Adversarial Substructured Representation Learning for Mobile User Profiling

Pengyang Wang[*]
Missouri Univ. of Sci. and Tech.
MO, USA
pwqt3@mst.edu

Yanjie Fu[*†]
Missouri Univ. of Sci. and Tech.
MO, USA
fuyan@mst.edu

Hui Xiong
Rutgers University
NJ, USA
hxiong@rutgers.edu

Xiaolin Li
Nanjing University
Nanjing, China
lixl@nju.edu.cn

ABSTRACT

Mobile user profiles are a summary of characteristics of user-specific mobile activities. Mobile user profiling is to extract a user's interest and behavioral patterns from mobile behavioral data. While some efforts have been made for mobile user profiling, existing methods can be improved via representation learning with awareness of substructures in users' behavioral graphs. Specifically, in this paper, we study the problem of mobile users profiling with POI check-in data. To this end, we first construct a graph, where a vertex is a POI category and an edge is the transition frequency of a user between two POI categories, to represent each user. We then formulate mobile user profiling as a task of representation learning from user behavioral graphs. We later develop a deep adversarial substructured learning framework for the task. This framework has two mutually-enhanced components. The first component is to preserve the structure of the entire graph, which is formulated as an encoding-decoding paradigm. In particular, the structure of the entire graph is preserved by minimizing reconstruction loss between an original graph and a reconstructed graph. The second component is to preserve the structure of subgraphs, which is formulated as a substructure detector based adversarial training paradigm. In particular, this paradigm includes a substructure detector and an adversarial trainer. Instead of using non-differentiable substructure detection algorithms, we pre-train a differentiable convolutional neural network as the detector to approximate these detection algorithms. The adversarial trainer is to match the detected substructure of the reconstructed graph to the detected substructure of the original graph. Also, we provide an effective solution for the

optimization problems. Moreover, we exploit the learned representations of users for the next activity type prediction. Finally, we present extensive experimental results to demonstrate the improved performances of the proposed method.

CCS CONCEPTS

• **Information systems** → **Data mining**;

KEYWORDS

Mobile User Profiling, Substructure, Representation Learning

1 INTRODUCTION

Mobile user profiles are a summary of characteristics of user-specific mobile activities. Mobile user profiling refers to the efforts of extracting a user's interest and behavioral patterns from mobile activities, e.g., shopping and commutes. Effective mobile user profiling can help understand users, provide customized products and services, improve user satisfactions, and, thus, have widely been applied in various applications, e.g., customer segmentation, user identification, fraud/intrusion detection, and recommendation systems. In this paper, we study the problem of learning to profile users with mobile activities data (i.e., mobile activities checkins).

Prior studies in mobile user profiling [6] can be categorized into: (1) *Explicit profile extraction*, in which users are profiled by content features that are explicitly defined and extracted from mobile behavioral data (e.g., demographics, website clicks, mobile purchases, in-App behaviors) [16]. Such profiling methods are highly depended on the collection of comprehensive user-related information. (2) *Implicit profile learning*, which includes collaborative methods, latent factor models, network embedding and deep learning. In particular, collaborative methods assume that users in the same group behave similarly, and, thus, share similar profiles [27]. Such methods suffers from the information sparsity of a target user's peers. Latent factor models, such as matrix/tensor factorization or topic

---

[*]These two authors contributed equally to this work.
[†]Corresponding author.

modeling based variants, are developed to model user profiles as latent factors or categorical distributions representations [18]. Such methods learn user profiles through optimization with large parameter space, thus, are easy to overfit. It highly needs domain knowledge inspired model regularization. More recent studies have applied deep neural networks to learn the network hidden layer (embeddings) of users from many end-to-end deep learning tasks, for example, deep learning based recommender systems [5, 7].

The recent emergence of graph mining, deep learning, adversarial training techniques provides great potentials for us to the improve existing studies. We next detail our research insights from four perspectives.

First, human activities, such as shopping, schooling, work, eat, travel, entertainments, are spatially, temporally, and socially structural. How can we identify a data structure to better describe a mobile user's activities? Traditional methods extract content based feature vectors, and are not sufficient to address the problem. We introduce an analogy. As we know, the click through rate of a webpage is highly depended on both webpage contents and webpage structure. Similarly, if we regard a mobile user as a webpage, then the user's activities can be regarded as webpage contents, and the user's transition patterns of activities can be regarded as webpage structures. Graphs are widely employed to describe structural and relational knowledge. A mobile user's profile is indeed an inherently interconnected composition of activities, and can be readily modeled as a graph. Therefore, we propose to construct a *user activity graph* to describe each mobile user, where vertexes are activity types (i.e., POI categories) and edges are the transition frequencies between activities (POI categories). In this way, we reformulate mobile user profiling as the problem of learning deep representations of users from user activity graphs.

Second, after studying many user activity graphs, we identify another important type of structural information, which we call substructure. Substructure refers to a subgraph with a specific topology. Such substructures indeed indicate the unique personalized activity patterns of a mobile user, and imply the social attributes and preferences of the user. For example, young people prefer to transit between work complex, restaurants, and cinemas, while entrepreneurs prefer to transit between business plazas. In this paper, we focus on two substructures: high-frequency vertexes and circle, of user activity graphs. Figure 1 shows two examples of user activity graphs that are extracted from users' mobile checkin event set: {< event id, user id, datetime, POI category, longitude, latitude > }. The red circle shows the user's transition between different POI categories routinely in the working days, and the cyan independent vertexes are visited frequently by this user during work. Different users show different substructures from their activity graphs. Apparently, Figure 1(a) shows the activity graph of a faculty with young kids, in which the user transits between home, office, and preschool, and frequently visit college-related POI categories during his work; while Figure 1(b) shows the activity graph of a financial professional, in which the user transits between home, commercial plaza, and restaurants, and frequently visit gyms and coffee shops after work.

Third, the identification of the substructure in user activity graphs creates a new challenge: how can we integrate the substructure patterns into representation learning from user activity graphs?
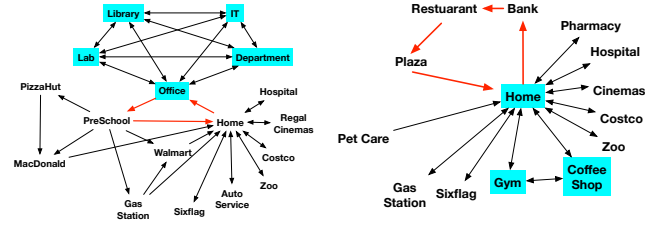
(a) The user activity graph of a faculty with young kids.

(b) The user activity graph of a financial professional.

Figure 1: The user activity graphs of two users. These two examples show that different users may have different substructures due to their personal unique patterns.

We highly need a unified learning framework to jointly model both the entire graph and the substructure information. An intuitive method is to use embedding techniques, e.g., Auto-Encoder, to learn the representations of the entire graph. Then, the substructure information is formulated as a regularization term of the optimization objective. However, as shown in Figure 1, the activity substructures of different users are of different topology (high-frequency vertexes, circles); the activity substructures of these users are dynamically distributed in different locations of a graph. A regularization term in a loss function can not tackle these challenges (varying topology and distributions) in training. The emergence of generative adversarial networks provides great potential to solve the problem. We propose to convert the integration of substructure into an adversarial substructured learning paradigm. This paradigm includes an auto-encoder that is to preserve the structure of entire graphs, a substructure detector that is to detect substructures in a graph, and an adversarial trainer that is to incorporate substructured regularization via adversarial attacks.

Fourth, it seems that the proposed adversarial substructured learning paradigm can strategically solve the challenges. But, we later find that if we use traditional subgraph detection algorithms (e.g., deep first search) to serve as a detector, these detection algorithms are usually not differentiable. This creates another challenge for the optimization of the adversarial substructured learning framework: it is impossible to back propagate the gradient. To tackle the problem, we propose to pre-train a convolutional neural network to capture the patterns of substructure in order to approximate traditional non-differentiable substructure algorithms.

Along these lines, in this paper, we develop an adversarial substructured learning framework for mobile user profiling. Specifically, our contributions are as follows: (1) We create user activity graphs to describe the characteristics, patterns, and preferences of mobile users. (2) We reformulate mobile user profiling as a problem of learning deep representations of user activity graphs. (3) We identify another structure information: substructures in user activity graphs and develop an adversarial substructured learning paradigm, including an auto-encoder, a detector, and an adversarial trainer, to preserve both the entire graph and substructure information. (4) We pre-train a convolutional neural network (CNN) to approximate traditional subgraph detection algorithms to solve the non-differentiable issue. (5) We apply the user profiling results to the application of next activity type prediction, and present extensive experiments to demonstrate the enhanced performance of the proposed method with real-world mobile checkin data.
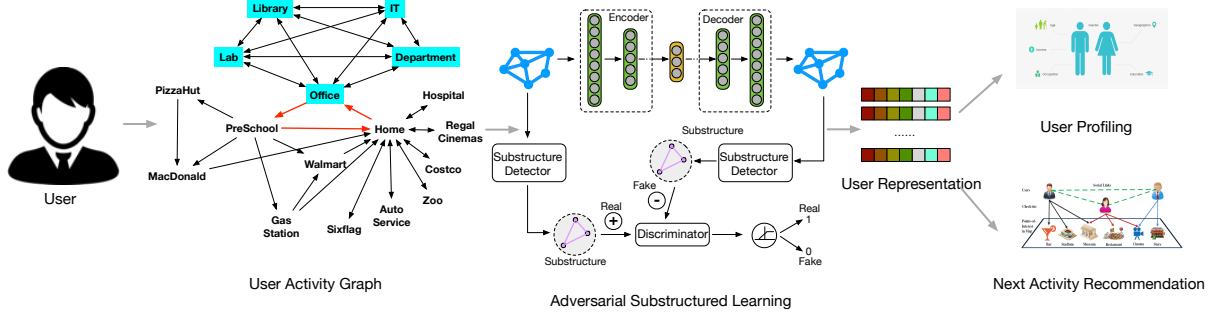
Figure 2: An overview of user profiling via adversarial substructured learning.

2 PRELIMINARIES

2.1 Definitions and Problem Statement

Definition 2.1. **User Activity Graph.** A mobile user's activities are represented as a user activity graph, in which vertexes are activity types (i.e., POI categories) and edges are the transit frequency between POI categories. Figure 1 shows user activity graphs can describe the behavioral structural information of user activities.

Definition 2.2. **Structure of the entire graph (entire-structure).** Given a user activity graph $G = (V, E)$, where V is the vertex set and E is the edge set, the structure of the entire graph (entire-structure for short) is defined as the global topological representation of the entire graph. The entire-structure preserves the relationships between vertexes and edges. For mobile users, the entire-structure can capture the general preferences patterns over all POI categories.

Definition 2.3. **Structure of the subgraph (substructure).** The structure of subgraphs (substructure for short) is defined as the topological representation of subgraphs that can feature the unique behavioral patterns of a user's activities. In this paper, we focus on two types of substructures: (1) *high-frequency vertexes*, of which the cumulative visit frequency is above the pre-defined threshold; (2) *circles*. Specifically, a high-frequency vertex in a user activity graph represents the personalized preference for a specific type of activities; a circle in a user activity graph represents the personalized preference for a close-loop consecutive activity pattern. Both high-frequency vertexes and circles can imply the unique activity patterns of a user in his/her daily life.

Definition 2.4. **Problem Statement.** In this paper, we study the problem of learning to profile users with mobile activity data. We aim to automatically learn a profile vector to represent the activity patterns of a user. We extract a user activity graph from mobile activity data, i.e., POI check-in data, to represent the activity patterns of a mobile user. We therefore formulate the problem as a task of learning deep representations of users with user activity graphs. While a user activity graph presents the overall activity profile of a user, a user's unique activity patterns are usually implied by various substructures of the activity graph. Therefore, this task is a joint objective of preserving both overall activity patterns and substructure patterns of a user in representation learning.

Formally, given a set of user activity graphs, we aim to find a mapping function $f : \mathbf{x} \rightarrow \mathbf{z}$ that takes a user activity graph $\mathbf{x}$ as the input, and outputs the vectorized representation $\mathbf{z}$ of the user, while

subject to the preservation constraints of both the entire-structure and substructure.

2.2 Framework Overview

Figure 2 shows an overview of our proposed framework that includes the following essential tasks: (i) constructing user activity graphs to represent the profiles of mobile users; (ii) developing an adversarial substructured learning framework to learn user representations from user activity graphs; (iii) evaluating the learned user representation in the application of next activity type prediction. In the first task, given the mobile checkin sequences of users, we construct a user activity graph for each user. In the second task, the adversarial substructured learning framework is developed for a joint objective of (1) modeling the entire-structure of user activity graphs, (2) constructing a differentiable substructure detector, (3) exploit adversarial training to integrate substruture regularization into representaton learning. In the last task, we apply our proposed method to profile mobile users for next activity type prediction.

3 ADVERSARIAL SUBSTRUCTURED LEARNING

Figure 3 shows the framework of the adversarial substructured learning for deep representation of user activity graphs.

3.1 Model Intuition

We represent a user as an activity graph. We learn representations of activity graphs on the following intuitions.

Intuition 1: Entire-Structure Preservation. The entire-structure of an activity graph represents how user activities interact with each other. The interaction can be strong link, weak link, no link. Consequently, we should preserve global behavioral patterns.

Intuition 2: Substructure Preservation. There are unique substructures in an activity graph, such as high-frequency activities and activity transition circles, which can uniquely feature a user's profiles. We should preserve substructural behavioral patterns.

Intuition 3: Integration of Entire-structures and Substructures. Intuitively, we can model entire-structure by network embedding and capture substructures by optimization regularizations. However, different users could exhibit different activity types, topologies, and spatial distributions in their substructures. We need a new learning paradigm to unify entire-structures and substructures.

Intuition 4: Differentiable Substructure Detector. Traditional subgraph detection algorithms are non-differentiable. If these detection algorithms are integrated into deep learning framework,

it is difficult to apply gradient descent for optimization. Therefore, we need a differentiable substructure detector to approximate non-defferentiable detection algorithms.

3.2 General Idea

Figure 3 shows our proposed adversarial substructured learning framework that includes a deep autoecoder, an approximated substructure detector, a discriminator, and an adversarial trainer. The autoencoder is to preserve entire-structure and derive the representation of a graph. We use traditional subgraph detection algorithms to detect substructure labels, and then use these labels to pre-train a CNN to approximate traditional subgraph detection algorithms. The discriminator is to classify the substructures from original graphs (real substructure set) and the substructures from reconstructed graphs (generated substructure set). The adversarial trainer is to integrate substructure awareness by forcing the autoencoder to pay specific attentions to preserving substructures in the reconstructed graph, in order to confuse the discriminator.

3.3 Preserving Entire-Structures

We exploit a deep auto-encoder [3] to preserve the global behavioral structures of users in the representation learning. Specifically, the auto-encoder includes an encoding step and a decoding step. The encoding step take a user activity graph as input and output a user feature vector. The decoding step use the user feature vector to reconstruct the user activity graph. The user feature vector captures the global behavioral structures via minimizing reconstruction loss.

Formally, given the ith graph, we flatten the graph by linking the neighbor connectivity information of each node into a single vector, denoted by $\mathbf{x_i}$, which indeeds capture the global structure of the graph. Let $\mathbf{y}^1, \mathbf{y}^2, \cdots, \mathbf{y}^o$ be the latent feature representations of the graph at hidden layers $1, 2, \cdots, o$ in the encoding step respectively. The embedding representation of the user activity graph is a d-dimentional vector, denoted by $\mathbf{z_i} \in \mathbb{R}^d$. The encoding step is formulated as:

$$\begin{cases} \mathbf{y}_i^1 & = \sigma(\mathbf{W}^1\mathbf{x}_i + \mathbf{b}^1), \\ \mathbf{y}_i^k & = \sigma(\mathbf{W}^k\mathbf{y}_i^{k-1} + \mathbf{b}^k), \forall k \in \{2, 3, \cdots, o\}, \\ \mathbf{z_i} & = \sigma(\mathbf{W}^{o+1}\mathbf{y}_i^o + \mathbf{b}^{o+1}). \end{cases} \quad (1)$$

The decoding step takes the embedding representation $\mathbf{z}_i$ (i.e., the output of the encode step) as input, and output a reconstructed graph, denoted by $\hat{\mathbf{x}}_i$. The latent feature vectors at each hidden layers are $\hat{\mathbf{y}}_i^o, \hat{\mathbf{y}}_i^{o-1}, \cdots, \hat{\mathbf{y}}_i^1$. The decoding step is formulated as:

$$\begin{cases} \hat{\mathbf{y}}_i^o & = \sigma(\hat{\mathbf{W}}^{o+1}\mathbf{z}_i + \hat{\mathbf{b}}^{o+1}), \\ \hat{\mathbf{y}}_i^{k-1} & = \sigma(\hat{\mathbf{W}}^k\hat{\mathbf{y}}_i^k + \hat{\mathbf{b}}^k), \forall k \in \{2, 3, \cdots, o\}, \\ \hat{\mathbf{x}}_i & = \sigma(\hat{\mathbf{W}}^1\hat{\mathbf{y}}_i^1 + \hat{\mathbf{b}}^1). \end{cases} \quad (2)$$

where $\mathbf{W}$s and $\mathbf{b}$s are the weight matrices and bias terms to be learned in the model.

We minimize the loss between the original graph $\mathbf{x}$ and the reconstructed graph $\hat{\mathbf{x}}$. Formally, the loss function is

$$\mathcal{L}_{AE} = \frac{1}{2}\sum_{i=1}^{m}\|(\mathbf{x}_i - \hat{\mathbf{x}}_i)\|_2^2 \quad (3)$$

3.4 Approximating Substructure Detector

Traditional substructure detection algorithms, e.g., deep first search based subgraph detection, are non-differentiable. The gradients of

neural networks cannot be passed by through back-propagation. Therefore, we propose use a pre-trained Convolutional Neural Network (CNN) to approximate the traditional substructure detector.

Formally, let F_{detr} denote a traditional substructure detector, and F_{cnn} denote a CNN based detector. The approximation objective is to let $F_{\text{cnn}} = F_{\text{detr}}$ through a two-step process:

Step 1: *Generating substructures (labels).* We take an activity graph $\mathbf{x}$ as the input of F_{detr} to generate corresponding real substructure $\mathbf{s}_{\text{real}}$ as a label.

Step 2: *Training F_{cnn} to approximate F_{detr}.* We take $\mathbf{x}$ as the input and $\mathbf{s}_{\text{real}}$ as the label to train F_{cnn}. Specifically, the architecture of F_{cnn} includes two patches of {Conv, Relu, MaxPooling}, where the kernel size of Conv is 5, and the kernel size of MaxPooling is 2. Let $\hat{\mathbf{s}}$ denote the output of F_{cnn}, the training objective is to minimize the loss:

$$\mathcal{L}_{cnn} = \frac{1}{2}\sum_{i=1}^{m}\|(\mathbf{s}_{\text{real}} - \hat{\mathbf{s}})\|_2^2 \quad (4)$$

Finally, we obtain the pre-trained F_{cnn} as a differentiable and approximated substructure detector.

3.5 Integrating Substructure Awareness via Adversarial Training

Figure 3(b) shows that we develop an adversarial learning strategy that includes a generator, a discriminator, and an adversarial trainer, in order to integrate substructure awareness.

The Generator. Figure 3(b) shows that the generator links a deep auto-encoder with a pre-trained CNN based detector. Specifically, we attach the pre-trained CNN to the last layer of the decoder, so that the CNN takes the reconstructed graph $\hat{\mathbf{x}}_i$, which is output by the decoder, as input. The CNN detects and outputs a substructure, denoted by $\hat{\mathbf{s}}_i$, from the reconstructed graph. Let G denote the generator, then the mapping procedure can be denoted as

$$\hat{\mathbf{s}}_i = G(\mathbf{x}_i) \quad (5)$$

The Discriminator. Figure 3(b) shows the discriminator is a multilayer percetron $D(\mathbf{s}; \theta_d)$, where θ_d is parameters, $D(\mathbf{s})$ outputs a probability, indicating how likely the substructure $\mathbf{s}$ is from the real substructure set $\mathbf{s}_{\text{real}}$ rather than the generated substructure set $\hat{\mathbf{s}}$.

The Adversarial Training Strategy. We simultaneously train G and D by playing the two-player minimax game. Specifically, D is trained to maximize the accuracy of classifying real substructures and reconstructed substructures generated from G. G is trained to minimize the accuracy of D for classifying the reconstructed substructure set generated from G.

Formally, let $p_{\text{real}}(\mathbf{s})$ denote the distribution over the real substructure set $\mathbf{s}_{\text{real}}$, and $p_{\mathbf{x}}(\mathbf{x})$ denote the distribution over the original graph set $\mathbf{x}$. Then, the minimax function of the adversarial training can be represented as

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{s}\sim p_{\text{real}}(\mathbf{s})}[\log D(\mathbf{s})] + \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}(\mathbf{x})}[\log(1 - D(G(\mathbf{x})))] \quad (6)$$

Specifically, the discriminator accuracy $\mathcal{L}_D$ can be represented as

$$\mathcal{L}_D = \frac{1}{m}\sum_{i=1}^{m}[\log D(\mathbf{s}_i) + \log(1 - D(G(\mathbf{x}_i)))], \quad (7)$$

and the generator loss $\mathcal{L}_G$ can be represented as

$$\mathcal{L}_G = \frac{1}{m}\sum_{i=1}^{m}\log(1 - D(G(\mathbf{x}_i))). \quad (8)$$

(a) Pre-train the CNN to approximate the substructure detector.

(b) Adversarial training process to integrate the substructure.

(c) Utilize the well-trained model to generate representations of mobile user profiles.
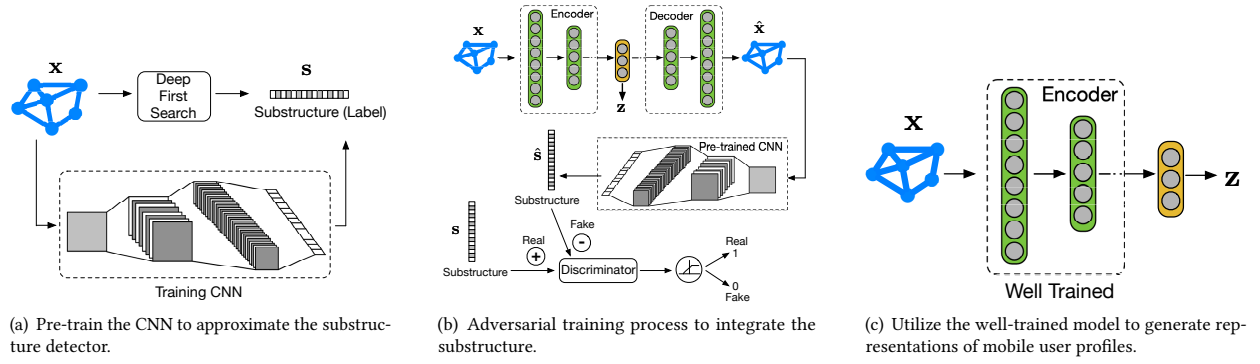
Figure 3: The framework of adversarial substructured learning that include three components. First, we pre-train a CNN as the differentiable substructure detector to approximate the the traditional substructure detection algorithm (i.e., DFS). Second, we integrate the substructure into the representation z via adversarial training. Third, we utilize the well-trained encoder to generate representations of mobile user profiles.

The objective is to maximize $\mathcal{L}_D$ and minimize $\mathcal{L}_G$ concurrently.

3.6 Solving The Optimization Problem

The loss function of the model includes: (i) minimizing the reconstruction loss (Equation 3); (ii) maximizing the discriminator accuracy (Equation 7), and (iii) minimizing the generator loss (Equation 8). The objective is to minimize the overall loss $\mathcal{L}$ as follows:

$$\mathcal{L} = -\lambda_D \mathcal{L}_D + \lambda_G \mathcal{L}_G + \lambda_{AE} \mathcal{L}_{AE} \tag{9}$$

where λ_D, λ_G, and λ_{AE} are the weights of $\mathcal{L}_D$, $\mathcal{L}_D$, and $\mathcal{L}_{AE}$ respectively.

In the training stage, we use stochastic gradient descent for optimizing $\mathcal{L}$. Specifically, we first update the Auto-Encoder by:

$$\nabla_{\theta_{AE}} \| (\mathbf{x}_i - \hat{\mathbf{x}}_i) \|_2^2. \tag{10}$$

We update the generator while keeping the parameters of the pre-trained detector F_{detr} fixed. The parameters of the generator θ_g are the same as the parameters of the Auto-Encoder θ_{AE}. In another word, $\theta_g = \theta_{AE}$. We update the generator by:

$$\nabla_{\theta_{AE}} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(\mathbf{x}^i))). \tag{11}$$

We update the discriminator by:

$$-\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [\log D(\mathbf{s}^i) + \log(1 - D(G(\mathbf{x}^i)))] \tag{12}$$

In the testing stage, Figure 3(c) shows we use the substructure-aware encoder to learn the representation of a user activity graph.

3.7 Discussion

In recent studies, there are some works related to representation learning with respect to preserving structural information. For example, *Wang et al.* propose to model the first-order proximity (i.e. the local pairwise similarity only between the vertexes linked by edges) and the second-order proximity (i.e., similarity of the vertexes' neighborhood structures) in the representation learning [38]. *Yu et al.* propose to capture the network structure through jointly

considering both locality-preserving and global reconstruction constraints with adversarially regularized auto-encoders [48]. The difference between these works and our paper is that our work aims to integrate the substructure (i.e., the structure of subgraphs) into the entire-structure (i.e., the structure of the entire graph), while these previous works jointly model the local structure (i.e., the structure of neighbors) and the global structure.

4 WHAT TO DO NEXT: USER PROFILING FOR FORECASTING NEXT ACTIVITY TYPE

As an application, we use the proposed method to profile user activity graphs and infer next activity types, in order to evaluate the performances. Specifically, we first regard a POI category as an activity type. Then, the preference for next activity type is represented by a visit probability distribution over POI categories, denoted by $P_{\mathrm{visit}}^i = \{P_{\mathrm{visit}}^{i,1}, P_{\mathrm{visit}}^{i,2}, \cdots, P_{\mathrm{visit}}^{i,k}\}$, which indeed is the normalized visit frequency over each POI category for next day. Here, $P_{\mathrm{visit}}^{i,k}$ is the probability that user u_i will visit the kth POI category.

Definition 4.1. **The What To Do Next Problem.** *Given the historical POI check-in records of users, the objective is to forecast a user's next activity type by inferring the probabilities of POI categories that a user will visit in next day.*

Specifically, for each user u_i, we first construct a user activity graph $\mathbf{G}_i$, where a vertex is a POI category, and the weight an edge is the frequency of visit transitions from a POI category to another POI category. Then, we exploit our method to learn the representation $\mathbf{z}_i$ of the user u_i. Later, we train a fully connected neural network NN with $\mathbf{z}_i$ as input, in order to forecast the preference for next activity type for each user: $NN : \mathbf{z}_i \rightarrow P_{\mathrm{visit}}^i$. Finally, we rank POI categories based on their visit probabilities to generate a candidate list R_i.

5 EXPERIMENTAL RESULTS

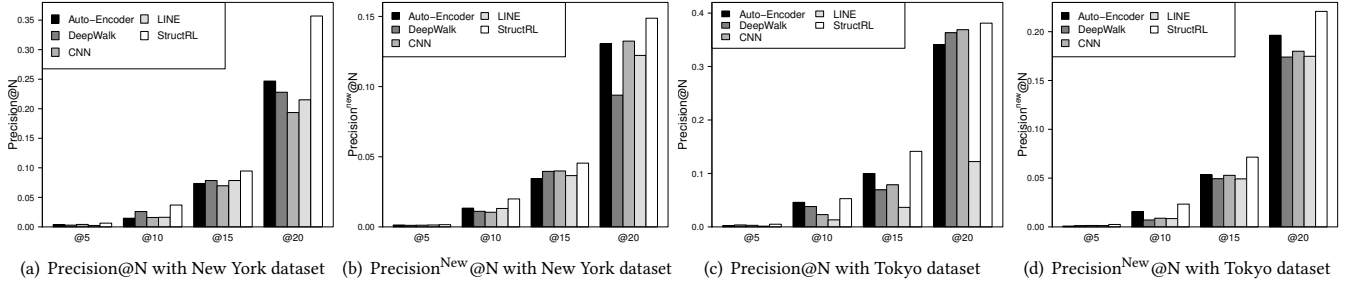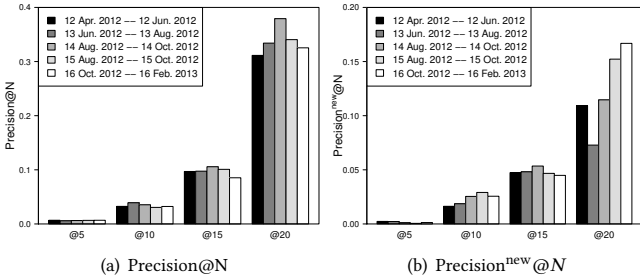This section details our empirical evaluation of the proposed method on real-world data.

(a) Precision@N with New York dataset (b) Precision^{New}@N with New York dataset (c) Precision@N with Tokyo dataset (d) Precision^{New}@N with Tokyo dataset

Figure 4: Overall comparison.

(a) Precision@N (b) Precision^{new}@N

Figure 5: Robustness check w.r.t. New York Data.
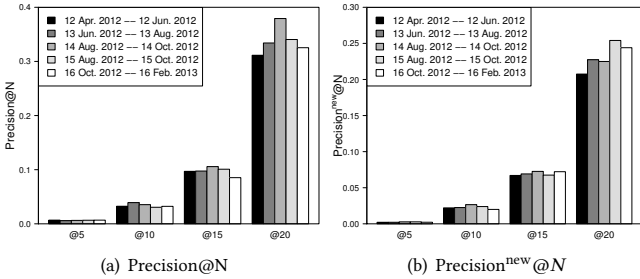

(a) Precision@N (b) Precision^{new}@N

Figure 6: Robustness check w.r.t. Tokyo Data.

5.1 Data Description

Table 1 shows the statistics of our two check-in datasets [45] from two cities: New York and Tokyo. Each dataset includes User ID, Venue ID, Venue Category ID, Venue Category Name, Latitude, Longitude, and Time.

Table 1: Statistics of the experimental data.

City	# Check-ins	# POI Categories	Time Period
New York	227428	251	12 April 2012 to 16 February 2013
Tokyo	573703	247	12 April 2012 to 16 February 2013

5.2 Evaluation Metrics

We evaluate the model performances in terms of two metrics:
(1) Precision@N: is the precision of forecasting next preexisted activity types that have been historically performed by a user. Let $\hat{R}_i$ denote the topK predicted POI category list ranked in a descending order based on the corresponding predicted visit probabilities, R_i denote the visited POI category list, and U denote the user set. Then, Precision@N can be represented as

$$\text{Precision@N} = \frac{1}{|U|} \sum_{u_i \in U} \frac{|\hat{R}_i^N \cap R_i|}{|R_i|} \qquad (13)$$

(2) Precision^{new}@N: is the precision of forecasting next new activity types that have NEVER been historically performed by a user. Let R_i^{new} denote the visited POI category list that user u_i has not visited in the training set, but will be visited in the testing set. Then, Precision^{new}@N can be denoted as

$$\text{Precision}^{new}\text{@N} = \frac{1}{|U|} \sum_{u_i \in U} \frac{|\hat{R}_i^N \cap R_i^{new}|}{|R_i^{new}|} \qquad (14)$$

5.3 Baseline Algorithms

We compare the performances of our method (namely "StructRL") against the following baseline algorithms.
(1) Auto-Encoder. The Auto-Encoder model [3] minimizes the loss between the original feature representations and reconstructed ones. In the experiments, we set the number of hidden layers = 3, the size of middle layer = 50.
(2) DeepWalk. The DeepWalk model [32] extends the word2vec model [28] to the scenario of network embedding. DeepWalk uses local information obtained from truncated random walks to learn latent representations. In the experiments, we set the number of walks = 50, the size of representation = 50, the walk length = 40, and the window size = 10.
(3) LINE. The LINE model optimizes the objective function that preserves both the local and global network structures with an edge-sampling algorithm [37]. In the experiments, we set the size of representation = 50, the number of negative samples = 5, and the starting value of the learning rate = 0.0005.
(4) CNN. The CNN model refers to Convolutional Neural Network, which projects original feature space into a new space via a variation of multilayer perceptrons [23]. In the experiments, the CNN has three patches of {Conv, Relu, MaxPooling}, where with the kernel size of Conv is 4, and the kernel size of MaxPooling is 2.

In the experiment, we split the datasets into two non-overlapping sets: for each user, the earliest 80% of check-ins are the training set and the remaining 20% check-ins are testing set. All the evaluations are performed on a x64 machine with Intel E5-1680 3.40GHz CPU and 128GB RAM. The operation system is CentOS 7.4.

5.4 Overall Performance

We compare our method with the baseline methods in terms of Precision@N and Precision@N^{new}. In general, Figure 4 shows our model achieves the best performance over both the New York and Tokyo dataset. In the task of forecasting next pre-existed activity types, we observe that the accuracy of our method is much better than the baseline algorithms when K is getting larger. In the task of forecasting next new activity types, our method still outperform
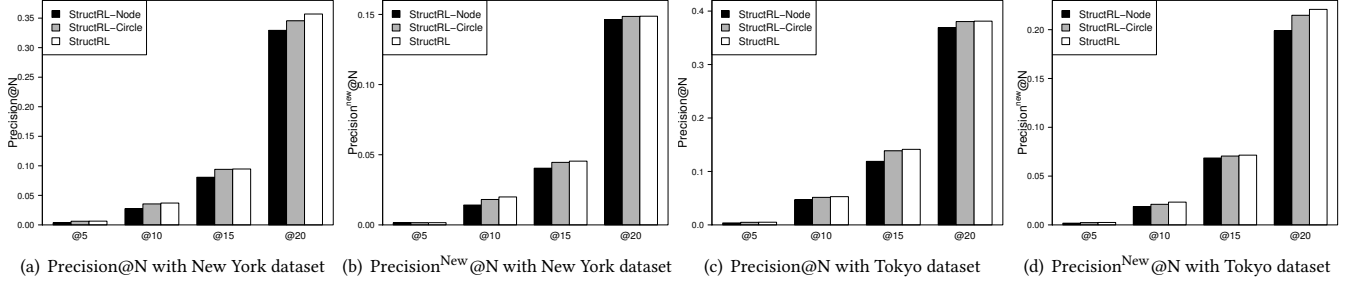
(a) Precision@N with New York dataset   (b) Precision^New@N with New York dataset   (c) Precision@N with Tokyo dataset   (d) Precision^New@N with Tokyo dataset

Figure 7: Study of Substructure Preserving.

(a) Training loss with New York dataset w.r.t. the Node Substructure.   (b) Training loss with New York dataset w.r.t. the Circle Substructure.   (c) Training loss with Tokyo dataset w.r.t. the Node Substructure.   (d) Training loss with Tokyo dataset w.r.t. the Circle Substructure.
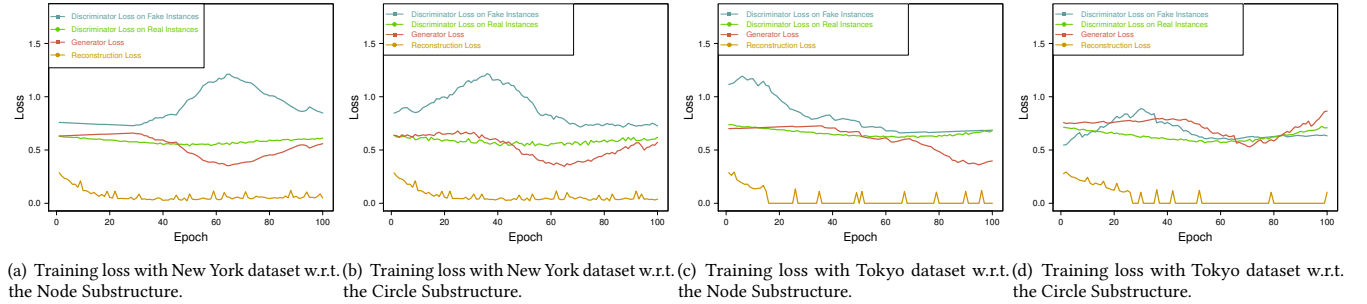
Figure 8: Training loss.

the baseline methods on discovering new POI categories that users never visit.

The results validate that the substructure is essential for user profiling. Specifically, the substructure of user activity graphs implies some particular patterns of user activities. For example, circle substructures can feature the activity transition patterns over specific POI categories, while node structures can feature a user's preferences over several independent POI categories. Our proposed framework effectively considers both substructure and entire-structure via adversarial training. However, Auto-Encoder, DeepWalk, CNN, and LINE are not able to learn representations with awareness of substructures, thus degrade their predictive performances.

5.5 Robustness Check

To conduct robustness check, we apply our method to different subgroups of data to examine the variance of our performances. Specifically, we equally split the dataset into five time periods, including (1) 12 Apr. 2012 – 12 Jun. 2012, (2) 13 Jun. 2012 – 13 Aug. 2012, (3) 14 Aug. 2012 – 14 Oct. 2012, (4) 15 Aug. 2012 – 15 Oct. 2012, and (5) 16 Oct. 2012 – 16 Feb. 2013. We set the last day's activities of each time period as a predictive target to conduct evaluations. We evaluate the forecasting performances in terms of Precision@N and Precision@N^new over both the New York and Tokyo dataset. Figure 5 and Figure 6 show that the performances of our method can achieve a small variance and are relatively stable, especially when K is small.

5.6 Study of Substructure Preserving

We introduce two types of substructures: *independent vertexes* and *circles*. Therefore, we examine how the two different substructure types impact the performances of our method on use profiling.

Specifically, we denote (1) *StructRL-Node*: a variant of our framework that only consider *discrete vertexes* substructure; (2) *StructRL-Circle*: a variant of our framework that only consider as *circle* substructure; (3) *StructRL*: our proposed method that consider both.

Figure 7 shows the performances of *StructRL-Circle* always slightly outperforms *StructRL-Node*; in other words, the substructure of *circle* is more effective than the substructure of *discrete vertexes* for describing user activity patterns. The substructure of *circle* shows a user's circle activity transition patterns, while the substructure of *independent vertexes* shows some independent POI categories that users highly and intensively prefer. Therefore, the substructure of *circle* can describe the correlations among POI categories to imply a user's particular lifestyle patterns, which is more informative than the substructure of *independent vertexes*.

5.7 Study of Training Loss

Figure 8 shows the training loss of our method with respect to different substructures and different dataset. The reconstruction loss indicates the effectiveness of preserving a graph's global structural patterns. The adversarial training loss indicates the learning process of integrating the substructure into the entire-structure. We can observe that the reconstruction loss will converge while conducting the adversarial training simultaneously. In other words, integrating substructure information into global structure can help to ensure the convergence of training loss.

6 RELATED WORK

Representation learning. Representation learning, also known as graph/network embedding, aims to learn a low-dimensional vector to represent vertexes or graphs. Technically, representation learning algorithms can be categorized into three main approaches: (1) the probabilistic models, (2) the geometrically motivated manifold-learning approaches, and (3) the reconstruction-based algorithms

related to auto-encoder. The key idea of the probabilistic model based approaches is to learn a hierarchy of features one level at a time by Bayesian Inference [35, 43, 49]. In the second category, the majority of the algorithms adopt a non-parametric approach, based on a training set nearest neighbor graph [2, 19]. In the third category, the auto-encoder based methods project the instances in original feature representations into a lower-dimensional feature space via a series of non-linear mappings, by minimizing the loss between original and reconstructed feature spaces [9, 20, 25]. Specifically, spatio-temporal representation learning is the elevation of representation learning in the spatio-temporal contexts [14, 15, 26, 41, 46]. *Wang et al.* developed a geographical learning method to find proper representations of communities to mimic the spatial structure [42]. *Wang et al.* proposed a collective embedding framework to learn the community structure from multiple periodic spatial-temporal graphs of human mobility [40].

User Profiling User profiling refers to the efforts of extracting a user's interest and behavioral patterns from users' activities. Generally speaking, the user profiling techniques can be categorized into two groups: (1) static profiling and (2) dynamic profiling. Static profiling refers to learn representations of users based on the aggregated dataset that depends on the temporal perspective [11]. For example, *Farseev et al.* proposed to learn user profile via integration of multiple data sources from Twitter, Foursquare and Instagram [13]. *Farseev et al.* proposed multi-source individual user profile learning framework named "TweetFit" that can handle data incompleteness and perform wellness attributes inference from sensor and social media data simultaneously [12]. On the other hand, dynamic profiling refers to modeling user representations considering the temporal effects that user profiles may change over time. *Akbari et al.* proposed an approach which directly learns the embedding from longitudinal data of users that simultaneously learns a low-dimensional latent space as well as the temporal evolution of users in the wellness space [1]. *Du et al.* proposed a framework which connects self-exciting point processes and low-rank models to capture the recurrent temporal patterns in a large collection of user-item consumption pairs [10]. *Zhao et al.* proposed a spatial-temporal latent ranking (STELLAR) method that capture the impact of time on successive POI recommendation [50]. *Xiao et al.* proposed to quantify user influence from user interactions in social networks tp explain price stock [44].

Generative Adversarial Networks Generative Adversarial Networks (GAN) [17] are deep neural net architectures comprised of two networks that simultaneously trained to compete against each other, where the discriminator is trained to distinguish between real samples and generated samples, and the generator is trained to make false examples to fool the discriminator. The adversarial learning paradigm has inspired many works, with applications on image generation [30], image super-resolution [24], image translation [21, 34], and network embedding [4, 39]. Technically, there are three types of variants, including (1) varying objective of the discriminator, (2) varying objective of the generator, and (3) varying architecture [47]. For varying objective of the discriminator, *Nowozin et al* proposed to utilize the f-divergence to as the objective function for for computing likelihoods or for marginalization [31]. *Springenberg et al.* proposed to utilize an entropy based objective to improve the robustness of the classifier [36]. For varying objective

of the generator, *Larsen et al.* combined a variational autoencoder with a generative adversarial network that replaces element-wise errors with feature-wise errors for better generation capability. [22]. *Mirza et al.* proposed to condition on the generator by simply feeding the data [29]. For varying architecture, *Radford et al.* proposed to introduce fully convolutional downsampling/upsampling layers in the architecture that can learn a hierarchy of representations [33]. *Denton et al.* constructed the adversarial training paradigm based on a cascade of convolutional networks within a Laplacian pyramid framework for generating images [8].

7 CONCLUSION REMARKS

We study the problem of automated mobile user profiling. We represent a user as a activity graph, and reformulate the user profiling problem as a task of representation learning from user activity graphs. After analyzing numerous user activity graphs, we found that it is essential to preserve both the entire-structure and the substructure of a graph. We observe that the contents, topology, and locations of substructures in a graph can dynamically vary over different users. We propose an adversarial substructured learning method to jointly model both the entire-structure and substructure (i.e., implying unique personalized activity patterns of a mobile user) in the representation learning. Specifically, we first adopt Auto-Encoder to model the entire-structure by minimizing the graph reconstruction loss. Besides, we pre-train a CNN to approximate the nondifferentiable substructure detectors, so the substructure detectors can cooperate with Auto-Encoder. Moreover, we design the generator by attaching the pre-trained CNN to the last layer of the Auto-Encoder to generate substructures. In addition, we integrate substructure awareness via adversarial training by jointly minimizing the graph reconstruction loss and generator loss and maximizing the discriminator loss. Moreover, we apply our method to the applications of forecasting next activity types. We present intensive experimental results with NYC and Tokyo data to demonstrate the effectiveness of our method.

ACKNOWLEDGMENTS

REFERENCES

[1] Mohammad Akbari, Xia Hu, Fei Wang, and Tat-Seng Chua. 2017. Wellness Representation of Users in Social Media: Towards Joint Modelling of Heterogeneity and Temporality. *IEEE Transactions on Knowledge & Data Engineering* 10 (2017), 2360–2373.

[2] Yoshua Bengio, Aaron C Courville, and James S Bergstra. 2011. Unsupervised models of images by spike-and-slab RBMs. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 1145–1152.

[3] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*. 153–160.

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*. 2172–2180.

[5] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 191–198.

[6] Ayse Cufoglu. 2014. User profiling-a short review. *International Journal of Computer Applications* 108, 3 (2014).

[7] Sudeep Das. 2015. Making Meaningful Restaurant Recommendations At OpenTable. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 235–235.

[8] Emily L Denton, Soumith Chintala, Rob Fergus, et al. 2015. Deep generative image models using aï£ij laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*. 1486–1494.

[9] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. 2017. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 5747–5756.

[10] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. 2015. Time-sensitive recommendation from recurrent user activities. In *Advances in Neural Information Processing Systems*. 3492–3500.

[11] Aleksandr Farseev, Mohammad Akbari, Ivan Samborskii, and Tat-Seng Chua. 2016. 360âŰe user profiling: past, future, and applications. *ACM SIGWEB Newsletter,(Summer)* (2016).

[12] Aleksandr Farseev and Tat-Seng Chua. 2017. TweetFit: Fusing Multiple Social Media and Sensor Data for Wellness Profile Learning.. In *AAAI*. 95–101.

[13] Aleksandr Farseev, Liqiang Nie, Mohammad Akbari, and Tat-Seng Chua. 2015. Harvesting multiple sources for user profile learning: a big data study. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 235–242.

[14] Yanjie Fu, Guannan Liu, Yong Ge, Pengyang Wang, Hengshu Zhu, Chunxiao Li, and Hui Xiong. 2019. Representing urban forms: A collective learning model with heterogeneous human mobility data. *IEEE transactions on knowledge and data engineering* 31, 3 (2019), 535–548.

[15] Yanjie Fu, Pengyang Wang, Jiadi Du, Le Wu, and Xiaolin Li. 2019. Efficient Region Embedding with Multi-view Spatial Networks: A Perspective of Locality-Constrained Spatial Autocorrelations. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*.

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

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[18] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 549–558.

[19] Geoffrey E Hinton et al. 2010. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2551–2558.

[20] Geoffrey E Hinton and Richard S Zemel. 1994. Autoencoders, minimum description length and Helmholtz free energy. In *Advances in neural information processing systems*. 3–10.

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. *arXiv preprint* (2017).

[22] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. 2015. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300* (2015).

[23] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. 1997. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks* 8, 1 (1997), 98–113.

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint* (2017).

[25] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057* (2015).

[26] Hao Liu, Ting Li, Renjun Hu, Yanjie Fu, Jingjing Gu, and Hui Xiong. 2019. Joint Representation Learning for Multi-Modal Transportation Recommendation. *AAAI, to appear* (2019).

[27] Heng Luo, Changyong Niu, Ruimin Shen, and Carsten Ullrich. 2008. A collaborative filtering framework based on both local user similarity and global user similarity. *Machine Learning* 72, 3 (2008), 231–245.

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[29] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[30] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. 2017. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4467–4477.

[31] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*. 271–279.

[32] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[33] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[34] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396* (2016).

[35] Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Deep boltzmann machines. In *Artificial Intelligence and Statistics*. 448–455.

[36] Jost Tobias Springenberg. 2015. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390* (2015).

[37] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1067–1077.

[38] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1225–1234.

[39] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. Graphgan: Graph representation learning with generative adversarial nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[40] Pengyang Wang, Yanjie Fu, Jiawei Zhang, Xiaolin Li, and Dan Lin. 2018. Learning urban community structures: A collective embedding perspective with periodic spatial-temporal mobility graphs. *ACM Transactions on Intelligent Systems and Technology (TIST)* 9, 6 (2018), 63.

[41] Pengyang Wang, Yanjie Fu, Jiawei Zhang, Pengfei Wang, Yu Zheng, and Charu Aggarwal. 2018. You are how you drive: Peer and temporal-aware representation learning for driving behavior analysis. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2457–2466.

[42] Pengyang Wang, Jiawei Zhang, Guannan Liu, Yanjie Fu, and Charu Aggarwal. 2018. Ensemble-Spotting: Ranking Urban Vibrancy via POI Embedding with Multi-view Spatial Graphs. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 351–359.

[43] Yue Wang, Dawei Yin, Luo Jie, Pengyuan Wang, Makoto Yamada, Yi Chang, and Qiaozhu Mei. 2016. Beyond ranking: Optimizing whole-page presentation. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 103–112.

[44] Keli Xiao, Qi Liu, Chuanren Liu, and Hui Xiong. 2018. Price shock detection with an influence-based model of social attention. *ACM Transactions on Management Information Systems (TMIS)* 9, 1 (2018), 2.

[45] Dingqi Yang, Daqing Zhang, Vincent. W. Zheng, and Zhiyong Yu. 2015. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2015), 129–142.

[46] Zijun Yao, Yanjie Fu, Bin Liu, Wangsu Hu, and Hui Xiong. 2018. Representing Urban Functions through Zone Embedding with Human Mobility Patterns.. In *IJCAI*. 3919–3925.

[47] Xin Yi, Ekta Walia, and Paul Babyn. 2018. Generative adversarial network in medical imaging: A review. *arXiv preprint arXiv:1809.07294* (2018).

[48] Wenchao Yu, Cheng Zheng, Wei Cheng, Charu C Aggarwal, Dongjin Song, Bo Zong, Haifeng Chen, and Wei Wang. 2018. Learning deep network representations with adversarially regularized autoencoders. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2663–2671.

[49] Chao Zhang, Liyuan Liu, Dongming Lei, Quan Yuan, Honglei Zhuang, Tim Hanratty, and Jiawei Han. 2017. Triovecevent: Embedding-based online local event detection in geo-tagged tweet streams. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 595–604.

[50] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R Lyu, and Irwin King. 2016. STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation.. In *AAAI*. 315–322.