

Heterogeneous Region Embedding with Prompt Learning

Silin Zhou^{1*}, Dan He^{2*}, Lisi Chen¹, Shuo Shang^{1†}, Peng Han^{1†}

¹ University of Electronic Science and Technology of China

² The University of Queensland, Australia

{zhousilinxy, jedi.shang}@gmail.com, d.he@uq.edu.au, lchen012@e.ntu.edu.sg, penghan_study@hotmail.com

Abstract

The prevalence of region-based urban data has opened new possibilities for exploring correlations among regions to improve urban planning and smart-city solutions. Region embedding, which plays a critical role in this endeavor, faces significant challenges related to the varying nature of city data and the effectiveness of downstream applications. In this paper, we propose a novel framework, HREP (Heterogeneous Region Embedding with Prompt learning), which addresses both intra-region and inter-region correlations through two key modules: Heterogeneous Region Embedding (HRE) and prompt learning for different downstream tasks. The HRE module constructs a heterogeneous region graph based on three categories of data, capturing inter-region contexts such as human mobility and geographic neighbors, and intra-region contexts such as POI (Point-of-Interest) information. We use relation-aware graph embedding to learn region and relation embeddings of edge types, and introduce self-attention to capture global correlations among regions. Additionally, we develop an attention-based fusion module to integrate shared information among different types of correlations. To enhance the effectiveness of region embedding in downstream tasks, we incorporate prompt learning, specifically prefix-tuning, which guides the learning of downstream tasks and results in better prediction performance. Our experiment results on real-world datasets demonstrate that our proposed model outperforms state-of-the-art methods.

Introduction

In recent years, the rapid growth of urban data has captured the attention of researchers in the field of urban studies (Crooks et al. 2015; Shang et al. 2016, 2017; Chen and Shang 2019; Chen et al. 2019). As cities are composed of diverse regions, including business districts, residential areas, etc, understanding the structures of cities requires effective learning of high-quality region embeddings. Developing such embeddings can facilitate the creation of smarter and more sustainable cities (Wang et al. 2018). Moreover, these embeddings have direct applications in various areas, such as crime prediction, traffic flow prediction, and real estate price estimation. With the increasing prevalence of mo-

bile computing technologies, an unprecedented amount of urban data has become available, including taxi trajectories, Point-of-Interest (POI). This abundance of data offers essential support for the study of region embeddings.

Many existing studies have attempted to introduce different urban data to facilitate region embedding (Pan et al. 2013; Wang and Li 2017; Yao et al. 2018; Zhang et al. 2019b; Fu et al. 2019; Hui et al. 2020; Zhang et al. 2020; Wu et al. 2022). For instance, (Wang and Li 2017) constructs flow graph and spatial graph through human mobility to learn region embedding since human mobility can directly reflect the commuting correlation between regions. (Yao et al. 2018) extracts human mobility patterns from taxi trajectories and uses the co-occurrence of source-destination regions to learn region embeddings. (Wu et al. 2022) analyzes the mobility data of different periods to establish the mobility pattern for region embedding. However, these methods only consider inter-region data but fail to incorporate the intra-region data, resulting in decreased effectiveness for region embedding.

Intuitively, more city data bring more available information on region attributes (Shang et al. 2015; Wang et al. 2019), which induces the researchers to learn region embeddings integrating with multiple sources of data. Specifically, (Fu et al. 2019) constructs two types of graphs using inter-region data human mobility and intra-region data POIs. These two graphs are simply flattened and concatenated as initial region vectors as input to the AutoEncoder for learning final region embedding. Additionally, (Zhang et al. 2019b) introduces the generative adversarial network in AutoEncoder to learn region embedding, but still uses a similar strategy to flatten and concatenate different graphs. Later, (Zhang et al. 2020) considers several types of data to construct different region-based graphs and adopts a multi-graph fusion mechanism, resulting in improved performance. However, almost methods fail to explore the heterogeneous graph learning of region embedding. Furthermore, all of the above methods fail to explore how the learned region embedding can be used more effectively in downstream tasks.

Inspired by the idea of prompt learning for NLP, we propose a novel framework named **HREP** (**H**eterogeneous **R**egion **E**mbedding with **P**rompt learning), which mainly includes two modules: **H**eterogeneous **R**egion **E**mbedding

*These authors contributed equally.

†Shuo Shang and Peng Han are corresponding authors.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

(HRE) and prompt learning for different downstream tasks. In the HRE module, we construct the region heterogeneous graph by incorporating multiple data sources, including human mobility, POI information, and the geographic neighbor of each region. More specifically, we design a relation-aware GCN by introducing relation embedding into GCN, which can learn different relation-specific region embedding. To extract global information for each relation-specific region embedding, we apply self-attention to learn the correlation between relation-specific region embeddings. Then, an attention-based fusion method is designed to fusion sharing embeddings to the final region embedding. For the prompt learning module, we apply the prefix-tuning method (Li and Liang 2021), which is the automated template continuous prompt learning. Prefix-tuning prepends a sequence of continuous task-specific vectors to the input, with the parameters of the HRE module frozen. By introducing prompt, the training process of downstream tasks becomes guided learning, which enables the downstream tasks to have better learning ability on the region embedding learned in the HRE module. To sum up, we have mainly three contributions listed as follows.

- We migrate prompt learning to region embedding learning and propose the **HREP** model, which contains heterogeneous region embedding module (**HRE**) and prompt learning module for downstream tasks.
- We design a continuous prefix-tuning prompt on region embedding, and the prompt is trained in the downstream task, making the learning of the downstream task as a guided process.
- We conduct extensive downstream experiments to evaluate our model with real-world datasets. The experimental results show the great superiority of our model compared with the state-of-the-art solutions.

Preliminaries

Human Mobility. Given a set of non-overlapping regions $R = \{r_1, r_2, \dots, r_{|R|}\}$ on a studied area, we define the trip record, denoted by $t_{r_a}^{r_b}$ (resp. $t_{r_b}^{r_a}$), for any given two regions r_a, r_b as the number of trajectories of all users from r_a to r_b (resp. from r_b to r_a). Accordingly, we derive a set of human mobility based on the trip records for any pair of regions in R , denoted by $M = \{m_1, m_2, \dots, m_{|M|}\}$, where each m_i represents a tuple with four elements, i.e., $m_i = (r_a, r_b, t_{r_a}^{r_b}, t_{r_b}^{r_a})$.

POI Information. Region Information usually refers to some social attributes of regions. In our work, we consider the POI information, which is denoted by p and can be obtained by category classification statistics. We denote the region information as $P = \{p_1, p_2, \dots, p_{|R|}\}$, $p_i \in \mathbb{R}^f$ where f is the number of categories.

Geographic Neighbor Information. Geographic neighbor information denotes the geographic relation between a region and its neighbors, which is notated as $N = \{N_1, N_2, \dots, N_{|R|}\}$, where N_i represents the set of all regions that are adjacent to region- i .

Region Representation Learning. Given the human mobility M of a set of regions R , POI information of regions, and geographic neighbor information of regions, we aim to learn a set of low dimensional embedding \mathcal{E} to represent each region: $\mathcal{E} = \{e_1, e_2, \dots, e_n\}$, $e_i \in \mathbb{R}^d$, where $e_i \in \mathcal{E}$ is the d -dimension embedding of the region $r_i \in R$ and n is the number of regions.

HREP Framework

Framework Overview

Figure 1 demonstrates the overall framework of our model, which includes four major components: 1) Relation-aware GCN is designed to learn relation-specific region embedding from the heterogeneous graph based on each relation; 2) Embedding sharing aims to learn global information by correlations between the same region with multiple relations; 3) Attention-based fusion method is aim to integrate relation-specific sharing embedding to the final region embedding; 4) Prompt learning is introduced for the prediction of various downstream tasks.

Region Heterogeneous Graph

Region heterogeneous graph, denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, includes multiple types of edges. The set of edge types is represented by \mathcal{R} , which includes the source edge type r_s , target edge type r_t , POI edge type r_p , and geographic neighbor edge type r_g . The source and target edge types (r_s and r_t) are constructed based on correlations from human mobility data, while the POI edge type r_p is constructed using POI information. To construct these three similarity matrices, we follow the same approach as presented in (Zhang et al. 2020). For each node, we obtain the k -nearest neighbors based on different similarity matrices as its neighbors in the heterogeneous graph. Regarding the edge type r_g , we connect each node to all of its geographic neighbors as defined in the previous section.

Relation-Aware Graph Embedding

To capture node information and edge type information from the heterogeneous graph, we design a novel graph embedding method named *relation-aware graph embedding*, which can generate different node embedding by different edge types. The graph embedding in our model is built upon the message-passing architecture of GCN (Kipf and Welling 2017). The basic GCN learns node embedding using symmetric normalization with node degrees over the graph. In our model, we incorporate the relation (i.e., edge type) embedding into GCN to support heterogeneous graph learning. To improve the performance of embedding, we apply the deep layer GCN. Formally, given the random initial node embedding set $E_{node}^{(0)} = \{e_1^{(0)}, e_2^{(0)}, \dots, e_{|R|}^{(0)}\}$ and initial relation embedding $E_{rel}^{(0)} = \{e_s^{(0)}, e_t^{(0)}, e_p^{(0)}, e_g^{(0)}\}$, we update node embedding of the l^{th} layer by follows:

$$e_u^{(l)} = \sigma \left(\sum_{v \in \mathcal{N}_r(u) \cup u} \frac{\mathbf{W}^{(l)} \phi(e_v^{(l-1)}, e_r^{(l-1)})}{\sqrt{|\mathcal{N}_r(u)|} |\mathcal{N}_r(v)|} \right), \quad (1)$$

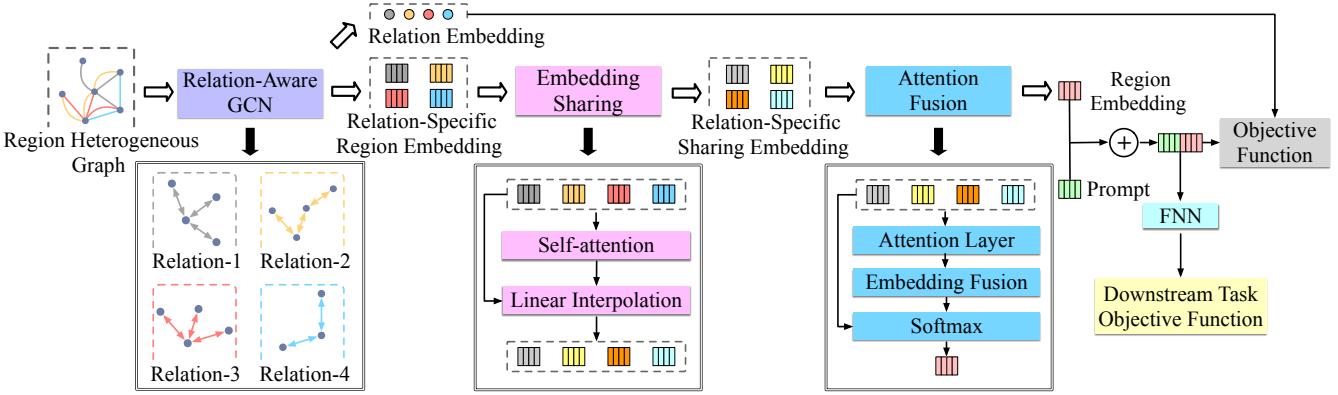


Figure 1: The overall of HREP

where $e_u^{(l)}, e_v^{(l)} \in E_{node}^{(l)}$, $e_r^{(l)} \in E_{rel}^{(l)}$, σ indicates the LeakyReLU activation function, $\mathbf{W}^{(l)}$ is the learnable parameter of the layer- l , and $\mathcal{N}_r(\cdot)$ is the neighbor set under relation type r . Additionally, ϕ represents the entity-relation composition operator, which is defined as follows:

$$\phi(e_v, e_r) = e_v \circ e_r, \quad (2)$$

where \circ denotes the element-wise product.

When updating the node in each layer of the relation-aware GCN, the node embedding is mapped to the new embedding space, thus the relational embedding also needs to be updated accordingly. We use Equation 3 to transform relation embeddings:

$$e_r^{(l)} = \mathbf{W}_r^{(l)} e_r^{(l-1)} + \mathbf{b}_r^{(l)}. \quad (3)$$

where $\mathbf{W}_r^{(l)}$, $\mathbf{b}_r^{(l)}$ are layer-specific parameters that project all relation embedding learned in the previous layer to the same embedding space and allow them to be used in the next layer of the relation-aware GCN. In graph neural network, the feature smoothing phenomenon will occur when the layer size of GNN increases, which causes a significant reduction in performance. To solve this problem, we introduce the ResNet (He et al. 2016) into our node embedding of relation-aware GCN.

So far, in the relation-aware GCN, for the given region embedding, we obtain four types of relation-specific region embeddings, denoted by $\mathcal{E} = \{\mathcal{E}_s, \mathcal{E}_t, \mathcal{E}_p, \mathcal{E}_g\}$, according to different types of relation, as well as four relation embeddings $\mathcal{E}_{rel} = \{e_s, e_t, e_p, e_g\}$.

Relation-Specific Region Embedding Sharing

In general, different attributes are correlated with each other in the same region. For instance, the region with dense human mobility might also have many POIs. Motivated by this, we apply the self-attention mechanism to capture global information. Note that self-attention is insensitive to the sequence order of the input. Specifically, we make use of the multi-head self-attention as suggested (Vaswani et al. 2017) to enhance the strength of correlation capturing.

Formally, given the relation-specific region embeddings set $\mathcal{E} = \{\mathcal{E}_s, \mathcal{E}_t, \mathcal{E}_p, \mathcal{E}_g\}$, we compute one-head self-

attention as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \quad (4)$$

$$\text{head}_h = \text{Attention}(\mathcal{E} \mathbf{W}_h^Q, \mathcal{E} \mathbf{W}_h^K, \mathcal{E} \mathbf{W}_h^V), \quad (5)$$

where \mathbf{W}_h^Q , \mathbf{W}_h^K and \mathbf{W}_h^V are learnable parameters for h^{th} head learning. And we compute multi-head self-attention by the following equation:

$$\text{MultiHead}(\mathcal{E}) = (\|_{h=1}^H \text{head}_h) \mathbf{W}^O, \quad (6)$$

where $\|$ denotes concatenation, \mathbf{W}^O is the learnable parameter for transformation and H is the number of heads. So far, our model learns the relation-specific region sharing embedding, denoted by $\mathcal{E}' = \{\mathcal{E}'_s, \mathcal{E}'_t, \mathcal{E}'_p, \mathcal{E}'_g\}$.

Then, we introduce the learning-based linear interpolation to adjust embedding, which is formulated by follows:

$$\tilde{\mathcal{E}}_i = \mathbf{c}_i \mathcal{E}'_i + (1 - \mathbf{c}_i) \mathcal{E}_i. \quad (7)$$

where $\mathcal{E}_i \in \mathcal{E}$, $\mathcal{E}'_i \in \mathcal{E}'$, and \mathbf{c}_i denotes the learning parameters. Afterwards, we obtain relation-specific region sharing embedding, denoted by $\tilde{\mathcal{E}} = \{\tilde{\mathcal{E}}_s, \tilde{\mathcal{E}}_t, \tilde{\mathcal{E}}_p, \tilde{\mathcal{E}}_g\}$.

Attention-Based Embedding Fusion

To integrate relation-specific region sharing embeddings, we propose *attention-based fusion* learning method where we introduce the attention vector to compute attention weights.

We first use a nonlinear transformation to transform each relation-specific region sharing embedding into hidden space. Then we introduce an attention vector \mathbf{q} to compute relation-based weight. Particularly, given a relation-specific region sharing embedding $\tilde{\mathcal{E}}_i = \{\tilde{e}_j^i\}_{j=1}^{|R|}$, where $\tilde{e}_j^i \in \tilde{\mathcal{E}}$, we average the weight of all node embedding to obtain the attention coefficient.

$$\omega_i = \frac{1}{|R|} \sum_{j=1}^{|R|} \mathbf{q}^T \cdot \sigma(\mathbf{W} \tilde{e}_j^i + \mathbf{b}), \quad (8)$$

where σ is the LeakyReLU activation function. Note that for the meaningful fusion, parameters \mathbf{q} , \mathbf{W} , \mathbf{b} are shared for all relation-specific region sharing embeddings, which can

project all embedding into the same space to compute the attention coefficient. Then we use the softmax function for the normalization.

$$\tilde{\omega}_i = \frac{\exp(\omega_i)}{\sum_{i=1}^4 \exp(\omega_i)}, \quad (9)$$

Next, with the learned coefficient $\tilde{\omega}_i$, we fuse the four relation-specific region sharing embeddings to obtain the final region embedding as follows:

$$\hat{\mathcal{E}} = \sum_{i=1}^4 \tilde{\omega}_i \cdot \tilde{\mathcal{E}}_i. \quad (10)$$

HRE Objective Function

To effectively train our model, we apply multi-task learning to design our objective function for the HRE module. We first use learned region embedding $\hat{\mathcal{E}} = \{e_i\}_{i=1}^{|R|}$ and learned relation embedding $\mathcal{E}_{rel} = \{e_s, e_t, e_p, e_g\}$ to generate multi-task embedding as follows:

$$\hat{e}_i = e_i \circ e_r, \quad (11)$$

where $e_r \in \mathcal{E}_{rel}$, \circ denotes the element-wise product. Then, we obtain four task embeddings $\hat{\mathcal{E}}_s$, $\hat{\mathcal{E}}_t$, $\hat{\mathcal{E}}_p$, and $\hat{\mathcal{E}}_g$.

Geographic Neighbor Loss. Intuitively, regions that are geographically adjacent to each other might have a higher similarity. Therefore, to preserve the property, we define geographic neighbor loss, notated by L_{geo} , as follows. Formally, given the fusion embedding $\hat{\mathcal{E}}_g = \{e_i^g\}_{i=1}^{|R|}$, we have:

$$L_{geo} = \sum_{i=1}^{|R|} \max \left\{ \|e_i - e_i^{g(pos)}\|_2 - \|e_i - e_i^{g(neg)}\|_2, 0 \right\}, \quad (12)$$

where $e_i^{g(pos)}$ (resp. $e_i^{g(neg)}$) is a positive (resp. negative) sample of geographic neighbors (resp. non-geographic neighbors) from the i -th region.

Mobility Loss. To reconstruct mobility, given the embeddings $\hat{\mathcal{E}}_s = \{e_i^s\}_{i=1}^{|R|}$, $\hat{\mathcal{E}}_t = \{e_i^t\}_{i=1}^{|R|}$ and human mobility M , we first compute original mobility distribution as follows:

$$p(r_j|r_i) = \frac{t_{r_i}^{r_j}}{\sum_{j=1}^{|R|} t_{r_i}^{r_j}} \quad (13)$$

Then, we reconstruct source and target region distribution $d_s(r_j|r_i)$, $d_t(r_i|r_j)$ as follows:

$$d_s(r_j|r_i) = \frac{\exp(e_i^{sT} e_j^t)}{\sum_j \exp(e_i^{sT} e_j^t)}, \quad (14)$$

$$d_t(r_i|r_j) = \frac{\exp(e_j^{tT} e_i^s)}{\sum_i \exp(e_j^{tT} e_i^s)}, \quad (15)$$

Therefore, mobility loss function by KL divergence can be defined as follows:

$$L_{st} = \sum_{i,j} -p_s(r_j|r_i) \log d_s(r_j|r_i) - p_t(r_i|r_j) \log d_t(r_i|r_j), \quad (16)$$

POI Loss. To reconstruct POI correlation, given the POI similarity matrix \mathcal{M}_p and POI task embedding $\hat{\mathcal{E}}_p = \{e_i^p\}_{i=1}^{|R|}$, we have the POI loss formulated by:

$$L_{poi} = \sum_{i=1}^{|R|} \sum_{j=1}^{|R|} (\mathcal{M}_p^{ij} - e_i^p T e_j^p)^2, \quad (17)$$

As a result, the final objective function can be formulated as follow:

$$\mathcal{L} = L_{geo} + L_{st} + L_{poi}. \quad (18)$$

Prompt Learning for Downstream Tasks

Recently, a new technique, named ‘prompt learning’ has been widely used in NLP and at the forefront of various NLP tasks, which is developed beyond fine-tuned models. Usually, in a fine-tuned model, embeddings are learned from the language model (LM) on a large dataset. Then, we take these embeddings as inputs for a consecutive model and keep learning these embeddings based on a specific downstream task. However, in prompt learning, we freeze these embeddings and introduce task-specific *prompt* into learned embedding to guide the learning process to adapt to different downstream tasks. The scale of embeddings learned from the LM model is usually large. In contrast, prompt learning freezes these embeddings for the downstream tasks and only introduces a few parameters, yet achieving great performance.

In our model, we try to migrate prompt learning to region embedding learning, and to the best of our knowledge we are the first to apply prompt learning to region embedding learning. Although we borrow the idea of prompt learning from NLP, the implementation is still very challenging and different from the NLP tasks. Specifically, pre-training and downstream tasks in NLP have sequential textual input, i.e., the input format in NLP is uniform. Thus the NLP downstream tasks only need to adjust the inputs and then apply them to the pre-training model. In contrast, the input type of our pre-training model (i.e. HRE module) and downstream tasks in region embedding are not uniform, so it is not possible to apply the input adjustment of the downstream task to the pre-training model.

In our work, we apply the prefix-tuning, which is the continuous prompt. Unlike handling text in NLP, we can not design a proper prompt for region embedding manually due to different input formats. Thus, our goal is to make the model able to understand and learn the prompt in the embedding space. Generally, the prefix-tuning prepends a prefix embedding as prompt and freezes the parameters from the pre-training module. However, the scenarios in our model are different. 1) Since the model for the downstream task is different from the pre-trained model, the region embeddings learned by pre-training is used both as part of the input and as the parameters to be frozen, while the prompt is the part to be updated. 2) The input to the pre-training model does not match the input format for the downstream task and also differs from the NLP sequence inputs. To address the above issues, we concatenate prompt embedding

Dataset	Description
Regions	180 regions based on the Manhattan community boards.
PoI data	20K PoIs in the studied areas, such as stations, stores, etc.
Check-in data	100K check-in locations of about 200 categories.
Taxi trips	10M taxi trips during one month.
Crime data	40K crime records during one year.

Table 1: Dataset description (K=10³, M=10⁶)

and learned region embedding from our HRE model. Formally, given the region embedding $\hat{\mathcal{E}} = \{e_i\}_{i=1}^{|R|}$ and prompt embedding $\mathcal{P} = \{p_i\}_{i=1}^{|R|}$ for each region, we have:

$$\tilde{p}_i = p_i \oplus e_i, \quad (19)$$

where \oplus is the concatenation operation. Note that we propose the prompt from each region instead of using a single prompt because the learned embeddings for regions are different. Then we apply prompt-specific embedding to different downstream models, such as the regression model, and classification model. In the model for the downstream tasks, we freeze e_i and update p_i with model parameters. Then we use a feedforward neural network (FNN) for the prediction, formulated as follows:

$$\tilde{y}_i = \text{FNN}(\tilde{p}_i), \quad (20)$$

where \tilde{y}_i is the prediction of the region- i in the downstream task. To optimize our prompt, we use the MSE loss function as follows:

$$\mathcal{L}_p = \frac{1}{|R|} \sum_{i=1}^{|R|} (\tilde{y}_i - y_i)^2. \quad (21)$$

where y_i is the ground truth of region- i in the downstream task.

Experiment

In our experimental study, we design three downstream applications to evaluate the performance of our model, including crime prediction, check-in prediction, and land usage classification. Our model is implemented with PyTorch on an Nvidia RTX3090 GPU. The details of implementation can be viewed on this URL ¹.

Experiment Setting

Dataset We collect a variety of real-world data from NYC Open Data ² specific for the Manhattan, New York area, where Taxi trips are used as human mobility. We divide the Manhattan area into 180 regions based on the community boards. The detailed description of datasets is shown in Table 1.

¹<https://github.com/slzhou-xy/HREP>

²<https://opendata.cityofnewyork.us>

	Crime Prediction			Check-in Prediction		
	MAE	RMSE	R ²	MAE	RMSE	R ²
GAE	96.55	133.10	0.19	498.23	803.34	0.09
LINE	117.53	152.43	0.06	564.59	853.82	0.08
node2vec	75.09	104.97	0.49	372.83	609.47	0.44
HDGE	72.65	96.36	0.58	399.28	536.27	0.57
ZE-Mob	101.98	132.16	0.20	360.71	592.92	0.47
MV-PN	92.30	123.96	0.30	476.14	784.25	0.08
MVURE	*69.28	96.51	0.57	312.63	513.02	0.61
MGFN	70.21	*89.60	*0.63	*292.60	*451.76	*0.69
HREP	65.66	84.59	0.68	270.28	406.53	0.75
Improve	5.23%	5.59%	7.93%	6.94%	10.01%	8.70%

Table 2: The main experiment performance. The mark * indicates the compared baseline for improvements.

Hyper Parameters The dimension of our model is 144. The dimension of prompt embedding is also set as 144. The layer of relation-aware GCN is set as 3. In the multi-head self-attention, we set the number of heads as 4. We adopt Adam to optimize our model, including HRE module and prompt learning module, and both learning rates are set as 0.001. The epoch is set as 2000 in HRE module and 6000 in prompt learning. Moreover, we set the value of k as 10.

Baseline Solutions. We compare our model with 8 baseline models.

- **GAE.** GAE (Kipf and Welling 2016) is an asymmetric network, which uses GCN as an encoder to learn node representations, and inner product as a decoder to reduce the adjacency matrix.
- **LINE.** LINE (Tang et al. 2015) uses the first-order proximity and the second-order proximity to learn node embedding.
- **node2vec.** Node2vec (Grover and Leskovec 2016) tunes random walk algorithm to adapt the graph structure, which captures homophily and structural equivalence of node in the graph.
- **HDGE.** HDGE (Wang and Li 2017) constructs traffic flow graph and spatial graph, then it samples node paths to jointly learn region embedding.
- **ZE-Mob.** ZE-Mob (Yao et al. 2018) uses co-occurrence to compute point-wise mutual information (PMI) to learn region embedding.
- **MV-PN.** MV-PN (Fu et al. 2019) uses PoI data and human mobility to construct a multi-view POI-POI network, and then AutoEncoder is used to learn region embedding.
- **MVURE.** (Zhang et al. 2020) adopts the intra-region and inter-region data to construct multi-view graphs, then applies multi-view fusion to learn region embedding.
- **MGFN.** MGFN (Wu et al. 2022) construct mobility patterns by human mobility to learn region embedding.

Main Performance Comparison

In this set of experiments, we mainly consider two downstream applications, including crime prediction and check-

	Crime Prediction			Check-in Prediction		
	MAE	RMSE	R^2	MAE	RMSE	R^2
HREP/G	73.30	95.15	0.56	303.58	466.01	0.68
HREP/R	67.95	95.25	0.59	328.62	556.57	0.54
HREP/F	67.39	88.96	0.65	313.29	480.07	0.66
HREP/P	66.89	85.91	0.67	273.56	409.98	0.74
HREP	65.66	84.59	0.68	270.28	406.53	0.75

Table 3: The performance of ablation experiment.

in prediction, which are widely used to evaluate the performance of region embedding (Zhang et al. 2020; Wu et al. 2022). Three general metrics are used for performance evaluation, i.e., Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and coefficient of determination (R^2). Table 2 show the experimental results, which overall demonstrate that our model outperforms all state-of-the-art methods dramatically. We present the detailed analysis as follows. 1) The graph embedding methods (e.g., LINE, GAE, node2vec) have the worse performance since these methods only can learn node embedding from graphs while failing to capture other region information. 2) With human mobility data (e.g., HDGE, ZE-Mob, MGFN), the performance of region embedding on downstream tasks can be improved a lot. In particular, MGFN constructs mobility patterns with spatial-temporal human mobility data earning the second best performance. 3) The average performance of methods with multiple urban data (e.g., MV-PN, MVURE) is better than those with a single type of data, which proves that multiple data types can involve different relations between regions. 4) Our model achieves the best performance. On the one hand, we not only consider multiple types of data, but also extend the region embedding problem further to the heterogeneous graph problem. On the other hand, we migrate the prompt learning to enhance the performance of region embedding for different downstream tasks.

Ablation Experiment

In this set of experiments, we conduct comprehensive experiments with various settings to verify the validity of different components in our model. The details of the ablation methods are as follows.

- **HREP/G.** We remove geographic correlations among regions and keep three other correlations instead.
- **HREP/R.** We use GCN to learn node embedding without relation embeddings.
- **HREP/F.** We replace attention-based fusion with element-wise fusion.
- **HREP/P.** We remove prompt learning and only make use of region embedding from heterogeneous graph learning in the feedforward neural network.

Table 3 present the ablation results of our model and its variants. From these results, we have the following observations. 1) Geographic neighbors can improve performance

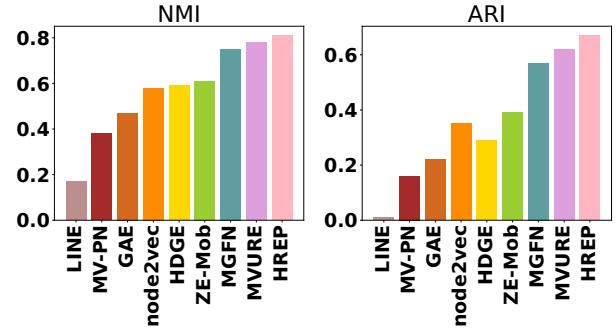


Figure 2: Land usage classification performance

greatly because adjacent regions will exhibit high similarity. 2) By introducing relational embeddings, the problem is introduced to the heterogeneous graph, enabling more effective representations to be learned from different correlations among regions. 3). Attention-based fusion method can provide more meaningful embedding fusion. 4). Compared to using region embedding directly in downstream tasks, prompt learning can provide substantiating guidance for different downstream tasks.

HRE Module Performance

To further investigate the effectiveness of our HRE model, we perform a set of experiments on land usage classification, which is a clustering-based experiment without labels, where prompt learning is unable to be applied. In this application, our goal is to cluster regions belonging to the same function together as much as possible by K-means. Note that the Manhattan area is divided into 12 categories of districts through the community boards (Berg 2010) (e.g., residual districts, business districts). We use two metrics to evaluate the performance in clustering: Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). Fig 2 shows that our heterogeneous graph learning module can achieve the best performance on land usage classification, which is contributed by three aspects: 1). We introduce new city data, region geographic neighbors, and constructed new node relationships in the heterogeneous graph based on this data. Specifically, the geographic neighbor can associate a region with its neighbors, which has a similar goal as clustering. 2). We design a relation-aware GCN by introducing relation embedding, which can learn relation-specific region embeddings based on relations. Then we use these relation embedding and region embedding to apply multi-task learning to optimize our model. 3). Compared to the previous studies, we use a more effective fusion method to integrate embedding from different correlations.

Prompt Update Strategy Evaluation

Prompt learning has different strategies for parameter updates (Liu et al. 2023). We conduct a set of experiments to study the impact of different parameter update strategies for prompt learning on downstream tasks.

	Crime Prediction			Check-in Prediction		
	MAE	RMSE	R ²	MAE	RMSE	R ²
RP	89.54	108.67	0.45	390.50	545.74	0.49
R	110.11	145.23	0.15	506.46	823.56	0.08
no-RP	105.87	138.56	0.18	450.32	702.25	0.11
HREP	65.66	84.59	0.68	270.28	406.53	0.75

Table 4: The performance of prompt update methods.

- **RP**: We update parameters for both the region embedding and the prompt.
- **R**: We freeze the parameters for prompt and only update parameters for region embedding.
- **no-RP**: We freeze parameters for both region embedding and prompt.

Table 4 show the experimental results, which demonstrate the strategy of parameter update for prompt learning has a significant impact on the results. 1) Comparing **R** and **no-RP**, we observe that updating region embedding will cause worse performance, which shows our heterogeneous graph learning module can learn effective region embedding. 2). Compare **RP** and **R**, updating prompt can improve performance a lot. 3). Comparing **RP** and **HREP** further demonstrates the validity of the region embedding learned in heterogeneous graph learning module and does not require to be updated in prompt learning. Instead, only updating the prompt can play a guiding role in the downstream task.

Related Work

Graph Neural Network

Graph neural networks are aimed to learn graph embedding on graph structure. In recent years, research on graph neural networks has been very popular. GCN (Kipf and Welling 2017) introduces the idea of convolution in CV into the graph neural network. GraphSage (Hamilton, Ying, and Leskovec 2017) makes graph neural network successfully applied to inductive representation learning. (Velickovic et al. 2018) design GAT after introducing the attention mechanism into the neighbor aggregation. GIN (Xu et al. 2019) demonstrates that GNN can achieve the performance of the Weisfeiler-Lehman graph isomorphism test. Meanwhile, graph studies have attempted to extend GNNs for modelling heterogeneous graphs. RGCN (Schlichtkrull et al. 2018) is designed to model knowledge graphs. Het-GNN (Zhang et al. 2019a) adopts different RNNs for different node types to integrate multi-modal features. HGT (Hu et al. 2020) introduces transformer into GNN.

Region Embedding

The rise of urban studies has led to an important research direction of urban region representation learning. Region embedding is closely related to the various characteristics of regions, such as POI, and human mobility. (Wang and Li 2017) proposes to use human mobility to construct the transition matrix of the graph. (Yao et al. 2018) uses human mobility to

count co-occurrences. These methods, while yielding good results, are all based on individual attributes of the region. Recently, researchers have tried to use multiple properties of the city to learn region embedding. (Fu et al. 2019) exploits the use of POI attributes and human mobility to obtain inter-region and intra-region relationships and uses AutoEncoder to learn region embedding. (Zhang et al. 2019b) adds generative adversarial networks (GAN) to the AutoEncoder on this basis and obtains better results. These methods consider multiple properties but do not consider the relation between properties. To solve this problem, (Zhang et al. 2020) adopts the multi-view method to learn region embedding. (Wu et al. 2022) construct mobility pattern by human mobility data to learn region embedding.

Prompt Learning

Prompting means prepending instructions and a few examples to the task input and generating the output (Liu et al. 2023). A single language model trained in entirely unsupervised learning can be used to solve many tasks (Sun et al. 2021). Prompt shapes can be separated into cloze (Petroni et al. 2019; Cui et al. 2021) and prefix prompt (Li and Liang 2021; Lester, Al-Rfou, and Constant 2021). The early prompt template engineering are hand-crafted prompts (Petroni et al. 2019; Brown et al. 2020). However, a manual template not only takes time and experience but also fails to discover optimal prompts. To solve these problems, an automated prompt is proposed. Automated prompt can be divided into discrete prompts (Wallace et al. 2019; Shin et al. 2020) and continuous prompt (Li and Liang 2021; Lester, Al-Rfou, and Constant 2021).

Conclusion

In this paper, we propose a novel model, named HREP, for region embedding, which considers both intra-region and inter-region correlations. Specifically, we make use of human mobility, POI data, and region geographic neighbors to construct a region heterogeneous graph. We first develop a relation-aware GCN to learn relation-specific region embedding from different relation types in the heterogeneous graph. Then, to capture the global correlation between different relation-specific region embedding, we apply the multi-head self-attention to learn sharing embedding. The attention-based fusion method is introduced to learn the final region embedding. Additionally, we design prompt learning to replace the direct use of region embedding in the downstream tasks. In particular, we apply the continuous prompt method prefix-tuning, which is able to have different guiding effects in different downstream tasks, and therefore can achieve better performance. The experiments on two downstream applications based on the real-world datasets show that our model outperforms state-of-the-art methods.

Acknowledgments

This work was supported by the NSFC (U2001212, U22B2037, U21B2046, 62032001, and 61932004).

References

Berg, B. F. 2010. *New York City Politics: Governing Gotham*. Rutgers University Press.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *NIPS*.

Chen, L.; and Shang, S. 2019. Region-Based Message Exploration over Spatio-Temporal Data Streams. In *AAAI*, 873–880.

Chen, L.; Shang, S.; Zheng, K.; and Kalnis, P. 2019. Cluster-Based Subscription Matching for Geo-Textual Data Streams. In *ICDE*, 890–901.

Crooks, A.; Pfoser, D.; Jenkins, A.; Croitoru, A.; Stefanidis, A.; Smith, D.; Karagiorgou, S.; Efentakis, A.; and Lampranidis, G. 2015. Crowdsourcing urban form and function. *Int. J. Geogr. Inf. Sci.*, 720–741.

Cui, L.; Wu, Y.; Liu, J.; Yang, S.; and Zhang, Y. 2021. Template-Based Named Entity Recognition Using BART. In *ACL*, 1835–1845.

Fu, Y.; Wang, P.; Du, J.; Wu, L.; and Li, X. 2019. Efficient Region Embedding with Multi-View Spatial Networks: A Perspective of Locality-Constrained Spatial Autocorrelations. In *AAAI*, 906–913.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable Feature Learning for Networks. In *SIGKDD*, 855–864.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*, 1024–1034.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Hu, Z.; Dong, Y.; Wang, K.; and Sun, Y. 2020. Heterogeneous Graph Transformer. In *WWW*, 2704–2710.

Hui, B.; Yan, D.; Ku, W.; and Wang, W. 2020. Predicting Economic Growth by Region Embedding: A Multigraph Convolutional Network Approach. In *CIKM*, 555–564.

Kipf, T. N.; and Welling, M. 2016. Variational Graph Auto-Encoders. *arXiv:1611.07308*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *EMNLP*, 3045–3059.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *ACL*, 4582–4597.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.*, 55(9): 195:1–195:35.

Pan, G.; Qi, G.; Wu, Z.; Zhang, D.; and Li, S. 2013. Land-Use Classification Using Taxi GPS Traces. *IEEE Trans. Intell. Transp. Syst.*, 113–123.

Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P. S. H.; Bakhtin, A.; Wu, Y.; and Miller, A. H. 2019. Language Models as Knowledge Bases? In *EMNLP*, 2463–2473.

Schlichtkrull, M. S.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC*, 593–607.

Shang, S.; Chen, L.; Jensen, C. S.; Wen, J.; and Kalnis, P. 2017. Searching Trajectories by Regions of Interest. *IEEE Trans. Knowl. Data Eng.*, 29(7): 1549–1562.

Shang, S.; Guo, D.; Liu, J.; Zheng, K.; and Wen, J. 2016. Finding regions of interest using location based social media. *Neurocomputing*, 173: 118–123.

Shang, S.; Zheng, K.; Jensen, C. S.; Yang, B.; Kalnis, P.; Li, G.; and Wen, J. 2015. Discovery of Path Nearby Clusters in Spatial Networks. *IEEE Trans. Knowl. Data Eng.*, 27(6): 1505–1518.

Shin, T.; Razeghi, Y.; IV, R. L. L.; Wallace, E.; and Singh, S. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *EMNLP*, 4222–4235.

Sun, Y.; Wang, S.; Feng, S.; Ding, S.; Pang, C.; Shang, J.; Liu, J.; Chen, X.; Zhao, Y.; Lu, Y.; Liu, W.; Wu, Z.; Gong, W.; Liang, J.; Shang, Z.; Sun, P.; Liu, W.; Ouyang, X.; Yu, D.; Tian, H.; Wu, H.; and Wang, H. 2021. ERNIE 3.0: Large-scale Knowledge Enhanced Pre-training for Language Understanding and Generation. *CoRR*.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. LINE: Large-scale Information Network Embedding. In *WWW*, 1067–1077.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*, 5998–6008.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Wallace, E.; Feng, S.; Kandpal, N.; Gardner, M.; and Singh, S. 2019. Universal Adversarial Triggers for Attacking and Analyzing NLP. In *EMNLP*, 2153–2162.

Wang, H.; and Li, Z. 2017. Region Representation Learning via Mobility Flow. In *CIKM*, 237–246.

Wang, P.; Zhang, J.; Liu, G.; Fu, Y.; and Aggarwal, C. C. 2018. Ensemble-Spotting: Ranking Urban Vibrancy via POI Embedding with Multi-view Spatial Graphs. In *SIAM*, 351–359.

Wang, Y.; Li, J.; Zhong, Y.; Zhu, S.; Guo, D.; and Shang, S. 2019. Discovery of accessible locations using region-based geo-social data. *World Wide Web*, 22(3): 929–944.

Wu, S.; Yan, X.; Fan, X.; Pan, S.; Zhu, S.; Zheng, C.; Cheng, M.; and Wang, C. 2022. Multi-Graph Fusion Networks for Urban Region Embedding. In *IJCAI*, 2312–2318.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *ICLR*.

Yao, Z.; Fu, Y.; Liu, B.; Hu, W.; and Xiong, H. 2018. Representing Urban Functions through Zone Embedding with Human Mobility Patterns. In *IJCAI*, 3919–3925.

Zhang, C.; Song, D.; Huang, C.; Swami, A.; and Chawla, N. V. 2019a. Heterogeneous Graph Neural Network. In *SIGKDD*, 793–803.

Zhang, M.; Li, T.; Li, Y.; and Hui, P. 2020. Multi-View Joint Graph Representation Learning for Urban Region Embedding. In *IJCAI*, 4431–4437.

Zhang, Y.; Fu, Y.; Wang, P.; Li, X.; and Zheng, Y. 2019b. Unifying Inter-region Autocorrelation and Intra-region Structures for Spatial Embedding via Collective Adversarial Learning. In *SIGKDD*, 1700–1708.