TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 01/01 pp1–15
DOI: 10.26599/TST.2025.9010061
Volume 1, Number 1, September 2026

Pre-Training Location Representations via Spatial-Temporal Trajectory Subgraph Contrastive Learning

Hepeng Gao, Funing Yang*, Xingliang Zhang, Yijun Su, and Yongjian Yang

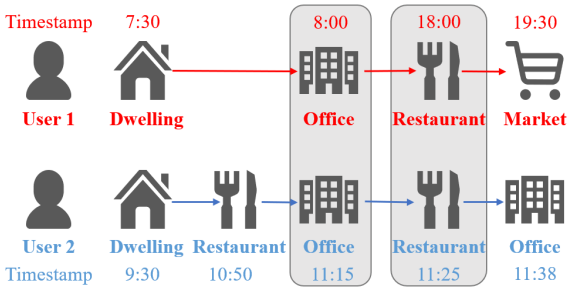
Abstract: Location representations from people's location-based service data are important and beneficial for urban downstream tasks. Locations usually have complex spatial-temporal contextual semantics, meaning that the same location has variable functionalities in different trajectories. Existing methods are mostly based on sequence or graph, where the former captures accurate temporal but limited spatial information (one trajectory), while the latter obtains a global spatial perspective (upstream and downstream nodes) but ignores the temporal information. Furthermore, the frequencies of visited locations are long-tail distributed, which is disadvantageous for infrequently visited locations. To that end, we propose a spatial-temporal trajectory subgraph contrastive learning framework entitled ST-TGCL, integrating comprehensive spatial-temporal information and relieving the long-tail issue with contrastive learning. Specifically, we construct contrastive trajectory subgraph pairs to stably learn variable functionalities and increase training opportunities for infrequently visited locations. To capture spatial-temporal contextual semantics, we design a trajectory network that formulates trajectories and a trajectory graph convolution network, which has the strengths of both sequence-based and graph-based models. Finally, we apply the location representations for downstream tasks to demonstrate our framework's effectiveness and generalization. ST-TGCL is evaluated over real-world datasets, and the results demonstrate that our framework significantly outperforms existing methods in location representation learning.

Key words: data mining; graph neural network; contrastive learning; location representation; trajectory

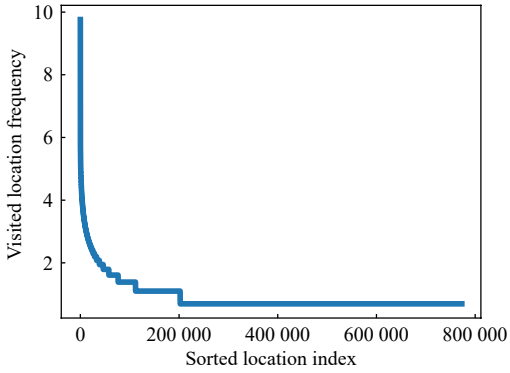
1 Introduction

Location representations have been widely used in a variety of urban tasks, such as point of interest (POI) recommendation[1], traffic flow prediction[2, 3], urban planning[4], urban safety[5, 6], etc. location-based service (LBS) data are generated in our daily lives, like GPS trajectories, check-ins at POIs, and cellular signaling records, which contain our behavior aims and location functionalities. Therefore, the trajectories are complex, where the same location in multiple trajectories has different contextual semantics. As shown in Fig.1a, the two users all visit the offices and restaurants. In their respective trajectories, User 1 goes to the work location and then gets off work to have dinner, while User 2 goes to the restaurant to pick up the takeaway and delivers it to the office. Meanwhile, as a real-world LBS dataset shows in Fig.1b, LBS data usually have a long-tailed distribution where numerous

• Hepeng Gao, Funing Yang, and Yongjian Yang are with Department of Computer Science and Technology, Jilin University, Changchun 130300, China. E-mail: gaohepeng13@hotmail.com; yfn@jlu.edu.cn; yyj@jlu.edu.cn.
• Xingliang Zhang is with China Mobile Communications Corporation Jilin Co., Ltd., Changchun 130300, China. E-mail: zhangxingliang@jl.chinamobile.com.
• Yijun Su is with JD iCity, Beijing 100000, China. E-mail: suyijun.ucas@gmail.com.
* To whom correspondence should be addressed.
Manuscript received: 2024-07-04; revised: 2024-09-03; accepted: 2025-04-02



(a) Trajectories of two users in daily life



(b) Distribution of a long-tailed dataset, namely a location-based service dataset

Fig. 1 Visited location pattern (we take the logarithm of the visited location frequency).

locations are visited infrequently, while a handful of locations are visited frequently. The trained model can be easily biased towards head locations with massive training data, leading to poor model performance on tail locations that have limited data.

By far, there are a variety of methods to learn location representations to address the above issues, among which the two most common options are sequence-based and graph-based methods. The two methods have their advantages for mining spatial-temporal data.

The sequence-based methods mainly capture temporal information from the LBS data, which arrange locations to generate a trajectory according to the visiting chronological order. Among them, some existing works[3] pre-train trajectories by using distributed word representations. The HrF-ZR[7] analyzes the location functionality by sequence embedding and clustering. With the success of self-supervised learning in many studies, sequence-based methods are suitable for the self-supervised learning framework. The CTLE[8] pre-trains representations by masking part locations of trajectories. The CityShield[5] generates contrastive samples in such a way that the same location is in multiple trajectories as a positive sample pair and vice versa for a negative sample pair. Sequence-based methods capture temporal information and pay attention to the spatial information of one trajectory. A training sample merely captures one functionality of the location, which is disadvantageous for stably learning representations of locations that have variable functionalities and designing self-supervised tasks.

In addition, the graph-based methods first construct graph structures that denote locations as nodes and rely on relations between locations to construct structures. The GLR[9] generates edges based on the rule that the transition time between the origin and destination locations is less than the set threshold, and divides the periods of the transition as the weight of edges. The GSD[10] constructs distance-based and transition-based graphs, i.e., distances and frequencies of origin-destination transitions. Target nodes could obtain all neighbors' information in one message passing and global information by stacked graph convolution layers, which capture the entire spatial semantics but ignore the temporal information of every trajectory.

In this paper, we propose a spatial-temporal trajectory subgraph contrastive learning framework (namely ST-TGCL) to learn location representations that are effective and general for downstream urban tasks. More specifically, our framework consists of three components: (1) Trajectory network generation is designed to denote the locations and the transfers between them. To address the shortcomings of the loss of temporal information in the graph-based methods and the limited spatial information in the sequence-based methods, the well-designed edges contain highly accurate and comprehensive temporal information similar to sequence data and the graph-based structure could obtain global spatial information. (2) Trajectory graph convolution network (TGCN) gets the best of both worlds, that is, the advantages of sequence-based and graph-based methods. It stacks temporal encoder layers and spatial graph convolution layers to integrate global spatial and comprehensive temporal information about locations and learn location representations. (3) Trajectory subgraph contrastive learning utilizes the spatial-temporal characteristics of trajectories to construct subgraph pairs as training samples, designs a multi-task loss function, and optimizes the parameters of TGCN. Our contributions are summarized as

follows:

- We design a novel network for trajectory, which formulates accurate and comprehensive spatial-temporal information about trajectories and constructs a graph structure to denote relations between locations.
- We design a trajectory graph convolution network as an encoder to stably capture complex spatial-temporal contextual semantics in trajectories and variable functionalities of locations and learn location representations.
- We propose a spatial-temporal trajectory subgraph contrastive learning framework that takes advantage of the spatial-temporal characteristics of trajectories to pre-train location representations and relieve the long-tail issue.
- We conduct experiments on real-world datasets. Evaluation results demonstrate that our framework, namely ST-TGCL, significantly outperforms existing sequence-based and graph-based methods, and relations between locations in representation space conform to our common sense.

2 Related Work

2.1 Sequence-based contrastive learning methods

Sequence-based contrastive learning methods have gained significant attention in the fields of urban computing and spatio-temporal machine learning. These methods leverage the temporal and spatial dependencies inherent in urban data to learn robust representations, which are crucial for various downstream tasks. One of the pioneering works in this area is CTLE[8], different from previous clustering and distributed word representation methods, which proposes a novel context- and time- aware location embedding method. This method design filling in the missing location based on the contextual location in a trajectory as a contrastive task to learning the location representations.

Furthermore, other notable sequence-based contrastive learning methods[11–16] have emerged, each addressing different challenges in urban computing. CCL[17] presents a novel curriculum contrastive learning framework for effectively modeling the sequential data, where a curriculum learning strategy conducts contrastive learning via an easy-to-difficult learning process. HCL[18] designs a novel hierarchical contrastive learning method for temporal point processes, which addresses overfitting and leads to unsatisfactory generalization power due to incomplete and sparse sequences that are common in practice. CoPPS[19] proposes a contrastive learning framework that aims to learn high-quality user representations by designing three data augmentation and contrastive learning strategies. The above methods address various challenges in urban computing, but they merely consider spatio-temporal dependencies between and within sequences. It may cause fluctuations in the model parameters and difficulty in convergence. In this paper, we integrate multiple sequences to generate subgraphs as contrastive instances, which allows for a more complete description of locations.

2.2 Graph-based contrastive learning methods

The main idea of contrastive learning is to make representations agree with each other under proper transformations and raises a recent surge of interest in representation learning and graph contrastive learning[20]. Existing methods[21–26] in graph contrastive learning can be categorized into two types: same-scale and cross-scale[27]. The former branch of methods discriminates graph samples on an equal scale, while the second type of method places the contrast across multiple granularities. For example, MICRO-Graph[28] proposes a learning-based sampling strategy to generate semantically informative subgraphs.

Recently, a series of studies have explored spatial-temporal representation learning with GNN[29, 30]. ST2Vec[31] proposes a trajectory representation architecture that considers spatial-temporal correlations between pairs of trajectories for similarity learning. ConST-CL[32] can effectively learn spatial-temporal representations by designing a region-based self-supervised task. This task requires the model to learn to transform sample representations from one view to another guided by context features. SPGCL[33] presents a general model for spatio-temporal learning that emphasizes the semantic relationships between monitored locations, and caters to graph construction when the deterministic adjacency is not available. AutoST[34] proposes a new region representation learning method with the automated spatio-temporal graph contrastive learning paradigm over the heterogeneous region graph generated from multi-view data sources. TKG-LDG[35] is an approach enhancing temporal knowledge graph for future entity prediction with long-term dense graph, modeling event evolution in an adaptive manner. Bi-LSTM-AM[36] proposes a

novel relational graph attention network that incorporates edge attributes. This method first builds a semantic dependency graph through dependency parsing, models a semantic graph that considers the edge attributes by using top-k attention mechanisms to learn hidden semantic contextual representations, and finally predicts event temporal relations. DBAG-GCN[37] proposes a novel deep bi-directional adaptive gating graph convolutional network for spatio-temporal traffic forecasting. HestGCL[38] proposes a novel heterogeneous spatio-temporal graph contrastive learning method to compensate for the shortcomings of the existing GNN-based methods for modeling spatio-temporal information. TAG-Net[39] proposes a model consisting of a new cooperative learning mechanism for target attention graphs, and also adopts self-supervised contrastive learning to alleviate the problem of excessive smoothing in graph learning, simulating a sociological phenomenon called the chameleon effect. Despite the efforts of these studies, they ignore the spatial-temporal characteristics of trajectories when constructing local subgraphs. In this paper, we utilize the spatial-temporal characteristics of trajectories to construct contrastive subgraphs of the trajectory network as training samples and elaborately design a contrastive loss function.

2.3 Urban computing

Our study also falls into the research category of urban computing. The location representations can be taken as pre-trained parameters to integrate into task models for assisting downstream tasks. Urban computing[40, 41] aims to build smart cities, which solve a series of urban issues[42–44]. The main research areas within urban computing can be categorized into four key aspects:

- **Transportation.** Researches[45–49] in this area focus on improving the efficiency and safety of urban transportation systems. The location representations can reflect population movement, road network density, and other information in different regions. In traffic flow prediction or route planning, location representations can provide spatial context for the model and improve prediction accuracy.
- **Environment.** Existing methods[50–52] aim to monitor and mitigate urban pollution, optimize energy consumption, and enhance green infrastructure. The location representations allow for a better understanding of the functional attributes, development potential, and needs of different areas. These representations can be combined with existing socio-economic data (e.g., population density, income levels, etc.) to help decision-makers optimize area use allocation.
- **Public safety.** Ensuring public safety is a fundamental aspect of smart city research. Urban computing has been applied to crime prediction[53] and emergency response optimization[54, 55]. The location representations allow the analysis of crime patterns and the probability of occurrence in different areas of the city. Combining historical crime data (e.g., burglary, violent crime, traffic accidents, etc.) with the spatial characteristics of the locations, the model can predict potential high-crime areas in the future, thus providing decision support to the police and security authorities.
- **Public health.** The integration of urban computing in healthcare aims to improve public health outcomes by leveraging data from urban environments. This includes monitoring the spread of infectious diseases[56, 57] and optimizing healthcare delivery[58]. The location representations can be combined with public health data, such as cases of infectious diseases, population movements, and distribution of healthcare resources, to help predict the pathways of disease transmission in cities.

In this work, we aim to learn generalization representations, which is beneficial for the prediction of traffic flow, the recommendation of people's next locations, and other urban tasks.

3 Preliminary

In this section, we give the mathematical definitions and problem statements discussed in this paper for convenience.

LBS data. LBS data are denoted as a quad-tuple $c^u = (u, l, t^{\text{in}}, t^{\text{out}})$, which indicates that user u ($u \in \{u_i | 0 \leqslant i < U\}$) arrives at location l ($l \in \{l_i | 0 \leqslant i < N\}$) at time t^{in} and leaves at time t^{out}.

User trajectory. A user trajectory is the sequence of LBS data, $\text{Traj}^u = [c_0^u, c_1^u, \ldots, c_{N^u}^u]$, where $c_i^u = (u, l_i, t_i^{\text{in}}, t_i^{\text{out}})$ is the i-th point of the user u's trajectory. The sequence is sorted by chronological order, i.e., $\forall i \geqslant 0$, $t_i^{\text{in}} < t_i^{\text{out}} < t_{i+1}^{\text{in}} < t_{i+1}^{\text{out}}$.

Trajectory segment. A trajectory segment comes from a continuous part of a user trajectory, that is, $\text{Traj}_{ij}^u = [c_i^u, c_{i+1}^u, \ldots, c_j^u]$, $0 \leqslant i \leqslant j \leqslant N^u$.

Trajectory network. We represent a trajectory network, a directed multigraph, as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$,

where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is a set of N nodes and corresponds to the locations, and $\mathcal{E} = \{e_{ij}^k | 0 \leqslant i, j < N\}$ is a set of directed edges and indicates that a transition is between the two nodes (v_i, v_j). k is an index to distinguish edges because multiple edges might be between v_i and v_j. The set $e_{ij}^k = (t_i^{\text{in}}, t_i^{\text{out}}, t_j^{\text{in}}, t_j^{\text{out}})$ is associated with the temporal feature of a transfer from location i to location j. The set $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ has a one-to-one correspondence with the node set $\mathcal{V}$, where x_{v_i} represents the representations of node v_i.

Problem statement. Given an LBS dataset $\{c_i^u | 0 \leqslant u < U, 0 \leqslant i \leqslant N^u\}$, this proposed framework aims to learn location representations. The end-to-end framework that randomly initializes the location representations $\mathcal{X}$, generates trajectory network $\mathcal{G}$, designs the TGCN, constructs contrastive trajectory subgraph pairs, and learns location representations.

4 Methodology

In this section, we detailedly describe the spatial-temporal trajectory subgraph contrastive learning framework, namely ST-TGCL, consisting of three components: trajectory network, TGCN, and trajectory subgraph contrastive learning.

4.1 Trajectory network

Existing works formulate trajectories as a graph or multiple sequences, where the former damages temporal information after graph generation, while the latter merely expresses a single functionality in one sample. Graphs can represent complex data structures and have good scalability. Hence, we utilize a graph to describe that users visit location sequences. The trajectory network is a directed multigraph whose edges denote transfers between locations. In a real-world scenario, numerous users could go from location l_i to location l_j. Therefore, multiple directed edges correspond from v_i to v_j to multiple origin-destination transitions (v_i, v_j).

The trajectory graph differs from both heterogeneous and dynamic graphs. Distinct from heterogeneous graphs, which have multiple types of nodes and edges, a trajectory graph contains only one type of node and edge. Additionally, unlike dynamic graphs, where nodes and edges can change over time, the structure of a trajectory graph remains static. To capture the sequential characteristics of user trajectories, trajectory graphs incorporate temporal features on the edges. This approach effectively combines the strengths of both graph-based and sequence-based methods.

We describe the trajectory network generation as follows: when obtaining LBS data, we first transform them into trajectories by sorting the user and time. Then, we take all locations as nodes and generate edges based on origin-destination relations in the trajectories. The features of the edge e_{ij}^k are timestamps of arrival and departure at location l_i and location l_j, i.e., t_i^{in}, t_i^{out}, t_j^{in}, and t_j^{out}. Finally, we randomly initialize the node features (location representations) $\mathcal{X} \in \mathbf{R}^{N \times K}$, where K is the dimension of the features. As shown in Fig. 2, two trajectories generate a trajectory network.

Through the above process, we generate the trajectory network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{X}\}$.

4.2 TGCN

The TGCN stacks temporal encoder layers and spatial graph convolution layers, where the former handles the shortcoming that graph convolution captures temporal information while the latter learns spatial structures. The temporal encoder layer first embeds the temporal features of edges, generates temporal embeddings, and then passes the embeddings to spatial graph convolution. The spatial graph convolution layer integrates the spatial-temporal information of neighboring nodes and generates location representations.

4.2.1 Temporal encoders

We utilize temporal encoder layers to capture the temporal information of trajectories. The temporal information of trajectories can be divided into three categories: visit time, stay time, and transfer time. To model the aforementioned three categories of temporal information, the features of edges are fed into three components, i.e., visitation temporal encoder, stay temporal encoder, and transfer temporal encoder.

Visitation temporal encoder. The time at which locations are visited is subject to tidal patterns.

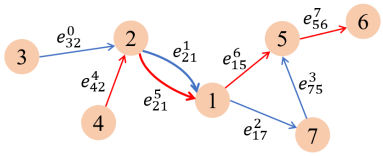


Fig. 2 Illustration of a generating trajectory network via trajectories. The colors (blue and red) of the arrows denote two trajectories. The blue trajectory is 3→2→1→7→5, and the red trajectory is 4→2→1→5→6. The trajectories generate a trajectory network that is a directed multigraph.

Locations with different functionalities have varying tidal patterns, and the functionalities of these locations change over time. The tidal patterns are affected by the hierarchical time stamps (hour, week, month, and year). For example, people's behaviors are very different on workdays and weekends, so the distribution of locations in the trajectory also varies greatly. In conclusion, visitation time is correlated with location functionalities and is beneficial for representation learning.

The visitation temporal encoders are calculated according to the arrival time of the origin or destination locations. We first transform the timestamps into hierarchical timestamps, i.e., minute, hour, and week. Then, the visitation temporal encoder embeds the hierarchical timestamps separately. Finally, the embeddings are aggregated to represent information on visitation time. The visitation time embedding (VTE) of the target the i-th node is expressed as

$$\mathrm{VTE}_i = \sum_{\mathrm{op}} F^{\mathrm{v}}_{\mathrm{op}}(\phi^{\mathrm{v}}_{\mathrm{op}}(t^{\mathrm{in}})) \tag{1}$$

where $\mathrm{Ht} = \{\text{minute, hour, week}\}$ is a set of hierarchical timestamp categories and $\mathrm{op} \in \mathrm{Ht}$, $\phi^{\mathrm{v}}_{\mathrm{op}}(\cdot)$ transforms the timestamps into hierarchical time stamps, and $F^{\mathrm{v}}_{\mathrm{op}}(\cdot)$ is an embedding function: $\mathbf{R} \to \mathbf{R}^K$.

Stay temporal encoder. The stay time at locations is distinguishable between different locations and different functions of the same location. For varying location categories, their stay time distributions are distinctive; for example, the stay time of visited offices and markets. Compared to the markets, the means and variances of office stay time are larger and smaller, respectively. For the same location, the stay time is related to varying functionalities, such as dining and takeout in a restaurant.

The input of stay temporal encoders is the arrival and departure times. Firstly, we calculate the stay time for locations, which is formulated as

$$t^{\mathrm{s}}_i = \phi^{\mathrm{s}}(t^{\mathrm{in}}_i, t^{\mathrm{out}}_i) \tag{2}$$

where the $\phi^{\mathrm{s}}(\cdot)$ calculates the interval between arrival and departure time. Next, we employ sine and cosine functions of different frequencies to represent the stay time embedding. The stay time embedding (STE) of target node i is formulated as

$$\begin{aligned}\mathrm{STE}_i &= [F^{\mathrm{s}}_1(t^{\mathrm{s}}_i) \| F^{\mathrm{s}}_2(t^{\mathrm{s}}_i) \| \ldots \| [F^{\mathrm{s}}_j(t^{\mathrm{s}}_i) \| \ldots \| F^{\mathrm{s}}_{K'}(t^{\mathrm{s}}_i)],\\ F^{\mathrm{s}}_j(t^{\mathrm{s}}_i) &= [\cos(w^{\mathrm{s}}_j t^{\mathrm{s}}_i) \| \sin(w^{\mathrm{s}}_j t^{\mathrm{s}}_i)]\end{aligned} \tag{3}$$

where the $w^{\mathrm{s}}_1, w^{\mathrm{s}}_2, \ldots, w^{\mathrm{s}}_{K'}$ are trainable parameters, "‖" is a concatenation operation, and the dimension of the STE_i is K, and K' is the number of the neighbors.

Transfer temporal encoder. The transfer time between locations often significantly affects message passing in representation learning. For the LBS data, the shorter the transfer time between two locations, the higher the correlation. The calculation method of the transfer temporal encoder is similar to that of the stay temporal encoder. We first calculate the transfer time between locations, which is expressed as

$$t^{\mathrm{t}}_i = \phi^{\mathrm{t}}(t^{\mathrm{out}}_i, t^{\mathrm{in}}_j) \tag{4}$$

where the $\phi^{\mathrm{t}}(\cdot)$ calculates the interval between the departure time of the origin location and the arrival time of the destination location. The target node i inputs its departure time when it is the origin location of the trajectory, and vice versa, inputs its arrival time. Then, we generate the transfer time embedding (TTE) of target node i, which is formulated as

$$\begin{aligned}\mathrm{TTE}_i &= [F^{\mathrm{t}}_1(t^{\mathrm{t}}_i) \| F^{\mathrm{t}}_2(t^{\mathrm{t}}_i) \| \ldots \| [F^{\mathrm{t}}_j(t^{\mathrm{t}}_i) \| \ldots \| F^{\mathrm{t}}_{K'}(t^{\mathrm{t}}_i)],\\ F^{\mathrm{t}}_j(t^{\mathrm{t}}_i) &= [\cos(w^{\mathrm{t}}_j t^{\mathrm{t}}_i) \| \sin(w^{\mathrm{t}}_j t^{\mathrm{t}}_i)]\end{aligned} \tag{5}$$

where the $w^{\mathrm{t}}_1, w^{\mathrm{t}}_2, \ldots, w^{\mathrm{t}}_{K'}$ are trainable parameters.

Through the above processing of three components, we obtain visitation time, stay time, and transfer time embeddings to denote the temporal information.

4.2.2 Spatial graph convolution

The temporal encoder models temporal information as edge embeddings to participate in representation learning. The spatial graph convolution aggregates upstream and downstream information to integrate the spatial-temporal information in nodes and edges. The spatial graph convolution consists of two processes: message passing and update embedding.

The message passing is to integrate temporal and spatial information, i.e., the edge and node features, which is expressed as

$$m^{\tau+1}_i = \sum_{(j,k)}^{(j,k)\in\mathcal{N}^{\mathrm{in}}_i} F_{\mathrm{in}}(h^{\tau}_j, e^{k}_{ij}) + \sum_{(j,k)}^{(j,k)\in\mathcal{N}^{\mathrm{out}}_i} F_{\mathrm{out}}(h^{\tau}_j, e^{k}_{ij}) \tag{6}$$

where $\mathcal{N}^{\mathrm{in}}_i$ is an edge set whose origin node is i and $\mathcal{N}^{\mathrm{out}}_i$ is an edge set whose destination node is i; j is an upstream or downstream node index of the target node and k is an edge index, which corresponds to one edge; $m^{\tau+1}_i$ denotes the aggregated message of the $(\tau+1)$-th layer; e^{k}_{ij} denotes the feature of the edge, i.e., VTE^k, STE^k, and TTE^k; $h^{\tau}_0, h^{\tau}_1, \ldots, h^{\tau}_N$ denotes node

embedding output by the τ-th layer, and h_0^0, h_1^0, …, h_N^0 are x_0, x_1, …, x_N.

The functions $F_{\text{in}}(\cdot)$ and $F_{\text{out}}(\cdot)$ aggregate upstream and downstream nodes, respectively. To begin with, we integrate the VTE and the STE into node embedding, which is expressed as

$$\hat{h}_i^\tau = \varphi^t(h_i^\tau, \text{VTE}_i^k, \text{STE}_i^k) \tag{7}$$

The node embedding integrates the VTE and the STE. A concatenation operation and an aggregation function are utilized to initialize $\varphi^t(\cdot)$. Afterward, the target node aggregates information about the upstream or downstream nodes based on the transfer time, which is formulated as

$$F_{\text{in}}(\cdot) = \frac{W_{\text{in}}^\alpha \text{TTE}_j^k}{\sum_{(j',k')\in\mathcal{N}_i^{\text{in}}} W_{\text{in}}^\alpha \text{TTE}_{j'}^{k'}} W_{\text{in}} \hat{h}_j^\tau \tag{8}$$

where W_{in}^α and W_{in} are trainable parameters. j' and k' are indexes for nodes and edges, respectively. $F_{\text{out}}(\cdot)$ is calculated in the same way as $F_{\text{in}}(\cdot)$, which is expressed as

$$F_{\text{out}} = \frac{W_{\text{out}}^\alpha \text{TTE}_j^k}{\sum_{(j',k')\in\mathcal{N}_i^{\text{out}}} W_{\text{out}}^\alpha \text{TTE}_{j'}^{k'}} W_{\text{out}} \hat{h}_j^\tau \tag{9}$$

where W_{out}^α and W_{out} are trainable parameters.

The update embedding is generated based on the aggregated message and previous embedding. We update the embedding of target node i, which is calculated as

$$h_i^{\tau+1} = F_u(h_i^\tau, m_i^{\tau+1}) \tag{10}$$

where $F_u(\cdot)$ is a function that stacks linear layers and activation functions to update the representation of target node i.

As shown in Fig. 2, when we take Node 1 as a target node, it would aggregate Nodes 2, 5, and 7 based on edges e_{21}^1, e_{21}^5, e_{15}^6, and e_{17}^2. It is noteworthy that Node 2 integrates the time information of blue and red trajectories and is aggregated to Node 1.

4.3 Trajectory subgraph contrastive learning

In this subsection, we mainly introduce trajectory subgraph contrastive learning, including trajectory contrastive sample construction and contrastive loss function. To address the long-tail issue, this component learns deep representations of locations by designing pretext tasks to optimize parameters. As shown in Fig. 3, we construct trajectory contrastive samples based on the trajectory network, input the samples to the trajectory graph convolution network, and optimize the parameters of the framework by the well-designed contrastive loss function.

4.3.1 Trajectory contrastive samples construction

The trajectory contrastive sample construction mainly obtains positive and negative samples by constructing subgraph pairs. It consists of two main steps: trajectory segment generation and contrastive trajectory subgraph construction.

Trajectory segment generation samples a trajectory segment whose length is $2N_{\text{seg}}+1$ in trajectories. We first select a location as the anchor node. In theory, every location visited at least twice is selected as an anchor node. Then, we sample a segment of the trajectory containing the anchor node. We set $[0, N_{\text{seg}})$, N_{seg}, and $(N_{\text{seg}}, 2N_{\text{seg}}+1]$ as upstream nodes, the

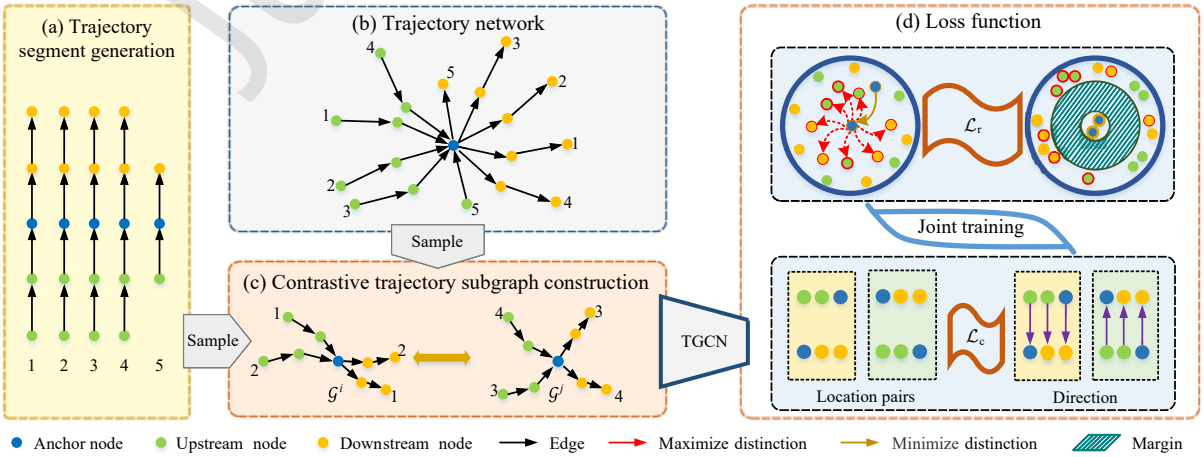


Fig. 3 Overall framework of spatial-temporal trajectory subgraph contrastive learning.

anchor node, and downstream nodes, respectively, and obtain a segment whose length is $2N_{\text{seg}}+1$. Finally, we create an index to quickly find the required sequence segments. As shown in Fig. 3a, the blue, green, and yellow nodes are the anchor, upstream nodes, and downstream nodes, respectively, and 1, 2, …, 5 are trajectory segments of the anchor node.

Contrastive trajectory subgraph construction takes trajectory segments and the trajectory network as input and outputs contrastive trajectory subgraph pairs, namely train samples. We first sample $2N_{\text{traj}}$ segments whose anchor node is the same location. Then, we evenly divide the sampled trajectory segments into two parts. Finally, we select the nodes and edges of the trajectory network corresponding to these trajectory segments to construct two subgraphs as a contrastive sample pair. As shown in Fig. 3b, the trajectory network merely includes the anchor node and its 2-hop neighboring nodes for convenience and clarity. In Fig. 3c, we set $N_{\text{seg}} = 2$ and $N_{\text{traj}} = 2$ and sample trajectory segments (1, 2) and (3, 4) to construct a contrastive trajectory subgraph pair.

In actual scenarios, each location could have multiple functionalities that usually vary in trajectories (trajectory segments), just as the same word could have different meanings in multiple sentences. For example, a factory is a place where various raw materials are received, various products are sent out during production, and workers work. In LBS data, the three functionalities of the factory mentioned above could be close to the warehouse, take-out restaurant, and office, respectively, in the representation space. Therefore, we combine trajectory segments to construct subgraph pairs to obtain a comprehensive and sufficient description of a location.

In contrastive trajectory subgraph construction, the settings of N_{traj} and N_{seg} are critical. The following factors are considered when setting the sampling trajectory segment length and the number of generated trajectory segments: Due to the periodicity of people visiting locations and functionalities, to avoid those anchor nodes reappearing upstream and downstream, the length of sampled trajectory segments $2N_{\text{seg}}+1$ should be less than the period of people's activity (e.g., day, week, etc.). Meanwhile, the number of trajectory segments N_{traj} constructed as contrastive trajectory subgraphs is enough to comprehensively describe the functionalities of locations. In a word, it is too large to introduce noise and unnecessarily compute. Considering the computational and redundancy aspects, N_{traj} and N_{seg} are not set to be too large.

4.3.2 Contrastive loss function

As shown in Fig. 3d, the contrastive loss function $\mathcal{L}$ consists of two parts: node representation loss $\mathcal{L}_{\text{r}}$ (representation distance) and node order loss $\mathcal{L}_{\text{c}}$ (edge direction), respectively. $\mathcal{L}$ is defined as

$$\mathcal{L} = \theta\mathcal{L}_{\text{r}} + (1-\theta)\mathcal{L}_{\text{c}} \tag{11}$$

where $\theta \in [0, 1]$ is a hyperparameter, and is the trade-off parameter of $\mathcal{L}_{\text{r}}$ and $\mathcal{L}_{\text{c}}$.

We sample a contrastive trajectory subgraph pair $\mathcal{G}^i = \{\mathcal{V}^i, \mathcal{X}^i, \mathcal{E}^i\}$ and $\mathcal{G}^j = \{\mathcal{V}^j, \mathcal{X}^j, \mathcal{E}^j\}$, whose nodes can be divided into three categories: (1) anchor nodes (the blue), (2) upstream nodes (the green), and (3) downstream nodes (the yellow), as shown in Fig. 3c. Through encoding of the TGCN, the node representations H^i and H^j of nodes in $\mathcal{G}^i$ and $\mathcal{G}^j$ are obtained, respectively.

The anchor representations of the contrastive trajectory subgraph pairs are indicated by h_a^i and h_a^j. Due to the complex behaviors and aims of people in actual scenarios, the nodes with similar functionality might wrongly are divided into different categories in a subgraph, which could make the parameters difficult to train. For more stable representations, a mean aggregation function MEAN $(\cdot)$ is employed to transform the representations of upstream and downstream nodes, $\bar{h}_{\text{u}}$ and $\bar{h}_{\text{d}}$, respectively, which are expressed as

$$\bar{h}_{\text{u}} = \text{MEAN}(\{h | h \in \mathcal{N}_{\text{u}}^{\mathcal{G}}\}) \tag{12}$$

$$\bar{h}_{\text{d}} = \text{MEAN}(\{h | h \in \mathcal{N}_{\text{d}}^{\mathcal{G}}\}) \tag{13}$$

where $\mathcal{N}_{\text{u}}^{\mathcal{G}}$ and $\mathcal{N}_{\text{d}}^{\mathcal{G}}$ are the N_{seg}-hop upstream and downstream neighbor nodes in $\mathcal{G}$, respectively.

In a word, the positive instance consists of anchor representations, h_{a}^i and h_{a}^j, while the negative instances consist of anchors and mean-aggregated upstream and downstream node representations that are $\bar{h}_{\text{u}}^i$, $\bar{h}_{\text{d}}^i$, $\bar{h}_{\text{u}}^j$, and $\bar{h}_{\text{d}}^j$.

Nodes representation loss $\mathcal{L}_{\text{r}}$ is used to estimate the distance of nodes in representation space. It takes two anchor nodes as a positive sample in a pair of subgraphs while taking anchor nodes and other nodes (upstream and downstream nodes) as negative samples. As shown in Fig. 3d, it is set to minimize the distance between anchor nodes and maximize the distance between anchor nodes and other nodes in a contrastive

trajectory subgraph pair. $\mathcal{L}_{\mathrm{r}}$ of the sample pair is defined as

$$\mathcal{L}_{\mathrm{r}} = \beta_{\mathrm{r}} l^{\mathrm{r}}(h_{\mathrm{a}}^{i}, h_{\mathrm{a}}^{j}) - (1-\beta_{\mathrm{r}}) \sum_{v \in i,\, j} (l^{\mathrm{r}}(h_{\mathrm{a}}^{v}, \bar{h}_{u}^{v}) + l^{\mathrm{r}}(h_{\mathrm{a}}^{v}, \bar{h}_{\mathrm{d}}^{v})) \tag{14}$$

where $\beta_{\mathrm{r}} \in [0, 1]$ is a hyperparameter, and is the trade-off parameter of minimizing and maximizing distance. We utilize mean squared error to instantiate $l^{\mathrm{r}}(\cdot)$, which is expressed as

$$l^{\mathrm{r}}(x, y) = (x-y)^2 \tag{15}$$

Nodes order loss $\mathcal{L}_{\mathrm{c}}$ mainly learns the temporal information to be designed to enhance temporal representation. As shown in Fig. 3d, $\mathcal{L}_{\mathrm{c}}$ is to predict edge direction, which takes anchor nodes, upstream nodes, or downstream nodes representations as inputs and predicts their chronological order. Moreover, its purpose is to make the nodes learn temporal information while making the nodes visited infrequently obtain sufficient training. If the number of trajectory segments of a location is not enough to be selected as an anchor, the location can participate in training as an upstream or downstream node of the contrastive sample.

This task is formulated as a classification problem, that is, if the input representations are arranged in an origin-to-destination manner, which is a positive sample, and vice versa, a negative sample. $\mathcal{L}_{\mathrm{c}}$ of the sample pair is defined as

$$\mathcal{L}_{\mathrm{c}} = \sum_{v \in i,\, j} (\beta_{\mathrm{c}} l^{\mathrm{c}}(\bar{h}_{\mathrm{u}}^{v}, h_{\mathrm{a}}^{v}) + \beta_{\mathrm{c}} l^{\mathrm{c}}(h_{\mathrm{a}}^{u}, \bar{h}_{\mathrm{d}}^{v}) + (1-\beta_{\mathrm{c}})(l^{\mathrm{c}}(\bar{h}_{\mathrm{u}}^{v}, \bar{h}_{\mathrm{d}}^{v})) \tag{16}$$

where $\beta_{\mathrm{c}} \in [0, 1]$ is a hyperparameter, and is the trade-off parameter of anchor and other nodes.

l^{c}: $\mathbf{R}^{2d} \rightarrow \mathbf{R}^{2}$ consists of stacked layers, which are the concatenation operation layers, linear layers, activation function layers, and log-likelihood loss function layers, which is formulated as

$$l^{\mathrm{c}}(x, y) = \log(\sigma(W^{\mathrm{c}}[x \| y])) + \log(1 - \sigma(W^{\mathrm{c}}[y \| x])) \tag{17}$$

where W^{c} is a trainable parameter, and $\sigma(\cdot)$ is an activation function.

5 Experiment

In this section, we describe our experimental setup and empirical results. We conduct extensive experiments to benchmark the effectiveness and generalization ability over multiple real datasets. Our experiments are designed to answer the following research questions:

- **RQ1:** Are the location representations useful for urban tasks in actual scenarios?
- **RQ2:** Do the location representations conform to our common sense?
- **RQ3:** What are the influences of various components in the ST-TGCL?

5.1 Dataset

We evaluate our framework on four real-world location-based datasets: Brighkite[‡], NYC[§], TKY[§], and MHK. Table 1 shows the statistics of the used datasets. The Brighkite, NYC, and TKY are location-based datasets where users share their locations by check-ins, including user ID, location, check-in time, etc.

The MHK contains POI and cellular signaling data. The POI data include name, location, and category. The cellular signaling data include user ID, connection time, and station location. Thus, we could obtain mobility traces of users when users arrive at and leave the station. The dataset covers the whole city, Meihekou, and all trip aims and modes to ensure comprehensive semantics. We have collected the data, whose capacity is 108.1 GB, from January 13 to 19, 2023.

5.2 Baselines.

We compare our framework with the following baselines:

- **Sequence-based methods.** POI2Vec[59] is an embedding method based on Word2Vec, which models sequence correlations through distributing locations. CTLE[8] implements the mapping function using a bidirectional transformer encoder and employs the masked language model pretraining objective to model the sequential correlation in trajectories. CL-TSim[60] employs a contrastive learning mechanism and proposes two augmentation strategies, including point

Table 1 Dataset statistics.

Dataset	Number of users	Number of locations	Number of check-ins
NYC	824	38 336	227 428
TKY	1939	61 858	573 703
Brighkite	5247	48 181	1 699 579
MHK	298 691	2356	514 525 418

[‡] https://snap.stanford.edu/data/loc-brightkite.html
[§] https://sites.google.com/site/yangdingqi/home/foursquare-dataset

down-sampling and point distorting, to learn the latent representations of trajectories. CoPPS[19] is a contrastive learning framework designed to enhance product search by generating high-quality representations through self-supervision signals, utilizing data augmentation strategies, and leveraging an external knowledge graph.

- **Graph-based methods.** DGI[61] leverages local mutual information maximization across the graph's patch representations to obtain powerful graph convolutional architectures. SUBG-CON[62] utilizes the strong correlation between central nodes and their regional subgraphs for model optimization. ST2Vec[31] is a spatio-temporal representation learning model that enhances trajectory similarity computation by encoding fine-grained spatial and temporal relations, using spatio-temporal co-attention fusion and curriculum learning for improved accuracy and efficiency. GraphMAE[63] proposes focusing on feature reconstruction with both a masking strategy and scaled cosine error that benefit robust training. MaskGAE[64] adopts a self-supervised method that masks a portion of edges and attempts to reconstruct the missing part with a partially visible, unmasked graph structure. AutoST[34] is a spatio-temporal graph contrastive learning framework that improves region embedding by capturing multi-view region dependencies and enhancing robustness against data noise and distribution heterogeneity.

5.3 Experiment setup

Model settings. Three categories of temporal encoders are applied in MHK, while the stay temporal encoders are not used due to the lack of leaving time in Brighkite, NYC, and TKY. Three TGCN layers are stacked to extract and aggregate representations of trajectory subgraphs in Brighkite, and two layers for the other dataset. For fairness, the number of parameters and computational complexity of the baselines are comparable to our framework.

Contrastive sample settings. In Brightkite, N_{seg} and N_{traj} are set to 3 and 8, respectively, while for other datasets they are set to 2 and 8. The dimension K of location representations is set to 128. We utilize a contrastive loss function with $\theta = 0.6$, $\beta_{\text{r}} = 0.3$, and $\beta_{\text{c}} = 0.3$.

Training settings. Adam optimizer with an initial learning rate of 0.001 is used to train our framework. Dropout $p = 0.5$ is applied to the outputs of the graph convolution layer. The loss function includes node representation and node order loss functions.

5.4 Next location recommendation

To demonstrate the effectiveness of the location representations, we conduct a next location recommendation as the downstream task. We select an acknowledged framework, SASRec[65], as a backbone network to fit the downstream task. The input of SASRec is a sequence of frozen location representations.

5.4.1 Metrics

We utilize two widely-used metrics, i.e., Hit Rate (HR) and Normalized Discounted Cumulative Gain (NDCG) to evaluate our framework. HR and NDCG at a cutoff k are denoted as HR@k and NDCG@k, respectively, which indicates counting the fraction of times that the target location is among the top k recommendation list. They are calculated as

$$\text{HR@}k = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \mathbb{1}(y_i \in \{\hat{y}_i^j | j = 1, 2, \ldots, k\}) \tag{18}$$

$$\text{NDCG@}k = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \sum_{j=1}^{k} \frac{2^{\mathbb{1}(y_i = \hat{y}_i^j)} - 1}{\log(j+1)} \tag{19}$$

where y_i is the ground truth, i.e., the next location, $\hat{y}_i^j$ is the j-th location in the recommendation list of the i-th instance, and $\mathcal{N}$ is the number of instances. $\mathbb{1}(x) \in \{0, 1\}$ is an indicator function that evaluates to 1 if x is true.

5.4.2 Results and analysis

Table 2 compares seven baselines and the proposed framework on the four datasets (RQ1). The results prove that our framework outperforms all the baselines in HR@10 and NDCG@10 on the next location recommendation. The methods are mainly divided into three categories: (1) The sequence-based methods merely consider temporal correlations in a single sequence, but they are not proficient in handling spatial relations. (2) The graph-based methods pay more attention to spatial correlations while ignoring temporal information. (3) Our framework employs graph structure to capture spatial information and designs temporal encoders for temporal information. Compared with sequence-based methods, the performances of most graph-based methods degrade due to the low correlation of self-supervised tasks with the next location recommendation task. MaskGAE designs the self-supervised task based on paths between nodes and

Table 2 Performance comparison of ST-TGCL and other baseline models. Bold text indicates the best model performance and Δ indicate the improvement ratio that is improvement of our method over the baseline methods with the best performance.

Method		NYC		TKY		Brighkite	
		HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
Sequence-based	POI2Vec	43.35	33.17	47.54	39.59	41.13	29.03
	CTLE	40.11	30.61	46.68	36.24	41.72	30.46
	CL-TSim	45.27	32.71	56.03	41.60	40.37	31.06
	CoPPS	47.42	32.86	57.11	41.92	41.68	31.14
Graph-based	DGI	41.87	30.72	54.76	39.89	40.92	30.79
	SUBG-CON	39.46	29.92	44.15	34.27	39.18	31.42
	ST2Vec	41.04	30.13	45.20	35.16	39.02	29.41
	GraphMAE	43.81	30.62	49.35	37.38	40.90	31.44
	MaskGAE	49.79	34.18	58.78	42.18	41.52	28.82
	AutoST	50.02	34.23	59.32	43.21	42.17	28.71
ST-TGCL		**50.37**	**34.73**	**60.65**	**43.80**	**42.35**	**31.46**
Δ (%)		0.7	1.5	2.2	1.4	0.4	0.1

captures spatial information with the graph structure, which is beneficial for this downstream task. AutoST generates multi-view graphs to capture spatio-temporal relationships and designs contrastive tasks to enhance representations, especially the trajectory-based region graph to integrate temporal information. Thus, MaskGAE and AutoST outperform the other baselines. In conclusion, the proposed ST-TGCL has significant results in location representation learning. The location representations are useful for urban tasks in actual scenarios.

5.5 Case study

We further give a case study on MHK to verify RQ2. In this case, we select three locations, i.e., *A*, *B*, and *C*, based on the POI category ratio within 500 m of locations. In terms of spatial distance, locations *A* and *B* are closer to each other, while location *C* is farther away from locations *A* and *B*. To compare the distance of two locations in our common sense (POI category ratio) and representation space for convenience, we design a metric. First, we calculate cosine similarities between locations and sort the similarities list S. Then the similarity metric is calculated as $\frac{S_{ij}}{\#S}$, where S_{ij} and $\#S$ are the order of the similarity between locations i and j in the list and the length of the list S, respectively. This metric is in [0, 1], where the larger the value, the more similar the two locations.

From Table 3, we observe that: (1) We select two locations *A* and *C*, that are similar in our common sense. Meanwhile, the similarity between the representations of *A* and *C* is high. (2) *B* is different from *A* and *C*. Similarly, *B* is far from *A* and *C* in representation space. As shown in Fig. 4, the POI category ratios of *B* in some categories are roughly the same as those of *A* and *C*. We believe *B* is different from *A* and *C* because there are tourist attractions around *B*. (3) *B* and *C* are more similar to *A* and *B* on the POI category ratio, but the distance between *A* and

Table 3 Similarities among locations *A*, *B*, and *C* on POI category ratio and representation space on MHK.

Location	POI			Representation space		
	A	*B*	*C*	*A*	*B*	*C*
A	–	0.625	0.967	–	0.477	0.989
B	0.625	–	0.576	0.477	–	0.758
C	0.967	0.576	–	0.989	0.758	–

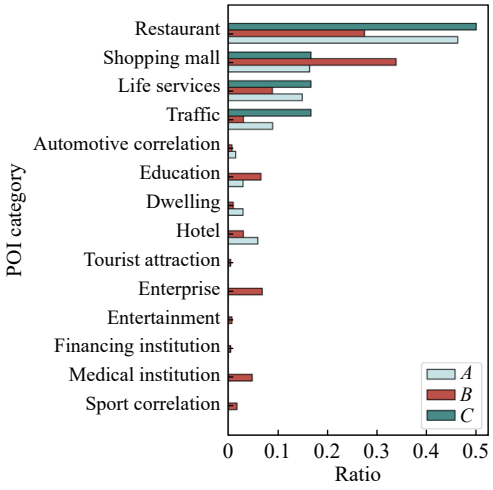


Fig. 4 POI category ratio around locations *A*, *B*, and *C*.

B is closer than between *B* and *C* in the representation space. This is because *A* and *B* are in spatial proximity, and *A* is affected by the tourist attractions around *B*. To sum up, the location representations that are trained by self-supervised tasks conform to our common sense. In addition, location representations are distinguished for downstream tasks.

5.6 Ablation study

This subsection describes ablation studies of ST-TGCL on the recommendation task to validate the effectiveness of key components to answer the RQ3. We consider the following variants of our base framework:

- w/o TE: We remove transfer temporal encoders.
- w/o VE: We remove visitation temporal encoders.
- w/o $\mathcal{L}_{\mathrm{r}}$: We optimize parameters without using $\mathcal{L}_{\mathrm{r}}$.
- w/o $\mathcal{L}_{\mathrm{c}}$: We optimize parameters without using $\mathcal{L}_{\mathrm{c}}$.

The performances of the variants and the base framework are on two datasets, as shown in Fig. 5. From Fig. 5, we can have the following findings:

- The performances of variants dropping temporal encoders (w/o TE and w/o VE) degrade due to the ability to perceive the temporal information in a sequence decreasing.
- $\mathcal{L}_{\mathrm{r}}$ and $\mathcal{L}_{\mathrm{c}}$ capture different semantics. $\mathcal{L}_{\mathrm{r}}$ prefers to learn about spatial correlation, while $\mathcal{L}_{\mathrm{c}}$ learns about temporal correlation. Thus, dropping one of the self-supervised tasks results in decreased performance.

6 Conclusion

This paper proposes a spatial-temporal trajectory subgraph contrastive learning for location representation learning. We generate the trajectory network to formulate the trajectory information and location relation, design the trajectory graph convolution network to integrate spatial-temporal information and generate the representations, and propose trajectory subgraph contrastive learning that samples contrastive subgraphs and optimizes parameters. The framework learns effective and generalization location representations for urban downstream tasks. On real-world datasets, the proposed approach achieves state-of-the-art results on the next location recommendation task, and location representations are verified to conform to our common sense.

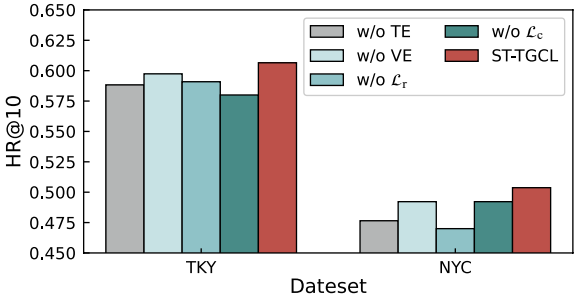


Fig. 5 Ablation studies on TKY and NYC.

Acknowledgment

This work was supported by the Jilin Province Science and Technology Development Program (No. 20240302093GX), the National Natural Science Foundation of China (No. 62072209), and the Sichuan Provincial Science and Technology Program Project (Provincial Academy-University Cooperation Project) (No. 2025YFHZ0015).

References

[1] E. Wang, Y. Jiang, Y. Xu, L. Wang, and Y. Yang, Spatial-temporal interval aware sequential POI recommendation, in *Proc. 2022 IEEE 38th Int. Conf. Data Engineering*, Kuala Lumpur, Malaysia, 2022, pp. 2086–2098.

[2] L. Zhao, M. Gao, and Z. Wang, ST-GSP: Spatial-temporal global semantic representation learning for urban flow prediction, in *Proc. 15th ACM Int. Conf. Web Search and Data Mining*, Virtual Event, 2022, pp. 1443–1451.

[3] C. Liu, Y. Yang, Z. Yao, Y. Xu, W. Chen, L. Yue, and H. Wu, Discovering urban functions of high-definition zoning with continuous human traces, in *Proc. 30th ACM Int. Conf. Information & Knowledge Management*, Virtual Event, 2021, pp. 1048–1057.

[4] M. Zhu, W. Chen, J. Xia, Y. Ma, Y. Zhang, Y. Luo, Z. Huang, and L. Liu, Location2vec: A situation-aware representation for visual exploration of urban locations, *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3981–3990, 2019.

[5] J. Ji, J. Wang, J. Wu, B. Han, J. Zhang, and Y. Zheng, Precision CityShield against hazardous chemicals threats via location mining and self-supervised learning, in *Proc. 28th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, Washington, DC, USA, 2022, pp. 3072–3080.

[6] S. Liu, Z. Xu, H. Ren, T. He, B. Han, J. Bao, K. Zheng, and Y. Zheng, Detecting loaded trajectories for hazardous chemicals transportation, in *Proc. 2022 IEEE 38th Int. Conf. Data Engineering*, Kuala Lumpur, Malaysia, 2022, pp. 3294–3306.

[7] C. Liu, Y. Yang, Z. Yao, Y. Xu, W. Chen, L. Yue, and H. Wu, Discovering urban functions of high-definition zoning with continuous human traces, in *Proc. 30th ACM Int. Conf. Information & Knowledge Management*, Virtual Event, 2021, pp. 1048–1057.

[8] Y. Lin, H. Wan, S. Guo, and Y. Lin, Pre-training context

and time aware location embeddings from spatial-temporal trajectories for user next location prediction, in *Proc. 35th AAAI Conf. Artificial Intelligence*, Palo Alto, CA, USA, 2021, pp. 4241–4248.

[9] Y. S. Lu and J. L. Huang, GLR: A graph-based latent representation model for successive poi recommendation, *Future Gener. Comput. Syst*., vol. 102, pp. 230–244, 2020.

[10] Z. Wang, Y. Zhu, Q. Zhang, H. Liu, C. Wang, and T. Liu, Graph-enhanced spatial-temporal network for next POI recommendation, *ACM Trans. Knowledge Discov. Data*, vol. 16, no. 6, p. 104, 2022.

[11] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, J. Zhang, B. Ding, and B. Cui, Contrastive learning for sequential recommendation, in *Proc. 2022 IEEE 38th Int. Conf. Data Engineering*, Kuala Lumpur, Malaysia, 2022, pp. 1259–1273.

[12] R. Qiu, Z. Huang, H. Yin, and Z. Wang, Contrastive learning for representation degeneration problem in sequential recommendation, in *Proc. 15th ACM Int. Conf. Web Search and Data Mining*, Virtual Event, 2022, pp. 813–823.

[13] X. Li, A. Sun, M. Zhao, J. Yu, K. Zhu, D. Jin, M. Yu, and R. Yu, Multi-intention oriented contrastive learning for sequential recommendation, in *Proc. 16th ACM Int. Conf. on Web Search and Data Mining*, Singapore, 2023, pp. 411–419.

[14] D. Babaev, N. Ovsov, I. Kireev, M. Ivanova, G. Gusev, I. Nazarov, and A. Tuzhilin, CoLES: Contrastive learning for event sequences with self-supervision, in *Proc. 2022 Int. Conf. Management of Data*, Philadelphia, PA, USA, 2022, pp. 1190–1199.

[15] H. Du, H. Shi, P. Zhao, D. Wang, V. S. Sheng, Y. Liu, G. Liu, and L. Zhao, Contrastive learning with bidirectional transformers for sequential recommendation, in *Proc. 31st ACM Int. Conf. Information & Knowledge Management*, Atlanta, GA, USA, 2022, pp. 396–405.

[16] Z. Wang, H. Liu, W. Wei, Y. Hu, X. L. Mao, S. He, R. Fang, and D. Chen, Multi-level contrastive learning framework for sequential recommendation, in *Proc. 31st ACM Int. Conf. Information & Knowledge Management*, Atlanta, GA, USA, 2022, pp. 2098–2107.

[17] S. Bian, W. X. Zhao, K. Zhou, J. Cai, Y. He, C. Yin, and J. R. Wen, Contrastive curriculum learning for sequential user behavior modeling via data augmentation, in *Proc. 30th ACM Int. Conf. Information & Knowledge Management*, Virtual Event, 2021, pp. 3737–3746.

[18] Q. Wang, M. Cheng, S. Yuan, and H. Xu, Hierarchical contrastive learning for temporal point processes, in *Proc. 37th AAAI Conf. Artificial Intelligence*, Washington, DC, USA, 2023, pp. 10166–10174.

[19] S. Dai, J. Liu, Z. Dou, H. Wang, L. Liu, B. Long, and J. R. Wen, Contrastive learning for user sequence representation in personalized product search, in *Proc. 29th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, Long Beach, CA, USA, 2023, pp. 380–389.

[20] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, Graph contrastive learning with augmentations, in *Proc. 34th Conf. Neural Information Processing Systems*, Vancouver, Canada, 2020, pp. 5812–5823.

[21] D. Kim and A. Oh, How to find your friendly neighborhood: Graph attention design with self-supervision, in *Proc. 9th Int. Conf. Learning Representations*, Virtual Event, https://dblp.org/db/conf/iclr/iclr2021.html#KimO21, 2021.

[22] D. Hwang, J. Park, S. Kwon, K. M. Kim, J. W. Ha, and H. J. Kim, Self-supervised auxiliary learning with meta-paths for heterogeneous graphs, in *Proc. 34th Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2020, p. 863.

[23] S. Suresh, P. Li, C. Hao, and J. Neville, Adversarial graph augmentation to improve graph contrastive learning, in *Proc. 35th Conf. Neural Information Processing Systems*, Virtual Event, 2021, p. 34.

[24] K. Hassani and A. H. Khasahmadi, Contrastive multi-view representation learning on graphs, in *Proc. 37th Int. Conf. Machine Learning*, Virtual Event, 2020, pp. 4116–4126.

[25] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, GCC: Graph contrastive coding for graph neural network pre-training, in *Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, Virtual Event, 2020, pp. 1150–1160.

[26] Y. You, T. Chen, Y. Shen, and Z. Wang, Graph contrastive learning automated, in *Proc. 38th Int. Conf. Machine Learning*, Virtual Event, 2021, pp. 12121–12132.

[27] Y. Liu, M. Jin, S. Pan, C. Zhou, Y. Zheng, F. Xia, and P. S. Yu, Graph self-supervised learning: A survey, *IEEE Trans. Knowledge Data Eng.*, vol. 35, no. 6, pp. 5879–5900, 2023.

[28] A. Subramonian, MOTIF-driven contrastive learning of graph representations, in *Proc. 35th AAAI Conf. Artificial Intelligence*, Palo Alto, CA, USA, 2021, pp. 15980–15981.

[29] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in *Proc. 33rd AAAI Conf. Artificial Intelligence*, Honolulu, HI, USA, 2019, pp. 922–929.

[30] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, Traffic flow prediction via spatial temporal graph neural network, in *Proc. Web Conf. 2020*, Taipei, China, 2020, pp. 1082–1092.

[31] Z. Fang, Y. Du, X. Zhu, D. Hu, L. Chen, Y. Gao, and C. S. Jensen, Spatio-temporal trajectory similarity learning in road networks, in *Proc. 28th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, Washington, DC, USA, 2022, pp. 347–356.

[32] L. Yuan, R. Qian, Y. Cui, B. Gong, F. Schroff, M. H. Yang, H. Adam, and T. Liu, Contextualized spatio-temporal contrastive learning with self-supervision, in *Proc. 2022 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, New Orleans, LA, USA, 2022, pp. 13957–13966.

[33] R. Li, T. Zhong, X. Jiang, G. Trajcevski, J. Wu, and F. Zhou, Mining spatio-temporal relations via self-paced graph contrastive learning, in *Proc. 28th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, Washington, DC, USA, 2022, pp. 936–944.

[34] Q. Zhang, C. Huang, L. Xia, Z. Wang, Z. Li, and S. Yiu, Automated spatio-temporal graph contrastive learning, in *Proc. ACM Web Conf. 2023*, Austin, TX, USA, 2023, pp. 295–305.

[35] B. Chen, J. Wu, X. Liu, F. Zhou, and G. Luo, Enhancing temporal knowledge graph for future event prediction with long-term dense graph, *Tsinghua Science and Technology*, doi: 10.26599/TST.2024.9010119.

[36] X. Xu, T. Gao, Y. Wang, and X. Xuan, Event temporal relation extraction with attention mechanism and graph neural network, *Tsinghua Science and Technology*, vol. 27, no. 1, pp. 79–90, 2022.

[37] X. Wang, J. Lv, M. O. Alassafi, F. E. Alsaadi, B. D. Parameshachari, L. Zou, G. Feng, and Z. Liu, Deep bi-directional adaptive gating graph convolutional networks for spatio-temporal traffic forecasting, *Tsinghua Science and Technology*, vol. 30, no. 5, pp. 2060–2080, 2025.

[38] J. Liu, H. Gao, C. Yang, C. Shi, T. Yang, H. Cheng, Q. Xie, X. Wang, and D. Wang, Heterogeneous spatio-temporal graph contrastive learning for point-of-interest recommendation, *Tsinghua Science and Technology*, vol. 30, no. 1, pp. 186–197, 2025.

[39] X. Liu, Y. He, W. Tai, X. Xu, F. Zhou, and G. Luo, Exploring the chameleon effect of contextual dynamics in temporal knowledge graph for event prediction, *Tsinghua Science and Technology*, vol. 30, no. 1, pp. 433–455, 2025.

[40] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, Urban computing: Concepts, methodologies, and applications, *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, p. 38, 2014.

[41] G. Jin, Y. Liang, Y. Fang, Z. Shao, J. Huang, J. Zhang, and Y. Zheng, Spatio-temporal graph neural networks for predictive learning in urban computing: A survey, *IEEE Trans. Knowledge Data Eng.*, vol. 36, no. 10, pp. 5388–5408, 2024.

[42] S. Ruan, X. Fu, C. Long, Z. Xiong, J. Bao, R. Li, Y. Chen, S. Wu, and Y. Zheng, Filling delivery time automatically based on couriers' trajectories, *IEEE Trans. Knowledge Data Eng.*, vol. 35, no. 2, pp. 1528–1540, 2023.

[43] Z. Pan, S. Ke, X. Yang, Y. Liang, Y. Yu, J. Zhang, and Y. Zheng, AutoSTG: Neural architecture search for predictions of spatio-temporal graph, in *Proc. Web Conf. 2021*, Ljubljana, Slovenia, 2021, pp. 1846–1855.

[44] X. Zhang, C. Huang, Y. Xu, L. Xia, P. Dai, L. Bo, J. Zhang, and Y. Zheng, Traffic flow forecasting with spatial-temporal graph diffusion network, in *Proc. 35th AAAI Conf. Artificial Intelligence*, Palo Alto, CA, USA, 2021, pp. 15008–15015.

[45] J. Ye, L. Sun, B. Du, Y. Fu, and H. Xiong, Coupled layer-wise graph convolution for transportation demand prediction, in *Proc. 35th AAAI Conf. Artificial Intelligence*, Palo Alto, CA, USA, 2021, pp. 4617–4625.

[46] G. Jin, L. Liu, F. Li, and J. Huang, Spatio-temporal graph neural point process for traffic congestion event prediction, in *Proc. 37th AAAI Conf. Artificial Intelligence*, Washington, DC, USA, 2023, pp. 14268–14276.

[47] Z. Wang, R. Jiang, H. Xue, F. D. Salim, X. Song, and R. Shibasaki, Event-aware multimodal mobility nowcasting, in *Proc. 36th AAAI Conf. Artificial Intelligence*, Virtual Event, 2022, pp. 4228–4236.

[48] K. Fu, F. Meng, J. Ye, and Z. Wang, CompactETA: A fast inference system for travel time prediction, in *Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, Virtual Event, 2020, pp. 3337–3345.

[49] X. Mo, Z. Huang, Y. Xing, and C. Lv, Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network, *IEEE Trans. Intell. Transp. Syst*, vol. 23, no. 7, pp. 9554–9567, 2022.

[50] J. Han, H. Liu, H. Zhu, H. Xiong, and D. Dou, Joint air quality and weather prediction based on multi-adversarial spatiotemporal networks, in *Proc. 35th AAAI Conf. Artificial Intelligence*, Palo Alto, CA, USA, 2021, pp. 4081–4089.

[51] H. Lin, Z. Gao, Y. Xu, L. Wu, L. Li, and S. Z. Li, Conditional local convolution for spatio-temporal meteorological forecasting, in *Proc. 36th AAAI Conf. Artificial Intelligence*, Virtual Event, 2022, pp. 7470–7478.

[52] S. Chen, J. A. Zwart, and X. Jia, Physics-guided graph meta learning for predicting water temperature and streamflow in stream networks, in *Proc. 28th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, Washington, DC, USA, 2022, pp. 2752–2761.

[53] L. Xia, C. Huang, Y. Xu, P. Dai, L. Bo, X. Zhang, and T. Chen, Spatial-temporal sequential hypergraph network for crime prediction with dynamic multiplex relation learning, in *Proc. 30th Int. Joint Conf. Artificial Intelligence*, Virtual Event, 2021, pp. 1631–1637.

[54] X. Yang, F. Zhang, P. Sun, X. Li, Z. Du, and R. Liu, A spatio-temporal graph-guided convolutional LSTM for tropical cyclones precipitation nowcasting, *Appl. Soft Comput.*, vol. 124, p. 109003, 2022.

[55] A. Doğan and E. Demir, Structural recurrent neural network models for earthquake prediction, *Neural Comput. Appl.*, vol. 34, no. 13, pp. 11049–11062, 2022.

[56] S. Deng, S. Wang, H. Rangwala, L. Wang, and Y. Ning, Cola-GNN: Cross-location attention based graph neural networks for long-term ILI prediction, in *Proc. 29th ACM Int. Conf. Information & Knowledge Management*, Virtual Event, 2020, pp. 245–254.

[57] G. Panagopoulos, G. Nikolentzos, and M. Vazirgiannis, Transfer graph neural networks for pandemic forecasting, in *Proc. 35th AAAI Conf. Artificial Intelligence*, Palo Alto, CA, USA, 2021, pp. 4838–4845.

[58] Z. Wang, T. Xia, R. Jiang, X. Liu, K. S. Kim, X. Song, and R. Shibasaki, Forecasting ambulance demand with profiled human mobility via heterogeneous multi-graph neural networks, in *Proc. 2021 IEEE 37th Int. Conf. Data Engineering*, Chania, Greece, 2021, pp. 1751–1762.

[59] S. Feng, G. Cong, B. An, and Y. M. Chee, POI2Vec: Geographical latent representation for predicting future visitors, in *Proc. 31st AAAI Conf. Artificial Intelligence*, San Francisco, CA, USA, 2017, p. 102–108.

[60] L. Deng, Y. Zhao, Z. Fu, H. Sun, S. Liu, and K. Zheng, Efficient trajectory similarity computation with contrastive

learning, in *Proc. 31st ACM Int. Conf. Information & Knowledge Management*, Atlanta, GA, USA, 2022, pp. 365–374.

[61] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, Deep graph infomax, in *Proc. 7th Int. Conf. Learning Representations*, New Orleans, LA, USA, https://dblp.org/db/conf/iclr/iclr2019.html#VelickovicFHLBH19, 2019.

[62] Y. Jiao, Y. Xiong, J. Zhang, Y. Zhang, T. Zhang, and Y. Zhu, Sub-graph contrast for scalable self-supervised graph representation learning, in *Proc. 2020 IEEE Int. Conf. Data Mining*, Sorrento, Italy, 2021, pp. 222–231.

[63] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, and J. Tang, GraphMAE: Self-supervised masked graph autoencoders, in *Proc. 28th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, Washington, DC, USA, 2022, pp. 594–604.

[64] J. Li, R. Wu, W. Sun, L. Chen, S. Tian, L. Zhu, C. Meng, Z. Zheng, and W. Wang, What's behind the mask: Understanding masked graph modeling for graph autoencoders, in *Proc. 29th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, Long Beach, CA, USA, 2023, pp. 1268–1279.

[65] W. C. Kang and J. McAuley, Self-attentive sequential recommendation, in *Proc. 2018 IEEE Int. Conf. Data Mining*, Singapore, 2018, pp. 197–206.

Hepeng Gao received the BEng degree from Inner Mongolia University, China in 2016, and the MEng degree from Jilin University, China in 2019. He is currently a PhD candidate at Jilin University, China. His research interests include data mining applications, recommender systems, urban computing, and graph neural networks.

Xingliang Zhang received the BEng degree from Jilin University, China in 2010. He is an engineer at China Mobile Group Jilin Co., Ltd. His main research interests include mobile communication data analysis and big data analysis of 5G.

Yongjian Yang received the BEng degree in automatization from Jilin University of Technology, China in 1983, the MEng degree in computer communication from Beijing University of Post and Telecommunications, China in 1991, and the PhD degree in softwareand theory of computer from Jilin University, China in 2005. He is currently a professor and a PhD supervisor at Jilin University, the director of Key lab under the Ministry of Information Industry, and a member of the Computer Science Academy of Jilin Province. His research interests include network intelligence management, wireless mobile communication and services, and wireless mobile communication.

Funing Yang received the BEng degree from Jilin University, China in 2010, the MEng degree from Beijing University of Posts and Telecommunications, China in 2013, and the PhD degree from Jilin University, China in 2022. She is currently an associate professor at Jilin University, China. Her research interests include mobile CrowdSensing, mobile computing, integration, and mining of massive data.

Yijun Su received the PhD degree from University of Chinese Academy of Sciences, China in 2020. He is currently a senior researcher at JD iCity, China. His research interests include spatio-temporal data mining, federated learning, and urban computing.