Learning Effective Road Network Representation with Hierarchical Graph Neural Networks

Ning Wu
Beijing Advanced Innovation Center for BDBC,
School of Computer Science and Engineering
Beihang University, Beijing, China
wuning@buaa.edu.cn

Wayne Xin Zhao
Gaoling School of Artificial Intelligence,
Renmin University of China
Beijing, China
batmanfly@gmail.com

Jingyuan Wang*
State Key Laboratory of Software Development
Environment, Beihang University
Beijing, China
jywang@buaa.edu.cn

Dayan Pan
MOE Engineering Research Center of ACAT,
School of Computer Science and Engineering,
Beihang University, Beijing, China
dayan@buaa.edu.cn

ABSTRACT

Road network is the core component of urban transportation, and it is widely useful in various traffic-related systems and applications. Due to its important role, it is essential to develop general, effective and robust road network representation models. Although several efforts have been made in this direction, they cannot fully capture the complex characteristics of road networks.

In this paper, we propose a novel Hierarchical Road Network Representation model, named *HRNR*, by constructing a three-level neural architecture, corresponding to "*functional zones*", "*structural regions*" and "*road segments*", respectively. To associate the three kinds of nodes, we introduce two matrices consisting of probability distributions for modeling segment-to-region assignment or region-to-zone assignment. Based on the two assignment matrices, we carefully devise two reconstruction tasks, either based on network structure or human moving patterns. In this way, our node presentations are able to capture both structural and functional characteristics. Finally, we design a three-level hierarchical update mechanism for learning the node embeddings through the entire network. Extensive experiment results on three real-world datasets for four tasks have shown the effectiveness of the proposed model.

CCS CONCEPTS

• **Information systems** → *Spatial-temporal systems*;

KEYWORDS

Graph Neural Network, Road Network, Representation Learning

*Corresponding author.

KDD '20, August 23–27, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403043

ACM Reference Format:
Ning Wu, Wayne Xin Zhao, Jingyuan Wang, and Dayan Pan. 2020. Learning Effective Road Network Representation with Hierarchical Graph Neural Networks. In *26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403043

1 INTRODUCTION

Nowadays, intelligent transportation systems are becoming increasingly important in daily life by providing traffic-related applications, such as route plan [2, 6, 24, 28, 31], arrival time estimation [10, 11] and next-location prediction [4, 14, 29]. A core component of such a system is *road network*, which consists of a network of interconnected road segments to accommodate vehicle and pedestrian traffic [6, 12]. Road network generally forms the most basic transport infrastructure within urban areas. It is widely useful in various traffic-related systems and applications [12, 20–22, 36, 39].

Due to its important role, it is essential to develop suitable methods to effectively characterize and model road networks, especially in a general way. Early research mainly considers road networks as constraints and adopt standard graph data structure for developing their algorithms [29, 38]. More recently, deep learning has shed light on the modeling of road network. Several recent studies start to utilize network or graph representation learning for obtaining node representations over road network [5, 24, 25]. In this way, the underlying characteristics of road network can be extracted and utilized, which is expected to improve the performance of downstream applications.

However, road network is a rather complex system, and it is not easy to design effective representation learning methods. There are at least three major issues, which have not been studied by previous works, to address. First, road network is not *flat*. It naturally organizes traffic units as "clusters", either structural (*e.g.,* transportation hub) or functional (*e.g.,* commercial area). Besides, some traffic units are more important and undertake more significant transportation task through the road network. While, previous studies [5, 24, 25] usually adopt standard graph neural networks and treat nodes equal, which cannot characterize the hierarchical structure. Second, road network might not be "*small-world*", tending to have long average paths. For example, the length of arterial roads typically increases with the growth of urban areas. However, in typical graph neural network [5, 7, 19], only messages from nearby nodes are aggregated, which cannot effectively capture long-range dependency among nodes. Third, road network mainly reflects structural characteristics, while other aspects of information might not be obtained through network structure. For example, it is usually difficult to determine the functional role (*e.g.,* shopping mall) of a traffic unit just based on its road connections.

To address these issues, the focus of this paper is to design a general, effective and robust road network representation method for various downstream applications. Our key idea is to develop hierarchical graph neural network for learning such representations. By taking a hierarchical organization, we can gradually form more abstractive clusters by aggregating fine-grained units, encoding useful characteristics at different levels. Especially, we expect that the hierarchical organization can correspond to actual aggregation of traffic units in road networks, such as the aforementioned structural or functional clusters. For this purpose, we incorporate two kinds of virtual nodes into the hierarchy, namely structural region and functional zone. Structural regions are mainly used to characterize spatially connected road segments, serving as some traffic role, *e.g.,* overpass and crossing. Furthermore, functional zones are formed on top of structural regions, providing some kind of functionality for traffic users, *e.g.,* shopping area. With such a three-level organization, we can alleviate the issue related to long-range node dependency, since we can first perform message sharing at a high level, and then propagate the information to low-level nodes. For the third issue, we consider incorporating real trajectory data of users for complementing the structure information. As shown in previous studies [23, 36, 40], users' trajectory data can be used to discover the underlying functional or lifestyle-related patterns.

To this end, in this paper, we propose a novel *Hierarchical Road Network Representation* model, named *HRNR*. We construct a three-level neural architecture by following the hierarchy "*functional zones*" $\longrightarrow$ "*structural regions*" $\longrightarrow$ "*road segments*". We first apply spectral clustering to construct structural regions by aggregating spatially connected road segments. Furthermore, we form functional zones by composing functionally related structural regions. To associate the three kinds of nodes, we introduce two assignment matrices, modeling segment-to-region or region-to-zone membership. The two matrices characterize the probability distributions of road segments in a structural region and probability distributions of structural regions in a functional zone, respectively. Based on the two assignment matrices, we carefully devise two reconstruction tasks, either based on network structure or human moving patterns. In this way, we can drive the learned node embeddings to capture both structural and functional characteristics. Finally, we design a three-level hierarchical update mechanism for learning the node embeddings through the entire network.

To our knowledge, it is the first time that road network representations have been learned with hierarchical graph neural networks, capturing both structural and functional characteristics. Our model is able to naturally model long-range dependencies between distant nodes on the road network, and utilize trajectory data to extract the functional characteristics. Our model provide a general representation learning method for various downstream traffic-related applications. We construct extensive experiments on four typical application tasks using three real-world datasets. Experimental results demonstrate the effectiveness of the proposed model.

2 RELATED WORK

Our work is related to the following research directions.

Modeling Road Networks. Since road network is the basic component of the transportation systems, various applications have incorporated it for developing the algorithms, such as next-location prediction [4, 14, 29], route plan [2, 6, 24, 28], arrival time estimation [10, 11] and destination prediction [8, 32, 33]. In order to utilize road network information, early studies mainly focus on designing heuristic constraints [6, 42] or constructing graph-based algorithms on the road network [30, 37, 38]. Later on, statistical models such as Hidden Markov Models have been used to model the location transitions over the road networks [15, 17]. With the rapid growth of deep learning techniques, several studies try to learn effective node representations from road network, including RNN-based models [29], graph convolution networks [5], graph attention network [24] and other types of networks [25]. Although these studies have improved the application performance with the enhanced data representations, they lack a comprehensive consideration of the proposed issues in Section 1. Specially, these methods usually focus on some specific tasks, which is not flexible to adapt to other tasks.

Graph Representation Learning. Recent years have witnessed the success of deep learning in modeling graph data. In specific, Graph Neural Networks (GNN) have been widely used for modeling complex graph data for learning effective node characteristics. Two classic models are graph convolution networks (GCN) [7] and graph attention network (GAT) [19]. The basic procedure is to perform message passing and aggregate the message from neighborhoods. Based on such a core architecture, various variants have been proposed to improve the original network [24]. Especially, the efforts on modeling hierarchical or structural characteristics are quite related to our work, including differentiable graph pooling [35], geometric aggregation scheme [18], and heterogeneous or meta-path-driven attention aggregation [27, 41]. However, all these studies are not tailored to road networks. It is not suitable to directly apply these studies to model road networks.

Our work is based on the extensive studies on traffic-based application tasks [10, 11, 29]. Instead of focusing on some specific task, we design a general, capable and robust road network representation learning model, so that it can provide effective representations for various downstream applications. To our knowledge, it is the first time that a comprehensive representation model has been proposed for road networks based on hierarchical graph neural networks.

3 PRELIMINARIES

In this section, we introduce the used notations throughout the paper and formally define our task.

Definition 1. ***Road Segment****. A road segment $s \in \mathcal{S}$ is a uniform section of road that is identified separately in transportation [26], and* *it is usually associated with some side features (*e.g.*, longitude and latitude, segment type, and length).*

Definition 2. ***Road Network****. A road network is characterized as a directed graph* $\mathcal{G} = \langle \mathcal{S}, \boldsymbol{A}_S \rangle$*, where* $\mathcal{S}$ *is a vertex set of* k_S *road segments and* $\boldsymbol{A}_S \in \mathbb{R}^{k_S \times k_S}$ *is the adjacency matrix. Each entry* $\boldsymbol{A}_S[s_i, s_j]$ *is a binary value indicating whether there exists a directed link from road segment* s_i *to road segment* s_j*.*

Here, we follow the widely adopted setting [5, 11, 24] by considering road segments as vertices. It will be equally feasible to define locations (*e.g.,* POI or location cell) as vertices. For bidirectional road segments, we simply add two directed links by inverting their start and end vertices. As motivated in Section 1, our aim is to incorporate hierarchical structure to better organize vertices on road network. We would like to capture and learn a three-level hierarchy, namely *functional zones* $\longrightarrow$ *structural regions* $\longrightarrow$ *road segment*. Next, we define the two new concepts.

Definition 3. ***Structural Region****. A structural region* $r \in \mathcal{R}$ *is composed of a set of spatially connected road segments [40], serving as some traffic role,* e.g., *overpass and crossing.*

Definition 4. ***Functional Zone****. A functional zone* $z \in \mathcal{Z}$ *consists of multiple structural regions, providing some kind of traffic functionality [36, 40],* e.g., *shopping areas and transportation hub.*

We assume that there are k_R structural regions and k_Z functional zones, denoted by region set $\mathcal{R}$ and zone set $\mathcal{Z}$, respectively. Throughout the paper, we use the lower-case alphabet s, r and z to denote a road segment, structural region and a functional zone, respectively, and their upper-cases S, R and Z indicate the index types for aggregated data. For convenience, we might call *segment*, *region* and *zone* for short in unambiguous cases. We further utilize such a hierarchical structure to organize road network.

Definition 5. ***Hierarchical Road Network****. A hierarchical road network is formally described as* $\mathcal{H} = \langle \mathcal{V}, \mathcal{E} \rangle$*, where* $\mathcal{V} = \mathcal{S} \cup \mathcal{R} \cup \mathcal{Z}$ *consisting of road segments, structural regions and functional zones, and* $\mathcal{E} = \{\boldsymbol{A}_S, \boldsymbol{A}_R, \boldsymbol{A}_Z, \boldsymbol{A}^{SR}, \boldsymbol{A}^{RZ}\}$*, where the five matrices* $\boldsymbol{A}_S \in \mathbb{R}^{k_S \times k_S}$*,* $\boldsymbol{A}_R \in \mathbb{R}^{k_R \times k_R}$*,* $\boldsymbol{A}_Z \in \mathbb{R}^{k_Z \times k_Z}$*,* $\boldsymbol{A}^{SR} \in \mathbb{R}^{k_S \times k_R}$ *and* $\boldsymbol{A}^{RZ} \in \mathbb{R}^{k_R \times k_Z}$ *denote (weighted or binary) adjacency matrices for capturing the links between (1) two segment nodes, (2) two regions nodes, (3) two zone nodes, (4) a segment node and a region node, and (5) a region node and zone node, respectively.*

Different from road segments, structural regions and functional zones are virtual nodes. Therefore, $\boldsymbol{A}_R$, $\boldsymbol{A}_Z$, $\boldsymbol{A}^{SR}$ and $\boldsymbol{A}^{RZ}$ are unknown parameters to learn. We also call $\boldsymbol{A}^{SR}$ and $\boldsymbol{A}^{RZ}$ *segment-to-region* and *region-to-zone* assignment matrices, respectively, which are used to associate segments with regions or associate regions with zones. We present an illustrative example for the hierarchical road network in Fig. 1. Now, we are ready to define our task.

Definition 6. ***Representation Learning on Road Networks****. Given a road network* $\mathcal{G}$*, we aim to construct the corresponding hierarchical road network* $\mathcal{H}$ *and meanwhile derive a* d*-dimensional representation* $\boldsymbol{n}_m \in \mathbb{R}^d$ *for each vertex on* $\mathcal{H}$*, where* $d \ll |\mathcal{V}|$ *and* m *is a placeholder for a vertex from* $\mathcal{V}$*.*

For the three kinds of nodes, we can aggregate their embeddings in a matrix form, namely $\boldsymbol{N}_S \in \mathbb{R}^{k_S \times d}$, and $\boldsymbol{N}_R \in \mathbb{R}^{k_R \times d}$ and $\boldsymbol{N}_Z \in \mathbb{R}^{k_Z \times d}$, which denote the segment, region and zone embedding matrices, respectively. Our task becomes how to form the virtual region and zone nodes and learn the embedding matrices ($\boldsymbol{N}_S$, $\boldsymbol{N}_R$, $\boldsymbol{N}_Z$) and the adjacency matrices ($\boldsymbol{A}_R$, $\boldsymbol{A}_Z$, $\boldsymbol{A}^{SR}$, $\boldsymbol{A}^{RZ}$).

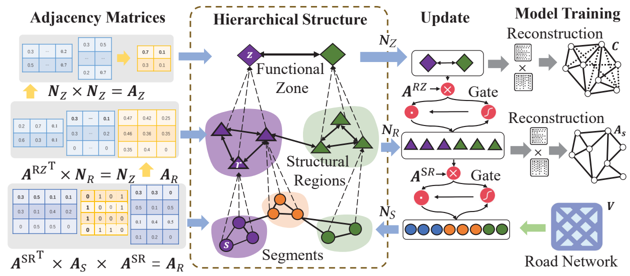


Figure 1: The overall architecture of the HRNR model.

4 MODEL

In this section, we present the proposed *Hierarchical Road Network Representation (HRNR)* model. Our core idea is to extend the graph neural network for road network representation learning by characterizing a three-level hierarchy. The overall architecture for the proposed model is presented in Fig. 1. We start with contextual embedding for location segments, then present how to model structural regions and functional zones, and finally discuss how to update the hierarchical model and train the entire network.

4.1 Contextual Embedding for Road Segments

As we introduced in Section 3, a road segment is associated with a set of useful context features. Here, we embed these side information and learn the contextual embeddings for road segments.

Different from the node representations using link information (*i.e.,* $\boldsymbol{n}_{l_i}$), we use $\boldsymbol{v}_{l_i} \in \mathbb{R}^d$ to denote the contextual embedding using the side features from itself. Given a road segment s_i, we consider five kinds of features for contextual embedding, namely road segment ID, road type (RT), lane number (LN), segment length (SL), and longitude and latitude (LL). For continuous features, we divide the entire value range into several consecutive bins, and utilize the bin number for feature coding. In this way, we set a unique embedding vector for each discrete value (or bin number), and then concatenate the associated vectors as the contextual embedding:

$$\boldsymbol{v}_{s_i} = \boldsymbol{v}_{ID} \| \boldsymbol{v}_{RT} \| \boldsymbol{v}_{LN} \| \boldsymbol{v}_{SL} \| \boldsymbol{v}_{LL}, \tag{1}$$

where "$\|$" is the vector concatenation operation and $\boldsymbol{v}_{(\cdot)}$ denotes the embedding vector for some kind of context feature.

Such a simple approach is flexible to include more side features. We adopt it for initializing the graph node embeddings:

$$\boldsymbol{N}_S^{(0)} \leftarrow \boldsymbol{V}, \tag{2}$$

where $\boldsymbol{V}$ is the aggregate matrix for the contextual embeddings of all the road segments.

4.2 Modeling Structural Regions

In our model, structural regions are mainly used to characterize the local connected patterns for some traffic purpose. We assume a road segment belongs to one single region, and different road segments correspond to different importance levels in a region. Next, we introduce how to model structural regions.

4.2.1 Constructing Structural Regions by Spectral Clustering. We adopt the classic spectral clustering algorithm [16] for deriving structural regions. It takes a graph cut view by splitting weak links, so that the yielded clusters achieve a more closely connected status. Such a clustering algorithm is particularly suitable for our task, since we aim to look for closely connected road segments. Formally, given the adjacency matrix $\boldsymbol{A}_S$ for road segments, we first derive its graph laplacian $\boldsymbol{L}_S$ by subtracting the diagonal matrix $\boldsymbol{D}_S$, so we have $\boldsymbol{L}_S = \boldsymbol{D}_S - \boldsymbol{A}_S$. By computing the first d' eigenvectors $\boldsymbol{u}_1, ..., \boldsymbol{u}_k$ of laplacian matrix $\boldsymbol{L}_S$, we obtain the matrix $\boldsymbol{U} \in \mathbb{R}^{k_S \times d'}$ consisting of the d' eigenvectors. By running standard K-means algorithm over the matrix $\boldsymbol{U}$, we can obtain a hard mapping from locations to clusters (*i.e.,* structural regions). We incorporate an membership matrix $\boldsymbol{M}_1 \in \mathbb{R}^{k_S \times k_R}$, where each entry is defined as

$$\boldsymbol{M}_1[l, r] = \begin{cases} 1 & l \in r, \\ 0 & other. \end{cases} \tag{3}$$

4.2.2 Learning Region Representations with Assignment Matrix. Since different road segments in a cluster are not equally important, we adopt the Graph Attention Network (GAT) [19] to model segment importance scores as follows:

$$\boldsymbol{W}_1 = \mathrm{GAT}\left(\boldsymbol{V}, \boldsymbol{A}_S\right), \tag{4}$$

where $\mathrm{GAT}(\cdot,\cdot)$ is a standard implementation of [19] detailed in supplementary documents, $\boldsymbol{V}$ is the contextual embedding matrix (Eq. 1), $\boldsymbol{A}_S$ is the adjacency matrix for road segments, and $\boldsymbol{W}_1 \in \mathbb{R}^{k_S \times k_R}$ is the learned output through GAT. Here, we set the column number of $\boldsymbol{W}_1$ to k_R (*i.e.,* the number of structural regions), and associate each latent dimension with a unique structural region. A column vector in $\boldsymbol{W}_1$ measures the importance levels of road segments *w.r.t.* some region. Since we have previously obtained a hard location-region mapping matrix $\boldsymbol{M}_1$ (Eq. 3), we further multiply the two matrices and derive the soft assignment of road segments in a structural region as:

$$\boldsymbol{A}^{SR} = \mathrm{softmax}\left(\boldsymbol{M}_1 \odot \boldsymbol{W}_1\right), \tag{5}$$

where "$\odot$" denotes the matrix-based element-wise product, and $\mathrm{softmax}(\cdot)$ is the standard softmax function for column normalization. Each entry $\boldsymbol{A}^{SR}[s, r]$ indeed models the conditional probability of a road segment s in a structural region r:

$$\boldsymbol{A}^{SR}[s, r] = \Pr(s|r). \tag{6}$$

We further utilize $\boldsymbol{A}^{SR}$ to associate region representations with segment representations:

$$\boldsymbol{N}_R = \boldsymbol{A}^{SR^\top} \boldsymbol{N}_S, \tag{7}$$

where $\boldsymbol{N}_S \in \mathbb{R}^{k_S \times d}$ is the representation matrix of road segments, and $\boldsymbol{N}_R \in \mathbb{R}^{k_R \times d}$ is the representation matrix of structural regions. It can be seen that $\boldsymbol{n}_r = \sum_{s \in r} \Pr(s|r)\boldsymbol{n}_s$. We can further obtain a weighted adjacency matrix $\boldsymbol{A}_R \in \mathbb{R}^{k_R \times k_R}$ for region nodes:

$$\boldsymbol{A}_R = \boldsymbol{A}^{SR^\top} \cdot \boldsymbol{A}_S \cdot \boldsymbol{A}^{SR}. \tag{8}$$

Such a formula can be explained as:

$$\boldsymbol{A}_R[r_i, r_j] = \sum_{s, s' \in \mathcal{S}} \Pr(s|r_i)\Pr(s'|r_j)\boldsymbol{A}_S[s, s']. \tag{9}$$

4.2.3 Learning the Assignment Matrix by Network Reconstruction. The assignment matrix $\boldsymbol{A}^{SR}$ plays the key role in associating segment representations $\boldsymbol{N}_S$ with region representations $\boldsymbol{N}_R$. Such a way is similar to hierarchical pooling technique with assignment matrix in DP-GCN [35]. However, it is difficult to directly learn $\boldsymbol{A}^{SR}$ without a suitable supervision signal for our task, since road network has its own unique features. Above, we have adopted spectral clustering to pre-construct region nodes. Here, we further present an enhanced learning method based on network reconstruction. Our core idea is to utilize region representations to fit segment representations based on assignment matrix, and reconstruct the road network with the approximated segment representations. Formally, we obtain the fitted segment representations $\boldsymbol{N}_S$ as follows:

$$\hat{\boldsymbol{N}}_S = \boldsymbol{A}^{SR} \boldsymbol{N}_R. \tag{10}$$

We can rewrite the above equation in a vector form: $\boldsymbol{n}_s = \Pr(s|r_s)\boldsymbol{n}_r$, where r_s is the assigned region for segment s. Furthermore, $\hat{\boldsymbol{N}}_S$ is utilized to reconstruct the original adjacency matrix $\boldsymbol{A}_S$:

$$\hat{\boldsymbol{A}}_S = \mathrm{sigmoid}(\hat{\boldsymbol{N}}_S \hat{\boldsymbol{N}}_S^\top), \tag{11}$$

where sigmoid function applies to each matrix element for transforming the value into the interval $(0, 1)$. Furthermore, the cross entropy function is employed to compute the reconstruction loss:

$$\begin{aligned} Loss_1 = & \sum_{s_i, s_j \in \mathcal{S}} -\boldsymbol{A}_S[s_i, s_j] \log(\hat{\boldsymbol{A}}_S[s_i, s_j]) \\ & -(1 - \boldsymbol{A}_S[s_i, s_j]) \log(1 - \hat{\boldsymbol{A}}_S[s_i, s_j]). \end{aligned} \tag{12}$$

A major merit with such network reconstruction is that it forces both $\boldsymbol{A}^{SR}$ and $\boldsymbol{N}_R$ to learn effective characteristics from original road network structure and enhances the association between regions and segments.

4.3 Modeling Functional Zones

In this part, we study how to model functional zones. A functional zone is constructed on top of functionally related structural regions. It aims to capture important functional characteristics, even for disconnected or distant regions.

4.3.1 Learning Zone Representations with Assignment Matrix. We adopt a similar strategy in Section 4.2.2 (Eq. 7) to learn function zone representations using a linear combination of region representations. Given the region-to-zone assignment matrix $\boldsymbol{A}^{RZ} \in \mathbb{R}^{k_R \times k_Z}$, each entry $\boldsymbol{A}^{RZ}[r, z]$ denotes the conditional probability of region r in a zone z. By aligning a latent dimension with a zone, we utilize the GAT network to derive $\boldsymbol{A}^{RZ}$:

$$\boldsymbol{A}^{RZ} = \mathrm{softmax}(\boldsymbol{M}_2), \tag{13}$$

$$\boldsymbol{M}_2 = \mathrm{GAT}\left(\boldsymbol{N}_R, \boldsymbol{A}_R\right), \tag{14}$$

where $\boldsymbol{M}_2 \in \mathbb{R}^{k_R \times k_Z}$ represents the region-to-zone mapping and we perform the softmax function by columns to derive the region-to-zone probability matrix $\boldsymbol{A}^{RZ}$. In this way, we can set zone representations as the linear combination of region representations:

$$\boldsymbol{N}_Z = \boldsymbol{A}^{RZ^\top} \boldsymbol{N}_R. \tag{15}$$

With $\boldsymbol{N}_Z$, we further derive the adjacency matrix for zone nodes:

$$\boldsymbol{A_Z} = \text{RELU}\left(\boldsymbol{N}_Z \boldsymbol{N}_Z^\top - \sigma\right), \tag{16}$$

where $\boldsymbol{A_Z}$ is the computed weighted adjacency matrix for zone nodes and σ is a scaling parameter set to 0.5.

4.3.2 Capturing Functional Characteristics with Trajectory Data. Road network itself mainly reflects the structural characteristics, containing very limited functional information. Therefore, we consider using real trajectory data for capturing functional characteristics. Previous studies on trajectory data mining [36, 40] have shown that trajectory behaviors can correspond to important functional patterns of underlying road network units. We collect trajectory sequence data of real users, which is a time-ordered road segment sequence visited by a user. In order to utilize the trajectory data, we construct a road segment transition matrix $\boldsymbol{T}^{(\lambda)} \in \mathbb{R}^{k_S \times k_S}$, in which entry $\boldsymbol{T}^{(\lambda)}[s_i, s_j]$ indicates the frequency that s_i has reached s_j with a step length λ in all trajectory sequences. With $\boldsymbol{T}^{(\lambda)}$, we can obtain a updated connectivity matrix $\boldsymbol{C} \in \mathbb{R}^{k_S \times k_S}$:

$$\boldsymbol{C} = \boldsymbol{A}_S + \sum_{j=1}^{\lambda} \boldsymbol{T}^{(j)}, \tag{17}$$

where $\boldsymbol{C}$ considers the connectivity in terms of both road network structure and human moving behaviors, and λ is a tunable parameter set to 5 in this work. Furthermore, we perform row-based normalization on $\boldsymbol{C}$. Then, we follow the similar way in Eq. 10 to fit segment representations based on zone representations:

$$\hat{\boldsymbol{N}}_S = \boldsymbol{A}^{SR} \boldsymbol{A}^{RZ} \boldsymbol{N}_Z, \tag{18}$$

where $\boldsymbol{A}^{SR}$ and $\boldsymbol{A}^{RZ}$ are the segment-to-region or region-to-zone assignment matrices, respectively. This can be explained in a vector form: $\boldsymbol{n}_s = \Pr(s|r_s) \sum_{z \in \mathcal{Z}} \Pr(r_s|z) \boldsymbol{n}_z$, which is a two-step fit process. Similar to Eq. 11, we reconstruct matrix $\boldsymbol{C}$ as:

$$\hat{\boldsymbol{C}} = \hat{\boldsymbol{N}}_S \hat{\boldsymbol{N}}_S^\top. \tag{19}$$

Instead of using sigmoid function on $\hat{\boldsymbol{C}}$ for deriving probabilities, we keep the original reachable degree from trajectory data. We use Mean Square Error (MSE) to measure the difference between the real and reconstructed matrices:

$$Loss_2 = \|\boldsymbol{C} - \hat{\boldsymbol{C}}\|^2, \tag{20}$$

where $\hat{\boldsymbol{C}}$ is the estimated connectivity matrix in Eq. 19.

4.4 Hierarchical Update Mechanism

Above, we have discussed how to learn the two assignment matrices that associate regions with zones or associate locations with regions. Next, we assume the two matrices are fixed and discuss how to update the node representations by designing a hierarchical update mechanism by levels.

4.4.1 Zone-level Update. We first perform the zone-level update. At this level, we update zone representations and prepare them for message passing to the next level. We adopt a standard Graph Convolutional Network (GCN) [7] to update the zone embeddings:

$$\boldsymbol{N}_Z^{(t+1)} = \text{GCN}\left(\boldsymbol{N}_Z^{(t)}, \boldsymbol{A_Z}\right), \tag{21}$$

where $\boldsymbol{A_Z}$ is the computed weighted adjacency matrix for zone nodes in Eq. 16. Since the $\boldsymbol{A_Z}$ is not a binary matrix, we do not adopt GAT here, and GCN's details can be found in supplementary documents. Then, it sends the zone embeddings to the next level for updating region embeddings:

$$\tilde{\boldsymbol{N}_R}^{(t)} = \boldsymbol{N}_R^{(t)} + \boldsymbol{g}^{ZR} \odot \left(\boldsymbol{A}^{RZ} \boldsymbol{N}_Z^{(t+1)}\right), \tag{22}$$

$$\boldsymbol{g}^{ZR} = \text{sigmoid}\left(\left(\boldsymbol{N}_R^{(t)} \| (\boldsymbol{A}^{RZ} \boldsymbol{N}_Z^{(t+1)})\right) \cdot \boldsymbol{w}_1\right), \tag{23}$$

where $\boldsymbol{g}^{ZR}$ is a gate vector controlling the information passing from zones to regions, and $\boldsymbol{w}_1$ is a parameter vector to learn.

4.4.2 Region-level Update. At the region level, it first updates its own embedding representations by adopting standard GCN:

$$\boldsymbol{N}_R^{(t+1)} = \text{GCN}\left(\tilde{\boldsymbol{N}_R}^{(t)}, \boldsymbol{A}_R\right), \tag{24}$$

where $\boldsymbol{A}_R$ is the weighted adjacency matrix in Eq. 8. Then, we forward the region embeddings to the next level for updating the segment representations:

$$\tilde{\boldsymbol{N}}_S^{(t)} = \boldsymbol{N}_S^{(t)} + \boldsymbol{g}^{RS} \odot \left(\boldsymbol{A}^{SR} \boldsymbol{N}_R^{(t+1)}\right), \tag{25}$$

$$\boldsymbol{g}^{RS} = \text{sigmoid}\left(\left(\boldsymbol{N}_S^{(t)} \| (\boldsymbol{A}^{SR} \boldsymbol{N}_R^{(t+1)})\right) \cdot \boldsymbol{w}_2\right), \tag{26}$$

where $\boldsymbol{g}^{RS}$ is a gate vector controlling the information passing from regions to segments, and $\boldsymbol{w}_2$ is a parameter vector to learn.

4.4.3 Segment-level Update. Finally, we employ a Graph Attention Network [19] (GAT) to model the relation between segment nodes as follows

$$\boldsymbol{N}_S^{(t+1)} = \text{GAT}\left(\tilde{\boldsymbol{N}}_S^{(t)}, \boldsymbol{A}_S\right), \tag{27}$$

where $\boldsymbol{A}_S$ is the binary adjacency matrix for the segment nodes.

4.5 Learning and Discussion

In our model, various kinds of node embeddings ($\boldsymbol{N}_S$, $\boldsymbol{N}_R$, $\boldsymbol{N}_Z$), assignment matrices ($\boldsymbol{A}^{SR}$, $\boldsymbol{A}^{RZ}$) and involved component parameters are the model parameters. Note that each GAT or GCN components have corresponded to a unique parameter set.

At each iteration, we first learn the assignment matrices $\boldsymbol{A}^{SR}$ and $\boldsymbol{A}^{RZ}$. For this purpose, we optimize the loss in $Loss_1$ (Eq. 12) for learning $\boldsymbol{A}^{SR}$, and then jointly optimize $Loss_1$ (Eq. 12) and $Loss_2$ (Eq. 20) for learning $\boldsymbol{A}^{RZ}$. Then, the assignment matrices $\boldsymbol{A}^{SR}$ and $\boldsymbol{A}^{RZ}$ are provided to the hierarchical update algorithm. Finally, we apply the hierarchical update mechanism in Section 4.4 for learning node embeddings. We provide detailed description for the algorithm flow, time complexity and training method in the supplementary materials.

Once our model has been learned, we can apply the node embeddings to various downstream applications. Interestingly, it is straightforward to add new task-specific loss according to some downstream application. In this way, we can re-tune the parameters in order to yield the best performance. Compared with previous studies on graph representation learning [7, 18, 19] or road network representation [5, 25], we design a three-level hierarchical architecture for learning effective representations from three kinds of nodes, namely segments, regions and zones. Our model can learn both structural and functional characteristics by utilizing both road network structure and human moving trajectory data.

Table 1: Statistics of the three datasets after preprocessing.

Statistics	Beijing	Chengdu	Xi'an
#tpyes	17	13	12
#trajectories	302,654	224,184	493,254
#records	16,040,662	9,632,481	6,672,027
#edges	47,082	8,224	7,341
#road segments	15,500	3,157	2,910
#label	708	303	291
graph diameter	131	71	47
average hop number	48	35	28

5 EXPERIMENTS

In this section, we construct experiments to demonstrate the effectiveness of our model.

5.1 Experimental Setup

5.1.1 Construction of the Datasets. To measure the performance of our proposed model, we use three real-world road network datasets with corresponding trajectory data. For the three datasets, we collect corresponding road network information from open street map [1]. The *Beijing* trajectory data is sampled every minute, while the *Chengdu* dataset and *Xi'an* dataset is sampled every 2-4 seconds. The *Xi'an* and *Chengdu* dataset are originally released in *GAIA Open Dataset* [2]. We further perform map matching [34] by aligning GPS points with locations in the road network. In this way, we transform the trajectory data into road segment sequences. With the boundary indicators provided by the three datasets, we split the location sequence into multiple trajectories. Table 1 lists statistics of the three datasets after preprocessing. We can see that the three road networks have a long graph diameter. Especially, the average hop number between segments is also significantly large.

5.1.2 Methods to Compare. We consider the following methods for comparison:

•*MDW* [3]: metapath2vec extends DeepWalk by constructing meta-path-guided paths. In our dataset, each road segment is associated with a road type. We manually create type-based meta-paths to guide the path generation in DeepWalk.

•*IRN2Vec* [25]: IRN2Vec is special road network model developed using shallow node representations, and which explores geo-locality and moving behaviors of road users. It defines and optimizes three parts of loss, namely location, type and tag.

•*GAT* [19]: It is a standard implementation of graph attention network for road network. Here, we adopt the same contextual embeddings (*i.e.,* $\boldsymbol{v}_s$) for road segments. Another similar baseline is GCN [7]. We omit it since the two methods have similar results.

•*Geo-GCN* [18]: Geo-GCN extends GCN by using a new geometrical aggregation scheme to solve the long dependency problem. We adapt it to road networks by aggregating the spatially closely road segments as neighbors.

•*DP-GCN* [35]: DP-GCN is a differentiable graph pooling model that can generate hierarchical representations of graphs. It adopts a hierarchical pooling way to construct the hierarchy. In our experiments, it is set to contain three pooling levels. Different from our method, it does not incorporate additional loss to supervise the learning of assignment matrices.

5.1.3 Application Tasks. We consider using four traffic-related applications for testing the effectiveness of the above comparison methods. For each application task, we construct a simple yet standard neural network architecture (*e.g.,* GRU or MLP) as the basic framework. Then, we incorporate the learned road network representations (mainly road segments) as embeddings to enhance the basic framework. Note that we do not construct very complicated neural architectures or adopt more data signals. Our focus is to learn generally useful road network representations and reduce the influence of other factors. Except MDW and IRN2Vec, the other comparison methods can be jointly optimized with the application tasks. The four application tasks are described as follows:

Next-Location Prediction. Next-location prediction aims to predict the next location to visit for a user [29]. A classic solution is to construct a GRU-based model, taking as input the historical trajectory and outputting a ranked list of candidate location(s). Here, we consider a road segment as a location. As a major motivation, we aim to capture long-range dependencies among locations. Therefore, we adopt a large down-sampling interval of ten minutes on the original trajectory data. A good method should rank the actual location at a high position in the candidate list.

Label Classification. Label classification is a standard task to test the performance of representation learning models [25]. Our dataset (Table 1) contains the labels for the road segments, such as *birdges* and *tunnel*. We develop a predictor based on the logistic regression model, taking as input the road segment representations and generating a label distribution. We adopt the label with the largest predictive probability as the final prediction.

Destination Prediction. Destination prediction [33] aims to predict the destination based on a partial trajectory. This task is useful to map navigation, POI recommendation, etc. Similar to next-location prediction, we construct a GRU based predictor and take as input the learned representations. While it is trained by optimizing the model using the destination as the ground-truth. We take the last location of a trajectory sequence as the destination.

Route Planning. Route plan aims to generate the actual route that connects source location with destination location [28]. It is more difficult than next-location or destination prediction. We construct a hierarchical predictor, which first encodes the seen trajectory with a GRU component and then predicts the next location with a MLP component. Given a trajectory sequence, the first and last locations are considered as source and destination, respectively, while the rest locations are hidden for prediction.

5.1.4 Evaluation Metrics. We adopt different evaluation metrics for the above four tasks. For next-location and destination prediction, we treat them as a ranking task and adopt top-1 and top-5 prediction accuracies as metrics, denoted by *ACC@1* and *ACC@5*. For label classification, we adopt *F1-score* and *AUC*: *F1-score* considers both precision and recall of binary classification, and *AUC* computes

[1]https://www.openstreetmap.org/

[2]https://outreach.didichuxing.com/appEn-vue/dataList

Table 2: Performance comparison for four tasks on three datasets. All the results are better with larger values except the EDT measure. Here, "BJ"=Beijing, "CD"=chengdu, and "XA"=Xi'an.

	Tasks	Next Location Prediction						Tasks	Label Classification					
Set		MDW	IRN2vec	GAT	Geo-GCN	DP-GCN	HRNR	Metric	MDW	IRN2vec	GAT	Geo-GCN	DP-GCN	HRNR
BJ	ACC@1	0.357	0.362	0.380	0.387	0.388	**0.413**	F1	0.728	0.732	0.770	0.775	0.772	**0.829**
	ACC@5	0.482	0.491	0.514	0.521	0.522	**0.551**	AUC	0.810	0.804	0.841	0.845	0.844	**0.888**
CD	ACC@1	0.370	0.368	0.385	0.396	0.396	**0.422**	F1	0.689	0.687	0.701	0.713	0.703	**0.748**
	ACC@5	0.503	0.496	0.534	0.540	0.541	**0.567**	AUC	0.692	0.690	0.722	0.739	0.733	**0.773**
XA	ACC@1	0.315	0.317	0.333	0.342	0.340	**0.372**	F1	0.619	0.622	0.636	0.643	0.637	**0.685**
	ACC@5	0.449	0.452	0.463	0.471	0.469	**0.503**	AUC	0.624	0.631	0.657	0.670	0.662	**0.716**
	Tasks	Destination Predition						Tasks	Route Planning					
Set		MDW	IRN2vec	GAT	Geo-GCN	DP-GCN	HRNR	Metric	MDW	IRN2vec	GAT	Geo-GCN	DP-GCN	HRNR
BJ	ACC@1	0.215	0.218	0.233	0.240	0.241	**0.273**	F1	0.269	0.274	0.298	0.300	0.305	**0.329**
	ACC@5	0.313	0.316	0.347	0.350	0.357	**0.396**	EDT	8.742	8.851	8.235	8.151	8.132	**7.851**
CD	ACC@1	0.239	0.235	0.256	0.267	0.263	**0.288**	F1	0.310	0.312	0.330	0.338	0.341	**0.357**
	ACC@5	0.343	0.346	0.375	0.394	0.389	**0.413**	EDT	8.142	8.013	7.869	7.731	7.664	**7.361**
XA	ACC@1	0.201	0.202	0.210	0.222	0.225	**0.251**	F1	0.259	0.254	0.271	0.278	0.282	**0.301**
	ACC@5	0.305	0.304	0.333	0.348	0.351	**0.370**	EDT	9.268	9.163	8.873	8.653	8.532	**8.138**

the area under the ROC curve. For route planning, given an actual route p, we predict a possible route p' with the same source and destination. Following [1, 13], we use *F1-score* as evaluation metrics: $Precision = \frac{|p \cap p'|}{|p'|}$, $Recall = \frac{|p \cap p'|}{|p|}$ and $F1 = \frac{2*P*R}{P+R}$. *F1-socre* measures the degree of overlapping locations *w.r.t.* the actual and predicted routes respectively. Besides, we use the *Edit distance* as a second measure [9], which is the minimum number of edit operations required to transform the predicted route into the actual route. For the four tasks, we divide all data into three parts with a ratio of 7 : 1 : 2, namely training set, validation set and test set. We train the model with training set, tune the parameters with validation set, and then report the performance on the test.

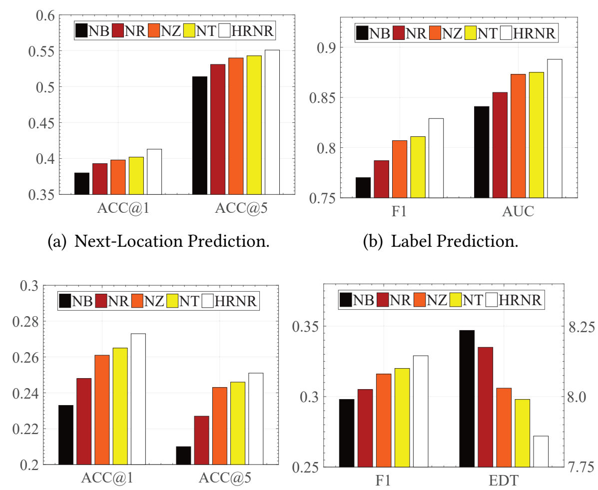
5.2 Results and Analysis

Table 2 presents the results of all the comparison methods.

First, network embedding based methods MDW and IRN2Vec perform worst among all the baselines. A possible reason is that they are not flexible to characterize rich context information. Besides, the two models cannot be jointly optimized with downstream applications. Comparing MDW and IRN2Vec, IRN2Vec gives a better performance. MDW only simply utilizes the type information via metapaths, while IRN2Vec incorporates geographical constraints into the random walk process and adopts a multi-task learning mechanism to capture more correlations among road segments.

Second, the three graph neural network variants (*i.e.,* GAT, Geo-GCN and DP-GCN) perform better than the MDW and IRN2Vec. A major merit of graph neural networks is that they are able to model network attribute information and characterize the node relation with deep neural networks. However, it is not suitable to capture either hierarchical characteristics or long-range dependency in road network. While, Geo-GCN and DP-GCN have made extensions to improve these issues, yielding a better performance. Geo-GCN mainly characterizes spatial information, and DP-GCN adopts a hierarchical pooling mechanism.

Finally, the proposed model HRNR is consistently better than all the baselines with a large margin in all cases. We carefully design



(a) Next-Location Prediction. (b) Label Prediction.

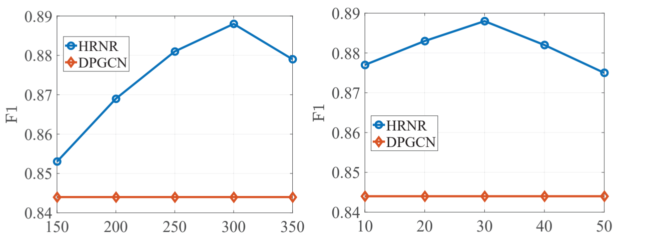
(c) Destination Prediction. (d) Route Planning.

Figure 2: Ablation study of our model on Beijing taxi dataset for four tasks.

a three-level hierarchy for organizing road network information. We devise effective reconstruction loss to associate components at different levels, and adopt a hierarchical update mechanism. Our model is able to explicitly learn both structural and functional characteristics using network structure and trajectory data. By modeling such a hierarchy in message passing, our model is more capable of learning long-range dependency between road segments. Compared with DP-GCN, our hierarchical structure is more interpretable and capture real-world "clusters".

5.3 Ablation Study

In our model, we have incorporated two additional kinds of nodes, namely structural regions and functional zones, respectively. Here, we would like to check how each part actually contributes to the final performance. We construct the ablation study experiment on the *Beijing taxi* dataset. The findings on the other datasets



(a) The # of structural regions (k_R). (b) The # of of functional zones (k_Z).

Figure 3: Parameter sensitivity on label classification on Beijing taxi dataset using F1 measure.

are similar and omitted for space limit. We report the result of F1 scores for label classification and route plan, and report the result of ACC@1 for next-location and destination prediction. We prepare five variants of the proposed HRNR model for comparisons, including (1) NB without either structural regions or functional zones, (2) NZ without functional zones (the reconstruction loss in Eq. 20 is also removed), (3)NR without structural regions and , (4) NT without using trajectory data (containing structural regions and functional zones), and (5) HRNR that is the complete model.

Figure 2 presents all the comparison results of the four variants. As we can see, the performance rank can be given as follows: $NB < NR < NZ < NT < HRNR$. These results indicate that the two parts are essential to improve the performance of our model. Besides, it seems that structural regions are more useful than functional zones. A possible reason is that it is a lower level and forms basic clusters of road segments for modeling functional zones. The basic variant NB removes both levels and performs worst among all the variants, which degenerates to a similar architecture of GAT. A final note is that the difference between *NZ* and *NT* is very small. This indicates that it might be less effective to simply incorporate more virtual cluster nodes. Instead, a suitable supervision signal (recall that we have designed the reconstruction loss in Eq. 20 using trajectory data) is more important to model road networks.

5.4 Parameter Sensitivity

In addition to the model components, there are several parameters to tune in our model. Here we incorporate the best baseline DP-GCN for comparison. We report the tuning results on label classification on the Beijing dataset.

We tune the number of structural regions and functional zones, respectively. We vary the number of regions k_R in the set $\{150, 200, 250, 300, 350\}$, and the number of zones k_Z in the set $\{10, 20, 30, 40, 50\}$. In Fig. 3(a) and Fig. 3(b), we can see that using 300 structural regions and using 30 functional zones are the optimal settings. Since functional zones are composed of structural regions, it is reasonable to have more regions than zones.

Overall, our model is relatively stable when varying the four parameters, consistently better than DP-GCN and GAT.

5.5 Qualitative Analysis

Previously, we have shown the effectiveness of our model on four tasks. In this part, we qualitatively analyze how the learned representations are useful in traffic-related applications.

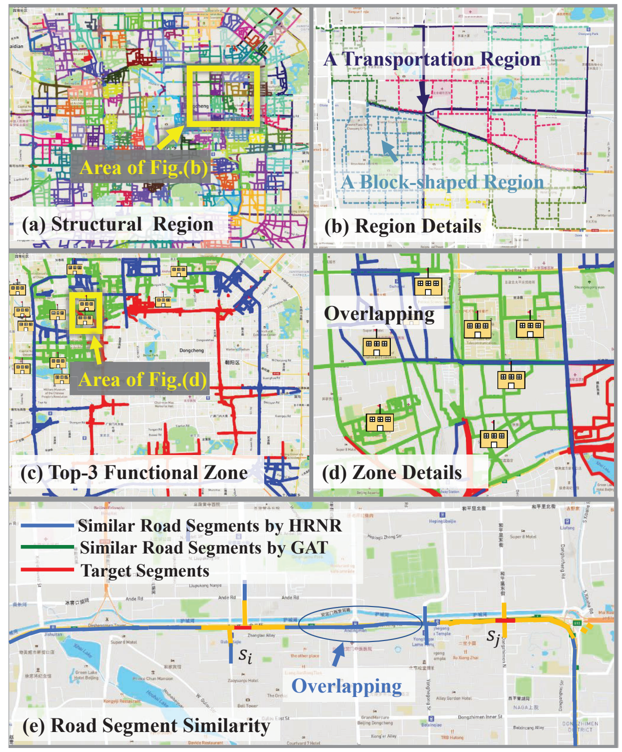


Figure 4: Visualization of the learned representations for road segments, structural regions and functional zones. The colored lines denote road segments in the road network. Road segments with the same color correspond to one structural region or one functional zone.

In our model, we incorporate two kinds of virtual nodes, namely structural regions and functional zones. Now, we examine whether they actually capture structural or functional characteristics in real world. In Fig. 4(a), we present all the identified structural regions in the Beijing dataset. Each color corresponds to a unique structural region. By zooming into a selected part of the entire road network, we present its enlarged view in Fig. 4(b). There are eight regions in total, corresponding to different colors. Interestingly, not all the regions are in a block shape. One can see that the region in dark violet indeed undertakes transportation function that connects other block-shaped regions.

Similarly, we present three sample functional zones (marked in different colors) and the magnified view of a functional zone in Fig. 4(c) and Fig. 4(d), respectively. The two ring-shaped zones correspond to the Beijing 2nd and 3rd ring roads, while the third functional zone in green is *educational* zone. By zooming into it, we can found that it contains many schools (marked with the house icon). These examples have shown that our generated regions and zones are indeed meaningful in real world, capturing structural or functional characteristics for the city.

Finally, we examine the representations for road segments. Figure 4(e) presents an example with two similar road segments (with the same type and labels) on a same main road, denoted by s_i and s_j. We select GAT [19] as a reference method. For both GAT and our model, we can compute the attention coefficient between any two road segments. Given s_i and s_j, we only highlight the neighbors with a large attention value (*i.e.,* lager than 0.8). We use different colors to discriminate the neighbors identified by the two methods. It is clear to see that GAT mainly focuses on very close neighbors in spatial position, while our model indeed captures influencing road segments in a long range. For s_i and s_j, the identified common neighbors mainly fall on the main road itself, which drive s_i and s_j to have similar representations in our model.

6 CONCLUSIONS

In this paper, we studied how to effectively represent road networks for general-purpose use in intelligent transportation systems. We proposed a hierarchical graph neural network by characterizing the hierarchy "*functional zones*" → "*structural regions*" → "*road segments*". We carefully devised two useful reconstruction loss functions to capture both structural and functional characteristics. A hierarchical update mechanism was also given tailored to our network architecture. Extensive experiment results on three real-world datasets for four tasks demonstrated the effectiveness and robustness of the proposed model.

Typically, road network is likely to change with time. As future work, we will consider extending our model to learn time-varying representations. Currently, we utilize trajectory data as supervision signal for network reconstruction. We will investigate how to explicitly incorporate trajectory data in the representation model.

ACKNOWLEDGMENTS

The work was supported by the National Key Research & Development Program of China (Grant No. 2019YFB2102100), and the National Natural Science Foundation of China (Grant No. 71531001 and 61872369). Wang's work was partially supported by the Fundamental Research Funds for the Central Universities (Grant No. YWF-20-BJ-J-839) and CCF-DiDi Gaia Collaborative Research Funds for Young Scholars. Zhao's work was partially supported by Beijing Academy of Artificial Intelligence (BAAI) (Grant No. BAAI2020ZJ0301), the Research Funds of Renmin University of China (Grant No. 18XNLG22) and the Fundamental Research Funds for the Central Universities (Grant No.19XNQ047).

REFERENCES

[1] Ge Cui, Jun Luo, and Xin Wang. 2018. Personalized travel route recommendation using collaborative filtering based on GPS trajectories. *IJED* 11, 3 (2018), 284–307.
[2] J. Dai, B. Yang, C. Guo, and Z. Ding. 2015. Personalized route recommendation using big trajectory data. In *ICDE*. 543–554.
[3] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *SIGKDD*.
[4] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *WWW*. 1459–1468.
[5] Tobias Skovgaard Jepsen, Christian S Jensen, and Thomas Dyhre Nielsen. 2019. Graph convolutional networks for road networks. In *SIGSPATIAL*. 460–463.
[6] Evangelos Kanoulas, Yang Du, Tian Xia, and Donghui Zhang. 2006. Finding fastest paths on a road network with speed patterns. In *ICDE*. IEEE, 10–10.
[7] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *ICLR* (2017).
[8] Vlaho Kostov, Jun Ozawa, Mototaka Yoshioka, and Takahiro Kudoh. 2005. Travel destination prediction using frequent crossing pattern from driving history. In *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.* IEEE, 343–350.
[9] Takeshi Kurashima, Tomoharu Iwata, Go Irie, and Ko Fujimura. 2010. Travel route recommendation using geotags in photo sharing sites. In *CIKM*. 579–588.
[10] Xiucheng Li, Gao Cong, Aixin Sun, and Yun Cheng. 2019. Learning travel time distributions with deep generative model. In *WWW*. 1017–1027.
[11] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. 2018. Multi-task representation learning for travel time estimation. In *SIGKDD*.
[12] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *ICLR* (2017).
[13] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2015. Personalized Tour Recommendation Based on User Interests and Points of Interest Visit Durations.. In *IJCAI*, Vol. 15. 1778–1784.
[14] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: a recurrent model with spatial and temporal contexts. In *AAAI*.
[15] Paul Newson and John Krumm. 2009. Hidden Markov map matching through noise and sparseness. In *SIGSPATIAL*. 336–343.
[16] Andrew Y Ng, Michael I Jordan, and Yair Weiss. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*. 849–856.
[17] Takayuki Osogami and Rudy Raymond. 2013. Map matching with inverse reinforcement learning. In *IJCAI*.
[18] Hongbin Pei, Bingzhe Wei, Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *ICLR*. 4800–4810.
[19] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *ICLR* 1, 2 (2017).
[20] Jingyuan Wang, Chao Chen, Junjie Wu, and Zhang Xiong. 2017. No longer sleeping with a bomb: a duet system for protecting urban safety from dangerous goods. In *SIGKDD*. 1673–1681.
[21] Jingyuan Wang, Xu He, Ze Wang, Junjie Wu, Nicholas Jing Yuan, Xing Xie, and Zhang Xiong. 2018. CD-CNN: a partially supervised cross-domain deep learning model for urban resident recognition. In *AAAI*.
[22] Jingyuan Wang, Xiaojian Wang, and Junjie Wu. 2018. Inferring metapopulation propagation network for intra-city epidemic control and prevention. In *SIGKDD*.
[23] Jingyuan Wang, Junjie Wu, Ze Wang, Fei Gao, and Zhang Xiong. 2019. Understanding urban dynamics via context-aware tensor factorization with neighboring regularization. *IEEE Transactions on Knowledge and Data Engineering* (2019).
[24] Jingyuan Wang, Ning Wu, Wayne Xin Zhao, Fanzhang Peng, and Xin Lin. 2019. Empowering A* Search Algorithms with Neural Networks for Personalized Route Recommendation. In *SIGKDD*. 539–547.
[25] Meng-xiang Wang, Wang-Chien Lee, Tao-yang Fu, and Ge Yu. 2019. Learning Embeddings of Intersections on Road Networks. In *SIGSPATIAL*. 309–318.
[26] Pu Wang, Timothy Hunter, Alexandre M Bayen, Katja Schechtner, and Marta C González. 2012. Understanding road usage patterns in urban areas. *Scientific reports* 2 (2012), 1001.
[27] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *WWW*. 2022–2032.
[28] Ling Yin Wei, Yu Zheng, and Wen Chih Peng. 2012. Constructing popular routes from uncertain trajectories. In *SIGKDD*. 195–203.
[29] Hao Wu, Ziyang Chen, Weiwei Sun, Baihua Zheng, and Wei Wang. 2017. Modeling Trajectories with Recurrent Neural Networks. In *ICJAI*. 3083–3090.
[30] Hao Wu, Jiangyun Mao, Weiwei Sun, Baihua Zheng, Hanyuan Zhang, Ziyang Chen, and Wei Wang. 2016. Probabilistic Robust Route Recovery with Spatio-Temporal Dynamics. In *SIGKDD*. 1915–1924.
[31] Ning Wu, Jingyuan Wang, Wayne Xin Zhao, and Yang Jin. 2019. Learning to Effectively Estimate the Travel Time for Fastest Route Recommendation. In *CIKM*. 1923–1932.
[32] Andy Yuan Xue, Jianzhong Qi, Xing Xie, Rui Zhang, Jin Huang, and Yuan Li. 2015. Solving the data sparsity problem in destination prediction. *The VLDB Journal* 24, 2 (2015), 219–243.
[33] Andy Yuan Xue, Rui Zhang, Yu Zheng, Xing Xie, Jin Huang, and Zhenghua Xu. 2013. Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. In *ICDE*. IEEE, 254–265.
[34] Can Yang and Gyozo Gidofalvi. 2018. Fast map matching, an algorithm integrating hidden Markov model with precomputation. *IJGIS* 32, 3 (2018), 547–570.
[35] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*. 4800–4810.
[36] Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering regions of different functions in a city using human mobility and POIs. In *SIGKDD*. 186–194.
[37] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. 2011. Driving with knowledge from the physical world. In *SIGKDD*. 316–324.
[38] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. 2011. T-drive: Enhancing driving directions with taxi drivers' intelligence. *IEEE TKDE* 25, 1 (2011), 220–232.
[39] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. 2010. T-drive: driving directions based on taxi trajectories. In *SIGSPATIAL*. ACM, 99–108.
[40] Nicholas Jing Yuan, Yu Zheng, Xing Xie, Yingzi Wang, Kai Zheng, and Hui Xiong. 2014. Discovering urban functional zones using latent activity trajectories. *IEEE Transactions on Knowledge and Data Engineering* 27, 3 (2014), 712–725.
[41] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *SIGKDD*. 793–803.
[42] Kai Zheng, Yu Zheng, Xing Xie, and Xiaofang Zhou. 2012. Reducing Uncertainty of Low-Sampling-Rate Trajectories. In *ICDE*. 1144–1155.