Automated Spatio-Temporal Graph Contrastive Learning

Qianru Zhang
The University of Hong Kong
qrzhang@cs.hku.hk

Chao Huang*
The University of Hong Kong
chaohuang75@gmail.com

Lianghao Xia
The University of Hong Kong
lhxia@cs.hku.hk

Zheng Wang
Huawei Singapore Research Center
wangzheng155@huawei.com

Zhonghang Li
South China University of Technology
cszhonghang.li@mail.scut.edu.cn

Siuming Yiu
The University of Hong Kong
smyiu@cs.hku.hk

ABSTRACT

Among various region embedding methods, graph-based region relation learning models stand out, owing to their strong structure representation ability for encoding spatial correlations with graph neural networks. Despite their effectiveness, several key challenges have not been well addressed in existing methods: i) Data noise and missing are ubiquitous in many spatio-temporal scenarios due to a variety of factors. ii) Input spatio-temporal data (*e.g.*, mobility traces) usually exhibits distribution heterogeneity across space and time. In such cases, current methods are vulnerable to the quality of the generated region graphs, which may lead to suboptimal performance. In this paper, we tackle the above challenges by exploring the Automated Spatio-Temporal graph contrastive learning paradigm (AutoST) over the heterogeneous region graph generated from multi-view data sources. Our AutoST framework is built upon a heterogeneous graph neural architecture to capture the multi-view region dependencies with respect to POI semantics, mobility flow patterns and geographical positions. To improve the robustness of our GNN encoder against data noise and distribution issues, we design an automated spatio-temporal augmentation scheme with a parameterized contrastive view generator. AutoST can adapt to the spatio-temporal heterogeneous graph with multi-view semantics well preserved. Extensive experiments for three downstream spatio-temporal mining tasks on several real-world datasets demonstrate the significant performance gain achieved by our AutoST over a variety of baselines. The code is publicly available at https://github.com/HKUDS/AutoST.

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; **Data mining**; • **Computing methodologies** → **Neural networks**;

KEYWORDS

Spatio-temporal data mining; Contrastive learning; Self-supervised learning; Graph neural networks; Urban computing

*Chao Huang is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WWW'23, May 1–5, 2023, Austin, TX, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583304

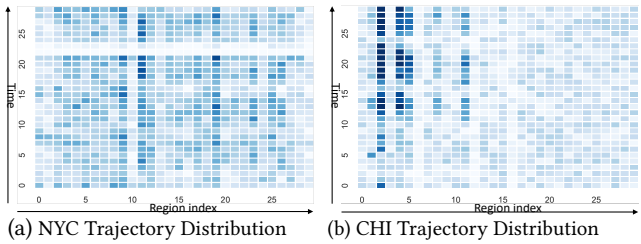
ACM Reference Format:
Qianru Zhang, Chao Huang, Lianghao Xia, Zheng Wang, Zhonghang Li, and Siuming Yiu. 2023. Automated Spatio-Temporal Graph Contrastive Learning. In *Proceedings of the ACM Web Conference 2023 (WWW'23), May 1–5, 2023, Austin, TX, USA.* ACM, Austin, TX, USA, 11 pages. https://doi.org/10.1145/3543507.3583304

1 INTRODUCTION

With the advancement of remote sensing technologies and large-scale computing infrastructure, different types of spatio-temporal data are being collected at unprecedented scale from various domains (*e.g.*, intelligent transportation [22], environmental science [18] and public security [5]). Such diverse spatio-temporal data drives the need for effective spatio-temporal prediction frameworks for various urban sensing applications, such as traffic analysis [28], human mobility behavior modeling [15], and citywide crime prediction [8]. For instance, motivated by the opportunities of building machine learning and big data driven intelligent cities, the discovered human trajectory patterns can help to formulate better urban planning mechanisms [3], or understanding the dynamics of crime occurrences is useful for reducing crime rate [12, 26].

To tackle the challenges in urban sensing scenarios, many efforts have been devoted to developing region representation learning techniques, for studying how to learn the complex region dependence structures come in both temporal and spatial dimensions from geographical datasets [27]. Instead of the extracting hand-crafted region feature design, these region embedding methods enable the automatic discovery of spatial relations among different regions from the collected spatio-temporal data [4, 34]. Among various region representation methods, graph-based learning models stand out owing to the strong feature representation effectiveness over the graph-based region relationships [4, 38, 40]. Towards this line, Graph Neural Networks (GNNs), have been utilized as a powerful modeling scheme for aggregating the complex relational information along with the graph connections, such as MV-PN [4] and MVURE [38]. Specifically, these methods take inspiration from Graph Auto-Encoder (GAE) [10] and Graph Attention Network [24], by following the graph-structured message passing to perform neighborhood feature aggregation. Despite the effectiveness of the existing region representation approaches, we identify two key challenges that have not been well addressed in previous work.

Data Noise and Incompleteness. Due to a variety of factors (*e.g.*, high cost of device deployment, or sensor failure), data noise and incompleteness are ubiquitous in the collected spatio-temporal data [11, 13]. That is, for graph-based representation methods, the pre-generated spatial graph often contain noisy information with



(a) NYC Trajectory Distribution (b) CHI Trajectory Distribution

Figure 1: Trajectory data with spatio-temporal distribution heterogeneity in New York and Chicago on Oct 2016.

weak dependency between connected regions. For example, a region may not have strong correlations with all its neighboring regions, due to different region functionalities in a city. Therefore, directly aggregating the spatial and temporal information over the generated spatial graph along with the less-relevant region connections will involve the task-irrelevant noisy signals for downstream spatio-temporal mining applications (*e.g.*, crime prediction, traffic analysis).

Spatio-Temporal Distribution Heterogeneity. spatio-temporal data (*e.g.*, human mobility flow) used in current methods for region graph construction may exhibit distribution heterogeneity across space and time dimension. Such data distribution heterogeneity hinders the representation capability of current neural network models, which leads to the suboptimal region representation performance. To get a better understanding of the skewed data distribution across different regions in urban space, we show the distribution of human mobility trajectories in New York City and Chicago in Figure 1. In those figures, the diverse distributions of human mobility data across different geographical regions can be observed, which brings challenges to generate accurate region graph connections to reflect complex spatial correlations based on the mobility traces.

Recently, a series of contrastive-based self-supervised learning (SSL) methods have achieved outstanding performance to address the data noise and label shortage issues, for various tasks in Computer Vision [21], Nature Language Processing [39], and Network Embedding [41]. The core idea of contrastive learning is to generate augmented SSL signals by i) shortening the distance between positive and anchor example in vector representation space; ii) widening the distance between negative and anchor examples [23, 31].

Additionally, the spatio-temporal data distributions are also influenced by heterogeneous external factors in urban data scenarios, such as region functionality, environmental attributes, and time-evolving traffic flow. These diverse datasets are exhibited with multi-view in nature. Each data dimension corresponds to view-specific characteristics and semantic. Therefore, how to effectively incorporate such heterogeneous data sources into the region representation framework and fuse knowledge from different views, remains a significant challenge.

In light of these aforementioned motivations and challenges, we propose an Automated Spatio-Temporal graph contrastive learning (AutoST) paradigm to provide effective and robust region representations with multi-view spatio-temporal data in urban sensing scenario. Specifically, the spatio-temporal heterogeneous graph neural encoder is firstly introduced to model spatio-temporal patterns by capturing region-wise dependencies from both the intra-view and inter-view data correlations. In our self-supervised learning paradigm, the encoded multi-view region embeddings will serve as the guidance for our augmentation schema to automatically distill the heterogeneous dependencies across all regions in urban space. Our graph contrastive learning component not only enhances the learning process of implicit cross-region relations, but also plays an important role in effectively incorporating auxiliary self-supervised signals in an adaptive manner. Conceptually, our AutoST is advantageous over the existing graph contrastive learning methods in enabling the automatic discovery of useful spatio-temporal SSL signals for more effective data augmentation. Some of current methods directly leverage the random mask operations, which may dropout some useful graph connections and is still vulnerable to noise perturbations.

We highlight the key contributions of this work as follows:

- **General Aspect**. We emphasize the importance of jointly tackling the data noise and distribution heterogeneity challenges in spatio-temporal graph learning task. Towards this end, we bring the superiority of contrastive learning into region representation with the automated distilled self-supervised signals, to enhance the robustness and generalization ability of our spatio-temporal heterogeneous graph neural architecture.
- **Methodologies**. We first design spatio-temporal heterogeneous graph neural network to simultaneously capture the intra-view and inter-view region dependencies with the modeling of various spatio-temporal factors. To address the data noise and distribution diversity issues, we perform the spatio-temporal contrastive learning over the region dependency graph for automated data augmentation. Our automated contrastive learning framework is built upon a variational spatio-temporal graph auto-encoder with our designed parameterized region dependency learner.
- **Experiments Findings**. We validate our AutoST in different spatio-temporal mining tasks, by competing with various region representation and graph neural network baselines.

2 PRELIMINARIES

In this work, we divide the entire urban area into I (indexed by i) geographical regions. To embed each region into latent representation space, we consider three types of spatio-temporal data sources generated from previous T time slots (*e.g.*, days) (indexed by t) as model inputs, which are elaborated with the following definitions:

Definition 1. ***Regional Point-of-Interests (POIs) Matrix*** $\mathcal{P}$. *In urban space, Point-of-Interest information describes the regional functionalities by referring to different categories of geographical places, e.g., restaurant, hotel, medical center and tourist attraction. We define a POI matrix* $\mathcal{P} \in \mathbb{R}^{I \times C}$, *in which* C *denotes the number of POI categories. In this work, 50 and 130 POI categories are utilized to generate matrix* $\mathcal{P}$ *in New York City and Chicago, respectively. Each element* $p_{i,c}$ *in* $\mathcal{P}$ *represents the number of places located in the region* r_i *with the* c*-th category of POIs. Given the generated* $\mathcal{P}$, *we can associate each region* r_i *with the POI vector* $\mathcal{P}_i \in \mathbb{R}^{1 \times C}$.

Definition 2. ***User Mobility Trajectories*** $\mathcal{M}$. *To reflect the urban dynamics with human mobility data, we collect a set of user mobility trajectories to record human movement. Particularly, each trajectory record is in the format of* (r_s, r_d, t_s, t_d), *where* r_s *and* r_d *denotes the source and destination regions of this trajectory.* t_s *and* t_d

indicate the starting and end timestamp of this trajectory, respectively.

Definition 3. ***Region-wise Geographical Distance Matrix*** *$\mathcal{D}$. We further define a geographical distance matrix $\mathcal{D} \in \mathbb{R}^{I \times I}$ to represent the geographical distance between each pair of regions (e.g., r_i and $r_{i'}$) in the spatial space of a city, based on the centre coordinates (latitude and longitude) of regions.*

Problem Statement: Our spatio-temporal graph learning task is: **Input**: Given the POI matrix $\mathcal{P} \in \mathbb{R}^{I \times C}$, the set of user mobility trajectories $\mathcal{M}$, and the geographical distance matrix $\mathcal{D} \in \mathbb{R}^{I \times I}$. **Output**: Our goal is to learn a mapping function $f : \boldsymbol{r_i} \rightarrow \mathbb{R}^d$ that encodes each region r_i into a low-dimensional embedding with the dimensionality of d. The spatial and temporal dependencies between different regions and time slots can be well preserved in the embedding space, benefiting for various urban sensing applications (*e.g.*, traffic forecasting, crime prediction).

3 METHODOLOGY

In this section, we present the technical details of our AutoST framework. The overall model architecture is illustrated in Figure 2.

3.1 POI Context Embedding Layer

To encode the Point-of-Interest context into region embeddings, we design a POI context embedding layer to reflect the regional functional information into latent representation space. Motivated by the POI token embedding with Skip-gram [17], we feed the region-specific POI vector into the Skip-gram model [2] for POI context embedding. After that, we concatenate the region-specific POI embeddings and generate the POI-aware representations: $\bar{\mathbf{E}} =$ MLP(Skip-gram($\mathcal{P}$)). Here, we adopt a Multi-Layer Perceptron to perform the embedding projection. $\mathbf{E} \in \mathbb{R}^{I \times d}$ denotes the embedding table for all regions, where d is the hidden dimensionality. The embedding $\mathbf{e}_i$ of each region preserves the region-specific POI contextual information into latent semantic space. To endow our POI embedding layer with the modeling of region interactions, we further design the region-wise self-attention mechanism as:

$$\mathbf{e}_i = \Big\Vert_{h=1}^{H} \sum_{j=1}^{I} \alpha_{i,j}^h \cdot \mathbf{V}^h \mathbf{e}_j; \quad \alpha_{i,j}^h = \delta\left(\frac{(\mathbf{Q}^h \mathbf{e}_i)^\top (\mathbf{K}^h \mathbf{e}_j)}{\sqrt{d/H}}\right) \tag{1}$$

where $\mathbf{V}^h, \mathbf{Q}^h, \mathbf{K}^h \in \mathbb{R}^{d/H \times d}$ denote the value, query, and key transformations for the h-th attention head, respectively. H denotes the number of heads and $\Big\Vert$ denotes concatenating the H vectors. The softmax function $\delta(\cdot)$ is applied. In our self-attention layer, the pairwise interaction between region r_i and r_j can be captured in embedding $\mathbf{e}_i$ with respect to region-wise POI semantics.

3.2 Spatio-Temporal Heterogeneous Graph Neural Network

To comprehensively capture the dynamic region dependencies from different views, *e.g.*, regional functions, traffic flow, and spatial locations, we propose a heterogeneous spatio-temporal GNN. In city-wide scenario, our AutoST firstly aims to model the dependencies among different regions in terms of individual representation view. In addition to the importance of cross-region relation modeling, another key dimension is to uncover the inter-correlations among view-specific embeddings of a specific region. For example, implicit dependencies between the POI information and traffic flow of a region are ubiquitous in urban space. Towards this end, in our region graph encoder, both the intra-view and inter-view message passing is performed over the constructed multi-view region graph.

3.2.1 **Heterogeneous Region Graph with Multi-View Relations.** We generate our heterogeneous region graph with multi-typed data views by integrating three view-specific region graphs:

- **POI-aware Region Graph** $\mathcal{G}_p$. We construct a region graph $\mathcal{G}_p$ with the previous encoded Point-of-Interest region embeddings $\mathbf{e}_i$ $(1 \leq i \leq I)$. To be specific, if the similarity between two region embeddings $\mathbf{e}_i$ and $\mathbf{e}_j$ is larger than the threshold ϵ^p, there exists an edge connecting region nodes r_i and r_j in graph $\mathcal{G}_p$. Here, the cosine function is applied to measure the embedding similarities.
- **Trajectory-based Region Graph** $\mathcal{G}_m$. In addition to the stationary POI information, we leverage the time-aware human mobility trajectories to correlate regions, to be reflective of urban flow dynamics over time. We decompose each region r_i into T time slot-specific region nodes (r_i^t). In particular, given the record of mobility trajectory (r_s, r_d, t_s, t_d), the source region $r_s^{t_s}$ within the t_s-th time slot and destination region $r_d^{t_d}$ within the t_d-th time slot will be connected with an edge.
- **Distance-based Region Graph** $\mathcal{G}_d$. To inject the geographical positional information into the graph encoder, we generate a distance-based region graph $\mathcal{G}_d$ by connecting regions in terms of their geographical distance estimated by their coordinates. Specifically, an edge will be added between region r_i and r_j if their distance is smaller than the threshold ϵ^d.

Heterogeneous Region Graph Construction. Given the above view-specific region graphs $\mathcal{G}_p$, $\mathcal{G}_m$ and $\mathcal{G}_d$, we construct the heterogeneous region graph $\mathcal{G}$ to enable the cross-view region connections and the region temporal self-discrimination connections.

3.2.2 **Relation-aware Message Passing Paradigm.** To capture both the intra-view and inter-view region dependencies based on different types of spatio-temporal data, AutoST conducts message passing over the heterogeneous region graph $\mathcal{G}$ for region embedding. Without loss of generality, we formally present our relation-aware message passing with spatio-temporal connections as follows:

$$\mathbf{h}_i^{(l)} = \sigma\Big(\sum_{\gamma \in \Gamma} \sum_{j \in \mathcal{N}_i^\gamma} \alpha_{i,\gamma} \mathbf{W}_\gamma^{(l-1)} \mathbf{h}_j^{(l-1)}\Big); \;\; \alpha_{i,\gamma} = \frac{1}{|\mathcal{N}_i^\gamma|} \tag{2}$$

where $\mathcal{N}_i^\gamma$ denotes neighbour indices set of region r_i via relation γ $(\gamma \in \Gamma)$. Γ is the relation set. $\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l-1)} \in \mathbb{R}^d$ represents the embedding vectors for the i-th region vertex in the l-th graph layer and the j-th region vertex in the $(l-1)$-th graph neural layer, respectively. Here, $\mathbf{h}_i^{(0)}$ is initialized with $\mathbf{e}_i$ which is derived by Eq 1. $\sigma(\cdot)$ denotes the ReLU activation function. $\alpha_{i,\gamma} \in \mathbb{R}$ denotes the normalization weight for region vertex pair (r_i, r_j), which is calculated using the degrees of the vertices. $\mathbf{W}_\gamma^{(l-1)} \in \mathbb{R}^{d \times d}$ denotes the learning weights for the $(l-1)-th$ iteration. To fully make use of the information gathered from multi-hop neighborhood, AutoST aggregates the multi-order embeddings together. The cross-view

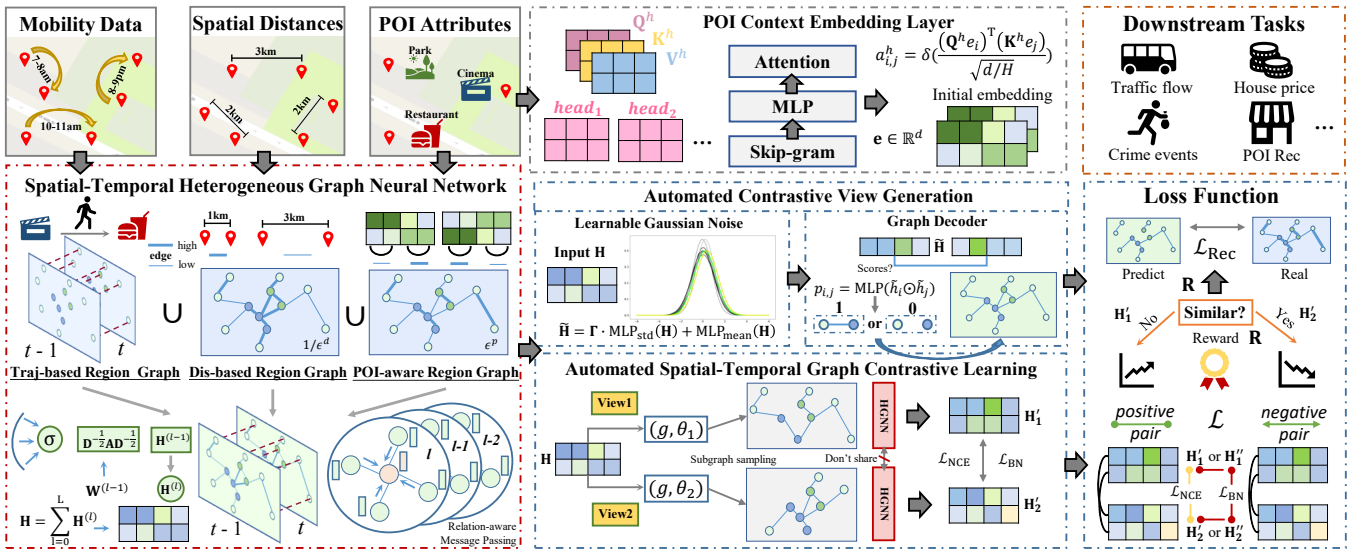


Figure 2: Architecture of our AutoST region representation model with automated spatio-temporal graph contrastive learning.

message passing process is given in matrix form shown below:

$$\mathbf{H} = \sum_{l=0}^{L} \mathbf{H}^{(l)}; \quad \mathbf{H}^{(l)} = \sigma(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{H}^{(l-1)}\mathbf{W}_{\gamma}^{(l-1)\top}) \tag{3}$$

where $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the embedding matrix, the rows of which are regional embedding vectors $\mathbf{h}_i$. L is the number of graph iterations. $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ denotes the adjacent matrix with self-loop. $\mathbf{D} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the corresponding diagonal degree matrix.

3.3 Spatio-Temporal GCL

In this work, AutoST focuses on exploiting the spatio-temporal graph contrastive learning paradigm to tackle the challenges of data noise and distribution heterogeneity. Specifically, our heterogeneous spatio-temporal multi-view graph neural network may be sensitive to the quality of the constructed multi-view region graph $\mathcal{G}$. The message passing scheme may easily aggregate noisy information, which hinders the spatio-temporal representation quality. For example, a region may be connected to another one with close distance but dissimilar region functions, *e.g.*, shopping mall and residential zone. Hence, aggregating the information from irrelevant neighboring nodes in graph $\mathcal{G}$ will impair the region embedding quality, and weaken the representation performance.

Inspired by graph contrastive learning (GCL) in [36], we propose to generate the contrastive views for spatio-temporal data augmentation in an automatic way. With the designed GCL principle, the graph connection noise issue can be alleviated by incorporating the automated self-supervised signals over the multi-view region graph. In our AutoST, we develop a variational graph auto-encoder as the graph structure generator for augmentation.

3.3.1 **Variational Graph Encoder.** Inspired by [36], our AutoST adopts Variational Graph Auto-Encoder (VGAE) [10] with random walk [14] for data augmentation, due to the strength of VGAE in considering the data distribution for reconstruction. To begin with, the VGAE module learns the encoding mapping $\mathcal{G} \rightarrow \mathbb{R}^{|\mathcal{V}|\times d}$ to capture the hierarchical graph structures in low-dimensional node embeddings $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}|\times d}$, as defined by Eq 3. Then, AutoST generates contrastive views in the original graph space by adding Gaussian noises to the low-dimensional graph representations (*i.e.* the encoded node embeddings), formally as: $\tilde{\mathbf{H}} = \mathbf{\Gamma} \cdot \text{MLP}_{\text{std}}(\mathbf{H}) + \text{MLP}_{\text{mean}}(\mathbf{H})$, where $\tilde{\mathbf{H}} \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the representations for the generated graph view. $\mathbf{\Gamma} \in \mathbb{R}^{|\mathcal{V}|\times d}$ denotes the noise matrix, whose elements $\gamma \sim \text{Gaussian}(\mu, \sigma)$, with μ and σ denoting hyperparameters for the mean and the standard deviation. To realize learnable view generation, AutoST employs two-layer MLPs with trainable parameters (*i.e.* $\text{MLP}_{\text{mean}}(\cdot)$ and $\text{MLP}_{\text{std}}(\cdot)$) to calculate the mean and the standard deviation from $\mathbf{H}$.

3.3.2 **Automated Contrastive View Generation.** With the generated representations for graph structures, AutoST then calculates the sampling scores *w.r.t* the generated view for each node pair, through element-wise product and a MLP. Formally, for node with indices i and j, the sampling score $p_{i,j} \in \mathbb{R}$ is calculated by: $p_{i,j} = \text{MLP}(\tilde{\mathbf{h}}_i \odot \tilde{\mathbf{h}}_j)$, where $\odot$ denotes the element-wise product. $\tilde{\mathbf{h}}_i, \tilde{\mathbf{h}}_j \in \mathbb{R}^d$ refer to the i-th and the j-th rows of the embedding matrix $\tilde{\mathbf{H}}$. The probability scores $p_{i,j}$ for all node pairs compose a sampling matrix $\mathbf{P} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$. Next, AutoST sparsify this matrix to acquire a binary adjacent matrix depicting the graph structures of the generated graph view, as $\tilde{\mathbf{P}} = \{\tilde{p}_{i,j}\}$, where $\tilde{p}_{i,j} = 0$ if $p_{i,j} < \epsilon$ else 1. ϵ is a hyperparameter for threshold. Here, $\tilde{\mathbf{P}}$ is the decoded graph given by our variational graph encoder-decoder. To conduct graph CL efficiently, we apply the random walk algorithm on the decoded graph to generate sub-graphs with less nodes and edges as the contrastive views. Specifically, for dual-view generation, AutoST employs two VGAE modules with the same architecture but non-shared parameters, targeting the two different contrastive views. The decoders generate two graphs, based on which the random walker generates sub-graphs using the same set of seed nodes. Formally, the augmented multi-view

graphs are denoted by $\mathcal{G}_1', \mathcal{G}_2'$ for the two adaptive contrastive representation views. Different from recent graph contrastive learning models with random augmentation operators (*e.g.*, edge perturbation, edge masking, and node dropping [29, 35]), we augment the spatio-temporal graph learning with the learnable region dependencies. Therefore, our AutoST enhances the robustness of region representation by adaptively offering auxiliary SSL signals.

3.3.3 **Contrastive Objective Optimization for Augmentation.** As discussed before, random contrastive view generation methods inevitably lose important structure information and cannot effectively alleviate the side effect caused by noisy and skewed data distribution. To this end, the view generation of our AutoST is augmented with parameterized (*i.e.*, MLP_{mean} and MLP_{std}) encoder-decoder architecture and is thus adjustable for contrastive augmentation. To train the learnable VGAE, we calculate the contrastive-based reward for the generated views and combine it with the reconstruction loss of the VGAE. Specifically, the optimization goal for the graph sampling can be formalized as:

$$\min_{\theta_1,\theta_2} \mathrm{R}(\mathcal{G}, \theta_1, \theta_2) \cdot \Big(\mathcal{L}_{\text{Rec}}(\mathcal{G}, \theta_1) + \mathcal{L}_{\text{Rec}}(\mathcal{G}, \theta_2) \Big) \tag{4}$$

where θ_1, θ_2 denote the learnable parameters of the two graph samplers for the contrastive views. $\mathrm{R}(\mathcal{G}, \theta_1, \theta_2)$ denotes the reward function based on mutual information maximization and minimization. $\mathcal{L}_{\text{Rec}}(\cdot)$ denotes the reconstruction loss of VGAE that optimizes the model to rebuild the node-wise connections and the edge-related features from the compressed low-dimensional embeddings $\tilde{\mathbf{H}}$.

The principle of assigning reward is to return larger values when the CL loss is big, in which case the view generator has identified challenging contrastive samples. When the CL loss is small, the reward function should return small values to indicate deficiency in the generated views. Instead of using the CL loss as reward directly, AutoST further expands the differences between reward values by minimizing the mutual information between the above two cases. With the Information Minimization (InfoMin) [20], the reward function is defined as the following discontinuous function:

$$\mathrm{R}_1(\mathcal{G}, \theta_1, \theta_2) = \begin{cases} 1, & \text{if } \mathcal{L}(\mathcal{G}, \theta_1, \theta_2) > \epsilon' \\ \xi \ll 1 & \text{otherwise} \end{cases} \tag{5}$$

where ϵ' denotes the hyperparameter for threshold, and ξ is a pre-selected small value. $\mathcal{L}(\mathcal{G}, \theta_1, \theta_2)$ denotes the contrastive loss function, which will be elaborated later. To enhance our contrastive learning, we further incorporate another dimension of reward signal presented as follows:

$$\mathrm{R}_2(\mathcal{G}, \theta_1, \theta_2) = 1 - \cos(\mathbf{H}_1', \mathbf{H}_2'). \tag{6}$$

In concrete, AutoST conducts the heterogeneous graph encoding (Eq 3) on the sub-graph samples in the two views (*i.e.* $\mathcal{G}_1', \mathcal{G}_2'$) respectively: $\mathbf{H}_1' = \text{HGNN}(\mathcal{G}_1')$ and $\mathbf{H}_2' = \text{HGNN}(\mathcal{G}_2')$ where $\text{HGNN}(\cdot)$ denotes the heterogeneous graph neural network. $\mathbf{H}_1' \in \mathbb{R}^{|\mathcal{V}_1'| \times d}, \mathbf{H}_2' \in \mathbb{R}^{|\mathcal{V}_2'| \times d}$ represent the resulting node embedding matrices, where $\mathcal{V}_1'$ and $\mathcal{V}_2'$ denote the two generated views, respectively. $\mathrm{R}(\mathcal{G}, \theta_1, \theta_2)$ is finally defined: $\mathrm{R}(\mathcal{G}, \theta_1, \theta_2) = w_1 \mathrm{R}_1(\mathcal{G}, \theta_1, \theta_2) + (1 - w_1) \mathrm{R}_2(\mathcal{G}, \theta_1, \theta_2)$. The reconstruction loss $\mathcal{L}_{\text{Rec}}(\cdot)$ in Eq 4 is calculated by minimizing the difference between the reconstructed graph information and the original graph. To be specific, to reconstruct edges, AutoST minimizes the loss: $\mathcal{L}_{\text{Rec}}(\mathcal{G}, \theta) = -\sum_{(i,j) \in \mathcal{E}} \log(\text{sigm}(p_{i,j})) - \sum_{(i,j) \notin \mathcal{E}} \log(1 - \text{sigm}(p_{i,j}))$. $\text{sigm}(\cdot)$ denotes the sigmoid function. $p_{i,j}$ is the non-binary sampling score for edge (i, j).

3.4 Model Training with Contrastive Learning

Our AutoST aims to obtain high-quality region embeddings by pretraining the model with self-supervised learning tasks. The optimization goal of AutoST can be formalized as follows:

$$\begin{aligned} \mathcal{L}(\mathcal{G}, \theta_1, \theta_2) = \beta \mathcal{L}_{\text{NCE}}(\mathcal{G}, \theta_1, \theta_2) \\ + (1 - \beta) \mathcal{L}_{\text{BN}}(\mathcal{G}, \theta_p, \theta_2) \end{aligned} \tag{7}$$

where $\mathcal{L}$ denotes the self-supervised loss, which is composed of two terms: the InfoNCE loss [16] $\mathcal{L}_{\text{NCE}}$, and the Information BottleNeck (InfoBN) $\mathcal{L}_{\text{BN}}$. The two terms are balanced by the scalar hyperparameter β. InfoNCE is a commonly-used contrastive learning goal which pulls close the representations for the same entity in two different views, and pushes away the embeddings for different entities in the two views. Based on the embeddings $\mathbf{H}_1'$ and $\mathbf{H}_2'$, the InfoNCE term is formally defined by:

$$\mathcal{L}_{\text{NCE}} = \sum_{i \in \mathcal{V}_1 \cap \mathcal{V}_2} -\log \frac{\exp\Big(\cos(\mathbf{H}_1'(i), \mathbf{H}_2'(i))/\tau\Big)}{\sum\limits_{j \in \mathcal{V}_1 \cap \mathcal{V}_2} \exp\Big(\cos(\mathbf{H}_1'(i), \mathbf{H}_2'(j))/\tau\Big)}$$

where τ is the temperature coefficient. $\mathbf{H}'(i) \in \mathbb{R}^d$ denotes the embedding vector in the i-th row. Here, we use cosine similarity to measure the distance between different representations. Only nodes in both graph views $\mathcal{G}_1$ and $\mathcal{G}_2$ are involved in this loss function.

The self-supervised learning of AutoST is additionally enhanced by the information bottleneck, which further reduce the redundancy in the embeddings from the two views. Specifically, InfoBN generates low-dimensional representations for both views, and minimizes the mutual information between the original contrastive views and their corresponding representations. Formally, InfoBN is to minimize the following loss:

$$\begin{aligned} \mathcal{L}_{\text{BN}} = &- \sum_{i \in \mathcal{V}_1} \log \frac{\exp\Big(\cos(\mathbf{H}_1'(i), \mathbf{H}_1''(i))/\tau\Big)}{\sum\limits_{j \in \mathcal{V}_1} \exp\Big(\cos(\mathbf{H}_1'(i), \mathbf{H}_1''(j))/\tau\Big)} \\ &- \sum_{i \in \mathcal{V}_2} \log \frac{\exp\Big(\cos(\mathbf{H}_2'(i), \mathbf{H}_2''(i))/\tau\Big)}{\sum\limits_{j \in \mathcal{V}_2} \exp\Big(\cos(\mathbf{H}_2'(i), \mathbf{H}_2''(j))/\tau\Big)} \end{aligned} \tag{8}$$

where $\mathbf{H}_1'' \in \mathbb{R}^{|\mathcal{V}_1| \times d}, \mathbf{H}_2'' \in \mathbb{R}^{|\mathcal{V}_2| \times d}$ denote the corresponding newly-generated views. This InfoBN regularization diminishes the superfluous information in both views by contrasting them with randomly-augmented representations.

4 EVALUATION

Evaluation is performed to answer the following research questions:

- **RQ1.** How does AutoST perform compared with various baselines on different spatio-temporal learning applications?
- **RQ2.** How the different data views and contrastive learning components affect the region representation performance?

- **RQ3.** How does our AutoST perform in representation learning over regions with different data sparsity degrees?
- **RQ4.** What are the benefits of our spatial and temporal dependency modeling across regions with learned representations?
- **RQ5.** How do different settings of hyperparameters affect AutoST's region representation performance? (Appendix Section)

4.1 Experimental Setup

4.1.1 **Datasets and Protocols.** We evaluate our AutoST framework on three spatio-temporal mining tasks, *i.e.*, crime prediction, traffic flow forecasting, house price prediction, with several real-world datasets collected from Chicago and New York City. Following the settings in [32], different types of crimes are included in Chicago (Theft, Battery, Assault, Damage) and NYC (Burglary, Larceny, Robbery, Assault) datasets. In Appendix Section, we present detailed data description and summarize the data statistics in Table 5. We adopt three commonly-used evaluation metrics *MAE*, *MAPE* and *RMSE*, to measure the accuracy in forecasting tasks on urban crimes, citywide traffic flow, and regional house price.

4.1.2 **Baselines for Comparison.** are presented as follows:
Graph representation methods:

- **Node2vec [6]**. It is a representative network embedding method to encode graph structural information using the random walk-based skip-gram model for embedding nodes.
- **GCN [9]**. The Graph Convolutional Network is a representative GNN architecture to perform the convolution-based message passing between neighbour nodes.
- **GraphSage [7]**. It enables the information aggregation from the sampled sub-graph structures, so as to improve the computational and memory cost of graph neural networks.
- **GAE [10]**. Graph Auto-encoder is designed to map nodes into latent embedding space with the input reconstruction objective.
- **GAT [24]**. Graph Attention Network enhances the discrimination ability of graph neural networks by differentiating the relevance degrees among neighboring nodes for propagation.

SOTA Region representation methods:

- **POI [38]**. It is a baseline method which utilizes the POI attributes to represent spatial regions. The TF-IDF algorithm is used to determine the relevance between regions over the POI vectors.
- **HDGE [27]**. It is a region representation approach which only relies on the human mobility traces to encode the embedding of each region from the constructed stationary flow graph.
- **ZE-Mob [34]**. This method explores the co-occurrence patterns to model the relations between geographical zones from both the taxi trajectories and human mobility traces.
- **MV-PN [4]**. It performs representation learning on regions by considering both the POI and human mobility data. The functionality proximities are preserved in region embeddings.
- **CGAL [40]**. It is an unsupervised approach to encode region embeddings based on the constructed POI and mobility graphs. The adversarial learning is adopted to integrate the intra-region structures and inter-region dependencies.
- **MVURE [38]**. This model firstly captures the region correlations based on region geographical properties and user mobility traces.
- **MGFN [30]**. It is a most recently proposed region representation method which constructs the mobility graph to perform the message passing for generating region embeddings.

Backbone models for crime and traffic prediction. We select two state-of-the-art methods as the backbone models, to evaluate the quality of the region representations learned by different methods over the tasks of crime prediction and traffic flow forecasting. The encoded region representations are used as the initialized embeddings for the following backbone models.

- **ST-SHN [32] for Crime Prediction**. It is a state-of-the-art crime prediction method which designs hypergraph structures to capture the global spatial correlations for modeling the crime patterns. Following the settings in the original paper, 128 hyperedges are configured in the hypergraph neural architecture.
- **STGCN [37] for Traffic Flow Forecasting**. It is a representative traffic prediction model built over a spatio-temporal GCN. The spatio-temporal convolutional blocks jointly encode the dependencies across regions and time slots. Two ST-Conv blocks are adopted to achieve the best performance.

4.1.3 **Hyperparameter Settings.** Due to space limit, we present the details of parameter settins in AutoST in Appendix Section.

4.2 Effectiveness Evaluation (RQ1)

We evaluate our method AutoST on three spatio-temporal mining tasks, including crime prediction, traffic prediction and house price prediction. We provide the result analysis as follows.

Crime Prediction. We present the evaluation results of crime prediction on both Chicago and NYC in Table 1 (averaged results) and category-specific results in Table 3 (shown in Appendix Section).

- AutoST achieves the best performance in predicting urban crimes on both datasets, which suggests the effectiveness of our automated spatio-temporal graph contrastive learning paradigm. Specifically, we attribute the improvements to: i) With the design of our hierarchical multi-view graph encoder, AutoST can capture both the intra-view and inter-view region dependencies from multi-typed data sources (*i.e.*, POI semantics, human mobility traces, geographical positions). ii) Benefiting from the adaptive region graph contrastive learning, AutoST offers auxiliary self-supervised signals to enhance the region representation learning against noise perturbation and skewed data distribution.
- While the current region representation methods (*e.g.*, CGAL, MVURE, MGFN) attempt to capture the region correlations based on the mobility graph, they are vulnerable to the noisy edges in their constructed mobility-based region graphs. The information propagation between less-relevant regions will impair the the performance of region representation learning. In addition, the skewed distribution of mobility data limits the spatial dependency modeling in those baselines. To address these limitations, our AutoST supplements the region embedding paradigm with the region self-discrimination supervision signals, which is complementary to the mobility-based region relationships.
- The significant performance improvement achieved by AutoST over the compared graph embedding methods (*e.g.*, GCN, GAT, GAE) further confirms the superiority of performing automated data augmentation with the adaptive self-supervision signals. In

Table 1: Overall performance comparison in urban crime forecasting, traffic prediction, and house price prediction.

Model	Crime Prediction				Traffic Prediction						House Price Prediction			
	CHI-Crime		NYC-Crime		CHI-Taxi		NYC-Bike		NYC-Taxi		CHI-House		NYC-House	
	MAE	MAPE	MAE	MAPE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	MAPE	MAE	MAPE
ST-SHN	2.0259	0.9987	4.4004	0.9861	–	–	–	–	–	–	–	–	–	–
ST-GCN	–	–	–	–	0.1395	0.5933	0.9240	1.8562	1.4093	4.1766	–	–	–	–
Node2vec	1.6334	0.8605	4.3646	0.9454	0.1206	0.5803	0.9093	1.8513	1.3508	4.0105	13137.2178	44.4278	4832.6905	19.8942
GCN	1.6061	0.8546	4.3257	0.9234	0.1174	0.5707	0.9144	1.8321	1.3819	4.0200	13074.2121	42.6572	4840.7394	18.3315
GAT	1.5742	0.8830	4.3455	0.9267	0.1105	0.5712	0.9110	1.8466	1.3746	4.0153	13024.7843	43.3221	4799.8482	18.3433
GraphSage	1.5960	0.8713	4.3080	0.9255	0.1196	0.5796	0.9102	1.8473	1.3966	4.0801	13145.5623	44.3167	4875.6026	18.4570
GAE	1.5711	0.8801	4.3749	0.9343	0.1103	0.5701	0.9132	1.8412	1.3719	4.0337	13278.3256	42.3221	4896.9564	18.3114
POI	1.3047	0.8142	4.0069	0.8658	0.0933	0.5578	0.8892	1.8277	1.3316	3.9872	12045.3212	33.5049	4703.3755	16.7920
HDGE	1.3586	0.8273	4.2021	0.7821	0.0865	0.5502	0.8667	1.8251	1.2997	3.9846	11976.3215	30.8451	4677.6905	12.5192
ZE-Mob	1.3954	0.8249	4.3560	0.8012	0.1002	0.5668	0.8900	1.8359	1.3314	4.0366	12351.1321	38.6171	4730.6927	16.2586
MV-PN	1.3370	0.8132	4.2342	0.7791	0.0903	0.5502	0.8886	1.8313	1.3306	3.9530	12565.0607	39.7812	4798.2951	17.0418
CGAL	1.3386	0.7950	4.1782	0.7506	0.1013	0.5682	0.9097	1.8557	1.3353	4.0671	12094.5869	36.9078	4731.8159	16.5454
MVURE	1.2586	0.7087	3.7683	0.7318	0.0874	0.5405	0.8699	1.8157	1.3007	3.6715	11095.5323	34.8954	4675.1626	15.9860
MGFN	1.2538	0.6937	3.5971	0.7065	0.0831	0.5385	0.8783	1.8163	1.3266	3.7514	10792.7834	29.9832	4651.3451	12.9752
AutoST	**1.0480**	**0.4787**	**2.1073**	**0.5241**	**0.0665**	**0.4931**	**0.8364**	**1.8145**	**1.2871**	**3.6446**	**10463.7715**	**25.7575**	**4517.7276**	**7.1660**

the graph neural paradigm, the noise effects will be amplified during the message passing over the task-irrelevant region edges. The information aggregated from all the mobility-based or POI-based graph neighboring regions may mislead the encoding of true underlying crime patterns in the urban space.

Traffic Prediction. Table 1 shows the results of predicting traffic volume for future period of 15 mins. (Full version in Table 4).

- AutoST consistently outperforms the compared baselines in all cases, which confirms that our proposed method produces more accurate pre-trained region representations. As the skewed distribution of traffic flow is prevalent across different regions in urban space, with the designed adaptive contrastive learning component over the hierarchical multi-view graph, AutoST overcomes the skewed data distribution issue.
- Region representation methods (*i.e.*, MVURE, MGFN) require sufficient data to discover traffic patterns or mine human behaviours, which means that they are unable to handle long-tail data. In contrast, contrastive learning with automated view generation has better adaptation on skewed data distribution.
- Specifically, region representation methods (i.e. MVURE, AutoST) obtain better performance on long-term traffic prediction (45 minutes) than other traditional methods (i.e. Node2vec, GAE). This suggests region representation methods are effective to capture time-evolving region-wise relations, which are crucial to encode long-term traffic patterns.

Regional House Price Prediction. Due to space limit, we summarize the key observations in Section A.3 of Appendix.

4.3 Ablation Study (RQ2)

In this section, we perform model ablation studies to evaluate the effects of different data views and contrastive learning components of AutoST in contributing the region representation performance. In particular, i) For different data views, we generate two variants: "w/o $\mathcal{G}_p$", "w/o $\mathcal{G}_d$" by removing POI-aware region graph $\mathcal{G}_p$, and distance-based region graph $\mathcal{G}_d$ respectively. ii) For the ablation study on contrastive learning modules, we generate the variant "w/o InfoMin" without the mutual information minimization.

(1) Analysis on Crime Prediction. From Table 2, we can observe that the designed component and the incorporated data views

Table 2: Ablation Study on Crime and Traffic Prediction

Model	NYC-Crime							
	Burglary		Larceny		Robbery		Assault	
	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
w/o $\mathcal{G}_p$	4.5058	0.7280	0.4257	0.3199	1.0524	0.5971	1.0884	0.9654
w/o $\mathcal{G}_d$	5.2415	0.9296	1.2058	0.9954	1.4762	1.0950	1.1370	0.9813
w/o InfoMin	4.3684	0.6666	0.3773	0.2967	0.8918	0.4227	0.7584	0.6554
AutoST	**4.2576**	**0.6424**	**0.3766**	**0.2905**	**0.8435**	**0.3738**	**0.7394**	**0.6343**

Model	NYC-Taxi (15/ 30/ 45 min)	
	MAE	RMSE
w/o $\mathcal{G}_p$	1.3753/ 1.4470/ 1.5094	3.9952/ 4.8177/ 5.3110
w/o $\mathcal{G}_d$	1.3932/ 1.4505/ 1.5131	4.0802/ 4.9238/ 5.4050
w/o InfoMin	1.2968/ 1.3622/ 1.3836	3.7405/ 4.7069/ 5.3480
AutoST	**1.2871/ 1.3134/ 1.3280**	**3.6446/4.2278/4.6836**

in our AutoST framework bring positive effects to the representation performance in the downstream crime prediction. Compared with w/o $\mathcal{G}_p$ and w/o $\mathcal{G}_d$, the performance improvement achieved by AutoST indicates the importance of incorporating region POI semantic and geographical information in region representations.

(2) Analysis on Traffic Prediction. Similar results can be observed for traffic prediction task. For example, by incorporating the POI semantic or geographical information into our graph neural encoder, the traffic forecasting performance becomes better with the learned region embeddings by AutoST. Therefore, with the effectively modeling of heterogeneous dynamic relationships among regions, our method can endow the region representation paradigm with the diverse spatio-temporal patterns.

4.4 Model Robustness Study (RQ3)

We also perform experiments to investigate the robustness of our framework AutoST against data sparsity. To achieve this goal, we separately evaluate the prediction accuracy of regions with different density degrees. Here, the density degree of each region is estimated by the ratio of non-zero elements (crime occurs) in the region-specific crime occurrence sequence Z_r. Specifically, we partition sparse regions with the crime density degree ≤ 0.5 into two groups, *i.e.*, $(0.0, 0.25]$ and $(0.25, 0.5]$. The evaluation results are shown in Figure 4. We observe that our AutoST consistently outperforms other methods. As such, this experiment again demonstrates that the spatio-temporal region representation learning benefits greatly from our incorporated self-supervised signals to offer accurate and robust crime prediction on all cases with different crime density

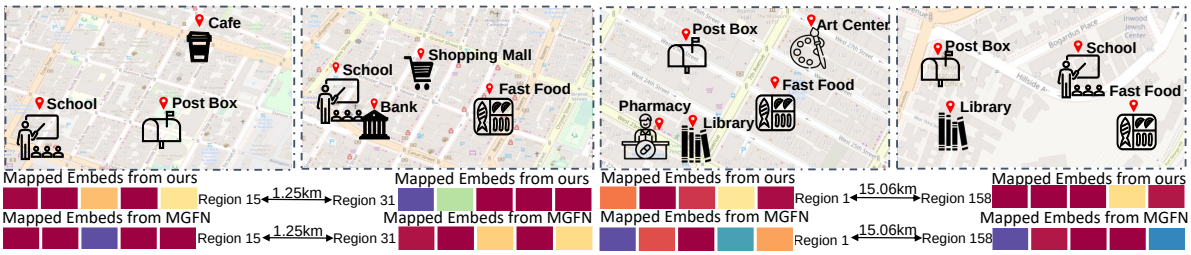


Figure 3: Case study of our AutoST method on New York City datasets.

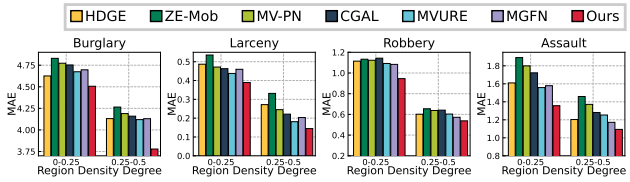


Figure 4: Results on NYC crime for Burglary, Larceny, Robbery and Assault *w.r.t* different data density degrees.

degrees. Existing region representation approaches (*e.g.*, MVURE and MGFN) need a large amount of data to mine human behaviours or mobility patterns, which leads lower performance on the sparse data. Another observation admits that current traditional graph learning methods (*e.g.*, GNN-based approaches) can hardly learn high-quality representations on regions with sparse crime data.

4.5 Case Study (RQ4)

In this section, we perform case study to show the capacity of our AutoST in learning the global region dependency with respect to geographical semantics. Specifically, we sample two region pairs, *i.e.*, 1) nearby regions: (region-15, region-31); 2) far away regions: (region-1, region-158). From the presented Figure 3, while region-15 and region-31 are spatially-adjacent with each other, they exhibit different urban functions. However, their embeddings learned by the baseline MGFN show high similarity. In contrast, the embedding diversity can be observed in the region embeddings learned by our AutoST method. Furthermore, although region-1 and region-158 are located with long geographical distance, their latent semantics are well preserved in our encoded embddings. Overall, these observations indicate the advantage of our AutoST in capturing global region dependencies in the entire urban space.

5 RELATED WORK

Region Representation Learning. Several studies [4, 27, 30, 34, 38, 40] investigate region representation learning problem. In particular, Fu et al. [4] propose to improve the quality of region representations by incorporating both intra-region information (*e.g.*, POI distance within a region) and inter-region information (*e.g.*, the similarity of POI distributions between two regions). Zhang et al. [40] extend the idea proposed in [4] by introducing a collective adversarial training strategy.

A recent study proposed in [38] develops a multi-view joint learning model for learning region representations. It first models the region correlations from different views (*i.e.*, human mobility view and region property view), and then a graph attention mechanism is used for each view in learning region representations. In addition, Wu et al. [30] propose to extract traffic patterns for learning region representations, but it only considers mobility data and ignores POI data, which is essential for capturing region functionalities. However, the effectiveness of above methods largely relies on generating the high quality region graphs, and may not be able to learn quality region representations under the noisy and skewed-distributed spatio-temporal data.

Graph Contrastive Learning. is widely investigated on graph data to learn SSL-enhanced representations [19, 25, 33, 35]. There are two categories of contrastive learning. One is to perform data augmentation with heuristics. For example, some methods [29, 35] leverage node/edge mask operations as data augmentor for contrastive self-supervision. However, two issues may exist in those approaches: first, they are specially designed with heuristics, which can hardly generalize to diverse environments. In addition, those solutions select the data augmentation via trial-and-error strategies, which is time-consuming. Inspired by the generative models for data reconstruction, our proposed AutoST performs automatic contrastive learning on spatio-temporal graph so as to distill the informative self-supervised signals for data augmentation.

6 CONCLUSION

In this paper, we propose a new region representation learning method with the automated spatio-temporal graph contrastive learning paradigm. We explore the adaptive self-supervised learning over the spatio-temporal graphs, and identify the key issues faced by current region representation models. Our work shows that the automated contrastive learning-based data augmentation can offer great potential to region graph representation learning with multi-view spatio-temporal data. We conduct extensive experiments on three spatio-temporal mining tasks with several real-life datasets, to validate the advantage of our proposed AutoST across various settings. In future work, we would like to extend our AutoST to perform the counterfactual learning for distill the causal factors underlying the implicit region-wise correlations.

ACKNOWLEDGMENTS

This project is partially supported by HKU-SCF FinTech Academy and Shenzhen-Hong Kong-Macao Science and Technology Plan Project (Category C Project: SGDX20210823103537030) and Theme-based Research Scheme T35-710/20-R. This research work is also supported by Department of Computer Science & Musketeers Foundation Institute of Data Science at the University of Hong Kong.

REFERENCES

[1] 2017. Real Estate Value in Chicago. *https://www.zillow.com/* (2017).

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

[3] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *International Conference on World Wide Web (WWW)*. 1459–1468.

[4] Yanjie Fu, Pengyang Wang, Jiadi Du, Le Wu, and Xiaolin Li. 2019. Efficient region embedding with multi-view spatial networks: A perspective of locality-constrained spatial autocorrelations. In *AAAI*, Vol. 33. 906–913.

[5] Nina Grgic-Hlaca, Elissa M Redmiles, Krishna P Gummadi, and Adrian Weller. 2018. Human perceptions of fairness in algorithmic decision making: A case study of criminal risk prediction. In *World Wide Web Conference (WWW)*. 903–912.

[6] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *International Conference on Knowledge Discovery and Data Mining (KDD)*. 855–864.

[7] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems (NIPS)* 30 (2017).

[8] Chao Huang, Junbo Zhang, Yu Zheng, and Nitesh V Chawla. 2018. DeepCrime: Attentive hierarchical recurrent networks for crime prediction. In *International Conference on Information and Knowledge Management (CIKM)*. 1423–1432.

[9] Thomas N Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

[10] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[11] Guanyao Li, Chih-Chieh Hung, Mengyun Liu, Linfei Pan, Wen-Chih Peng, and S-H Gary Chan. 2021. Spatial-temporal similarity for trajectories with location noise and sporadic sampling. In *International Conference on Data Engineering (ICDE)*. IEEE, 1224–1235.

[12] Zhonghang Li, Chao Huang, Lianghao Xia, Yong Xu, and Jian Pei. 2022. Spatial-Temporal Hypergraph Self-Supervised Learning for Crime Prediction. In *International Conference on Data Engineering (ICDE)*. IEEE, 2984–2996.

[13] Haoxing Lin, Rufan Bai, Weijia Jia, Xinyu Yang, and Yongjian You. 2020. Preserving dynamic attention for long-term spatial-temporal prediction. In *International Conference on Knowledge Discovery & Data Mining (KDD)*. 36–46.

[14] László Lovász. 1993. Random walks on graphs. *Combinatorics, Paul erdos is eighty* 2, 1-46 (1993), 4.

[15] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. Stan: Spatio-temporal attention network for next location recommendation. In *ACM Web Conference (WWW)*. 2177–2185.

[16] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[17] Hossein A Rahmani, Mohammad Aliannejadi, Rasoul Mirzaei Zadeh, Mitra Baratchi, Mohsen Afsharchi, and Fabio Crestani. 2019. Category-aware location embedding for point-of-interest recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 173–176.

[18] Vanessa Jine Schweizer, Jude Herijadi Kurniawan, and Aidan Power. 2022. Semi-automated Literature Review for Scientific Assessment of Socioeconomic Climate Change Scenarios. In *Companion Proceedings of the Web Conference 2022*. 789–799.

[19] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2019. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *International Conference on Learning Representations (ICLR)*.

[20] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020), 6827–6839.

[21] Kevin Swersky Ting Chen, Simon Kornblith et al. 2020. A simple framework for contrastive learning of visual representations. In *Advances in Neural Information Processing Systems (NIPS)*.

[22] Patara Trirat and Jae-Gil Lee. 2021. Df-tar: a deep fusion network for citywide traffic accident risk prediction with dangerous driving behavior. In *The Web Conference (WWW)*. 1146–1156.

[23] Puja Trivedi, Ekdeep Singh Lubana, Yujun Yan, Yaoqing Yang, and Danai Koutra. 2022. Augmentations in graph contrastive learning: Current methodological flaws & towards better practices. In *ACM Web Conference (WWW)*. 1538–1549.

[24] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

[25] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax.. In *International Conference on Learning Representations (ICLR)*.

[26] Hongjian Wang, Daniel Kifer, Corina Graif, and Zhenhui Li. 2016. Crime rate inference with big data. In *International Conference on Knowledge Discovery and Data Mining (KDD)*. 635–644.

[27] Hongjian Wang and Zhenhui Li. 2017. Region representation learning via mobility flow. In *International Conference on Information and Knowledge Management (CIKM)*. 237–246.

[28] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. 2020. Traffic flow prediction via spatial temporal graph neural network. In *The Web Conference (WWW)*. 1082–1092.

[29] Lirong Wu, Haitao Lin, Cheng Tan, Zhangyang Gao, and Stan Z Li. 2021. Self-supervised learning on graphs: Contrastive, generative, or predictive. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* (2021).

[30] Shangbin Wu, Xu Yan, Xiaoliang Fan, Shirui Pan, Shichao Zhu, Chuanpan Zheng, Ming Cheng, and Cheng Wang. 2022. Multi-Graph Fusion Networks for Urban Region Embedding. *arXiv preprint arXiv:2201.09760* (2022).

[31] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. 2022. SimGRACE: A Simple Framework for Graph Contrastive Learning without Data Augmentation. In *The Web Conference (WWW)*. 1070–1079.

[32] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Liefeng Bo, Xiyue Zhang, and Tianyi Chen. 2021. Spatial-Temporal Sequential Hypergraph Network for Crime Prediction with Dynamic Multiplex Relation Learning. In *International Joint Conferences on Artificial Intelligence (IJCAI)*. 1631–1637.

[33] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 70–79.

[34] Zijun Yao, Yanjie Fu, Bin Liu, Wangsu Hu, and Hui Xiong. 2018. Representing urban functions through zone embedding with human mobility patterns. In *International Joint Conferences on Artificial Intelligence (IJCAI)*.

[35] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems (NIPS)* 33 (2020), 5812–5823.

[36] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. 2022. Bringing your own view: Graph contrastive learning without prefabricated data augmentations. In *International Conference on Web Search and Data Mining (WSDM)*. 1300–1309.

[37] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *International Joint Conferences on Artificial Intelligence (IJCAI)*.

[38] Mingyang Zhang, Tong Li, Yong Li, and Pan Hui. 2021. Multi-view joint graph representation learning for urban region embedding. In *International Joint Conferences on Artificial Intelligence (IJCAI)*. 4431–4437.

[39] Tong Zhang, Wei Ye, Baosong Yang, Long Zhang, Xingzhang Ren, Dayiheng Liu, Jinan Sun, Shikun Zhang, Haibo Zhang, and Wen Zhao. 2022. Frequency-aware contrastive learning for neural machine translation. In *International Conference on Artificial Intelligence (AAAI)*, Vol. 36. 11712–11720.

[40] Yunchao Zhang, Yanjie Fu, Pengyang Wang, Xiaolin Li, and Yu Zheng. 2019. Unifying inter-region autocorrelation and intra-region structures for spatial embedding via collective adversarial learning. In *International Conference on Knowledge Discovery & Data Mining (KDD)*. 1700–1708.

[41] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *The Web Conference (WWW)*. 2069–2080.

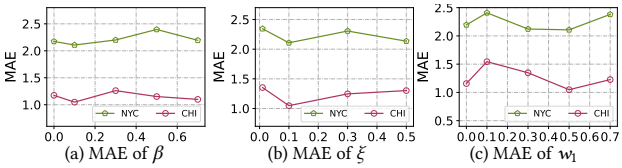


Figure 5: Effect study for hyperparameters in the performance of AutoST on NYC crime data, in terms of MAE.

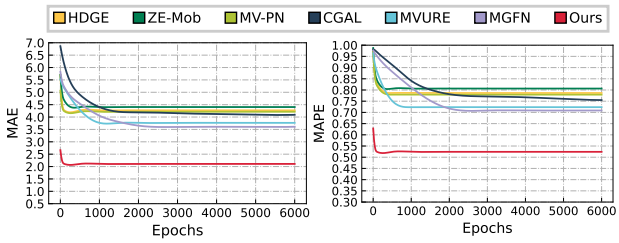


Figure 6: MAE and MAPE of crime prediction with epochs.

A APPENDIX

In this section, we provide some supplementary materials to support our methodology and evaluation sections with further details. Specifically, we first present the data description and hyperparameter settings. Then, the impacts of key parameters are studied in Section A.4. Furthermore, the model convergence analysis of our proposed AutoST and baselines are investigated in Section A.5.

A.1 Data Description Details

In our experiments, we partition New York City and Chicago into 180 and 234 disjoint geographical regions based on the census blocks with street boundaries. For the traffic flow datasets, we collect the taxi trips spanning the time period of two weeks from Chicago and New York City to generate the set of user mobility trajectories $\mathcal{M}$. For the collected urban crime datasets, each crime record is formatted as < crime, category, timestamp, coordinates > and will be mapped into a specific region based on the longitude and latitude information. Following the settings in [32], different types of crimes are included in Chicago (Theft, Battery, Assault, Damage) and NYC (Burglary, Larceny, Robbery, Assault) datasets. For the house price dataset, we follow the data pre-processing strategy in [27] to generate region-specific price information by considering 22,540 and 44,447 houses [1] in New York City and Chicago, respectively.

A.2 Hyperparameter Settings

For fair comparison, the dimensionality d of region representation is set as 96 to be consistent with the settings in [30, 38]. The depth of convolutional layers in GCN is set as 3. The learning rate is initialized as 0.0005 with the weight decay of 0.01. For the crime prediction backbone model, ST-SHN [32] is configured with the learning rate of 0.001 and the weight decay of 0.96. The depth of the spatial path aggregation layers is set as 2. For the traffic prediction backbone model ST-GCN [37], the historical time window of all tests are set as 60 minutes with 12 observed data points that are utilized to forecast traffic conditions in the next 15, 30, 45 minutes. Most baselines are implemented with their released codes.

A.3 House Price Prediction

The learned region representations with different methods are used as the input embeddings to a Lasso regression method. Table 1 shows the results on the house price data in Chicago and NYC.

- We notice that AutoST has the best performance in all cases. Meanwhile, MVURE and MGFN have better performance on Chicago house price data and New York house price data. This is mainly due to they mine human patterns, which build connections between similar or neighbour regions. We also observe that POI method has good performance, since this method builds connections between POIs of regions and central shopping regions including different POIs from that of remote towns.
- Region representations methods achieve better performance than network embeddings methods. The reasons behind this observation is that traditional graph-based methods (*e.g.*, Node2vec, GCN) lack the effective encoding of region-wise dependencies from both spatial and temporal dimensions.

A.4 Hyperparameter Studies (RQ5)

To show the effect of different parameter settings, we conduct experiments to evaluate the performance of our framework AutoST with different configurations of important hyperparameters (*e.g.*, (a) # β, # ξ and # w_1)). When varying a specific hyperparameter for effect investigation, other parameters are fixed with default values. The results are shown in Figure 5. We summarize the observations below to analyze the influence of different hyperparameters:

- We vary β from the range of $\{0.0, 0.1, 0.3, 0.5\}$. The best performance is achieved with $\beta = 0.1$. The prediction performance decreases as we further increase the value of β, which suggests larger β does not always bring better model representation ability. The further increase of β leads to little gap issue between positive samples and negative samples.
- When ξ is set as 0.1, the best performance can be achieved. Additionally, w_1 is the weight for the auxiliary supervised signal, which is utilized to calculate the distance between the positive samples and the negative samples. We observe that the best performance is achieved when $w_1 = 0.5$.

Algorithm 1: Automated Contrastive View Generation

Input: The multi-view spatio-temporal graph $\mathcal{G}$ consisting of three different graph layers;
Output: Graph views $\mathcal{G}'_1, \mathcal{G}'_2$ for contrastive learning;
Apply the cross-layer graph message passing on $\mathcal{G}$ for the low-dimensional graph representation $\mathbf{H}$ (Eq 3);
Employ the adjustable variational graph encoder for low-dimensional data generation $\tilde{\mathbf{H}}$;
Decode the hidden representations to graph structures and sparsify the adjacent matrix $\tilde{\mathbf{P}}$;
Use random walker to sample small sub-graph pairs $\mathcal{G}'_1, \mathcal{G}'_2$ for the two generated graph views;
Return $\mathcal{G}'_1$ and $\mathcal{G}'_2$.

Table 3: Overall performance comparison in crime prediction on both Chicago and NYC datasets.

Model	Chicago								New York City							
	Theft		Battery		Assault		Damage		Burglary		Larceny		Robbery		Assault	
	MAE	MPAE	MAE	MPAE	MAE	MPAE	MAE	MPAE	MAE	MPAE	MAE	MPAE	MAE	MPAE	MAE	MPAE
ST-SHN	1.2100	0.9995	1.9275	0.9993	2.2018	0.9998	2.1349	0.9985	5.6100	0.9975	1.2000	0.9987	1.7551	0.9998	1.2909	0.9985
Node2vec	1.1378	0.9862	1.7655	0.8970	1.9631	0.9714	1.9015	0.9657	4.9447	0.8092	0.7272	0.6532	1.0566	0.8040	1.2411	0.9967
GCN	1.1065	0.9643	1.3012	0.8094	1.5431	0.8094	1.5031	0.8056	4.6993	0.7912	0.49994	0.4178	1.0655	0.8004	1.2407	0.9890
GAT	1.1123	0.9759	1.3215	0.8344	1.5892	0.8241	1.5387	0.8277	4.7055	0.7944	0.5023	0.4019	1.0653	0.8027	1.2403	0.9949
GraphSage	1.1231	0.9790	1.3574	0.8561	1.6016	0.8563	1.5761	0.8432	4.7313	0.8066	0.5213	0.4314	1.0719	0.8110	1.2418	0.9965
GAE	1.1043	0.9614	1.3065	0.7984	1.5379	0.7914	1.4986	0.8033	4.7013	0.7910	0.5012	0.4289	1.0679	0.8012	1.2405	0.9958
POI	0.9733	0.9341	1.1065	0.7513	1.4089	0.7541	1.4076	0.7697	4.6939	0.7825	0.4969	0.4172	1.0660	0.7970	1.2400	0.9943
HDGE	0.9545	0.9012	1.0887	0.7389	1.3970	0.7217	1.3768	0.7349	4.5658	0.7160	0.4734	0.3930	1.0507	0.6731	1.1551	0.9870
ZE-Mob	1.0983	0.9547	1.3142	0.8236	1.5345	0.8163	1.5138	0.8273	4.7570	0.8013	0.5186	0.4307	1.0722	0.8093	1.1403	0.9975
MV-PN	0.9613	0.9146	1.0946	0.7452	1.4013	0.7393	1.3582	0.7218	4.6329	0.7502	0.4213	0.3708	1.0642	0.7840	1.1091	0.9982
CGAL	0.9589	0.9014	1.0897	0.7403	1.3995	0.7345	1.3698	0.7296	4.6013	0.7203	0.4113	0.3651	1.0714	0.7765	1.1009	0.9894
MVURE	0.9365	0.8910	1.0631	0.6957	1.3709	0.6375	1.3037	0.6567	4.5907	0.7144	0.4077	0.3262	1.0578	0.5889	0.8410	0.6943
MGFN	0.9231	0.9015	1.0804	0.5824	1.3016	0.6072	1.2563	0.6503	4.5646	0.7994	0.4285	0.3084	1.0475	0.6310	0.8319	0.7096
AutoST	**0.9075**	**0.8711**	**0.8915**	**0.5710**	**1.1296**	**0.4254**	**0.9158**	**0.4542**	**4.2576**	**0.6424**	**0.3766**	**0.2905**	**0.8435**	**0.3738**	**0.7394**	**0.6343**

Table 4: Overall performance comparison in traffic prediction with different periods of time.

Model	CHI-Taxi (15/ 30/ 45 min)		NYC-Bike (15/ 30/ 45 min)		NYC-Taxi (15/ 30/ 45 min)	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
STGCN	0.1395/ 0.2051/ 0.2385	0.5933/ 0.6198/ 0.6330	0.9240/ 1.0601/ 1.2984	1.8562/ 2.1823/ 2.8658	1.4093/ 1.4513/ 1.5159	4.1766/ 4.7814/ 5.4858
Node2vec	0.1206/ 0.1407/ 0.1694	0.5803/ 0.6054/ 0.63327	0.9093/ 0.9875/ 1.1512	1.8513/ 2.0917/ 2.8019	1.3508/ 1.4078/ 1.4637	4.0105/ 5.0608/ 5.4043
GCN	0.1174/ 0.1368/ 0.1639	0.5707/ 0.6016/ 0.6278	0.9144/ 1.0238/ 1.2175	1.8321/ 2.1036/ 2.8359	1.3819/ 1.4309/ 1.4934	4.0200/ 4.8711/ 5.3463
GAT	0.1105/ 0.1317/ 0.1619	0.5712/ 0.6028/ 0.6305	0.9110/ 1.0245/ 1.2163	1.8466/ 2.0912/ 2.8277	1.3746/ 1.4425/ 1.5039	4.0153/ 4.8510/ 5.3264
GraphSage	0.1196/ 0.1345/ 0.1708	0.5796/ 0.6047/ 0.6346	0.9102/ 1.0530/ 1.2903	1.8473/ 2.0977/ 2.7651	1.3966/ 1.4577/ 1.5106	4.0801/ 4.9626/ 5.2976
GAE	0.1103/ 0.1312/ 0.1607	0.5701/ 0.6013/ 0.6297	0.9132/ 1.0118/ 1.2664	1.8412/ 2.1632/ 2.7061	1.3719/ 1.4307/ 1.5006	4.0337/ 4.9795/ 5.2080
POI	0.0933/ 0.1204/ 0.1578	0.5578/ 0.5903/ 0.6198	0.8892/ 0.9911/ 1.0362	1.8277/ 2.1333/ 2.5391	1.3316/ 1.3646/ 1.4297	3.9872/ 3.9189/ 4.8996
HDGE	0.0865/ 0.1203/ 0.1524	0.5502/ 0.5840/ 0.6134	0.8667/ 0.9889/ 1.0243	1.8251/ 1.9357/ 2.5376	1.2997/ 1.3572/ 1.4214	3.9846/ 4.4336/ 4.8316
ZE-Mob	0.1002/ 0.1245/ 0.1576	0.5668/ 0.5932/ 0.6154	0.8900/ 0.9982/ 1.0443	1.8359/ 2.1669/ 2.5718	1.3314/ 1.4008/ 1.4707	4.0366/ 4.5656/ 4.9116
MV-PN	0.0903/ 0.1235/ 0.1571	0.5502/ 0.5843/ 0.6132	0.8886/ 0.9791/ 1.0490	1.8313/ 1.9775/ 2.5687	1.3306/ 1.3874/ 1.4525	3.9530/ 4.5368/ 4.9061
CGAL	0.1013/ 0.1267/ 0.1502	0.5682/ 0.5998/ 0.6197	0.9097/ 1.0504/ 1.0910	1.8557/ 1.9845/ 2.5870	1.3353/ 1.4209/ 1.4833	4.0671/ 4.5850/ 4.9606
MVURE	0.0874/ 0.1196/ 0.1503	0.5405/ 0.5710/ 0.6013	0.8699/ 0.9364/ 0.9502	1.8157/ 1.9751/ 2.4356	1.3007/ 1.3266/ 1.3347	3.6715/ 4.2534/ 4.7200
MGFN	0.0831/ 0.1145/ 0.1492	0.5385/ 0.5671/ 0.5917	0.8783/ 0.9376/ 0.9771	1.8163/ 1.9907/ 2.4198	1.3266/ 1.3497/ 1.3561	3.7514/ 4.3200/ 4.7311
AutoST	**0.0665/ 0.0675/ 0.0684**	**0.4931/ 0.4956/ 0.4971**	**0.8364/ 0.8767/ 0.8801**	**1.8145/ 1.8864/ 2.3510**	**1.2871/ 1.3134/ 1.3280**	**3.6446/4.2278/4.6836**

Algorithm 2: The AutoST Learning Algorithm

Input: The hierarchical spatio-temporal graph $\mathcal{G}$, the maximum epoch number E, the learning rate η;
Output: Regional embeddings $\mathbf{H}$

Initialize all parameters;
for $e = 1$ *to* E **do**
  Generate contrastive views $\mathcal{G}'_1, \mathcal{G}'_2$ following Alg 1;
  Use the hierarchical graph encoder for embeddings $\mathbf{H}'_1, \mathbf{H}'_2$ *w.r.t* the generated views;
  Compute the CL loss $\mathcal{L}$ containing the InfoNCE $\mathcal{L}_{\text{NCE}}$ and the InfoBN $\mathcal{L}_{\text{BN}}$ (Eq 7, 8);
  Minimize $\mathcal{L}$ using Adam with learning rate η;
  Calculate the reward $\text{R}(\mathcal{G}, \theta_1, \theta_2)$ for view generation with InfoMin (Eq 5);
  Calculate the reconstruction loss $\mathcal{L}_{\text{Rec}}$ of VGAE;
  Optimize the VGAE-based graph sampler by minimizing $\text{R}(\cdot)$ combining $\mathcal{L}_{\text{Rec}}$ (Eq 4);
end
Return the learned region embeddings $\mathbf{H}$.

Table 5: Data Description of Experimented Datasets

Data	**Description of Chicago Data**	**Description of NYC data**
Census	Boundaries of 234 regions split by streets in a certain district, Chicago	Boundaries of 180 regions split by streets in Manhattan, New York
Taxi trips	Total 386,272 taxi trips during a month	Total 1,445,285 taxi trips during a month
Crime data	Total 321,876 crime records during 1 year	Total 108,575 crime records during 1 year
POI data	Total 3,680,125 POI locations of 130 categories	Total 20,569 POI locations of 50 categories
House price	Total 44,447 house price data in a certain district, Chicago	Total 22,540 house price data in Manhattan, New York

A.5 Model Convergence Study

To show our model convergence, Figure 6 presents the crime prediction performance using the encoded embeddings by different region representation methods *w.r.t* epochs. From the results, we can find that our method AutoST converges faster than other region representation methods, meanwhile achieving the best performance. It validates the scalability of our AutoST method with our proposed automated spatio-temporal contrastive learning framework.