Unifying Inter-region Autocorrelation and Intra-region Structures for Spatial Embedding via Collective Adversarial Learning

Yunchao Zhang*
Missouri Univ. of Sci. and Tech.
Missouri, USA
yzgv7@mst.edu

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

Pengyang Wang
Missouri Univ. of Sci. and Tech.
Missouri, USA
pwqt3@mst.edu

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

Yu Zheng
JD Intelligent Cities Research,
JD Intelligent Cities Business Unit,
Xidian University
Beijing, China
msyuzheng@outlook.com

ABSTRACT

Unsupervised spatial representation learning aims to automatically identify effective features of geographic entities (i.e., regions) from unlabeled yet structural geographical data. Existing network embedding methods can partially address the problem by: (1) regarding a region as a node in order to reformulate the problem into node embedding; (2) regarding a region as a graph in order to reformulate the problem into graph embedding. However, these studies can be improved by preserving (1) *intra-region geographic structures*, which are represented by multiple spatial graphs, leading to a reformulation of collective learning from relational graphs; (2) *inter-region spatial autocorrelations*, which are represented by pairwise graph regularization, leading to a reformulation of adversarial learning. Moreover, field data in real systems are usually lack of labels, an unsupervised fashion helps practical deployments. Along these lines, we develop an unsupervised **C**ollective **G**raph-regularized dual-**A**dversarial Learning (**CGAL**) framework for multi-view graph representation learning and also a **G**raph-regularized dual-**A**dversarial Learning (**GAL**) framework for single-view graph representation learning. Finally, our experimental results demonstrate the enhanced effectiveness of our method.

KEYWORDS

Urban Computing, Data Mining, Representation Learning

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

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330972

ACM Reference Format:
Yunchao Zhang*, Yanjie Fu†*, Pengyang Wang, Xiaolin Li, and Yu Zheng. 2019. Unifying Inter-region Autocorrelation and Intra-region Structures for Spatial Embedding via Collective Adversarial Learning. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'19), June 22–24, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330972

1 INTRODUCTION

A geographic entity is a spatial unit, such as a state, county, or neighborhood, and is socially a place where people live, work, consume, and entertain. For convenience, we refer geographic entities as regions unless stated otherwise. Spatial representation learning aims to identify effective features to represent regions with heterogeneous yet structural geographical data. Studying the representations of regions can help us better understand the structures and dynamics of cities, support regional planning, improve city governance, and ultimately make our cities smarter and more sustainable [18]. However, almost all field data are unlabeled, which creates significant challenges toward practical deployment [31]. In this study, we investigate the problem of *deep unsupervised spatial representation learning*.

Existing network and word embedding techniques (e.g., Skip-gram [5], LINE [22], DeepWalk [19], AutoEncoder [13]) can partially solve the problem. Moreover, some preliminary studies [25, 27, 32] have been developed for spatial representation learning. These studies typically tackle the problem from three perspectives: (1) Learn spatial representations from homogeneous network, where a region representation is regarded as a node embedding [25]; (2) Learn node embeddings from heterogeneous networks, where nodes are mixture of regions, mobility events, or texts [30]; (3) Regard a region as a network, and learns representations from the entire network [27]. However, the three strategies have some limits. For example, the first strategy only captures inter-region spatial autocorrelations. The second strategy ignore intra-region structural information. The third strategy captures intra-region structural information, but ignores inter-region or cross geo-type correlations.

Indeed, the emergence of representation learning and adversarial learning techniques provide great potential to overcome the limitations of prior literatures. However, several unique challenges arise toward this goal. (1) How can we preserve intra-region geographic structures? (2) How can we preserve inter-region spatial autocorrelations that are described by pairwise graph regularization? (3) How can we simultaneously address (1) and (2) in a unified unsupervised learning framework?

First, what differentiate a region from other regions? To answer this question, we analogize a region with a webpage, which is uniquely identified by not just contents, but also webpage layout. Similarly, a region is uniquely identified not only by POIs (contents), but also by geographic structures. Graphs with nodes and edges have been proved to effectively describe network structures. We propose to regard POIs as nodes and construct multi-view graphs: (1) a POI-POI distance graph, and (2) a POI-POI mobility graph. This leads to a new problem reformulation: Given a region that is described by two spatial graphs, how can we learn region representations *collaboratively* from the two relational graphs to *preserve graph structures*? We propose to revise the auto-encoder network architecture through employing an assemble-disassemble strategy: an assemble step to aggregate multiple graphs into a fused embedding and a disassemble step to disaggregate a fused embedding into multiple graphs. To preserve the graph structures, we adopt the deep adversarial autoencoder to encode input graphs into embeddings that can minimize reconstruction loss as well as constrain the embedding space to mach a desired statistical characteristics.

Secondly, geographic entities exhibit spatial autocorrelations among them. If two regions show higher spatial autocorrelations, their embeddings are more likely to be close to each other as well. The pairwise inter-region spatial correlations of all regions can be described by a similarity matrix calculated with prior edge information (e.g., demographic data, geo-tagged posts). Traditional method of employing spatial autocorrelations as a graph regularization term (GRT) requires extensive parameter tuning efforts to identify the weight of GRT. Thus, we propose to translate GRT into a batch adversarial learning reformulation. We add a second adversarial learning component into the training process. The second attacker constrains the representations by matching the inter-region similarities of learned latent representations to the graph regularization term. Standard adversarial learning strategy draws the true samples from a prior distribution. We analogize this process with sampling subgraphs from the entire inter-region autocorrelation matrix. In this way, we exploit a dual adversarial strategy to learn the attack feedbacks benchmarked by not just intra-region geographic structures, but also inter-region spatial autocorrelations. Unfortunately, the feedbacks of the second attacker are from the inconsistency between embedding-based inter-region similarities and GRT, which means the second attacker relies on the encoder outputs of all inputs. Thus, a batch adversarial method is proposed to address this issue.

Along these lines, we develop an unsupervised collective graph-regularized dual-adversarial framework to learn deep spatial representations. Our main contributions are the following: (1) We construct multi-view spatial graphs to characterize the geographic structures of a region. (2) We reformulate the spatial representation learning problem as a joint objective of collaborative learning from multi-view graphs, preserving intra-region geographic structures, and preserving inter-region spatial autocorrelations. (3) To tackle the reformulation, we integrate an auto-encoder framework with an assemble-disassemble strategy and a dual-adversarial learning strategy as a unified unsupervised model. (4) We examine the impacts of different calculation methods of inter-region spatial autocorrelations on the quality of representations. (5) As applications, we exploit the learned representations of geographic regions for predicting urban community popularity. We conduct extensive experiments to demonstrate the enhanced performance of the proposed method with real-world big geo-tagged data (POIs, checkins, GPS trajectories).

2 PROBLEM STATEMENT AND FRAMEWORK OVERVIEW

We first introduce some important definitions and the problem statement, and then present an overview of the proposed method.

2.1 Definitions and Problem Statement

Definition 2.1. **Spatial Region** In this paper, a spatial region consists of a residential complex and its neighborhood area (e.g., area within a one-kilometer radius). In each residential complex, there are multiple apartment buildings and each apartment building has many apartments. Additionally, the neighborhood area of each residential complex consists of various POIs that serve different functions for residents and visitors in the community. For each region, we have access to its POIs, road networks, checkin texture comments, crowd flows at different time (taxi, bus, and bike GPS trajectories). Without loss of generality, a spatial region can be generalized to any spatial entity, that refers to a geographic unit which occupies a position in space with data describing its features and geographic location.

Definition 2.2. **Intra-region Geographic Structure** The intra-region geographic structure describes the geographic configuration and layout within a region, which are often represented by graphs with POIs as nodes and POI-POI relations as edges. There are multiple relations from different viewpoints between two POIs. For instance, we can measure the geographic distance between the POIs to form a distance graph and count trajectories between the POIs to form a mobility connectivity graph.

Definition 2.3. **Inter-region Spatial Correlations** The inter-region spatial autocorrelations refer to the pairwise geographic mutual information among all regions, which are represented by a correlation matrix $R^{K\times K}$ where K is the total number of regions and R_{ij} shows the correlation between region i and j. To describe the inter-region spatial autocorrelations, we learn three region-region correlation matrices by exploiting (i) geo-tagged textual description (i.e. property category, special features, business district, etc), (ii) temporal dynamics of crowd flows, and (iii) urban function topics.

Definition 2.4. **Problem Statement.** In this paper, we study the problem of unsupervised spatial representation learning by considering both intra-region geographic structures and inter-region spatial autocorrelations. We aim to learn the vector representations of regions which can preserve not only the geographic structures,

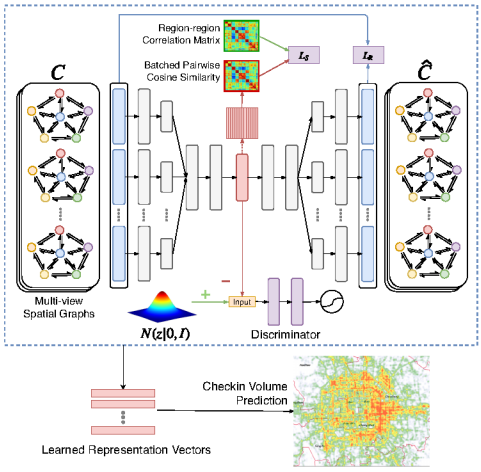


Figure 1: An overview of the proposed framework for unsupervised spatial representation learning

but also the region-region correlations. Formally, consider the existence of a set of K regions $C = \{c_k\}_{k=1}^K$. Each region $c_k \in C$ is described by a distance graph $\mathcal{G}_k^d \in G^d$, where $G^d = \{\mathcal{G}_k^d\}_{k=1}^K$, and a mobility connectivity graph $\mathcal{G}_k^m \in G^m$, where $G^m = \{\mathcal{G}_k^m\}_{k=1}^K$. The inter-region autocorrelation matrix is denoted by S. The objective is to learn a mapping function $f : \{G^d, G^m \rightarrow \mathbb{R}^n\}$, where n is the dimension of embedding vectors, to map multiple spatial graphs into the embedding space that preserves intra-region geographic structures and inter-region spatial autocorrelations. We formulate this problem as a task of unsupervised collective graph-regularized dual-adversarial representation learning. The reasons are: (1) Field data in real systems are lack of labels, which creates significant challenges for practical deployment. This task intends to build a feature learning framework with unlabeled, heterogeneous, and relational spatial data in an unsupervised fashion. (2) The intra-region geographic structures are represented by multiple spatial graphs, leading to a reformulation of collective learning from relational graphs. (3) The inter-region spatial autocorrelations are represented by pairwise graph regularization, which can be converted into an adversarial learning reformulation by regarding graph regularization as attacks.

2.2 Framework Overview

Figure 1 shows an overview of our proposed framework that includes the following essential tasks: (i) constructing multi-view spatial graphs for each region; (ii) unsupervised spatial representation learning via a collective graph-regularized dual-adversarial encoding-decoding framework; (iii) applications to checkin volume prediction. Specifically, in the first task, we collect large-scale POIs and human mobility data, construct both geographic distance graphs and human mobility connectivity graphs for each region. In the second task, we develop a collective graph-regularized dual-adversarial encoding-decoding framework to achieve a joint objective of intra-region structural preservation, relational learning among multi-view spatial graphs, and inter-region spatial correlations. In particular, for each region, the framework takes multi-view graphs as inputs. An assemble encoding step will fuse multi-view graphs into a fused latent embedding. Later, an disassemble step will dis-aggregate the fused embedding to reconstruct the original multiple graphs. In addition, we incorporate two adversarial components. The first adversarial component imposes a prior distribution on the embedding space to achieve peer preservation, in which two structurally similar graphs share similar representations. The second adversarial component matches the pairwise cosine similarities of embedding vectors with given inter-region spatial autocorrelations. In the third task, we exploit the learned representations of multi-view region graphs to support important applications such as checkin volume prediction.

3 CONSTRUCTING MULTI-VIEW GRAPHS OF SPATIAL REGIONS

We aim to learn the representations of the structural information of a spatial region from multi-view spatial graphs that describe a region from different perspectives: (1) geographical distance and (2) human mobility activity.

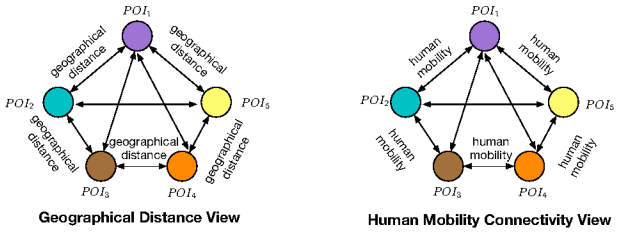


Figure 2: An example of multi-view graphs [8].

A View of Geographical Distance. The geographical distance view can demonstrate the static geographical distribution of POIs within regions. In the geographical distance based spatial graph, the vertexes denote POI categories, and the edges denote the average distance among POI categories [18]. We use POI categories instead of just POIs due to the reason that traditional graph embedding methods, such as autoencoder, cannot take inputs with dynamic sizes. Number of POIs in each community varies but the amount of POI categories stays fixed.

A View of Human Mobility Connectivity. The human mobility view can demonstrate the dynamic human mobility patterns within regions. In the human mobility connectivity based spatial graph, the vertexes denote POI categories, and the edges denote the human mobility connectivities among POI categories. To quantify the human mobility connectivity, we adopt the four-step method [27] as follows: Specifically, we first estimate POI visited probabilities as follows: Given the drop-off point dp of a taxi trace, we model the probability of a POI p visited by a passenger as a parametric function, whose input x is the road network distance between the drop-off point dp and the POI p: $P(x) = \frac{\beta_1}{\beta_2} \cdot x \cdot exp(1 - \frac{x}{\beta_2})$, where $\beta_1 = \max_x P(x)$ and $\beta_2 = \arg\max_x P(x)$; Second, we aggregate all probabilities from all drop-off points in taxi traces: $\tau(p) = \sum_{dp \in \mathcal{D}} P(dis(dp, p))$, where $\mathcal{D}$ is the drop-off point set of taxi traces in the region; Then, for each POI category i in a region, the POI category-level aggregated visit probability is given by: $\phi_i = \sum_{p \in i} \tau(p)$, where $p \in i$ denotes the POI p belongs to the i^{th} POI category; Finally, we calculate the flow probabilities from the i^{th} POI category to the j^{th} POI category:

$\phi_{\vec{ij}} = \begin{cases} \phi_i \cdot \phi_j, & if\ i \neq j \\ 0, & if\ i = j \end{cases}$. We use the flow probabilities to represent the human mobility connectivities among POI categories.

4 COLLECTIVE GRAPH-REGULARIZED DUAL-ADVERSARIAL REPRESENTATION LEARNING

We present a collective graph-regularized dual-adversarial representation learning framework to model both intra-region structures and inter-region autocorrelations.

4.1 Model Intuitions

There are intra-region multi-view geographic structures and inter-region spatial autocorrelations among regions. Therefore, in our approach, we build the representation learning framework of geographic regions based on the following intuitions.

Intuition 1: Structural Preservation After reducing regions into graphs, we need a representation learning model to convert graphs into embedding vectors for automated quantification and profiling. As a result, the model should be capable of projecting graphs into latent space while preserve corresponding features and graph structures.

Intuition 2: Collective Learning from Multi-view Graphs. Each region can be represented by more than one spatial graph as its features can be characterized from different perspectives. Each independent view of the regions as well as their interactions can contribute to the structure representation. In consequence, the model should be able to collaboratively learn representations of regions from multiple relational graphs.

Intuition 3: Inter-region Spatial Autocorrelation. Different regions can also contain underlying spatial autocorrelations. For example, those regions with similar intrinsic properties (e.g., similar urban functions) should share similar feature representations as well. In another word, the proposed method should have the ability to model region-region spatial autocorrelation in representation learning.

4.2 Base Models

Autoencoder [2] is an unsupervised neural network model that is trained to project input signals through layers of linear mapping into lower-dimension embedding space and can map embedding vectors back to input space such that the reconstructed input is as close to the original input as possible.

The network consists of two main parts: encoder and decoder. The encoder takes an input x and maps it to an embedding vector z. The process can be viewed as an encoder function $z = \sigma(Wx + b)$ where b is the bias term, W is the weight matrix, and σ is the activation function. The decoder reconstruct the embeddings back. This step can be viewed as a decoder function $\hat{x} = \sigma(W'z+c)$ where c is the bias term, W' is the weight matrix, and σ is the activation function. The entire autoencoder network is trained to minimizing the reconstruction error $\mathcal{L}_R(x, x') = \|x - x'\|^2$.

Recent autoencoders, such as Variational Autoencoder and Adversarial Autoencoder, incorporate encoder and decoder as stochastic mappings, $p_{encoder}(z|x)$ and $p_{decoder}(\hat{x}|z)$, instead of deterministic functions. In Variational Autoencoder, a KL divergence penalty is used to impose an arbitrary prior distribution on the embedding vectors. While in Adversarial Autoencoder, generative adversarial network is used to adversarially learn an encoder that can map the aggregated posterior to a prior distribution. For our base model, we utilize the adversarial autoencoder mode.

4.3 Integrating Graph Regularization via Dual Adversarial Learning

Traditional adversarial autoencoder network constrains the aggregated posterior distribution of encoder output to match an arbitrary prior distribution. The model does that by attaching an adversarial network to the embedding vector of the autoencoder. The encoder also functions as a generator and the adversarial network ensures that the output of encoder can trick the discriminator to think that the output of encoder is from the true prior distribution.

We expect to achieve inter-region graph regularization by adding a second adversarial learning network to the embedding layer of the autoencoder. To measure the inter-region spatial correlations, we can calculate the pairwise cosine similarity matrix for learned embedding vectors. The true prior similarity matrix is provided as the target matrix. Meanwhile, in our second adversarial network, we introduce a new loss function $\mathcal{L}_S$, which measures the normalized Frobenius distance between two matrices, $R \in \mathcal{R}^{n \times n}$, the target similarity matrix given as prior, and $M \in \mathcal{R}^{n \times n}$, the calculated similarity matrix obtained through taking normalized inner product of learned embedding vectors. The goal is to minimize the loss function $\mathcal{L}_S$ to match the inter-region correlation matrix with prior correlation matrix.

We implement $\mathcal{L}_S$ as follows,

$$\mathcal{L}_S(M, R) = \|M - R\|_F \tag{1}$$

where each element M_{ij} is calculated by $M_{ij} = \frac{z_i \cdot z_j}{|z_i||z_j|}$ and z_k represents the learned embedding vector of kth region.

Since we typically train our model using mini batches due to computational resource limitations, we can tweak the loss function a little to adapt the batch training method. Given b, the mini-batch size, $\mathcal{I}_B$, the set of region indices in the batch, K, total number of input regions, the loss function $\mathcal{L}_S$ is defined as

$$\mathcal{L}_S(M_b, R, U) = \left\| M_b - URU^T \right\|_F \tag{2}$$

where $M_b \in R^{b \times b}$ includes pairwise similarity among the learned embedding vectors of batched inputs, $U \in R^{b \times K}$ is the selection matrix including kth row of identity matrix for $k \in \mathcal{I}_B$ such that the product URU^T represents the prior sub-similarity matrix with matched indices as the batched inputs.

4.3.1 Inter-region Autocorrelations. One key question raised here is how we calculate the similarity matrix. The pairwise similarity of embedding vectors we learn should reveal underlying spatial autocorrelations among regions that are not described by structural similarities. Thus, we show three methods of defining the similarity $S_{i,j}$ between region i and region j.

POI Distribution Based Autocorrelation Since regions serve diverse functions, we can assume that the distribution of POI categories of each region varies too. As a result, based on the difference

between two regions' POI distributions, we can acquire the similarity between those two regions. We construct the POI distribution vector for every region by setting each element in the vector to the frequency of the corresponding POI category.

Text Based Autocorrelation Each region comes along with textual information that describes some properties of the region in detail, such as property category, special features of housing, business district the region belongs to, surrounding environment, school district, etc. Those information are essential to depict a region yet they cannot simply be revealed from geographical structure. Therefore, we adopt Word2Vec to obtain document vectors from description word lists and calculate the pairwise cosine similarity between document vectors to construct the similarity matrix.

Temporal Mobility Dynamics Based Autocorrelation Temporal patterns of human mobility can reflect a region's region type as well as its vitality. We model human mobility of each region based on the amount of arriving and leaving activities at that region within different time intervals. In order to represent temporal mobility dynamics of each region, we count the frequency of arriving and leaving events during each 2-hour interval and build two 12-dimensional vectors M_A for arriving mobility and M_L for leaving mobility. Then, we calculate the similarity between two regions based on their mobility vectors.

4.4 Collective Adversarial Learning

4.4.1 ***Collective Learning from Multi-view Spatial Graphs via assemble-disassemble Strategy.*** The basic autoencoder is not capable of learning region representations collaboratively from multiple relational graphs. So we propose to incorporate an assemble-disassemble strategy in the framework. In the assemble step, the vector of each spatial graph, denoted by $(\mathcal{G}_1, \mathcal{G}_2, .., \mathcal{G}_k)$, is first transformed to a lower-dimension vector representation individually and then all low-dimension vector representations of multi-view graphs are aggregated together into a fused input.

The detailed assembling and disassembling procedures of multi-view spatial graphs are shown in Figure 3. Given a spatial region,

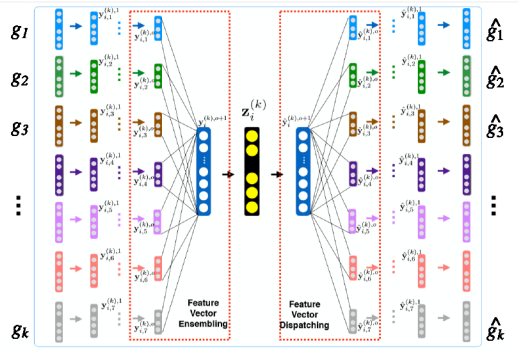


Figure 3: Illustration of Assemble-disassemble Strategy

we have k-view graphs which are represented by $\mathcal{G}_k$. We transform all $\mathcal{G}_k$ into a series of vectors. Each vector is first projected to lower dimension through multi-layer perceptrons. Then these low-dimensional vectors of different views, serving as multi-inputs, are mapped together into one vector space through one layer of network. The vector resulted from this assemble step is then further projected into embedding space. The disassembling process inverts the above procedures to reconstruct the original multi-view graphs from the embedding vector.

4.4.2 ***Integrating Inter-Region Spatial Autocorrelation via Dual-Adversarial Learning.*** In adversarial autoencoder, the model aims to reduce the reconstruction loss as well as minimize the Jensen-Shannon divergence, $JSD(p(z)||q(z))$, on the embedding vectors,

$$JSD(p(z)||q(z)) = \frac{1}{2}KL\left(p\middle\|\frac{p+q}{2}\right) + \frac{1}{2}KL\left(q\middle\|\frac{p+q}{2}\right) \quad (3)$$

where $p(z)$ is the prior distribution and $q(z)$ represents the aggregated posterior distribution defined as follows,

$$q(z) = \int_x q(z|x)p_{data}(x)dx \quad (4)$$

where $p_{data}(x)$ is the data distribution.

However, minimizing JSD is intractable. Recall that maximizing the likelihood estimation is also equivalent to minimizing KL divergence. So the adversarial autoencoder leverages a discriminant network and the objective function of the adversarial network becomes the following,

$$\min_G \max_D E_{p(z)}[\log D(z)] + E_{p_{data}(x)}[\log(1 - D(G(x)))] \quad (5)$$

In our framework, CGAL, in order to incorporate both intra-region structural preservation and inter-region spatial correlation regularization, we append another adversarial network to the embedding layer of adversarial autoencoder. Formally, we obtain the objective function of graph-regularized dual-adversarial network as follows:

$$\min_G \max_{D_1, D_2} E_{p(M|\mathcal{K})}[\log D_1(M)] + E_{p_{data}(X)}[\log(1 - D_1(G(X)^T G(X)))] + E_{p(z)}[\log D_2(z)] + E_{p_{data}(x)}[\log(1 - D_2(G(x)))]$$

where G refers to the generator, D_1 measures the gap between calculated autocorrelation matrix M and the true prior autocorrelation matrix R, D_2 discriminates generated embedding vectors against real embedding vectors coming from prior distribution, and $\mathcal{K}$ refers to the hidden prior knowledge of all regions from which we calculated regions' pairwise similarity. D_1, along with G, aims to minimize the difference between calculated pairwise similarity matrix of embedding vectors and the given inter-region similarity matrix. D_2, along with G, imposes a prior distribution on the aggregated posterior distribution of generated embedding vectors. To train the graph-regularized dual-adversarial network, we alternate between learning the generator, G, and the discriminators, D_1 and D_2. The objective follows a minimax game where D_1 and D_2 aim to distinguish embedding vectors coming out of real distribution from generated ones while G intends to fool the discriminative models. To train the whole CGAL model, we also include the reconstruction error of the autoencoder network, i.e., $E_{q(z|x)}[-logp(x|z)]$. During training, we alternate between optimizing the adversarial network and minimizing the reconstruction error.

4.4.3 ***Solving the Optimization Problem.*** In practice, to solve the optimization problem, we need to optimize the discriminator D_2 with parameters θ, the generator G with parameters ψ, and the autoencoder α. Even though the prior autocorrelation matrix is sampled from a prior knowledge distribution, we can not actually generate it for more times during the experiment. In another word, the prior autocorrelation matrix we obtain is deterministic. However, standard adversarial learning requires sampling true data points from a prior distribution. To tackle this problem, we propose to leverage the mini-batch training strategy. ***We can view the process of drawing random sub-matrix from the whole inter-region autocorrelation matrix equivalent to the process of sampling from true distribution.*** And then, we try to minimize the gap between the calculated autocorrelation matrix and the sampled sub-matrix. Since the encoder part of autoencoder also acts as the generator, ψ and α will share part of the parameters. The updating rules of parameters are as follows,

$$\theta_{t+1} \leftarrow \theta_t - \epsilon_D \frac{\partial}{\partial \theta}\Big[E_{z\sim p_{real}} \log D_2(z) + E_{x\sim p(x)} \log(1 - D_2(G(x)))\Big] \tag{6}$$

$$\psi_{t+1} \leftarrow \psi_t - \epsilon_G \frac{\partial}{\partial \psi}\Big[E_{x\sim p(x)} \log(1 - D_2(G(x)))\Big] \tag{7}$$

$$\alpha_{t+1} \leftarrow \alpha_t - \epsilon_{ae} \frac{\partial}{\partial \alpha}\Big[\sum_{i\in b} \|x_i - \hat{x_i}\|^2\Big] \tag{8}$$

$$\psi_{t+1} \leftarrow \psi_t - \epsilon_G \frac{\partial}{\partial \psi}\Big[\Big\|G(X)^T G(X) - URU^T\Big\|_F\Big] \tag{9}$$

where ϵ_D, ϵ_G, and ϵ_{ae} are learning rates for discriminator, generator, and autoencoder respectively. Equation (6) and (7) together play a min-max game, aiming to impose a normal distribution on embedding vectors by updating generator and discriminator iteratively. Equation (8) trains the autoencoder to minimize the reconstruction error and Equation (9) trains the generator to approximate the generated autocorrelation matrix with the prior autocorrelation matrix.

5 AN APPLICATION: PREDICTING REGIONAL POPULARITY

The representations learned from *CGAL* are capable of capturing the intra-region geographic configuration, structural information, as well as inter-region spatial autocorrelations. And thus, we can feed them into many important downsteam applications. One such example is to predict the popularity of regions. More specifically, we can represent the popularity of an urban region with the number of mobile check-in events, such as hotel check-ins, mall check-ins, restaurant check-ins, etc, occured in that region. Intuitively, a region with a larger amount of check-in events indicates that people are more willing to pay to transport to that region, to shop, to consume, to live, or to carry out many other activities in that region.

For this task, we take our learned representations of geographical regions as inputs of a linear regression model. Formally, let z_i denotes the embedding feature vector of region r_i, P_i denotes the ground truth of region i's popularity, and $\hat{P}_i$ indicates the predicted measure of region i's popularity. Our objective is to fit a linear regression model:

$$\hat{P}_i = \mathbf{W}^T z_i + \mathbf{b} \tag{10}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the weights and biases respectively. Through learning the regression model with representations obtained from our framework and other baseline methods, we can conduct performance analysis with experimental results.

6 EXPERIMENTAL RESULTS

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

6.1 Data Description

Table 1 shows the statistics of four data sources used in the experiment. The taxi GPS traces are collected from a Beijing taxi company. Each trajectory contains trip id, distance(m), travel time(s), average speed(km/h), pick-up time and drop-off time, pick-up location and drop-off location. Also, we extract POIs related data from Dianping.COM which is a business review site in China. Moreover, we crawl the Beijing residential region data from www.soufun.com which is the largest real-estate online system in China. Furthermore, the check-in data of Beijing is crawled from www.jiepang.com which is a Chinese version of Fourquare. Each check-in event includes name, category, address, longitude and latitude of POIs.

Table 1: Statistics of the Experimental Data

Data Sources	Properties	Statistics
Taxi Traces	Number of taxis	13,597
	Effective days	92
	Time period	Apr. - Aug. 2012
	Number of trips	8,202,012
	Number of GPS points	111,602
	Total distance(km)	61,269,029
Residential Regions	Number of residential regions	2,990
	Latitude and Longitude	
	Time period of transactions	04/2011 - 09/2012
POIs	Number of POIs	328668
	Number of POI categories	20
	Latitude and Longitude	
Texts	Number of textual descriptions	2,990
	Time Period	04/2011 - 09/2012
Check-Ins	Number of check-in events	2,762,128
	Number of POI categories	20
	Time Period	01/2012-12/2012

6.2 Evaluation Metrics

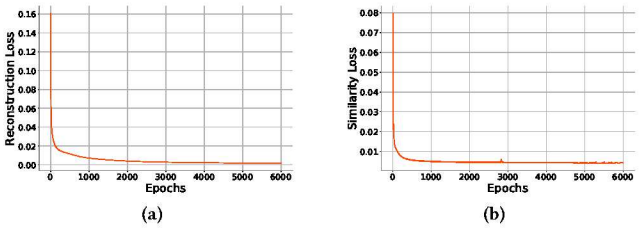
To evaluate our proposed representation learning framework, we perform a regression task on regional check-in volume. Each region k is associated with a benchmark check-in volume v_i and a predicted check-in volume $\hat{v}_i$. To show the effectiveness of the proposed model, we use the following metrics for evaluation.

Mean Squared Logarithmic Error. Since both actual and predicted check-in volumes are likely to be huge numbers, we utilize *Mean Squared Logarithmic Error* (MSLE) to measure regression errors. $MSLE = \frac{1}{N}\sum_{i=1}^{N}(\log(\hat{v_i}+1) - \log(v_i+1))^2$, where N is the number of regions. The lower the MSLE is, the better the learned representation is.

6.3 Baseline Algorithms

We compare the performances of our method against the following baseline algorithms. For all methods, we set the representation size as 32.

(1) Adversarial Autoencoder. The Adversarial Autoencoder (AAE) model [15] minimizes the loss between the original data representations and reconstructed ones while also imposing an aggregated



(a) (b)

Figure 4: (a) Convergence of reconstruction loss. (b) Convergence of similarity loss.

posterior distribution on the latent vectors through adversarial learning. In the experiments, we set the number of hidden layers = 4.

(2) DeepWalk. The DeepWalk model [19] performs network embedding task by leveraging local information learned from short random walks on vertices in graphs. In the experiments, we set the number of random walks = 10, the length of random walks = 40, and the window size of skip-gram model = 10.

(3) GraphFactorization. GraphFactorization is a matrix factorization based graph embedding model. It represents graphs in the form of matrices and matrices are factorized to obtain embedding vectors.

(4) GraRep. GraRep [4] learns low dimensional embedding vectors of graphs by aggregating representations obtained through matrix factorization method applied on all k-step transition probability matrix. By considering all k-step transitions, GraRep is able to learn both local structural information and global structural properties of the graph. In the experiments, we set the maximum number of steps $K = 4$.

(5) HOPE. HOPE [16] is a scalable graph embedding algorithm that is developed to preserve high-order proximities of large graphs while also capturing the asymmetric transitivity.

(6) Node2Vec. Node2Vec [9] learns continuous feature representations for nodes in a graph while preserving network neighborhoods of nodes. It leverages a biased random walk procedure to help efficiently explore neighborhoods of nodes. We set the number of walks = 10, the length of walks = 80, the window size of skip-gram model = 10, the return hyperparameter = 0.25, and the inout hyperparameter = 0.25.

(7) SDNE. SDNE [24] aims to learn graph representations that can capture highly non-linear structure of networks. It preserves such network structure by collectively exploiting the network's first-order and second-order proximity. We set the dimension of intermediate layer as 1000, hyperparameter that controls the first order proximity loss as $1e-6$, the batch size = 200, and the learning rate = 0.01.

(8) PCA. PCA aims to find low dimensional representations of graphs by applying orthogonal transformations on original graph representation vectors such that the result has the largest variance.

Baseline models (2)-(7) are all implemented with OpenNE (https://github.com/thunlp/OpenNE), an open source toolkit for network embedding. For our proposed framework CGAL, we set the dimensions of assemble layers and corresponding disassemble layers as 512 and 256. The dimensions of hidden layers in the encoder are set to be 256 and 128. Tensorflow implementation of Adam algorithm is used to optimize the model. We set the batch size as 281, the learning rate to be 0.00001 and the training epochs as 6000. For representation learning on single-view graph, we use GAL model

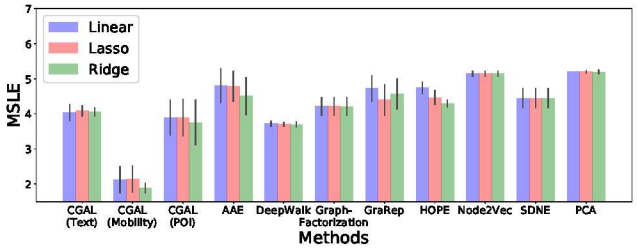


Figure 5: Overall performance comparison with CGAL models and other baseline methods using Linear, Lasso, and Ridge regression models.

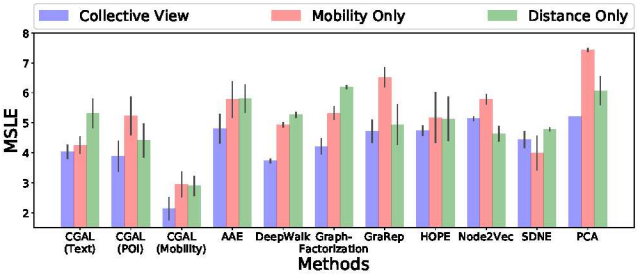


Figure 6: Performance comparison with models using single-view and collective-view relational graphs.

which does not contain assemble and disassemble networks. The hyperparameters remain the same. In each experiment, we randomly select 80% of data as training dataset and the rest 20% as testing dataset. For each regression model on every representation learning method, we run the experiments for 100 trials and take the mean MSLEs as our results.

6.4 Study of Convergence

In Figure 4, we analyze the convergence rate of CGAL in terms of its reconstruction loss and similarity loss. Both losses reduce rapidly with a few epochs but takes some more time to finally converge. We observe that there is fluctuation in the similarity loss but only to a small extent. The models takes approximately 70 minutes to run 6000 epochs.

6.5 Overall Performance Comparison

In the first study, the prediction errors of three different linear regression models using representation vectors learned from CGAL models are compared with those obtained from other baseline representation learning methods as mentioned above. Figure 5 presents the results of comparison.

CGAL models outperforms most baseline methods. However, CGAL models regularized with POI based similarity matrix and with text based similarity matrix result in greater MSLEs than DeepWalk. Whereas CGAL regularized with mobility-based similarity matrix performs significantly better than any other method. We will analyze the impact of different similarity matrix on CGAL in a later study.

6.6 Study of Performance in Different Views

This experiment studies the effect of learning representations from multi-view relational graphs collectively and from single-view graph independently. We demonstrate the comparison results in

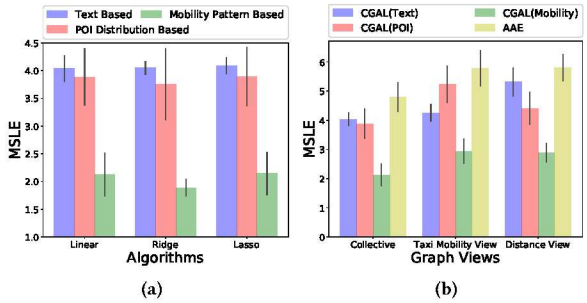


Figure 7: (a) Regression performance comparison of CGAL models using different similarity matrices. (b) Performance comparison between dual-adversarial learning and single adversarial.

Figure 6. As we can see, models learning representation vectors from multi-view relational graphs significantly outperform those learning from only mobility view or only distance view except in Node2Vec and SDNE models. By learning from two views collaboratively, models are able to capture information of the regions from each viewpoint as well as from the interaction of both perspectives.

6.7 Study of Performance with Similarity Matrix Constructed using Different Priors

As discussed in section 4.3.1, we calculate our similarity matrix used in CGAL with three types of prior information - text based autocorrelation, POI distribution based autocorrelation, and temporal mobility dynamics based autocorrelation. In this study, we examine the performance of CGAL models regularized with those three inter-region similarity matrices. The comparison of regression results using all three regression methods are shown in 7a. We notice that there is no significant difference between the performance of CGAL models with text based regularization and POI distribution based regularization. However, we observe significant improvement in regression performance with CGAL regularized with mobility based similarity regularization. This is likely due to that the information our model obtained from mobility based similarity matrix has strong association with checkin volumes. Thus representation vectors of regions with close checkin volumes are enforced to be similar to each other as well.

6.8 Study of Performance in Dual-Adversarial Learning

CGAL integrates dual-adversarial learning rather than a single adversarial component in AAE. Here we compare the linear regression performance with representation vectors obtained from CGAL models and AAE using different views, as shown in Figure 7b. As we can see, with dual-adversarial learning implemented, CGAL models consistently performs better than AAE. With the second adversarial network, CGAL not only preserves the intra-region geographic structure, but also maintains the inter-region spatial correlations which helps learn more effective and powerful representations.

6.9 Impact of Regularization Term in Parameter Learning

We notice that when comparing the performance of three linear regression models using representation vectors learned from CGAL,

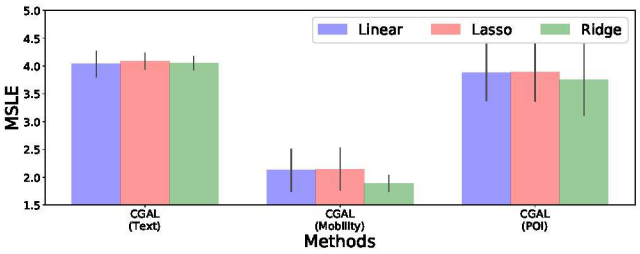


Figure 8: Comparison of regression performance using three different linear regression models

Lasso generally performs a bit worse than the other two, as illustrated in Figure 8.

Our interpretation is that, as Lasso regularizes the parameters of learned linear model with $\mathcal{L}_1$ norm, it forces some impactful parameters to be zeros, which, in turn, degrades the model performance. Further, this finding actually reflects that CGAL can very effectively compress all useful information of the graphs into the learned representation vectors.

7 RELATED WORK

Graph representation learning and its variants. Graph representation learning algorithms can be categorized into three main approaches: (1) the probabilistic models, (2) the manifold-learning approaches, and (3) the reconstruction-based algorithms. Probabilistic model approaches use unsupervised feature learning to learn a hierarchy of features one level at a time [11, 20]. For example, *Wang et al.* used a regression learner to learn the optimized layout of heterogeneous elements on the search result page (SERP) [29]. In the second category, the large majority of the algorithms adopt a non-parametric approach, based on a training set nearest neighbor graph [1]. The auto-encoder based methods projects the instances in original feature representations into a lower-dimensional feature space via a series of non-linear mappings, by minimize the loss between original and reconstructed spaces [10, 12].

Adversarial Learning Methods. More advanced methods integrate adversarial strategies and multiple-input to improve representation learning. For example, the work in [17] proposed adversarially regularized graph autoencoder (ARGA) and adversarially regularized variation graph autoencoder (ARVGA) to encode the topological structure and node features of a graph into a latent representation which can be used by the decoder to reconstruct the graph structure and is enforced to match a prior distribution. [3]proposed Adversarial Learning for Knowledge Graph Embeddings (KBGAN) which has a generator that generates negative samples to fool the discriminator and aims to adversarially train a good discriminator that produces final knowledge graph embeddings.

Collective Multi-View Learning. Real-world data often provide more than one set of information over the same set of entities. For graphs, this is usually represented by multiple sets of edges associated with the same nodes. One simple method to learn graph representations collectively is to embed each view of the graph first and then concatenate all learned embeddings for each node [21]. There are also probabilistic collective matrix factorization (PCMF) methods that learn collective representations over multi-view data to fully extract complementary information. Some more complex

methods are based on matrix factorization [23] and spectral embedding [6] which focus more on clustering multi-view graphs.

Spatial Representation learning and Applications. Our work is relevant to spatio-temporal representation learning, which is the elevation of graph representation learning in the spatio-temporal contexts. Graph representation learning, also known as graph embedding, aims to learn a low-dimensional vector to represent vertexes or graphs[7, 14, 16, 24, 26, 28, 29]. For spatio-temporal representation learning, *Wang et al.* proposed a collective embedding framework to learn the community structure from multiple periodic spatial-temporal graphs of human mobility [27]. *Yao et al.* developed an embedding method to learn the urban functions by exploring human mobility patterns.

8 CONCLUSION REMARKS

We studied the problem of of deep unsupervised spatial representation learning. We constructed multi-view spatial graphs to characterize the geographic structures of a region. Then, we reformulated the spatial representation learning problem as a joint objective of collaborative learning from multi-view graphs, preserving intra-region geographic structures, and preserving inter-region spatial autocorrelations. Specifically, we developed an assemble-dissemble paradigm to take multiple graph views as inputs. Moreover, we incorporated two adversarial components. The first adversarial component imposes a prior distribution on the embedding space to achieve peer preservation, and the second adversarial component matches the pairwise cosine similarities of embedding vectors with given inter-region spatial autocorrelations. In addition, we applied our method to the applications of predicting regional popularity. The extensive experimental results with real-world data demonstrated the effectiveness of our method.

9 ACKNOWLEDGMENT

Dr. Xiaolin Li was partially supported by the National Science of Foundation China via grant number: 61773199.

REFERENCES

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

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

[3] Liwei Cai and William Yang Wang. Kbgan: Adversarial learning for knowledge graph embeddings. *arXiv preprint arXiv:1711.04071*, 2017.

[4] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 891–900. ACM, 2015.

[5] Winnie Cheng, Chris Greaves, and Martin Warren. From n-gram to skipgram to concgram. *International journal of corpus linguistics*, 11(4):411–433, 2006.

[6] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. Clustering on multi-layer graphs via subspace analysis on grassmann manifolds. *IEEE Transactions on signal processing*, 62(4):905–918, 2014.

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

[8] Yanjie Fu, Pengyang Wang, Jiadi Du, Le Wu, and Li Xiaolin. Efficient region embedding with multi-view spatial networks: A perspective of locality-constrained spatial autocorrelations. In *Proceedings of the 33th AAAI Conference on Artificial Intelligence*, page to appear. AAAI, 2019.

[9] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

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

[11] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, pages 609–616. ACM, 2009.

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

[13] Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014.

[14] Hao Liu, Ting Li, Renjun Hu, Yanjie Fu, Jingjing Gu, and Hui Xiong. Joint representation learning for multi-modal transportation recommendation. In *AAAI*, page to appear, 2019.

[15] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[16] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114. ACM, 2016.

[17] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder. *arXiv preprint arXiv:1802.04407*, 2018.

[18] Guannan Liu Yanjie Fu Charu Aggarwal Pengyang Wang, Jiawei Zhang. Ensemble-spotting: Ranking urban vibrancy via poi embedding with multi-view spatial graphs. In *Proceedings of 2018 SIAM International Conference on Data Mining (SDM'18)*. SIAM, 2018.

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

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

[21] Yu Shi, Fangqiu Han, Xinran He, Carl Yang, Jie Luo, and Jiawei Han. mvn2vec: Preservation and collaboration in multi-view network embedding. *arXiv preprint arXiv:1801.06597*, 2018.

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

[23] Wei Tang, Zhengdong Lu, and Inderjit S Dhillon. Clustering with multiple graphs. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 1016–1021. IEEE, 2009.

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

[25] Hongjian Wang and Zhenhui Li. Region representation learning via mobility flow. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 237–246. ACM, 2017.

[26] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. Multi-task representation learning for demographic prediction. In *European Conference on Information Retrieval*, pages 88–99. Springer, 2016.

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

[28] Pengyang Wang, Yanjie Fu, Jiawei Zhang, Pengfei Wang, Yu Zheng, and Charu Aggarwal. 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*, pages 2457–2466. ACM, 2018.

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

[30] Zijun Yao, Yanjie Fu, Bin Liu, Wangsu Hu, and Hui Xiong. Representing urban functions through zone embedding with human mobility patterns. In *IJCAI*, pages 3919–3925, 2018.

[31] Hengtong Zhang, Yaliang Li, Fenglong Ma, Jing Gao, and Lu Su. Texttruth: an unsupervised approach to discover trustworthy information from multi-sourced text data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2729–2737. ACM, 2018.

[32] Liang Zhang, Keli Xiao, Hengshu Zhu, Chuanren Liu, Jingyuan Yang, and Bo Jin. Caden: A context-aware deep embedding network for financial opinions mining. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 757–766. IEEE, 2018.