POI-Enhancer: An LLM-based Semantic Enhancement Framework for POI Representation Learning

Jiawei Cheng[1,2], **Jingyuan Wang**[1,3,4,*], **Yichuan Zhang**[1], **Jiahao Ji**[1], **Yuanshao Zhu**[2], **Zhibo Zhang**[1], **Xiangyu Zhao**[2,*]

[1]SKLCCSE, School of Computer Science and Engineering, Beihang University, Beijing, China
[2]Department of Data Science, City University of Hong Kong, Hong Kong, China
[3]MIIT Key Laboratory of Data Intelligence and Management, Beihang University, Beijing, China
[4]School of Economics and Management, Beihang University, Beijing, China

Abstract

POI representation learning plays a crucial role in handling tasks related to user mobility data. Recent studies have shown that enriching POI representations with multimodal information can significantly enhance their task performance. Previously, the textual information incorporated into POI representations typically involved only POI categories or check-in content, leading to relatively weak textual features in existing methods. In contrast, large language models (LLMs) trained on extensive text data have been found to possess rich textual knowledge. However leveraging such knowledge to enhance POI representation learning presents two key challenges: first, how to extract POI-related knowledge from LLMs effectively, and second, how to integrate the extracted information to enhance POI representations. To address these challenges, we propose POI-Enhancer, a portable framework that leverages LLMs to improve POI representations produced by classic POI learning models. We first design three specialized prompts to extract semantic information from LLMs efficiently. Then, the Dual Feature Alignment module enhances the quality of the extracted information, while the Semantic Feature Fusion module preserves its integrity. The Cross Attention Fusion module then fully adaptively integrates such high-quality information into POI representations and Multi-View Contrastive Learning further injects human-understandable semantic information into these representations. Extensive experiments on three real-world datasets demonstrate the effectiveness of our framework, showing significant improvements across all baseline representations.

Code — https://github.com/Applied-Machine-Learning-Lab/POI-Enhancer

Extended version — https://github.com/JarvisOrange/POI-Enhancer

1 Introduction

With the advancement of smart city technology (Ji et al. 2022b; Wang et al. 2021b; Wang, Wang, and Wu 2018) and the widespread adoption of smart devices, the volume of location-based mobile data, such as POI (Points of Interest) check-in data and user trajectory data, has surged (Ding et al. 2018; Zhu et al. 2023). Predicting user destinations (Zhao et al. 2020), forecasting visit flow (Song et al. 2020), and similar tasks (Han et al. 2024) have become key research focuses. In tackling these difficulties, POI representation learning, which can be trained via self-supervised methods and utilized across various tasks like traffic forecasting (Liu et al. 2024b; Ji et al. 2022a; Wang et al. 2022) and trajectory predction (Jiang et al. 2023b; Wu et al. 2019; Wang et al. 2018), stands as a particularly meaningful and promising direction.

To enhance the diversity of information within POI representation vectors and achieve superior performance in complex downstream tasks, researchers are exploring the integration of various information beyond basic geographic data. For example, they incorporated user preference data (Dai et al. 2022) and visual information (Balsebre et al. 2023) into POI representations. Although related textual information, such as POI categories (e.g. restaurants and hotels) and check-in content on social media like Twitter, provides some insights into the social functions and other aspects of POIs, the semantic richness and depth of these data are limited. When compared to the vast amount of descriptive information available on the internet regarding POIs, these data sources fall short in both content richness and coverage. In recent years, large language models (LLMs) trained on extensive volumes of internet data have been applied across numerous fields, demonstrating remarkable capabilities, particularly in the domain of spatial-temporal data (Li et al. 2024). Although LLMs have proven beneficial in addressing challenges in this area, leveraging LLMs to enhance POI representation presents two specific challenges.

The first challenge lies in effectively **extracting the geographical knowledge within LLMs**. A common idea (Golkar et al. 2023) is to provide LLMs with prompts related to geographic information and then obtain text output. However, LLMs have limitations in handling numerical input, and for representation learning, we need vectors that are versatile across tasks, which makes this method not suitable. Some studies (Chen, Wang, and Xu 2023; Liu et al. 2024a) have also experimented with feeding extracted spatial-temporal features to a partially or fully frozen LLM, using the LLM as the backbone to solve specific problems. But these works are typically tailored to a single spatial-temporal task and extract information specific to that task only. However, POI representation learning aims to capture

*Corresponding authors.

versatile information to address diverse tasks. Clearly, task-specific extraction is insufficient for this requirement.

The second challenge is how to effectively **integrate the extracted textual information into POI representation** for enhancement. Since the information extracted by LLMs is versatile, combining these diverse aspects information with the POI representation is difficult. Most researchers (Dai et al. 2022) adopted the approach of one-hot encoding the corresponding POI category features and then concatenating them with the representation vectors, which overlooks the interactions between features. For example, the POI category and time are related: a restaurant's lunch hours and lunch break times exhibit different visitor flow patterns. This limits the ability to exploit the richness of semantic information to enhance POI representations.

To address the challenges, we propose a POI representation enhancement framework, called POI-Enhancer, which is designed to leverage textual information in LLM to strengthen embedding vectors. Specifically, to better utilize LLMs for extracting textual features of POIs, we develop unique prompts to separately extract features related to POI addresses, visit patterns, and surrounding environments. Following this, we design the Dual Feature Alignment module to leverage the relationships between textual features, enabling the acquisition of higher-quality semantic information. The Semantic Feature Fusion module is specifically designed to ensure the preservation of high-quality semantic information. Then, to fully integrate the extracted information with the representation vectors, we introduce the Cross Attention Fusion module based on the attention mechanism. Finally, we incorporate Multi-View Contrastive Learning to further inject human-understandable semantic information into POI representations to enhance its capability of capturing real-world patterns.

We summarize our main contributions as follows:

- To the best of our knowledge, POI-Enhancer is the first to introduce LLM-based textual knowledge to enhance POI representations of POI learning models. We demonstrate that leveraging knowledge from LLMs is crucial for boosting the performance of POI embedding models.
- We design three kinds of specialized prompts to thoroughly extract textual information from LLMs, and employ a cross-attention mechanism to adaptively integrate these information into POI representations. We also introduce temporal, spatial, and functional contrastive learning to refine the POI representations.
- We conducted extensive experiments on three public real-world datasets across various downstream tasks. The results demonstrate that our approach significantly enhances the performance of POI embedding methods.

2 Preliminaries

Definition 1 (**Point of Interest (POI)**). *A POI is a specific geographic location with some basic attributes* $p = (id, pn, c, lon, lat)$*, where* id *indicates index,* pn *means name of POI,* c *denotes category,* lon *and* lat *represent longitude and latitude coordinates respectively. Besides, each POI has some extra attributes such as visit pattern, address, and surrounding environment. An example of the attributes of a POI in New York City is provided in Tab. 1.*

Definition 2 (**Check-in Record**). *A check-in record is a triplet* $r = (u, p, t)$ *which means a user* u *visits POI* p *at time* t*. A user's movement behavior over a period of time can be modeled by a sequence of check-in records, which we define as a Check-in Record Sequence. It can be represented by* $R = \{r_1, r_2, ..., r_L\}$*, where the check-in records are arranged in the order of time sequence and* L *is the length of the Check-in Record Sequence. We also denote the set of all users' check-in record sequences as* S*.*

Definition 3 (**POI Representation**). *Given the set of all POIs* $P = \{p_1,\ p_2,\ \ldots,\ p_N\}$*, where n is the number of the set, a mapping function* f *is used to generate a fixed vector representation* $E_{p_i} = f(p_i)$ *for each POI.*

Problem Statement. Given a POI Representation function $\mathcal{F}$, POIs set $P = \{p_1,\ p_2,\ \ldots,\ p_N\}$ and other related data *e.g.,* check-in record sequences S, with the aid of LLM, the aim of our framework is to learn a function g that enhance the capability of the function F, *i.e.,* $E_{p_i} = g(\mathcal{F}(p_i)),\ E_{p_i} \in \mathbb{R}^d$, where d is a uniform dimension.

Attribute	Value
POI ID	22337
Name	New York Stock Exchange
Longitude	74.011154
Latitude	40.706806
Category	Stock Exchange
Street Name	Wall Street
House Number	11
Postal Code	10005
Surrounding	Office, Building and Road
Visit Pattern	Between 6 am and 9 am, Weekday

Table 1: An Example of the POI attributes.

3 Methodology

This section provides a comprehensive demonstration of the technical details of POI-Enhancer framework and Fig. 1 presents the overall architecture. In Fig. 1, part (a) is the Prompt Generation and Feature Extraction phase, where specialized prompts are generated and used to extract relevant semantic information from the LLM. The second phase, Embedding Enhancement, corresponds to part (b), where the extracted information is further refined and integrated with the POI representations to be enhanced. Finally, part (c) represents Multi-View Contrastive Learning, where we designed three sampling strategies to select positive and negative samples for contrastive learning. Besides, to assist LLMs in more accurately capturing POI-related knowledge, we additionally processed and derived three kinds of extra attributes mentioned above. A detailed description of this procedure can be found in the Supplementary Material.

Prompt Generation and Feature Extraction

Generate prompt Due to the LLM's low sensitivity to numbers, we need to bundle basic attributes like latitude, longitude, and name with extra attributes when inputting them, to

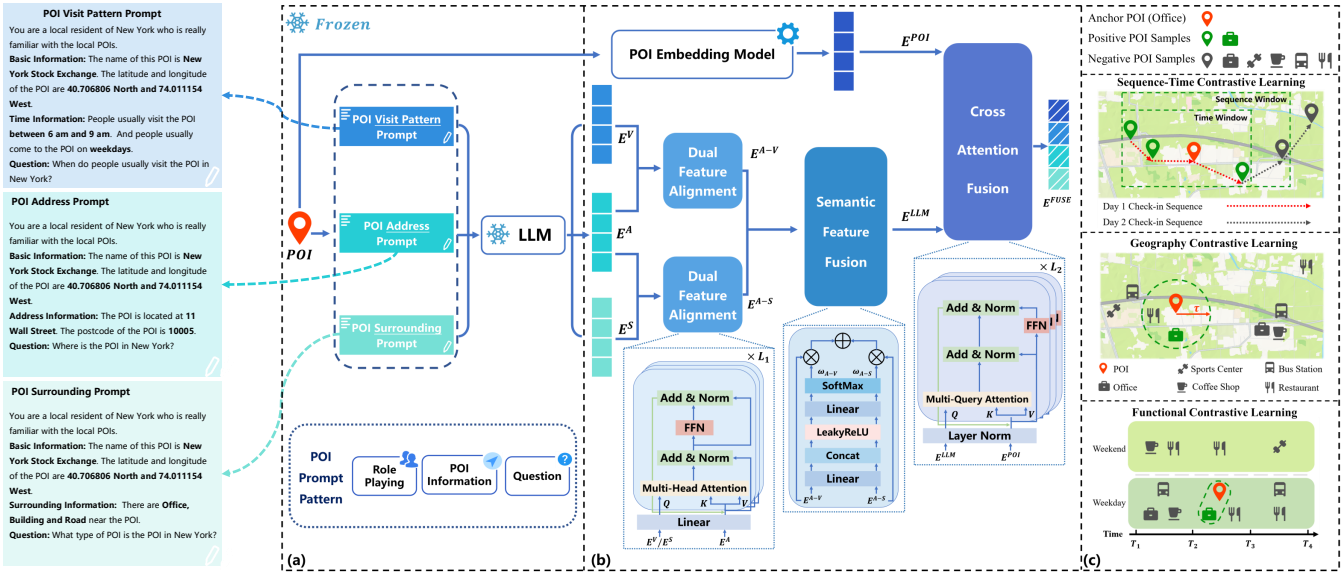

Figure 1: (a): Prompt Generation and Feature Extraction are used to obtain prompts and get textual features from the LLM. (b): Embedding Enhancement is designed to enhance POI embeddings by leveraging textual features. (c): Multi-View Contrastive Learning enables the sampling of more diverse positive and negative examples during training

.

help the LLM accurately target the desired POI. Besides, simply stacking various features into a prompt can make it difficult for the LLM to focus on key points and effectively extract information. Hence, the proposed prompt pattern consists of three parts: (1) Role-Playing, (2) POI Information, and (3) the Question. The POI Information part encompasses basic information and extra information, corresponding to the basic and extra attributes, respectively. Firstly, the design purpose of role-playing at the beginning of the prompt is to allow the LLM to fully unleash its knowledge, enabling the LLM to embody a role familiar with geographical knowledge. An attribute header is added in front of the POI information to help the LLM accurately capture the information of input attributes. Next, we generate multiple sentences based on combinations of the basic attributes and three extra attributes. Lastly, inspired by the (Gurnee and Tegmark 2024), we design the question at the end of each prompt about the content to trigger the relevant knowledge. Consequently, we generate three types of prompts for each POI p_i:POI Visit Pattern Prompt, POI Address Prompt, and POI Surrounding Prompt, denoted as $T^V_{p_i}, T^A_{p_i}, T^S_{p_i}$. An Example of the prompt we generated is shown in Fig. 1.

Extract from LLM In POI-Enhancer, we input the prompts into the LLM and take the final hidden layer state from the LLM as the semantic feature. It is worth noting that the LLM serves as a frozen encoder when training. So, for a POI p_i, the feature extraction process can be denoted as:

$$\boldsymbol{E}^V_{p_i} = \mathcal{H}(T^V_{p_i}),\ \boldsymbol{E}^A_{p_i} = \mathcal{H}(T^A_{p_i}),\ \boldsymbol{E}^S_{p_i} = \mathcal{H}(T^S_{p_i}), \quad (1)$$

where $\boldsymbol{E}^V_{p_i}, \boldsymbol{E}^A_{p_i}, \boldsymbol{E}^S_{p_i} \in \mathbb{R}^D$ are the corresponding semantic feature of three kinds of prompts, $\mathcal{H}$ is the process of extracting the last hidden state from the LLM, and D is the dimension size of the hidden state vector.

Embedding Enhancement

Dual Feature Alignment leverages the intricate connections between address and visit patterns, as well as between address and surrounding environment to obtain higher-quality semantic features. *Semantic Feature Fusion* uses attention score-weighted merging to ensure the quality of the features when fusing the semantic features into a single semantic vector. Afterward, *Cross Attention Enhancement*, based on the cross-attention method, employs the semantic vector obtained earlier to fully integrate and enhance the POI representations, resulting in the final output vector.

Dual Feature Alignment A POI's address, a key factor of geography information, is closely linked to its visit patterns and surrounding environment. For example, as shown in Tab. 1, the New York Stock Exchange is on Wall Street, a well-known hub of financial firms. People often visit there during daytime working hours, and the surrounding environment mainly consists of office spaces. If we align the address textual feature with the visit pattern and surrounding environment textual features, we can obtain higher-quality textual information. Thus, we designed Dual Feature Alignment. First, Given a batch of textual information of n POIs $\{E^V, E^A, E^S\}$, they will be fed into a linear layer to transform them into a hidden space with the same dimension as the POI embedding to be enhanced, denoted as:

$$\tilde{\boldsymbol{E}}^V = \boldsymbol{W}^{V'}\boldsymbol{E}^V,\ \tilde{\boldsymbol{E}}^A = \boldsymbol{W}^A\boldsymbol{E}^A,\ \tilde{\boldsymbol{E}}^S = \boldsymbol{W}^S\boldsymbol{E}^S, \quad (2)$$

where $\tilde{\boldsymbol{E}}^V, \tilde{\boldsymbol{E}}^A, \tilde{\boldsymbol{E}}^S \in \mathbb{R}^{n\times d}$, d is the dimension of the hidden space and $\boldsymbol{W}^{V'}, \boldsymbol{W}^A, \boldsymbol{W}^S$ are all learnable matrices.

Next, to leverage the relationships between textual features and obtain higher-quality information, multiple layers of the Transformer encoder are introduced. Each layer consists of multi-head attention (MHA), residual connections,

and layer normalization operations (LN) and the number of layers is L_1. Formally, take the relation between address and visit patterns as an example, given the vectors $\{\tilde{E}^V, \tilde{E}^A\}$, we computed a MHA as follows:

$$\boldsymbol{Q} = \tilde{\boldsymbol{E}}^V \boldsymbol{W}^Q,\ \boldsymbol{K} = \tilde{\boldsymbol{E}}^A \boldsymbol{W}^K,\ \boldsymbol{V} = \tilde{\boldsymbol{E}}^A \boldsymbol{W}^V, \tag{3}$$

$$head_h = \phi(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}})\boldsymbol{V}, \tag{4}$$

$$\mathrm{MHA}(\tilde{\boldsymbol{E}}^V, \tilde{\boldsymbol{E}}^A) = (\|_{h=1}^{H} head_h)\boldsymbol{W}^O, \tag{5}$$

where $\boldsymbol{W}^Q, \boldsymbol{W}^K, \boldsymbol{W}^V \in \mathbb{R}^{d \times d_h}$ are learnable parameters, ϕ is softmax activation function, d_h is the dimension of a single head. And $\|$ is the concatenation operation, $\boldsymbol{W}^O \in \mathbb{R}^{(d_h \cdot H) \times d}$ is a trainable parameter and H denotes the number of heads. The output of the first layer Z_1' is denoted as:

$$\boldsymbol{Z} = \mathrm{LN}(\tilde{\boldsymbol{E}}^A + \mathrm{MHA}(\tilde{\boldsymbol{E}}^V, \tilde{\boldsymbol{E}}^A)), \tag{6}$$

$$\boldsymbol{Z}_1' = \mathrm{LN}(\boldsymbol{Z} + \mathrm{FFN}(\boldsymbol{Z})), \tag{7}$$

where FFN is a feed-forward neural network. Then, vector $\boldsymbol{Z}_1'$, along with $\boldsymbol{E}^A$, will be fed back into the next layer as input, and after repeating this process $L_1 - 1$ times, the final layer result $\boldsymbol{Z}_{L_1}'$ is the vector $\boldsymbol{E}^{A-V} \in \mathbb{R}^{n \times d}$. It should be noted that $\{Z_k' | k \in [1, L_1]\}$ is transformed into K and V, while $\tilde{E}^A$ is converted into $\boldsymbol{Q}$ in the subsequent repetition process. Similarly, to deal with the connection between address and surrounding by replacing $\tilde{\boldsymbol{E}}^V$ with $\tilde{\boldsymbol{E}}^S$ in Formula (3), (5) and (6). We can get the output E^{A-S} accordingly.

Semantic Feature Fusion Considering that visit patterns are related to the surrounding environment, for example, POIs near entertainment venues are mostly accessed on weekends. We got a comprehensive semantic feature by combining the two feature vectors from the previous module into one. To integrate two vectors into one while maintaining the quality of the vector, we designed the Semantic Feature Fusion based on a weighted sum method. Accordingly, the computation process can be represented as follows:

$$\begin{aligned} \theta^{A-V} &= \boldsymbol{W}_2 \cdot LeakyReLU([\boldsymbol{W}_1 \boldsymbol{E}^{A-V} || \boldsymbol{W}_1 \boldsymbol{E}^{A-S}]), \\ \theta^{A-S} &= W_2 \cdot LeakyReLU([\boldsymbol{W}_1 \boldsymbol{E}^{A-S} || \boldsymbol{W}_1 \boldsymbol{E}^{A-V}]), \end{aligned} \tag{8}$$

where θ^{A-V} and θ^{A-S} are the attention scores for $\boldsymbol{E}^{A-V}$ and $\boldsymbol{E}^{A-S}$. $\boldsymbol{W}_1 \in \mathbb{R}^{d \times d'}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{2d' \times 1}$ are used to project the features into the same hidden space and to transform them into attention weights, respectively. $LeakyReLU$ is an activation function, and d' is the dimension of the latent space. After that, a softmax activation is employed to get the normalized weight, and a weighted sum fusion of two semantic features is applied to get the output $E^{LLM} \in \mathbb{R}^{n \times d}$, which can be represented as:

$$[\omega^{A-V}, \omega^{A-S}] = \phi([\theta^{A-V}, \theta^{A-S}]), \tag{9}$$

$$\boldsymbol{E}^{LLM} = \omega^{A-V} \cdot \boldsymbol{E}^{A-V} + \omega^{A-S} \cdot \boldsymbol{E}^{A-S}. \tag{10}$$

Cross Attention Fusion Cross-attention techniques have been employed to fully fuse features from different views (Sun et al. 2024). Hence, inspired by (Yan et al. 2024), to enhance other embedding methods by making use of the vector E^{LLM}, a Cross Attention Fusion is developed.

Here, we also employ a multi-layer transformer encoder architecture but in each layer we use the multi-query attention (Shazeer 2019) plus parallel attention and feed-forward net (PAF) (Sonkar and Baraniuk 2023) to combine E^{LLM} and $E^{POI} \in \mathbb{R}^{n \times d}$. The multi-query attention (MQA) is almost the same as the multi-head attention except all heads share the same set of K and V, which is proved to be faster with minor quality degradation in the calculation. Additionally, PAF can be effective in improving the performance of transformer-based models. As shown in Fig. 1, the first layer of the Cross Attention Fusion can be presented formally as:

$$\boldsymbol{X} = \mathrm{LN}(\boldsymbol{E}^{POI} + MQA(\boldsymbol{E}^{LLM}, \boldsymbol{E}^{POI})), \tag{11}$$

$$\boldsymbol{X}_1' = \mathrm{LN}(\boldsymbol{X} + \mathrm{FFN}(\boldsymbol{X})), \tag{12}$$

Then, the vector $\boldsymbol{X}_1'$ and $\boldsymbol{E}^{POI}$ will be fed into the next layer, and after repeating this process in $\boldsymbol{L}_2 - 1$ times, the outcome of the last layer $\boldsymbol{E}_{FUSE}$ is obtained. It is worth to noticed that $\{\boldsymbol{X}_k' | k \in [1, L_2]\}$ is transformed into $\boldsymbol{K}$ and $\boldsymbol{V}$, and $\boldsymbol{E}^{LLM}$ is converted into $\boldsymbol{Q}$ in the following repetition.

Multi-View Contrastive Learning

Our Multi-View Contrastive Learning approach is designed to enhance the similarity between the anchor POI and positive samples, while simultaneously reducing the similarity with negative samples. This strategy aims to strengthen the robustness and effectiveness of the embedding vector. However, Unlike previous works that only use distance as the sampling criterion (Li et al. 2023), we incorporated temporal, spatial, and functional views into our considerations and designed three sampling strategies. Besides, the formal definitions of the following three sampling strategies are presented in the Supplementary Material.

Sequence-Time Contrastive Learning The visit context of a POI *i.e.,* the neighboring check-in records in the check-in record sequence is often considered an important factor. However, if the duration of a check-in record sequence is very long, two adjacent consecutive check-in records may be separated by several days. Considering such neighbors as positive samples will reduce the effectiveness of contrastive learning. Therefore, to avoid this situation, we propose a Sequence-Time sampling strategy. The positive samples are required not only to be neighbors of the check-in record but also to have the same visit date as the anchor sample.

Geography Contrastive Learning From a spatial perspective, our strategy incorporates both local spatial similarity and category similarity as criteria. Specifically, for a given POI, we define a square area centered around it and consider POIs of the same category in that area as positive samples.

Functional Contrastive Learning Apart from the two types of contrastive learning mentioned above, we aim to identify groups of POIs that are similar in social function. Therefore, based on the category and visit patterns of POIs, We propose the following principle for selecting positive samples: only POIs that share the same category and visit pattern as the anchor sample are regarded as positive samples.

In summary, based on the above three criteria, we sampled more high-quality positive samples for subsequent contrastive learning training. This approach helps enhance the comprehensive capability of the representation vectors.

Model Training

Given a POI p_i and a set of all the POIs P, we derive its positive set P_i^+ through the above strategies. And for each pair in $\{(p_i, p_i^+)|p_i^+ \in P_i^+\}$, we will randomly choose $m-2$ negative samples from the negative set $\{p_i^-|p_i^- \in P_i^-, P_i^- = P - p_i - P_i^+\}$, to form a training batch.

Then we use InfoNCE as the loss for contrastive learning:

$$\mathcal{L}_{Cont} = -log\frac{e^{\frac{1}{\gamma}sim(p_i,p_i^+)}}{\sum_{i=0}^{m} e^{\frac{1}{\gamma}sim(p_i,p_i^-)}} \tag{13}$$

where $sim(\cdot,\cdot)$ is a similarity measure function, γ is a temperature parameter and m is number of POIs in the batch.

Apart from this, in order to maintain the similarity between the origin vectors and enhanced vectors, a loss based on cosine similarity is constructed, which can be defined as:

$$\mathcal{L}_{Sim} = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}|cos(E_i^{FUSE}, E_j^{FUSE}) - cos(E_i^{POI}, E_j^{POI})|, \tag{14}$$

where cos is the cosine similarity function.

Ultimately, the loss of POI-Enhancer can be denoted:

$$\mathcal{L} = \mathcal{L}_{Cont} + \mathcal{L}_{Sim}. \tag{15}$$

4 Experiments

Experiment Setup

Datasets We conducted experiments on three check-in datasets provided by (Yang et al. 2014): Foursquare-NY, Foursq-SG, and Foursquare-TKY, sampled from New York, Singapore, and Tokyo, respectively. We remove all POIs with less than 5 check-ins in the dataset and check-in sequences with less than 10 POIs. The statistics of the processed dataset are in Supplementary Material. Then we shuffled the dataset and split it into a ratio of 2:1:7 for the test set, validation set, and training set. It should be noted that the training set for the POI Recommendation task will also be used as the dataset for sampling in contrastive learning.

Baselines

• **Skip-Gram** (Tomas et al. 2013), a simple variant of Word2Vec model, and is widely used in sequential tasks.

• **POI2Vec** (Feng et al. 2017), a latent representation model for both POI and user, which introduces geographical information into output via splitting the map into rectangle regions and building binary tree on it.

• **Geo-Teaser** (Zhao et al. 2017), an embedding model based on Skip-Gram, but take temporal influence and neighboring locations in the trajectory dataset into consideration and design two separate loss functions for them.

• **TALE** (Wan et al. 2021), a POI embedding pre-training method based on CBOW, constructing a temporal tree structure based on user trajectories to acquire time awareness.

• **Hier** (Shimizu, Yabe, and Tsubouchi 2020), a method to obtain fine-grained place embedding via extracting spatial information from trajectory dataset in a hierarchical way.

• **CTLE** (Lin et al. 2021), the state-of-the-art model of POI representation which incorporates both temporal and spatial information of user activities into its embedding.

LLM-based baselines are also included like Llama2 (Touvron et al. 2023), ChatGLM2 (GLM et al. 2024), and GPT-2 (Radford et al. 2019).

Downstream Tasks & Metrics To evaluate POI-Enhancer and make a comprehensive comparison, we set up three downstream tasks based on LibCity (Wang et al. 2021a).

• **POI Recommendation**, Based on a user's historical check-in sequence, the POI Recommendation task aims to predict the next POI the user would visit.

• **Check-in Sequence Classification**, Given an arbitrary check-in sequence, this task requires the downstream model to detect which user this sequence belongs to.

• **POI Visitor Flow Prediction**, POI visitor flow prediction requires the downstream model to forecast the future volume of visitor flow based on historical visitor data.

In the POI Recommendation task, we use Hit@k as the evaluation metric (value equals 1 if the ground truth is among the top k in the recommendation list, otherwise 0, $k = 1,\ 5$). The Check-in Sequence Classification task is evaluated using Accuracy (ACC) and Macro-F1 while the Visitor Flow Prediction task is assessed with Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

Implementation Details In our framework, we use the Llama-2-7B as the LLM backbone. The complete implementation details are provided in the Supplementary Material.

Overall Result Analysis

The result of downstream tasks is presented in Tab. 2, demonstrating that POI-Enhancer significantly improved the performance of all baselines across all datasets. Skip-Gram and POI2Vec incorporate spatial information differently: Skip-Gram uses co-occurrence frequencies, while POI2Vec employs a geographic binary tree, both ignoring temporal features. Geo-Teaser includes spatial and temporal data with coarse granularity, while TALE, Hier, and CTLE integrate finer-grained spatiotemporal data. However, all six methods overlook POI semantic knowledge. Our framework addresses this gap, significantly enhancing performance.

Judging from the enhancement effects on downstream tasks, POI-Enhancer shows the most significant improvement in the Check-in Sequence Classification task. This could be because the textual knowledge provided by POI-Enhancer is more beneficial for handling classification tasks. And among the datasets, New York achieved the highest performance boost, while Tokyo showed the least improvement. The weaker results for Tokyo may be due to the fewest contrastive learning samples and the LLM's unfamiliarity with Japanese, resulting in lower-quality information extraction.

For the POI Recommendation task, POI2Vec achieves the greatest improvement on the New York dataset, with both metrics increasing by over 20%. This is due to its focus on capturing check-in sequence patterns and spatial similarity while neglecting other modalities. And POI-Enhancer compensates for these limitations by enriching textual knowledge, leading to excellent performance. As for the Check-in Sequence Classification task, our findings indicate that Skip-Gram shows the weakest improvement. This is likely because Skip-Gram focuses on modeling representations from

Task	POI Recommendation						Check-in Sequence Classification						POI Visitor Flow Prediction					
Dataset	NY		TKY		SG		NY		TKY		SG		NY		TKY		SG	
Metric	Hit@1	Hit@5	Hit@1	Hit@5	Hit@1	Hit@5	Acc	F1	Acc	F1	Acc	F1	MAE	RMSE	MAE	RMSE	MAE	RMSE
Skip-Gram	6.984	17.356	15.037	33.305	9.194	23.652	48.967	0.224	59.982	0.413	43.768	0.229	0.371	0.747	0.494	0.691	0.665	0.994
Skip-Gram+	7.610	18.032	15.557	34.197	10.747	24.468	52.151	0.251	62.936	0.438	47.285	0.255	0.336	0.514	0.492	0.668	0.621	0.890
Impr.	**8.96%**	**3.89%**	**3.46%**	**2.68%**	**16.89%**	**3.45%**	**6.5%**	**12.07%**	**4.92%**	**5.97%**	**8.04%**	**11.37%**	**9.43%**	**31.14%**	**0.47%**	**3.31%**	**6.66%**	**10.45%**
POI2Vec	6.417	14.684	15.195	33.214	8.828	21.729	32.057	0.113	51.499	0.331	31.736	0.139	0.343	0.574	0.531	0.764	0.625	0.918
POI2Vec+	7.851	18.353	15.800	34.768	10.630	24.030	52.151	0.245	62.358	0.438	46.521	0.264	0.326	0.492	0.490	0.696	0.602	0.868
Impr.	**22.35%**	**24.99%**	**3.98%**	**4.68%**	**20.41%**	**10.59%**	**62.68%**	**117.35%**	**21.09%**	**32.39%**	**46.59%**	**89.32%**	**4.78%**	**14.27%**	**7.8%**	**8.99%**	**3.66%**	**5.36%**
Geo-Teaser	6.174	15.355	14.956	33.814	8.768	22.851	38.296	0.149	54.852	0.355	39.511	0.182	0.394	0.778	0.498	0.696	0.623	0.913
Geo-Teaser+	7.116	16.657	15.500	34.475	10.122	23.532	49.910	0.233	62.647	0.437	50.064	0.279	0.341	0.524	0.483	0.669	0.588	0.854
Impr.	**15.27%**	**8.48%**	**3.64%**	**1.95%**	**15.45%**	**2.98%**	**30.33%**	**55.84%**	**14.21%**	**23.11%**	**26.71%**	**52.98%**	**13.35%**	**32.64%**	**3.07%**	**3.87%**	**5.57%**	**6.41%**
TALE	6.025	13.618	13.608	30.612	7.555	19.238	33.950	0.127	51.521	0.330	33.112	0.140	0.336	0.645	0.523	0.716	0.639	0.926
TALE+	6.690	15.208	14.940	33.223	8.694	20.342	50.689	0.240	63.380	0.448	47.719	0.263	0.320	0.482	0.510	0.701	0.610	0.903
Impr.	**11.04%**	**11.67%**	**9.79%**	**8.53%**	**15.08%**	**5.74%**	**49.3%**	**88.82%**	**23.02%**	**35.82%**	**44.11%**	**87.29%**	**4.76%**	**25.17%**	**2.49%**	**2.08%**	**4.56%**	**2.49%**
Hier	6.982	15.631	15.120	32.091	9.181	22.174	37.436	0.143	50.189	0.316	41.269	0.196	0.361	0.584	0.536	0.733	0.634	1.000
Hier+	8.009	19.197	16.187	35.715	10.592	24.079	51.893	0.254	63.380	0.441	47.795	0.258	0.313	0.483	0.510	0.719	0.574	0.804
Impr.	**14.72%**	**22.81%**	**7.06%**	**11.29%**	**15.37%**	**8.59%**	**38.62%**	**77.33%**	**26.28%**	**39.59%**	**15.81%**	**31.5%**	**13.09%**	**17.33%**	**4.88%**	**1.91%**	**9.39%**	**19.58%**
CTLE	6.653	14.594	14.859	31.852	8.625	20.218	40.103	0.181	55.030	0.369	41.805	0.206	0.337	0.566	0.515	0.703	0.697	1.061
CTLE+	7.093	17.032	15.479	34.138	10.315	24.027	50.430	0.234	61.848	0.434	51.440	0.287	0.291	0.456	0.495	0.689	0.610	0.892
Impr.	**6.61%**	**16.71%**	**4.18%**	**7.18%**	**19.59%**	**18.84%**	**25.75%**	**29.59%**	**12.39%**	**17.36%**	**23.05%**	**39.62%**	**13.65%**	**19.44%**	**4.01%**	**2.02%**	**12.4%**	**15.92%**

Table 2: The overall performance of downstream tasks and (+) means being enhanced by POI-Enhancer. Hit@1, Hit@5 and Acc are in %, and F1 means macro-F1. For MAE and RMSE, lower is better, while for other metrics, higher is better.

user trajectories, which limits the potential for improvement in the Check-in Sequence Classification task after enhancement. In the POI Visitor Flow Prediction task, CTLE shows significant improvement and strong performance after enhancement. CTLE effectively captures contextual neighbors and temporal information in trajectories, and when combined with the high-quality textual information extracted by POI-Enhancer, it greatly improves the performance of representation vectors in complex flow prediction tasks.

Besides, comparison experiments with LLM-based baselines reveal that, with the aid of POI-Enhancer, the POI representation method still holds a considerable advantage. This advantage stems from the fact that the POI representation method captures the fundamental spatial-temporal features, and when further enhanced with textual knowledge, it outperforms the text-centric LLM-based baselines. The results of this experiment are in the Supplementary Material.

Further Analysis on POI-Enhancer

Ablation Experiment In this subsection, we conduct comprehensive experiments with four variant settings to evaluate the effectiveness of the components we design:

• **POI-Enhancer/P** We remove the special prompt design including the role-playing, the attribute headers, and the question. • **POI-Enhancer/D** We removed the Dual Feature Alignment and Semantic Embedding Fusion. Instead, we generated a single prompt, which accumulates the content of the previous three kinds of prompts while maintaining the same format. The features extracted from this prompt by the LLM will be directly input into the Cross Attention Fusion. • **POI-Enhancer/F** We remove Cross Attention Fusion and concatenate the E_{POI} and E_{LLM} as the final vector instead. • **POI-Enhancer/C** We only consider the spatial perspective. Specifically, given a POI, we define a square area centered around it to collect positive samples, with the parameters consistent with Geography Contrastive Learning.

We tested them on three downstream tasks using the New York dataset, with Hit@1, ACC, and MAE as evaluation metrics. As shown in the Fig. 2, POI-Enhancer outperforms

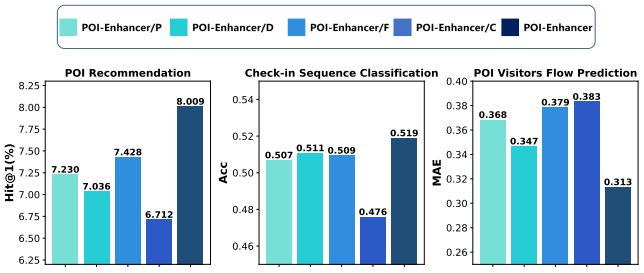

Figure 2: The result of ablation experiment.

all variant settings and we can draw the following conclusions: (1) The specialized prompts can enhance the framework's performance because they stimulate the LLM to extract spatial-temporal knowledge more efficiently. (2) The Dual Feature Alignment and the Semantic Feature Fusion help obtain and maintain high-quality semantic vectors and improve the capabilities of the POI representation. (3) The Cross Attention Fusion enables a more thorough integration, allowing the final vector to capture richer semantic information, resulting in improved performance. (4) Compared to distanced-based positive samples, Multi-View Contrastive Learning selects richer samples from different perspectives, enhancing the ability of the embedding vectors.

Parameters Analysis In this subsection, we study the effect of different L_1 and L_2 parameter settings in our framework. Specifically, we focus on enhancing the Hier model using the New York dataset, with POI recommendation as the downstream task. When evaluating the impact of one parameter, we keep the other parameters fixed at their optimal values. As shown in the Fig. 3, we can observe that for both L_1 and L_2, the performance initially improves with the increasing number of layers, reaches optimal performance, and then deteriorates. So, in other experiments, we set L_1 to 4 and L_2 to 2. On the one hand, this indicates that when L_1 is too low, our alignment method fails to fully utilize the relational information between features, while an excessively

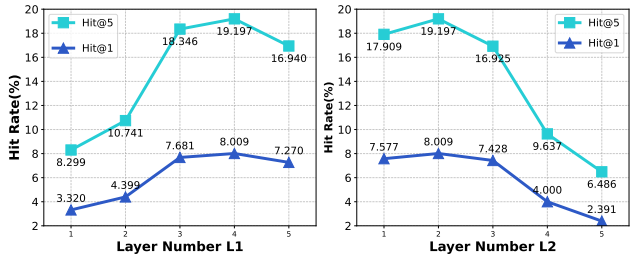

Figure 3: The effect of L_1 and L_2.

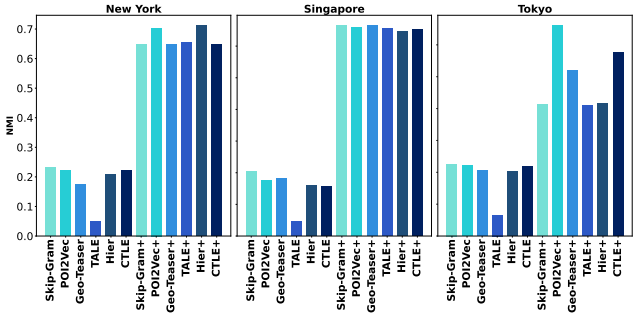

Figure 4: The result of POI cluster task.

high number of L_1 layers tends to cause over-fitting. On the other hand, this suggests that when L_2 is below the optimal value, our fusion method fails to effectively incorporate the knowledge from LLM into the original representations. However, when L_2 exceeds a certain threshold, the semantic knowledge will overshadow the original vectors.

Quality Analysis To further evaluate the quality of the enhanced vectors produced by POI-Enhancer, we conducted clustering tasks using the K-means algorithm on three datasets. We applied this algorithm to all types of representation vectors, both before and after enhancement. The number of clusters was set to match the number of POI categories in each dataset. We then assessed the clustering performance using the Normalized Mutual Information (NMI) metric. The results depicted in the Fig. 4 demonstrate the effectiveness of our framework, as all evaluation metrics for the representation vectors across the three datasets have shown significant improvement. This indicates that: (1) We successfully extracted high-quality textual features, and the rich textual information helps similar representation vectors to cluster more closely together. (2) We effectively integrated textual information into the initial representations, further enhancing the quality of the original vectors. (3) The Multi-View Contrastive Learning approach encouraged vectors of the same class to be closer together while pushing vectors of different classes further apart.

5 Related Work

LLMs in Spatial-temporal Tasks Considerable efforts have been dedicated to using LLMs to improve the performance of spatial-temporal tasks (Yu et al. 2024). For instance, GeoGPT (Zhang et al. 2023a) introduced an LLM-based tool capable of automating the processing of geographic data, but it does not delve into extracting detailed information about locations. GEOLLM (Manvi et al. 2023) designed prompts that include coordinates, address, and surrounding buildings, but it can only address simple questions in a Q&A format and are unable to handle complex tasks like POI recommendation. Besides, they fail to fully extract the semantic information of POIs. Some researchers have used LLMs as backbones to tackle complex real-world tasks. For example, GATGPT (Chen, Wang, and Xu 2023) input spatial-temporal features into a frozen LLM to predict traffic speeds, while ST-LLM (Liu et al. 2024a) used a partially frozen LLM to forecast traffic flow. However, these methods are designed for specific individual problems and cannot be applied across multiple tasks. To solve these limitations, we designed three types of special prompts to extract the semantic information of POIs from LLMs effectively.

POI Representation with Semantic Information POI representation aims to turn each POI into a vector that can be utilized in various downstream tasks like traffic forecasting tasks (Zhang et al. 2023b; Jiang et al. 2023a; Ji et al. 2023) and trajectory tasks (Zhu et al. 2024; Jiang et al. 2023c,d). Most existing methods like (Dai et al. 2022), leverage textual features typically using one-hot code to encode POI categories and then concatenate them with the embedding vectors. For data types like check-in content, (Xie et al. 2016) model the similarity between POIs by constructing a POI-Word relationship graph, while (Chang et al. 2018) draws inspiration from Word2Vec method, simultaneously training word vectors and POI vectors. However, these methods often fall short in preserving semantic information and achieving a more comprehensive integration during the fusion process. To address this issue, we designed three modules within the Embedding Enhancement stage, aimed at improving the preservation and integration of semantic information in the POI embedding vectors.

6 Conclusion and Future Work

We propose a framework called POI-Enhancer, which enhances all POI representation methods by leveraging the LLM. To introduce textual information into POI representations, we designed three specialized prompts to extract features from the LLM. To utilize the relationships between address features and other features for improving representation quality, we introduced Dual Feature Alignment and Semantic Feature Fusion, which help obtain and preserve high-quality textual features. For better integration of the extracted knowledge into POI representations, we further developed the Cross Attention Fusion. Lastly, to enhance the generalization and representation capabilities of the vectors, we proposed Multi-View Contrastive Learning, using three different strategies to sample positive and negative examples. The experiment results demonstrate that our framework consistently and significantly improves the performance of POI representation vectors across various downstream tasks in three real-world datasets. Further ablation experiments and quality analysis experiments also reveal the effectiveness of our module design and the robustness and generalization of the output results.

7 Acknowledgments

Prof. Jingyuan Wang's work was partially supported by the National Natural Science Foundation of China (No. 72171013, 72222022, 72242101), the Special Fund for Health Development Research of Beijing (2024-2G-30121) and State Key Laboratory of Complex & Critical Software Environment (SKLSDE-2023ZX-04). Prof. Xiangyu Zhao's work was partially supported by Research Impact Fund (No.R1015-23), APRC - CityU New Research Initiatives (No.9610565, Start-up Grant for New Faculty of CityU), CityU - HKIDS Early Career Research Grant (No.9360163), Hong Kong ITC Innovation and Technology Fund Midstream Research Programme for Universities Project (No.ITS/034/22MS), Hong Kong Environmental and Conservation Fund (No. 88/2022), and SIRG - CityU Strategic Interdisciplinary Research Grant (No.7020046), Huawei (Huawei Innovation Research Program), Tencent (CCF-Tencent Open Fund, Tencent Rhino-Bird Focused Research Program), Ant Group (CCF-Ant Research Fund, Ant Group Research Fund), Alibaba (CCF-Alimama Tech Kangaroo Fund No. 2024002), CCF-BaiChuan-Ebtech Foundation Model Fund, and Kuaishou.

References

Balsebre, P.; Huang, W.; Cong, G.; and Li, Y. 2023. Cityfm: City foundation models to solve urban challenges. *arXiv preprint arXiv:2310.00583*.

Chang, B.; Park, Y.; Park, D.; Kim, S.; and Kang, J. 2018. Content-aware hierarchical point-of-interest embedding model for successive poi recommendation. In *IJCAI*, volume 20, 3301–3307.

Chen, Y.; Wang, X.; and Xu, G. 2023. Gatgpt: A pre-trained large language model with graph attention network for spatiotemporal imputation. *arXiv preprint arXiv:2311.14332*.

Dai, S.; Yu, Y.; Fan, H.; and Dong, J. 2022. Spatio-temporal representation learning with social tie for personalized POI recommendation. *Data Science and Engineering*, 7(1): 44–56.

Ding, X.; Chen, L.; Gao, Y.; Jensen, C. S.; and Bao, H. 2018. UlTraMan: A unified platform for big trajectory data management and analytics. *Proceedings of the VLDB Endowment*, 11(7): 787–799.

Feng, S.; Cong, G.; An, B.; and Chee, Y. M. 2017. Poi2vec: Geographical latent representation for predicting future visitors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

GLM, T.; Zeng, A.; Xu, B.; Wang, B.; Zhang, C.; Yin, D.; Zhang, D.; Rojas, D.; Feng, G.; Zhao, H.; et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Golkar, S.; Pettee, M.; Eickenberg, M.; Bietti, A.; Cranmer, M.; Krawezik, G.; Lanusse, F.; McCabe, M.; Ohana, R.; Parker, L.; et al. 2023. xval: A continuous number encoding for large language models. *arXiv preprint arXiv:2310.02989*.

Gurnee, W.; and Tegmark, M. 2024. Language Models Represent Space and Time. arXiv:2310.02207.

Han, X.; Zhou, D.-X.; Shen, G.; Kong, X.; and Zhao, Y. 2024. Deep Trajectory Recovery Approach of Offline Vehicles in the Internet of Vehicles. *IEEE Transactions on Vehicular Technology*, 73(11): 16051–16062.

Ji, J.; Wang, J.; Huang, C.; Wu, J.; Xu, B.; Wu, Z.; Zhang, J.; and Zheng, Y. 2023. Spatio-temporal self-supervised learning for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 4356–4364.

Ji, J.; Wang, J.; Jiang, Z.; Jiang, J.; and Zhang, H. 2022a. STDEN: Towards physics-guided neural networks for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 4048–4056.

Ji, J.; Wang, J.; Wu, J.; Han, B.; Zhang, J.; and Zheng, Y. 2022b. Precision CityShield against hazardous chemicals threats via location mining and self-supervised learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3072–3080.

Jiang, J.; Han, C.; Zhao, W. X.; and Wang, J. 2023a. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 4365–4373.

Jiang, J.; Pan, D.; Ren, H.; Jiang, X.; Li, C.; and Wang, J. 2023b. Self-supervised trajectory representation learning with temporal regularities and travel semantics. In *2023 IEEE 39th international conference on data engineering (ICDE)*, 843–855. IEEE.

Jiang, J.; Pan, D.; Ren, H.; Jiang, X.; Li, C.; and Wang, J. 2023c. Self-supervised trajectory representation learning with temporal regularities and travel semantics. In *2023 IEEE 39th international conference on data engineering (ICDE)*, 843–855. IEEE.

Jiang, W.; Zhao, W. X.; Wang, J.; and Jiang, J. 2023d. Continuous trajectory generation based on two-stage GAN. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4374–4382.

Li, Y.; Huang, W.; Cong, G.; Wang, H.; and Wang, Z. 2023. Urban region representation learning with openstreetmap building footprints. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1363–1373.

Li, Z.; Xia, L.; Tang, J.; Xu, Y.; Shi, L.; Xia, L.; Yin, D.; and Huang, C. 2024. Urbangpt: Spatio-temporal large language models. *arXiv preprint arXiv:2403.00813*.

Lin, Y.; Wan, H.; Guo, S.; and Lin, Y. 2021. Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4241–4248.

Liu, C.; Yang, S.; Xu, Q.; Li, Z.; Long, C.; Li, Z.; and Zhao, R. 2024a. Spatial-temporal large language model for traffic prediction. *arXiv preprint arXiv:2401.10134*.

Liu, Z.; Wang, J.; Li, Z.; and He, Y. 2024b. Full Bayesian Significance Testing for Neural Networks in Traffic Forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*.

Manvi, R.; Khanna, S.; Mai, G.; Burke, M.; Lobell, D.; and Ermon, S. 2023. Geollm: Extracting geospatial knowledge from large language models. *arXiv preprint arXiv:2310.06213*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners.

Shazeer, N. 2019. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*.

Shimizu, T.; Yabe, T.; and Tsubouchi, K. 2020. Enabling finer grained place embeddings using spatial hierarchy from human mobility trajectories. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, 187–190.

Song, C.; Lin, Y.; Guo, S.; and Wan, H. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 914–921.

Sonkar, S.; and Baraniuk, R. G. 2023. Investigating the Role of Feed-Forward Networks in Transformers Using Parallel Attention and Feed-Forward Net Design. *arXiv preprint arXiv:2305.13297*.

Sun, F.; Qi, J.; Chang, Y.; Fan, X.; Karunasekera, S.; and Tanin, E. 2024. Urban region representation learning with attentive fusion. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 4409–4421. IEEE.

Tomas, M.; Kai, C.; Greg, C.; and Jeffrey, D. 2013. Efficient estimation of word representations in vector space. arXiv:1301.3781v3.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wan, H.; Lin, Y.; Guo, S.; and Lin, Y. 2021. Pre-training time-aware location embeddings from spatial-temporal trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 34(11): 5510–5523.

Wang, J.; He, X.; Wang, Z.; Wu, J.; Yuan, N. J.; Xie, X.; and Xiong, Z. 2018. CD-CNN: a partially supervised cross-domain deep learning model for urban resident recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Wang, J.; Ji, J.; Jiang, Z.; and Sun, L. 2022. Traffic flow prediction based on spatiotemporal potential energy fields. *IEEE Transactions on Knowledge and Data Engineering*, 35(9): 9073–9087.

Wang, J.; Jiang, J.; Jiang, W.; Li, C.; and Zhao, W. X. 2021a. Libcity: An open library for traffic prediction. In *Proceedings of the 29th international conference on advances in geographic information systems*, 145–148.

Wang, J.; Lin, X.; Zuo, Y.; and Wu, J. 2021b. DGeye: Probabilistic risk perception and prediction for urban dangerous goods management. *ACM Transactions on Information Systems (TOIS)*, 39(3): 1–30.

Wang, J.; Wang, X.; and Wu, J. 2018. Inferring metapopulation propagation network for intra-city epidemic control and prevention. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 830–838.

Wu, N.; Wang, J.; Zhao, W. X.; and Jin, Y. 2019. Learning to effectively estimate the travel time for fastest route recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1923–1932.

Xie, M.; Yin, H.; Wang, H.; Xu, F.; Chen, W.; and Wang, S. 2016. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, 15–24.

Yan, Y.; Wen, H.; Zhong, S.; Chen, W.; Chen, H.; Wen, Q.; Zimmermann, R.; and Liang, Y. 2024. Urbanclip: Learning text-enhanced urban region profiling with contrastive language-image pretraining from the web. In *Proceedings of the ACM on Web Conference 2024*, 4006–4017.

Yang, D.; Zhang, D.; Zheng, V. W.; and Yu, Z. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1): 129–142.

Yu, X.; Wang, J.; Yang, Y.; Huang, Q.; and Qu, K. 2024. BIGCity: A Universal Spatiotemporal Model for Unified Trajectory and Traffic State Data Analysis. *arXiv preprint arXiv:2412.00953*.

Zhang, Y.; Wei, C.; Wu, S.; He, Z.; and Yu, W. 2023a. GeoGPT: understanding and processing geospatial tasks through an autonomous GPT. *arXiv preprint arXiv:2307.07930*.

Zhang, Z.; Huang, Z.; Hu, Z.; Zhao, X.; Wang, W.; Liu, Z.; Zhang, J.; Qin, S. J.; and Zhao, H. 2023b. MLPST: MLP is All You Need for Spatio-Temporal Prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 3381–3390.

Zhao, P.; Luo, A.; Liu, Y.; Xu, J.; Li, Z.; Zhuang, F.; Sheng, V. S.; and Zhou, X. 2020. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34(5): 2512–2524.

Zhao, S.; Zhao, T.; King, I.; and Lyu, M. R. 2017. Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In *Proceedings of the 26th international conference on world wide web companion*, 153–162.

Zhu, Y.; Ye, Y.; Wu, Y.; Zhao, X.; and Yu, J. 2023. Synmob: Creating high-fidelity synthetic gps trajectory dataset for urban mobility analysis. *Advances in Neural Information Processing Systems*, 36: 22961–22977.

Zhu, Y.; Yu, J. J.; Zhao, X.; Liu, Q.; Ye, Y.; Chen, W.; Zhang, Z.; Wei, X.; and Liang, Y. 2024. Controltraj: Controllable trajectory generation with topology-constrained diffusion model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4676–4687.

Technical Appendix

POI Attributes Preprocess

Visit Pattern of POI We conduct a statistical analysis of the check-in data to identify the visit patterns of a POI, which consists of the weekly visit pattern and daily visit pattern. On the one hand, we first divide the week into weekdays (Monday to Friday) and weekends (Saturday and Sunday). Whether a POI is visited more during the weekdays or weekends will determine its weekly visit pattern. On the other hand, We part every day into seven time periods: early morning (6 to 9 AM), morning (9 to 11 AM), noon (11 AM to 1 PM), afternoon (1 PM to 5 PM), evening (5 to 7 PM), night (7 to 12 PM), and midnight (0 to 6 AM). Afterward, we categorize each POI's check-in records into the corresponding time slot. Finally, the time slot with the highest number of check-in records will represent the daily visit pattern. Take the POI from Tab. 1 as an example: its weekly visit pattern is weekdays, and its daily visit pattern is 6 to 9 AM.

Address of POI Address features are essential for a POI. Typically, a POI's address includes the street name, house number, and postal code. By utilizing geographic revere search API provided by Nominatim [1], we can input latitude and longitude coordinates to retrieve the corresponding address details. For each POI in the dataset, we leverage this API to query its address information as well as the basic attribute $Name$.

Surrounding of POI For any given POI, we survey the category of other POIs nearby and consider this as the surrounding attributes. Specifically, we search for all POIs within a square area centered on a certain POI and the square's side length is set to 0.5 km. Then, we count the number of categories within these POIs and sort them in descending order. Finally, we select the top three categories as its surrounding attribute. For example, the top three categories near the POI in Tab. 1 are office, building, and road, which are its surrounding attributes.

Multi-View Contrastive Learning

Sequence-Time Contrastive Learning A formal constraint is that given a check-in record sequence R and a check-in record $r = (u, p, t)$, for any $r' = (u, p', t')$ in the R, we have the following two requirements for r':

- $|index(r') - index(r)| \leq \lambda$, where $index$ is a function indicating the position in R, and λ is a threshold.
- $date(r') = date(r)$, where $date$ is a function that extracts the date from a timestamp.

The p' in r' which meets the above requirements is considered a positive sample.

Geography Contrastive Learning Given a POI p, the positive samples p' have to meet the following two criteria:

- $Area(p)$ is defined as $Square(p, \tau)$, where $Square$ is a function that generates the corresponding square region with a side length of τ. The point p' lies within the $Area(p)$.
- $p'.c = p.c$, and c is the category attribute of a POI introduced in Section 2.

Functional Contrastive Learning Given a POI p, the positive samples p' must satisfy the following two conditions:

- $p'.visit\_pattern = p.visit\_pattern$, and $visit\_pattern$ is the visit pattern attribute of a POI.
- $p'.c = p.c$, where c is the category attribute of a POI.

[1] Nominatim: https://nominatim.openstreetmap.org

Dataset	#User	#POI	#Check-in
Foursquare-NY	15,171	24,118	641,005
Foursquare-SG	10,909	20,154	696,306
Foursquare-TKY	2,293	15,164	496,459

Table 3: Statistics of Datasets.

Experiments

In our experiments section, the training processes of the POI-Enhancer and all downstream tasks are implemented with Pytorch on the Nvidia RTX 3090 GPU. For LLM-based baselines, We use the POI Address Prompts to extract features from the LLM as representations.

Statistics of Processed Datasets After processing the data, the details of the datasets are presented in Tab.3.

Downstream Task Implementation In our framework, the dimension d is uniformly set to 256. Moreover, the number of encoder layers L_1 in Dual Feature Alignment and the number of encoder layers L_2 in Cross Attention Fusion is 4 and 2, respectively. The temperature parameter γ in InfoNCE is set to 0.1. We train POI-Enhancer for 100 epochs using a learning rate of 0.001 and the AdamW optimizer with a decay of 0.001.

• **POI Recommendation**, To evaluate this task, we train a two-layer LSTM model to process the input check-in sequence and feed the output into an MLP to make the prediction. Note that we cut long sequences in the dataset into slices with less than 128 check-in records to improve model efficiency.

• **Check-in Sequence Classification**, Similar to the POI Recommendation task, we use a two-layer LSTM connected to an MLP to classify check-in sequences.

• **POI Visitor Flow Prediction**, We set the time window to 1 hour and calculate the check-in count of every POI based on the check-in dataset. Due to the sparsity of the check-in dataset, we only select non-zero visitor flow sequences with a length greater than 5 when building the dataset. Then, we trained a Seq2seq model with MSE loss to do the prediction. Before input into the model, the visitor flow series is normalized and appended with two additional information, the embedding of the current POI and the hour of the day at which the series starts.

Parameter Settings

In POI-Enhancer, the distance τ is set to 0.5 km when obtaining POI Surrounding features. The feature dimension D extracted from the Llama2 and ChatGLM2 is 4096, while the feature dimension D extracted from GPT-2 is 768. Within each encoder layer of the Dual Feature Alignment

and Cross Attention Fusion, the number of attention heads H is set to 8, and the dimension of each head d_h is set to 32.

Besides, λ in Sequence-Time Contrastive Learning is set to 2 and the sampling distance of Geography Contrastive Learning τ is set to 0.5 km.

In the downstream task training, the hidden size of LSTM is uniformly set to 512. We use the Adam optimizer and a learning rate of 0.0001 to train all the downstream models for 100 epochs, except for the model of the POI Recommendation task, which has a learning rate of 0.001.

Performance of LLM-based baselines

Here, we utilize three LLM-based baselines including Llama2, ChatGLM2, and GPT-2. Specifically, we used the POI Address Prompts to extract features from the LLM as representations, just as we discussed in Section 3. Then, we compare the LLM-based baselines with the POI embedding models improved by POI-Enhancer. The result of this experiment is shown in Tab. 4.

Task	**POI Recommendation**						**Check-in Sequence Classification**						**POI Visitors Flow Prediction**					
Dataset	NY		TKY		SG		NY		TKY		SG		NY		TKY		SG	
Metric	Hit@1	Hit@5	Hit@1	Hit@5	Hit@1	Hit@5	Acc	F1	Acc	F1	Acc	F1	MAE	RMSE	MAE	RMSE	MAE	RMSE
Skip-Gram+	7.610	18.032	15.557	34.197	10.747	24.468	52.151	0.251	62.936	0.438	47.285	0.255	0.336	0.514	0.492	0.668	0.621	0.890
POI2Vec+	7.851	18.353	15.800	34.768	10.630	24.030	52.151	0.245	62.358	0.438	46.521	0.264	0.326	0.492	0.490	0.696	0.602	0.868
Geo-Teaser+	7.116	16.657	15.500	34.475	10.122	23.532	49.910	0.233	62.647	0.437	50.064	0.279	0.341	0.524	0.483	0.669	0.588	0.854
TALE+	6.690	15.208	14.940	33.223	8.694	20.342	50.689	0.240	63.380	0.448	47.719	0.263	0.320	0.482	0.510	0.701	0.610	0.903
Hier+	8.009	19.197	16.187	35.715	10.592	24.079	51.893	0.254	63.380	0.441	47.795	0.258	0.313	0.483	0.510	0.719	0.574	0.804
CTLE+	7.093	17.032	15.479	34.138	10.315	24.027	50.430	0.234	61.848	0.434	51.440	0.287	0.291	0.456	0.495	0.689	0.610	0.892
GPT-2	0.6639	2.7265	6.9243	17.3328	2.3632	8.0120	1.0327	0.0006	0.4664	0.0003	0.6628	0.0004	0.409	0.773	0.539	0.766	0.637	0.923
Llama2	1.6491	5.1294	8.0065	19.0356	3.9478	11.8564	1.0327	0.0005	1.3324	0.0039	0.9177	0.0019	0.389	0.694	0.520	0.738	0.639	0.901
ChatGLM2	4.2386	11.4755	12.4734	28.2362	5.4692	16.1045	17.1687	0.0487	31.3347	0.1680	11.5728	0.0372	0.439	0.795	0.536	0.738	0.629	0.952

Table 4: The performance of LLM-based baselines in downstream tasks.