691	0.717	<u>0.744</u>	0.709	0.716	0.742	**0.789**
		60%	0.689	0.693	0.718	0.714	0.724	<u>0.761</u>	0.721	0.751	0.752	**0.799**
		70%	0.709	0.711	0.723	0.722	0.756	0.776	0.755	0.764	<u>0.778</u>	**0.804**
SH	Micro-F1	40%	0.635	0.638	0.627	0.622	0.676	<u>0.718</u>	0.688	0.678	0.713	**0.753**
		50%	0.657	0.659	0.674	0.664	0.714	0.742	0.694	0.694	<u>0.744</u>	**0.778**
		60%	0.677	0.681	0.679	0.669	0.744	<u>0.761</u>	0.742	0.746	0.756	**0.790**
		70%	0.702	0.704	0.694	0.695	0.752	<u>0.778</u>	0.768	0.771	0.774	**0.806**

5.3 Model Scalability

Given that the above datasets are relatively small, we study the model scalability on a large POI dataset in Singapore that consists of 251,219 POIs. Due to the lack of ground-truth relationships, for each POI we randomly assign 8 relationships to others. We evaluate the scalability for POIs ranging from 50K to 250K.

First, we evaluate the training efficiency for GNN based methods. The results are shown in Figure 5. It can be observed that the models that handle homogeneous graphs achieve the best efficiency. Apart from R-GCN, all the models that can handle multiple relationships have comparable training efficiency. Furthermore, as discussed in our time complexity analysis earlier, our proposed method grows linearly against the input size in terms of the training time. The results also show that our model achieves a great trade-off between effectiveness and efficiency in training phrase.

Next, we evaluate the prediction efficiency. Given POI pairs as queries, we first obtain the embeddings for these POIs according to Section 4.2-4.4. After that, we only need to index the generated embeddings and apply scoring functions in Equation (11) & (12) to get predictions. In this case, the prediction time does not change with the increase of POI size for all the GNN based methods. Based on such a procedure, we first generate the POI embeddings and then conduct 10K queries to test the prediction efficiency. The experiments show that the average prediction time per query is 1.57ms for our proposed method. In addition, when distance-specific hyperplane projection is not performed (Equation (11)), which is the case for the other GNN based methods, the average query time reduces to 0.61ms. In practice, no matter we use additional operation (Equation (11)) or not, the prediction time can satisfy the requirement in company services.

In summary, the training and prediction efficiency results demonstrate that our proposed method can scale to large-scale datasets and can be applied in practical use.

5.4 Ablation Study

We conduct an ablation study by removing different components of PRIM to demonstrate their contributions to the performance. Specifically, we compare with the following model variants: 1)-**T**: the category taxonomy constraint is removed and we learn each category embeddings independently (Section 4.3); 2)-**S**: the spatial context is removed (Section 4.4); 3)-**D**: the distance-specific hyperplane projection (Equation 11) is removed (Section 4.5). We also combine these three variants to test the model with two or more components being removed, i.e., -**DS**, -**DT**, -**ST** and -**DST**. In addition, we present the results of the best baseline model for comparison, denoted by **Base**.
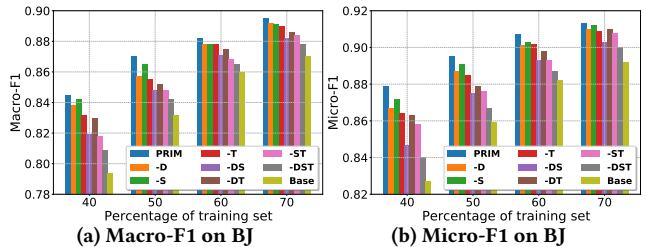

Figure 4: Results of ablation study

Figure 4 presents the results of all variants of our model, and we have the following observations: (1) The removal of different components leads to degradation of model performance, which demonstrates the effectiveness of each component. Moreover, with more components removed, we can see a larger decrease in model performance. This shows that different components capture different aspects of characteristics in POIs which are complementary to enhance the performance. (2) The performance gap between model variants becomes larger when dealing with less training data. It reveals that context information extracted from different components

would be important to alleviate the issue when the data is relatively small. (3) The model with all the three components removed (equivalent to WRGNN) still outperforms the best baseline in most cases, which validates the superiority of WRGNN over baseline models.

5.5 Model Analysis

5.5.1 Analysis on sparse cases. To achieve practical usage in applications, PRIM is expected to be robust to all cases. That is to say, the model should deal with POIs effectively even if they have very few relationships (i.e., sparsity). For this purpose, we construct the test set for POIs with fewer than 3 relationships in the training data to check their performance. We only report the results for 4 best performing baselines, since others show consistently worse results.
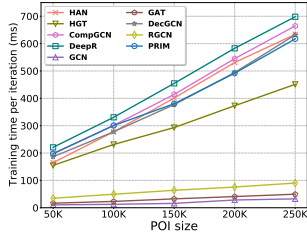
Figure 5: Training scalability   Figure 6: Results on sparse cases

The results of sparse POI cases are reported in Figure 6. We can observe that our model outperforms all the baselines. It demonstrates that our model is more capable of handling sparse cases. Furthermore, the performance of the baselines degrades more significantly than our model. These results show the superiority of PRIM for exploiting the context information specific to POIs to compensate for the lack of relationships.

5.5.2 Analysis on unseen cases. Apart from model robustness on sparse cases, it is also important that the model should be effective in inductive settings. In other words, the model should generalize well on POIs which are not seen during training. Such a capability is desired in real scenarios since the model can be directly applied to newly arrived POIs without being frequently updated. To compare the model performance in inductive settings, we randomly hide 20% of the POIs and remove the relation edges connecting to them to serve as unseen cases. Then we train all the models with the remaining POIs and the same parameter settings. Same as sparse cases, we only report the results for 4 best performing baselines.

The results of unseen POI cases are listed in Figure 7. We observe that all the compared models achieve relatively good results. This is because they are all based on GNN models, which have been shown to handle inductive settings well [10]. Specifically, DeepR performs the worst among all the baselines, while HAN, HGT and CompGCN have comparable performance. In addition, PRIM outperforms all the compared methods, which verifies the effectiveness of our model on unseen POIs.

5.5.3 Analysis on different regions. To gain a better understanding of our model's performance on different regions, we split Beijing into core area and suburb. The core area is less than 15% of the area in Beijing with over 53% POIs, which is much denser than the suburb. We report the model's performance for these two regions in Table 4. The results show that the performance gap is very small
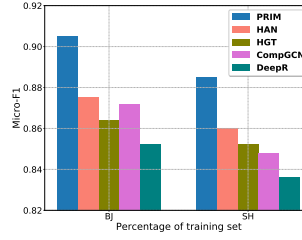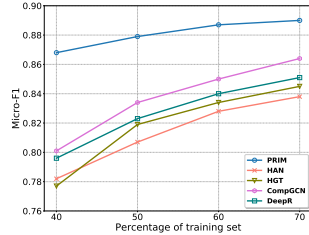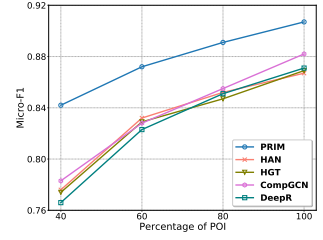
**Figure 7: Results   Figure 8: Results on datasets
on unseen case   with different characteristics**

between the core area and the suburb, which demonstrates the robustness of our model on regions with different sparsity.

We further investigate the performance on regions with more complex differences by directly applying the model learned from Beijing on Shanghai. The results are presented in Table 4. In the column of Shanghai dataset, the first and second values indicate the performance of model trained on Beijing and model trained on Shanghai itself respectively. We can observe larger performance drop compared to regions within a city. However, the overall performance is still good where most Micro-F1 metrics are over 0.8.

Table 4: PRIM performance on different regions

Metric	Train%	Area			
		BJ core area	BJ suburb	BJ overall	SH
Micro-F1	40%	0.874	0.890	0.879	0.797/0.886
	50%	0.890	0.906	0.895	0.813/0.896
	60%	0.903	0.915	0.907	0.801/0.909
	70%	0.911	0.918	0.913	0.823/0.920

5.5.4 Analysis on datasets with different characteristics. We conduct experiments to evaluate the model performance on datasets with different characteristics. To this end, we propose to generate datasets by sampling subsets of the Beijing dataset. Specifically, we randomly select 40%/60%/80% POIs in Beijing and only keep the edges among the selected POIs to serve as datasets with different scale/density/spatial distances. As a result, sparser datasets have lower density and larger spatial distances. Then we split the relational edges into 60%/20%/20% as train/validation/test data. The results are shown in Figure 8. We can observe that PRIM outperforms all the baselines, thus validating the superiority of our proposed model in handling datasets with different characteristics.

6   CONCLUSION

In this paper, we proposed a new model, PRIM, to solve the POI relationship inference problem with multiple relation types. It is featured with several new components, including the weighted relational graph neural networks, incorporating structural constraints from category taxonomy, a self-attentive spatial context extractor, and self-attentive spatial context extractor. Extensive experiments on two real-world datasets have demonstrated the superiority of the PRIM model over state-of-the-art baselines.

REFERENCES

[1] Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. 2018. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Trans. Knowl. Data Eng.* 30, 9 (2018), 1616–1637.

[2] Yue Chen, Zhida Chen, Gao Cong, Ahmed R. Mahmood, and Walid G. Aref. 2020. SSTD: A Distributed System on Streaming Spatio-Textual Data. *Proc. VLDB Endow.* 13, 11 (2020), 2284–2296.

[3] Zhida Chen, Lisi Chen, Gao Cong, and Christian S. Jensen. 2021. Location- and keyword-based querying of geo-textual data: a survey. *VLDB J.* 23, 4 (2021), 381–400.

[4] Gao Cong, Christian S. Jensen, and Dingming Wu. 2009. Efficient Retrieval of the Top-k Most Relevant Spatial Web Objects. *Proc. VLDB Endow.* 2, 1 (2009), 337–348.

[5] Le Dai, Yu Yin, Chuan Qin, Tong Xu, Xiangnan He, Enhong Chen, and Hui Xiong. 2020. Enterprise Cooperation and Competition Analysis with a Sign-Oriented Preference Network. In *KDD*. 774–782.

[6] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. In *WWW*. 2331–2341.

[7] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *KDD*. 855–864.

[8] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A Survey on Knowledge Graph-Based Recommender Systems. *TKDE* (2020).

[9] Tao Guo, Kaiyu Feng, Gao Cong, and Zhifeng Bao. 2018. Efficient Selection of Geospatial Data on Maps for Interactive and Visualized Exploration. In *SIGMOD*. 567–582.

[10] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*. 1024–1034.

[11] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous Graph Transformer. In *WWW*. 2704–2710.

[12] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[13] Shuangli Li, Jingbo Zhou, Tong Xu, Hao Liu, Xinjiang Lu, and Hui Xiong. 2020. Competitive Analysis for Points of Interest. In *KDD*. 1265–1274.

[14] Yiding Liu, Yulong Gu, Zhuoye Ding, Junchao Gao, Ziyi Guo, Yongjun Bao, and Weipeng Yan. 2020. Decoupled Graph Convolution Network for Inferring Substitutable and Complementary Items. In *CIKM*. 2621–2628.

[15] Yiding Liu, Tuan-Anh 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.

[16] Hui Luo, Jingbo Zhou, Zhifeng Bao, Shuangli Li, J. Shane Culpepper, Haochao Ying, Hao Liu, and Hui Xiong. 2020. Spatial Object Recommendation with Hints: When Spatial Granularity Matters. In *SIGIR*. 781–790.

[17] Julian J. McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring Networks of Substitutable and Complementary Products. In *KDD*. 785–794.

[18] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*. 3111–3119.

[19] Harvey J. Miller. 2004. Tobler's First Law and Spatial Analysis. *Annals of the Association of American Geographers* 94, 2 (2004), 284–289.

[20] Maximilian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *NIPS*. 6338–6347.

[21] Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2016. Holographic Embeddings of Knowledge Graphs. In *AAAI*. 1955–1961.

[22] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. In *KDD*. 701–710.

[23] Vineeth Rakesh, Suhang Wang, Kai Shu, and Huan Liu. 2019. Linked Variational AutoEncoders for Inferring Substitutable and Supplementary Items. In *WSDM*. 438–446.

[24] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC*, Vol. 10843. 593–607.

[25] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. 2019. Heterogeneous Information Network Embedding for Recommendation. *IEEE Trans. Knowl. Data Eng.* 31, 2 (2019), 357–370.

[26] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In *NIPS*. 926–934.

[27] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *ACL*. 1556–1566.

[28] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *ICLR*.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*. 5998–6008.

[30] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[31] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. In *KDD*. 968–977.

[32] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In *WWW*. 2022–2032.

[33] Zihan Wang, Ziheng Jiang, Zhaochun Ren, Jiliang Tang, and Dawei Yin. 2018. A Path-constrained Framework for Discriminating Substitutable and Complementary Products in E-commerce. In *WSDM*. 619–627.

[34] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Networks Learn. Syst.* 32, 1 (2021), 4–24.

[35] Dong Xie, Feifei Li, Bin Yao, Gefei Li, Liang Zhou, and Minyi Guo. 2016. Simba: Efficient In-Memory Spatial Analytics. In *SIGMOD*. 1071–1085.

[36] Kaiquan Xu, Stephen Shaoyi Liao, Jiexun Li, and Yuxia Song. 2011. Mining comparative opinions from customer reviews for Competitive Intelligence. *Decis. Support Syst.* 50, 4 (2011), 743–754.

[37] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*.

[38] Yang Yang, Jie Tang, Jacklyne Keomany, Yanting Zhao, Juanzi Li, Ying Ding, Tian Li, and Liangwei Wang. 2012. Mining competitive relationships by learning across heterogeneous networks. In *CIKM*. 1432–1441.

[39] 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 *WSDM*. 263–272.

[40] 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 *KDD*. 1305–1314.

[41] Chao Zhang, Guangyu Zhou, Quan Yuan, Honglei Zhuang, Yu Zheng, Lance M. Kaplan, Shaowen Wang, and Jiawei Han. 2016. GeoBurst: Real-Time Local Event Detection in Geo-Tagged Tweet Streams. In *SIGIR*. 513–522.

[42] Le Zhang, Tong Xu, Hengshu Zhu, Chuan Qin, Qingxin Meng, Hui Xiong, and Enhong Chen. 2020. Large-Scale Talent Flow Embedding for Company Competitive Analysis. In *WWW*. 2354–2364.

[43] Shijie Zhang, Hongzhi Yin, Qinyong Wang, Tong Chen, Hongxu Chen, and Quoc Viet Hung Nguyen. 2019. Inferring Substitutable Products with Deep Network Embedding. In *IJCAI*. 4306–4312.

[44] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban Computing: Concepts, Methodologies, and Applications. *ACM Trans. Intell. Syst. Technol.* 5, 3 (2014), 38:1–38:55.

[45] Jingbo Zhou, Tao Huang, Shuangli Li, Renjun Hu, Yanchi Liu, Yanjie Fu, and Hui Xiong. 2020. Competitive Relationship Prediction for Points of Interest: A Neural Graphlet Based Approach. *IEEE Trans. Knowl. Data Eng.* (2020).

[46] Xiangmin Zhou and Lei Chen. 2014. Event detection over twitter social media streams. *VLDB J.* 23, 3 (2014), 381–400.

[47] Rong Zhu, Kun Zhao, Hongxia Yang, Wei Lin, Chang Zhou, Baole Ai, Yong Li, and Jingren Zhou. 2019. AliGraph: A Comprehensive Graph Neural Network Platform. *Proc. VLDB Endow.* 12, 12 (2019), 2094–2105.