Spatial-Temporal Graph Learning with Adversarial Contrastive Adaptation

Qianru Zhang [1] Chao Huang* [1] Lianghao Xia [1] Zheng Wang [2] Siuming Yiu [1] Ruihua Han [1]

Abstract

Spatial-temporal graph learning has emerged as a promising solution for modeling structured spatial-temporal data and learning region representations for various urban sensing tasks such as crime forecasting and traffic flow prediction. However, most existing models are vulnerable to the quality of the generated region graph due to the inaccurate graph-structured information aggregation schema. The ubiquitous spatial-temporal data noise and incompleteness in real-life scenarios pose challenges in generating high-quality region representations. To address this challenge, we propose a new spatial-temporal graph learning model (GraphST) for enabling effective self-supervised learning. Our proposed model is an adversarial contrastive learning paradigm that automates the distillation of crucial multi-view self-supervised information for robust spatial-temporal graph augmentation. We empower GraphST to adaptively identify hard samples for better self-supervision, enhancing the representation discrimination ability and robustness. In addition, we introduce a cross-view contrastive learning paradigm to model the inter-dependencies across view-specific region representations and preserve underlying relation heterogeneity. We demonstrate the superiority of our proposed GraphST method in various spatial-temporal prediction tasks on real-life datasets. We release our model implementation via the link: `https://github.com/HKUDS/GraphST`.

1. Introduction

Spatial-temporal graph representation learning aims to provide a meaningful latent embedding for each region in a city, which facilitates various spatial-temporal prediction tasks such as crime forecasting for public security (Li et al., 2022b), traffic flow prediction in intelligent transportation systems (Pan et al., 2019; Zheng et al., 2020), and Point-of-Interest (POI) recommendation in location-based services (Zhou et al., 2019). One of the most promising solutions to this problem is the adoption of Graph Neural Networks (GNNs) (Zhang et al., 2021a; Wu et al., 2022). These GNN approaches model region correlations with message passing over region graphs generated based on collected spatial-temporal data, such as human mobility traces and citywide traffic flow.

Although spatial-temporal graph neural models are highly effective, their reliance on the quality of the constructed region graph presents significant challenges. The current schema (Zhang et al., 2020; Jin et al., 2020; Zhou et al., 2023) of embedding propagation and refinement along graph structural connections is limited and often unable to capture the complexities of real-world urban environments. Data noise and incompleteness are also common in spatial-temporal data analysis, which further undermines the quality of the constructed region graph. For example, sensor readings may be lost or inaccurate (Yi et al., 2016), and human mobility data from crowd sensors is frequently noisy (Feng et al., 2019). In addition, spatially adjacent regions may not be strongly correlated due to differences in urban function, while similar regions may be far apart geographically (Zhou et al., 2020). These factors contribute to the questionable quality of region-wise connection graphs, making it difficult to effectively perform spatial-temporal graph representation learning in the presence of noise perturbation.

With the recent success of self-supervised learning in mitigating data scarcity and noise challenges, we draw inspiration to propose a spatial-temporap graph pre-training framework with effective data augmentation, by exploring the following questions in model design.

- **Q1**: How can we offer self-supervised signals as spatial-temporal graph pre-training tasks?
- **Q2**: How can we automatically identify hard samples during contrastive learning to enhance model robustness?
- **Q3**: How can we model the inter-dependencies across different region relation views?

To address these challenges, we propose a novel model

[1] The Univeristy of Hong Kong, Hong Kong [2] Nanyang Technological University, Singapore. Correspondence to: Chao Huang <chaohuang75@gmail.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

called GraphST, which advances spatial-temporal graph representation learning by distilling self-supervisory information for pre-training. To generate self-supervisory signals over our multi-view region graph in an adaptive manner, we design a learnable function that reflects the global urban context across different regions in a city. Furthermore, we develop an adversarial contrastive learning model that endows our GraphST with the ability to automatically identify hard positive and negative samples under a contrastive min-max optimization framework. Lastly, we introduce a cross-view contrastive learning method that captures inter-view dependencies and enhances representation uniformity of different regions through a self-discrimination scheme.

Our work makes the following contributions:

- We investigate the drawbacks of existing spatial-temporal graph learning methods with non-robustness against the perturbation of noisy and incomplete urban data.
- We propose a spatial-temporal graph pre-training model GraphST, in which an adversarial contrastive paradigm is designed to augment region relationship learning with graph structure-adaptive self-supervision.
- We demonstrate the significant improvements that GraphST achieves over state-of-the-art baselines in different settings and spatial-temporal prediction tasks.

2. Preliminaries

Our approach begins by partitioning the geographical area of a city into J spatial regions, indexed by j (e.g., r_j). To incorporate diverse urban contextual information into our latent embedding space, we propose a region embedding paradigm that draws on different heterogeneous data sources in urban space. Specifically, our model input is elaborated:

Region Point-of-Interests (POIs). To characterize the region urban functions (*e.g.*, accommodation, entertainment, education, medical and health), we construct a POI matrix $\mathcal{P} \in \mathbb{R}^{J \times C}$, in which C is the number of POI categories.

Human Mobility Trajectories. To capture the dynamic urban flow among citywide regions in T time slots, we generate a human trajectory set denoted as $\mathcal{M}$. This set contains individual human mobility traces (r_s, r_d, t_s, t_d), which record the source and destination region r_s and r_d along with the timestamp information t_s and t_d, respectively.

Problem Statement. Using the input of the POI matrix $\mathcal{P}$, human trajectory set $\mathcal{M}$, and spatial distance matrix $\mathcal{D}$, the output of this task is the low-dimensional embeddings of all regions in a city. These embeddings represent the encoded representation of each region and can be used for downstream spatial-temporal mining tasks, such as traffic prediction and urban crime forecasting.

3. Methodology

We introduce our spatial-temporal graph learning method GraphST with the key components illustrated in Figure 1.

3.1. Graph-enhanced Spatial-Temporal Learning

We develop a graph-enhanced spatial-temporal learning framework that is specifically designed to extract the underlying spatial-temporal relationships between regions by comprehensively capturing dynamic region dependencies across space and time from different perspectives such as POI semantics, urban flow transitions, and geo-locations.

Our approach to preserving POI semantics in the embedding space involves using Skip-gram and MLP to embed the POI matrix $\mathcal{P} \in \mathbb{R}^{J \times C}$ into a set of latent representations. Specifically, we apply Skip-gram to capture the context of each POI and then use an MLP to transform these contextual embeddings into the final POI embeddings. This approach allows us to capture the semantic relatedness between POIs and maintain this information in the embedding space. In the resulting matrix $\bar{\mathbf{E}} \in \mathbb{R}^{J \times d}$, each row corresponds to the d-dimensional vector of an individual region r_j.

Multi-View Region Graph. Prior to integrating multi-view information in a unified framework, we initially process each view independently and create view-specific graphs: i) *The POI-based region graph* $\mathcal{G}_p$ is generated by measuring the similarities between region POI semantic embeddings $\mathbf{e}_j$ and $\mathbf{e}_{j'}$. In this graph, two regions are connected if their cosine similarity $\cos(\mathbf{e}_j, \mathbf{e}_{j'}) > \varepsilon$ exceeds a threshold ε. ii) *The time-aware mobility-based graph* $\mathcal{G}_m$ is constructed with nodes and edges across both space and time. For each region r_j, a sequence of time-aware region nodes is generated corresponding to different time slots ($t \in T$). The source region node $r_s^{t_s}$ at time slot t_s is connected to the destination region node $r_d^{t_d}$ at time slot t_d given the trajectory (r_s, r_d, t_s, t_d). iii) *Geographic-based graph* $\mathcal{G}_s$ connects spatially adjacent regions based on their distance.

We generate a unified multi-view region graph $\mathcal{G}$ by stacking the view-specific region graphs, namely $\mathcal{G}_p$, $\mathcal{G}_m$, $\mathcal{G}_s$, to reflect the region-wise relation heterogeneity. To achieve this, we add self-connection edges between nodes of the same region in different view-specific graph structures.

Information Propagation Paradigm. To capture both intra-region and inter-region relations in multi-view data, GraphST utilizes message passing over the multi-view region graph $\mathcal{G}$ in both time and space dimensions. The embedding propagation from the $(l-1)$-th graph layer to the (l)-th layer is defined as follows:

$$\mathbf{h}_j^{(l)} = \sigma\Big(\sum_{j' \in \mathcal{N}_j} \beta_{j,j'} \mathbf{W}^{(l-1)} \mathbf{h}_{j'}^{(l-1)}\Big) \tag{1}$$

$\mathbf{h}_j$ represents the refined embedding vector of region r_j.

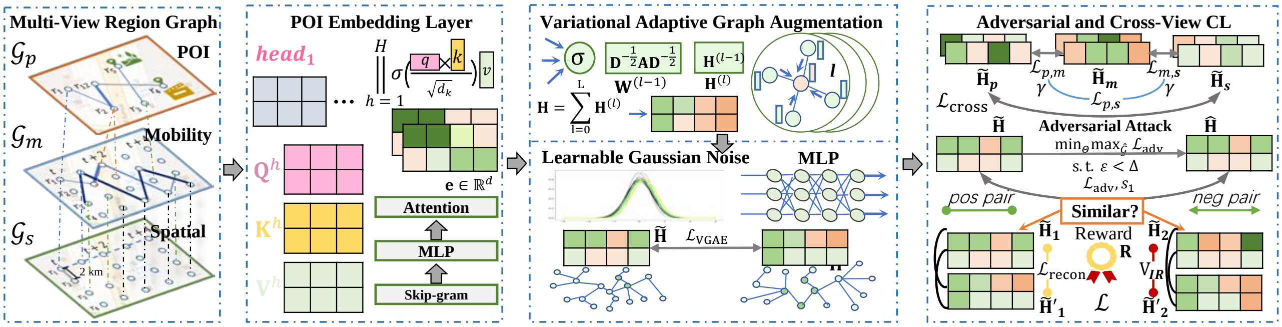


Figure 1. Illustration of GraphST's overall architecture. (i) POI embedding layer is a parameterized function which maps spatial functional semantics into latent embedding space. ii) Variational adaptive graph augmentation is performed over multi-view spatial-temporal graph $\mathcal{G}$ with learnable structure corruption. iii) Adversarial contrastive augmentation generates self-supervision signals with hard samples for enhancing model optimization. The quality of training samples with contrastive augmentation will be improved in spatial-temporal graph space. iv) Cross-view contrastive learning aims to preserve the implicit region inter-dependencies from heterogeneous relational views.

The coefficient $\beta_{j,j'}$ is computed as $1/\sqrt{|\mathcal{N}_j||\mathcal{N}_{j'}|}$, where $\mathcal{N}_j$ denotes the set of neighboring nodes of region r_j in graph $\mathcal{G}$. The initial embedding vector $\mathbf{h}_j^{(0)}$ is derived from the POI embedding layer. The ReLU activation function is denoted by $\sigma(\cdot)$, and $\mathbf{W}^{(l-1)} \in \mathbb{R}^{d\times d}$ represents the learnable transformation weights for the $(l-1)$-th iteration. Information aggregation is performed as:

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

$\mathbf{H} \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the embedding matrix of all nodes in graph $\mathcal{G}$, where $|\mathcal{V}|$ is the number of nodes. The variable L is the depth of the graph neural layers. The matrices $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times\mathcal{V}|}$ and $\mathbf{D} \in \mathbb{R}^{|\mathcal{V}|\times\mathcal{V}|}$ is the adjacent matrix of $\mathcal{G}$ with self-connections, and the diagonal degree matrix, respectively.

3.2. Graph Augmentation

Many existing graph contrastive learning methods use handcrafted contrastive views, such as stochastic graph structure corruption and masking, to simplify the model design. To make graph contrastive learning adaptable to global spatial-temporal relation heterogeneity, we propose an adaptive graph augmentation scheme for the multi-view region graph $\mathcal{G}$. Our scheme is based on the variational graph auto-encoder (VGAE) for self-supervision. This approach allows us to improve the performance of spatial-temporal representations by unifying contrastive and generative SSL to incorporate global spatial-temporal relation heterogeneity.

We propose a method to automatically learn the dependencies among different regions with global context and improve the region graph structures against noise perturbations such as low correlations between adjacent regions and strong dependencies between long-range regions. Our approach is based on the embedding mapping function $\mathcal{G} \to \mathbb{R}^{|\mathcal{V}|\times d}$ (Eq 2) and we perform Gaussian noise-based augmentation (Rusak et al., 2020) with the following design:

$$\tilde{\mathbf{H}} = \mathbf{\Gamma} \cdot \text{MLP}_{\text{std}}(\mathbf{H}) + \text{MLP}_{\text{mean}}(\mathbf{H}); \tilde{\mathcal{G}} = \eta(\tilde{\mathbf{H}}) \quad (3)$$

The augmented region embeddings for graph structure learning over all regions are denoted by $\tilde{\mathbf{H}} \in \mathbb{R}^{|\mathcal{V}|\times d}$. The structure learning function, $\eta(\cdot)$, estimates the region-wise dependencies. The noise matrix, $\mathbf{\Gamma} \in \mathbb{R}^{|\mathcal{V}|\times d}$, is represented by elements γ that are drawn from a Gaussian distribution with mean value μ and standard deviation σ.

To enable learnable contrastive view generation, GraphST uses two-layer MLPs with trainable parameters (*i.e.* $\text{MLP}_{\text{mean}}(\cdot)$ and $\text{MLP}_{\text{std}}(\cdot)$) to calculate the mean and standard deviation from the original embedding matrix $\mathbf{H}$. Following this, the region graph structure learning function $\eta(\cdot)$ is repeatedly applied, and the adaptive graph augmentation is performed using a contrastive loss.

$$\mathcal{L}_{\text{VGAE}} = \sum_j -\log \frac{\exp(\cos(\tilde{\mathbf{h}}_j, \tilde{\mathbf{h}}'_j)/\tau)}{\sum_{j'} \exp(\cos(\tilde{\mathbf{h}}_j, \tilde{\mathbf{h}}'_{j'})/\tau)} \quad (4)$$

τ is the temperature parameter to control the gradient effect.

3.3. Adversarial Contrasting with Hard Samples

To enhance the robustness of our graph augmentation against perturbations, we incorporate adversarial self-supervision to identify hard negative and positive node samples. This improves our contrastive learning paradigm by providing auxiliary self-supervision signals that are beneficial for gradient learning during model training. By performing adversarial contrastive learning, we aim to improve the performance of our model and make it more robust to perturbations.

To achieve this goal, we perform adversarial data augmentation by maximizing the contrastive loss between the generative autoencoder view $\tilde{\mathcal{G}}$ and a constrained adversarial view $\hat{\mathcal{G}}$. The adversarial contrastive learning is performed under the mini-max adversarial optimization paradigm as follows:

$$\min_{\mathbf{\Theta}} \max_{\hat{\mathcal{G}}} \; \mathcal{L}_{\text{adv}}(\tilde{\mathcal{G}}, \hat{\mathcal{G}}), \quad \text{s.t. } \varepsilon(\tilde{\mathcal{G}}, \hat{\mathcal{G}}) < \triangle \tag{5}$$

In the above equation, $\mathcal{L}_{\text{adv}}$ is the InfoNCE-based contrastive loss. $\mathbf{\Theta}$ represents the trainable model parameters of our variational graph auto-encoder. The error function $\varepsilon(\cdot)$ is defined as the absolute error between the adjacent matrices and the node features. $\triangle$ denotes hyperparameters for the error with respect to the adjacent matrix $\hat{\mathbf{A}}, \tilde{\mathbf{A}}$ and node embeddings $\hat{\mathbf{H}}, \tilde{\mathbf{H}}$.

To enable adversarial augmentation over the encoded node embeddings and graph structures, we adopt the projected gradient descent (PGD) attack (Feng et al., 2022) for data perturbation. We define a supplement matrix for the adjacency matrix, denoted as $\hat{\mathbf{A}}''$, and $\hat{\mathbf{A}}' = \mathbf{1}_{|\hat{\mathcal{V}}'|\times|\hat{\mathcal{V}}'|} - \mathbf{I}_{|\hat{\mathcal{V}}'|} - \hat{\mathbf{A}}''$, where $\mathbf{1}_{|\hat{\mathcal{V}}'|\times|\hat{\mathcal{V}}'|}$ is an all-one matrix with the size of $|\hat{\mathcal{V}}'| \times |\hat{\mathcal{V}}'|$. After PGD attack, the perturbed adjacency matrix is given by $\hat{\mathbf{A}} = \hat{\mathbf{A}}' + (\bar{\mathbf{A}} - \hat{\mathbf{A}}') \circ \mathbf{L}_{\hat{\mathbf{A}}'}$, where $\circ$ denotes element-wise product. Each element $\mathbf{L}_{\hat{\mathbf{A}}'}[i, j]$ in $\mathbf{L}_{\hat{\mathbf{A}}'} \in \{0, 1\}_{|\mathcal{V}|\times|\mathcal{V}|}$ represents the corresponding modification of the edge between node v_i and node v_j. The perturbation on $\hat{\mathbf{H}}'$ is presented as $\hat{\mathbf{H}} = \hat{\mathbf{H}}' + \mathbf{L}_{\hat{\mathbf{H}}'}$. where $\mathbf{L}_{\hat{\mathbf{H}}} \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the perturbation on feature matrix. For optimization relaxation, the convex hull $\ddot{\mathbf{L}}_{\hat{\mathbf{A}}'}$ replaces $\mathbf{L}_{\hat{\mathbf{A}}'}$, subject to the following constraints:

$$\mathcal{K}_{\hat{\mathbf{A}}'} = \left\{ \ddot{\mathbf{L}}_{\hat{\mathbf{A}}'} | \sum_{i,j} \ddot{\mathbf{L}}_{\hat{\mathbf{A}}'} \leq \triangle_{\hat{\mathbf{A}}'}, \ddot{\mathbf{L}}_{\hat{\mathbf{A}}'} \in [0, 1]^{|\hat{\mathcal{V}}'|\times|\hat{\mathcal{V}}'|} \right\}$$
$$\mathcal{K}_{\hat{\mathbf{H}}'} = \left\{ \mathbf{L}_{\hat{\mathbf{H}}'} | \|\mathbf{L}_{\hat{\mathbf{H}}'}\|_\infty \leq \delta_{\hat{\mathbf{H}}'}, \mathbf{L}_{\hat{\mathbf{H}}'} \in \mathbb{R}^{|\hat{\mathcal{V}}'|\times d} \right\} \tag{6}$$

where $\delta_{\hat{\mathbf{H}}'}$ denotes the constraint on feature perturbation. During each iteration, the update is presented as:

$$\ddot{\mathbf{L}}^e_{\hat{\mathbf{A}}'} = \prod_{\mathbf{K}_{\hat{\mathbf{A}}'}} [\ddot{\mathbf{L}}^{e-1}_{\hat{\mathbf{A}}'} + \zeta \mathbf{G}^e_{\hat{\mathbf{A}}'}]; \mathbf{G}^e_{\hat{\mathbf{A}}'} = \bigtriangledown_{\ddot{\mathbf{L}}^{e-1}_{\hat{\mathbf{A}}'}} \mathcal{L}'$$
$$\mathbf{L}^e_{\hat{\mathbf{H}}'} = \prod_{\mathbf{K}_{\hat{\mathbf{H}}'}} [\mathbf{L}^{e-1}_{\hat{\mathbf{H}}'} + \eta \text{sgn}(\mathbf{G}^e_{\hat{\mathbf{H}}'})]; \mathbf{H}^e_{\mathbf{X}} = \bigtriangledown_{\mathbf{L}^{e-1}_{\hat{\mathbf{H}}'}} \mathcal{L}'$$

where $\mathbf{G}^e_{\hat{\mathbf{A}}'}$ and $\mathbf{G}^e_{\hat{\mathbf{H}}'}$ denotes the gradient of loss in terms of $\ddot{\mathbf{L}}^{e-1}_{\hat{\mathbf{A}}'}$ and $\mathbf{L}^{e-1}_{\hat{\mathbf{H}}'}$ on e-th iteration. In addition, $\mathcal{L}'$ represents $\mathcal{L}_{\text{adv}}(\tilde{\mathbf{H}}, \hat{\mathbf{H}}^{e-1})$. $\prod_{\mathbf{K}_{\hat{\mathbf{H}}'}}$ denotes the projection head and maps $\mathbf{L}_{\hat{\mathbf{H}}'}$ into $[-\delta_{\hat{\mathbf{H}}'}, \delta_{\hat{\mathbf{H}}'}]$. Besides, Let $z = \sum_{i,j} P_{[0,1]}[\mathbf{Y} - \vartheta \mathbf{1}_{|\hat{\mathcal{V}}'|\times|\hat{\mathcal{V}}'|}]$ and $\prod_{\mathcal{K}_{\hat{\mathbf{A}}'}}(\mathbf{Y})$ is defined:

$$\prod_{\mathcal{K}_{\hat{\mathbf{A}}'}}(\mathbf{Y}) = \begin{cases} P_{[0,1]}[\mathbf{Y} - \vartheta \mathbf{1}_{|\hat{\mathcal{V}}'|\times|\hat{\mathcal{V}}'|}], & \text{if } z = \triangle_{\hat{\mathbf{A}}'}, \vartheta > 0 \\ P_{[0,1]}[\mathbf{Y}], & z \leq \triangle_{\hat{\mathbf{A}}'} \end{cases}$$

where $P_{[0,1]}[\mathbf{Y}]$ maps $\mathbf{Y}$ into $[0, 1]$. Following (Feng et al., 2022), we adopt bisection (Boyd et al., 2004) to address the equation $z = \triangle_{\hat{\mathbf{A}}'}$ with the dual variable ϑ. Each element is sampled from the Bernoulli distribution as $\mathbf{L}_{\hat{\mathbf{A}}'}[i:j] \sim \text{Bernoulli}(\ddot{\mathbf{L}}_{\hat{\mathbf{A}}'}[i:j])$ to obtain $\mathbf{L}_{\hat{\mathbf{A}}'}$ from $\ddot{\mathbf{L}}_{\hat{\mathbf{A}}'}$.

3.4. Cross-View Graph Contrastive Learning

To enhance the ability of GraphST to capture interdependencies among different data views and preserve heterogeneous region relations, we introduce a contrastive learning component in addition to our structure-level augmentation over the region graph. We accomplish this by splitting the embedding matrix $\tilde{\mathbf{H}}$ into three view-specific matrices $\tilde{\mathbf{H}}_p$, $\tilde{\mathbf{H}}_m$, and $\tilde{\mathbf{H}}_s$, which correspond to three different data views: POI semantic relatedness, urban flow transitions, and geographical locations. To illustrate, let us consider the contrastive loss $\mathcal{L}_{p,m}$ between the POI graph view ($\mathcal{G}_p$, $\tilde{\mathbf{H}}_p$) and the mobility graph view ($\mathcal{G}_m$, $\tilde{\mathbf{H}}_m$).

$$\mathcal{L}_{p,m} = \sum_{j \in \mathcal{V}_p \cap \mathcal{V}_m} - \log \frac{\exp(\cos(\tilde{\mathbf{h}}^p_j, \tilde{\mathbf{h}}^m_j)/\tau)}{\sum_{j'} \exp(\cos(\tilde{\mathbf{h}}^p_j, \tilde{\mathbf{h}}^m_{j'})/\tau)} \tag{7}$$

The remaining two contrastive loss terms, $\mathcal{L}_{m,s}$ ($\mathcal{G}_m$-$\mathcal{G}_s$) and $\mathcal{L}_{p,s}$ ($\mathcal{G}_p$-$\mathcal{G}_s$), can be obtained similarly. To weight these terms, we use $\gamma_{p,m} = \sigma(\text{MLP}(\tilde{\mathbf{H}}_p \odot \tilde{\mathbf{H}}_m))$, where $\odot$ denotes the element-wise product and σ is the ReLU function. The joint cross-view contrastive learning loss is:

$$\mathcal{L}_{\text{cross}} = \gamma_{p,m} \cdot \mathcal{L}_{p,m} + \gamma_{m,s} \cdot \mathcal{L}_{m,s} + \gamma_{p,s} \cdot \mathcal{L}_{p,s} \tag{8}$$

3.5. Contrastive Model Optimization with InfoMin

To make our data augmentation adaptive to the contrastive learning tasks, we enhance the reconstruction loss of generative variational autoencoder using a contrast-aware reward function with mutual information minimization (Tian et al., 2020; Wang et al., 2019). This function is formally defined:

$$\text{R}(\tilde{\mathcal{G}}) = \begin{cases} 1, & \text{if } \mathcal{L}_{\text{VGAE}} > \epsilon \\ \xi \ll 1 & \text{otherwise} \end{cases} \tag{9}$$

where ϵ is a threshold parameter. To enable robust model training, we incorporate information regularization (IR) into the model as follows:

$$V_{\text{IR}} = \max(2\text{s}_1 - \text{s}_2 - \text{s}_3, 0); \; \text{s}_1 = \exp(\frac{\cos(\tilde{\mathbf{h}}_{ij}, \tilde{\mathbf{h}}'_{ij})}{\tau})$$
$$\text{s}_2 = \exp(\frac{\cos(\tilde{\mathbf{h}}_{ij}, \mathbf{h}_{ij})}{\tau}); \; \text{s}_3 = \exp(\frac{\cos(\tilde{\mathbf{h}}'_{ij}, \tilde{\mathbf{h}}_{ij})}{\tau}) \tag{10}$$

where $\tilde{\mathbf{H}}'$ denotes the node embedding matrix of $\tilde{\mathcal{G}}'$, which is obtained via Eq 3. With the reward, the overall optimization

Table 1. Statistics of Experimented Datasets

Dataset	**Description of Chicago Data**	**Description of NYC data**
Blocks	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
Crimes	Total 321,876 2016/01/1-2017/12/30	Total 108,575 crime 2021/01/1-2022/02/24
POI	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

objective is to minimize the following loss:

$$\mathcal{L} = \mathcal{L}_{\text{VGAE}} + \mathcal{L}_{\text{cross}} + \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{recon}} \cdot \mathrm{R}(\tilde{\mathcal{G}}) + \lambda V_{\text{IR}} \quad (11)$$

where $\mathcal{L}_{\text{recon}}$ denotes the reconstruction loss between the VGAE-derived graph structures $\tilde{G}$ and the original graph $\mathcal{G}$.

3.6. Model Complexity Analysis

The GraphST framework incurs computational costs mainly in three aspects: Firstly, the multi-view space-time message passing paradigm has a complexity of $\mathcal{O}(N \times L \times d)$, where N and L denote the number of edges and graph propagation layers, respectively. Secondly, the complexity of our variational adaptive graph augmentation is $\mathcal{O}(|\mathcal{V}|^2 \times d)$. Thirdly, for self-supervised loss calculation, GraphST takes $\mathcal{O}(S \times |\tilde{\mathcal{V}}| \times d)$, where S denotes the number of samples included in each batch, and $|\tilde{\mathcal{V}}|$ denotes the average number of nodes in each sampled graph. Despite these costs, our GraphST achieves comparable model efficiency to existing spatial-temporal representation methods. This has been validated in our computational cost evaluation.

4. Evaluation

In this section, we evaluate the performance of our GraphST on various spatial-temporal mining tasks, including urban crime forecasting, traffic prediction, and house price prediction. We perform experiments on various datasets, which are described in detail in Table 1. Our goal is to validate the performance of GraphST against state-of-the-art (SOTA) methods on these three tasks.

4.1. Baselines and Parameter Settings

To evaluate our GraphST method, we consider baselines from three research lines for comparison. The descriptions of the baselines are elaborated in Appendix A.1.

(i) Network Embedding/GNN Models. We compare GraphST with several representative network embedding and graph neural models, including Node2vec (Grover & Leskovec, 2016), GCN (Kipf & Welling, 2017), GraphSage (Hamilton et al., 2017), GAE (Kipf & Welling, 2016), GAT (Veličković et al., 2018). We apply these methods to our constructed region graph to generate region embeddings.

(ii) Graph Contrastive Learning Methods. In our evaluation, we conduct experimental comparisons between our proposed GraphST method and two recently proposed graph contrastive learning frameworks, GraphCL (You et al., 2020) and RGCL (Li et al., 2022a).

(iii) Spatial-Temporal Region Representation. We further evaluate the performance of our proposed GraphST approach by comparing it with state-of-the-art methods for learning spatial-temporal embeddings of regions, such as POI (Rahmani et al., 2019), HDGE (Wang & Li, 2017), ZE-Mob (Yao et al., 2018), MV-PN (Fu et al., 2019), CGAL (Zhang et al., 2019), MVURE (Zhang et al., 2021a), and MGFN (Wu et al., 2022).

Hyperparameter Settings. To ensure a fair comparison, we set the dimensionality of the region representation d to 96, which is consistent with the settings used in previous works such as (Zhang et al., 2021a; Wu et al., 2022). We explore the number of graph propagation layers in the range of $\{1,2,3,4,5\}$ and tune the learning rate to 0.0005 with weight decay of 0.01. We also tune the temperature parameter τ in the range of $\{0.2,0.4,0.6,0.8\}$. The baselines are implemented using their original source code. Finally, we tune the weights of the augmented SSL loss in the range of (0,1).

4.2. Urban Crime Forecasting

Setup. The urban crime forecasting task aims to predict the number of crime events that will occur in future time slots. We conduct experiments on two real-world crime datasets collected from Chicago (2018) and New York City (2019). Since our GraphST is model-agnostic for downstream tasks, we utilize it as a pre-trained module to integrate with the state-of-the-art crime prediction model ST-SHN (Xia et al.). The encoded region embeddings from GraphST are then fine-tuned using ST-SHN. We follow the same settings as ST-SHN, including the region partition strategy (Chicago: 234, NYC: 180 regions), training/test data split, and evaluation metrics (MAE and MAPE). To conserve space, we reported the overall prediction accuracy across different crime categories, such as Theft, Assault, and Robbery. Category-specific prediction results and detailed settings are included in the supplementary material.

Results. Based on the reported results in Table 2, it is evident that our proposed GraphST outperforms state-of-the-art spatial-temporal region representation methods in all cases. This significant improvement in performance validates the effectiveness of our approach in addressing the challenges posed by sparse and skewed crime data. Our model achieves this by providing effective self-augmented signals for the region representation paradigm. Furthermore, applying standard graph neural networks directly over the region graph can be susceptible to noisy spatial-temporal

Table 2. Performance comparison in spatial-temporal learning tasks of 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
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
GraphCL	1.2332	0.6293	3.3075	0.6771	0.0812	0.5364	0.8582	1.8180	1.3022	3.7029	10752.5693	28.8374	4562.7279	11.3055
RGCL	1.1946	0.6081	3.1026	0.6323	0.0797	0.5052	0.8319	1.8107	1.2973	3.6865	10673.3289	27.5279	4439.0733	10.2375
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
Ours	**1.1285**	**0.5740**	**2.1060**	**0.5203**	**0.0747**	**0.4938**	**0.7787**	**1.8079**	**1.2892**	**3.6616**	**10461.5321**	**26.5092**	**4347.9815**	**7.0510**

region relations generated from heterogeneous data sources. This vulnerability can result in suboptimal representations.

4.3. Traffic Flow Prediction

Setup. To assess the effectiveness of our proposed GraphST in predicting citywide traffic volume, we evaluated all compared methods on the backbone of ST-GCN (Yu et al., 2018). The results, evaluated in terms of MAE and MAPE on three traffic benchmark datasets, are reported in Table 2. Following the settings in ST-GCN, we set the historical time periods to 60 minutes with 12 traffic volume records to make predictions for the next 15, 30, and 45 minutes.

Results. Our experimental results demonstrate that our proposed GraphST consistently outperforms all baselines on different datasets, indicating its effectiveness in capturing region-wise traffic dependencies with noisy region graph connections. This is in contrast to current region embedding models, which are limited in accurately capturing such dependencies, particularly in spatially adjacent areas with different urban functions (POI semantic relatedness). Furthermore, while some methods, such as MV-PN, CGAL, and MGFN, attempt to consider mobility data for region correlation modeling, the skewed mobility and traffic distribution make them easily biased towards regions with dense data, sacrificing the representation performance of unpopular spatial areas. This highlights the advantages of our proposed GraphST in addressing these limitations and improve the quality of spatial-temporal graph representation learning.

4.4. House Price Prediction

Setup. To further evaluate the effectiveness of GraphST, we conduct experiments to predict region-specific house prices using data collected from 22,540 houses in NYC and 44,447 houses in Chicago. We follow the same house price mapping method as in (Wang & Li, 2017). The region embeddings learned by different methods are then fed into Lasso regression to make house price predictions.

Results. Our experimental results demonstrate that our GraphST outperforms all baselines in the house price prediction task, confirming its superior region representation capacity. Benefiting from our adversarial adaptive graph augmentation, GraphST can preserve the holistic semantics across time and space in the latent region embedding space, despite noisy and incomplete multi-view spatial-temporal data. This is an advantage over previous approaches, which fail to effectively capture such dependencies

4.5. Ablation Study and Effectiveness Analyses

In this section, we investigate the impact of each component of GraphST on model performance by constructing four variants of the model. Our goal is to answer the question: *Is it important to incorporate GraphST's key components in boosting the model performance?* The five variants include: (1) "w/o adversarial view (ADS)" which disables the adversarial learning component against hard samples; (2) "w/o VGAE" which replaces VGAE with random edge dropout operators; (3) "w/o information regularization (IR)" which disables GraphST's ability to identify hard negative samples; (4) "w/o information minimization (IM)" which removes mutual information minimization for redundant information alleviation. By comparing the performance of these variants with the original GraphST, we aim to determine the importance of each component in improving performance.

We report the category-specific crime forecasting results in Figure 2 and traffic prediction results on NYC-Taxi and NYC-bike in Table 3. Our results demonstrate that all key components of GraphST contribute significantly to learning expressive and robust region representations, as evidenced by the improved performance. Furthermore, some simplified GraphST variants that include only the adversarial adaptive augmentation achieve promising performance, indicating the effectiveness of the region representation with multi-

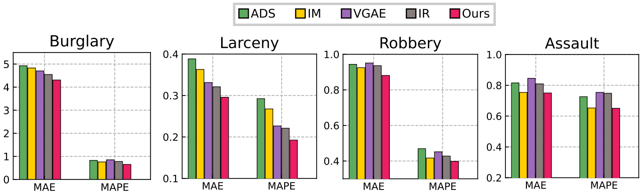


Figure 2. Ablation study of GraphST for crime prediction.

Table 3. Ablation study of GraphST for traffic prediction

Model	NYC-Taxi		NYC-Bike	
	MAE	RMSE	MAE	RMSE
w/o ADS	1.3731	3.8764	0.9067	1.9516
w/o IM	1.3519	3.8533	0.9132	1.9304
w/o VGAE	1.3486	3.8457	0.8941	1.9072
w/o IR	1.3418	3.7904	0.8837	1.8705
Ours	**1.2892**	**3.6616**	**0.7787**	**1.8079**

view graph self-supervised augmentation underlying spatial-temporal dynamics for urban computing applications.

4.6. Performance over Sparse Data

In this section, we investigate the robustness of GraphST in alleviating the effects of data sparsity. Specifically, we evaluate the representation performance on regions with different crime data density degrees, ranging from $(0.0, 2.5]$ to $(2.5, 5.0]$, based on the non-zero value ratio of region-specific crime sequences. The results, presented in Figure 3, demonstrate that our GraphST consistently improves performance under different degrees of data sparsity. This can be attributed to the self-supervision signals that are derived from the intrinsic structure of multi-view spatial-temporal data, which allows GraphST to learn effective representations even in regions with sparse crime data.

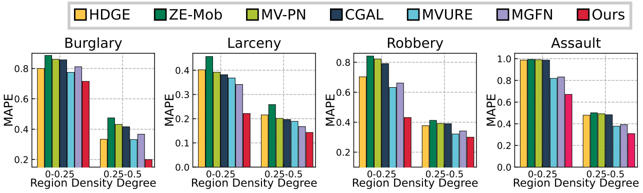
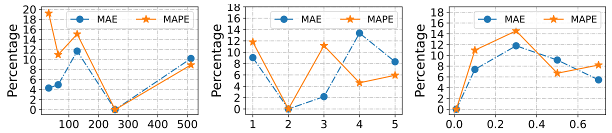


Figure 3. Evaluation Results on New York Data for different criminal offense types *w.r.t* different data density degrees.

4.7. Effects of Hyperparameters

We investigate the effects and sensitivity of our proposed GraphST to different hyperparameters. Figure 4 shows the percentage of performance degradation compared to the best performance when varying each hyperparameter, while keeping the other parameters at their default values. Our results show that increasing the embedding dimensionality beyond a certain point (tuning range $\{2^5, 2^6, 2^7, 2^8, 2^9\}$) leads to more noise and overfitting, resulting in degraded performance. Similarly, increasing the number of GNN layers beyond a certain point (tuned from $\{1,2,3,4,5\}$) can lead to over-smoothing and indistinguishable representations, which also results in degraded performance. Finally, we find that the best performance is achieved with a value of $\xi = 0.01$ from $\{0.01,0.1,0.3,0.5,0.7\}$.



(a) # Hidden (b) # Layer (c) ξ

Figure 4. Hyperparameter effects of GraphST on NYC crime data.

4.8. Case Study

This section presents case studies to demonstrate the effectiveness of our proposed GraphST in modeling cross-region dependencies and distilling global spatial-temporal knowledge while alleviating view-specific noise perturbation. Figure 5 visualizes the region embeddings mapped from the original representations encoded by GraphST. We observe that the long distance between two functionally similar regions, r_{10} and r_{66}, may lead to weak connections in the multi-view region graph. However, our adversarial self-supervised augmentation paradigm enables GraphST to distill the implicit region dependencies from the noisy information. Furthermore, while regions r_7 and r_{109} exhibit different function semantics in the urban space, their spatially adjacent relationship results in direct connections over the graph structure. Our GraphST can overcome this bias by avoiding misleading the graph neural network with heavy propagation between these less relevant regions.

4.9. Model Efficiency Study

Finally, we investigate the model efficiency of GraphST by comparing it with state-of-the-art region representation methods in terms of training time, as shown in Table 4. All methods are implemented in Python 3.8, PyTorch 1.7.0 (GPU version), and TensorFlow 1.15.3 (GPU version) (ST-SHN). The experiments are conducted on a server with 10 cores of Intel(R) Core(TM) i9-9820X CPU @ 3.30GHz, 64.0GB RAM, and one Nvidia GeForce RTX 3090 GPU. We observe that GraphST achieves competitive model efficiency compared to the baselines, demonstrating its potential in handling large-scale spatial-temporal data. The augmented self-supervised learning paradigms do not involve much additional computational cost but significantly boost the region representation performance.

5. Related Work

5.1. Spatial-Temporal Region Representation Learning

Many efforts have been devoted to proposing various region representation solutions based on spatial-temporal data. For example, an early study (Wang & Li, 2017) utilized net-

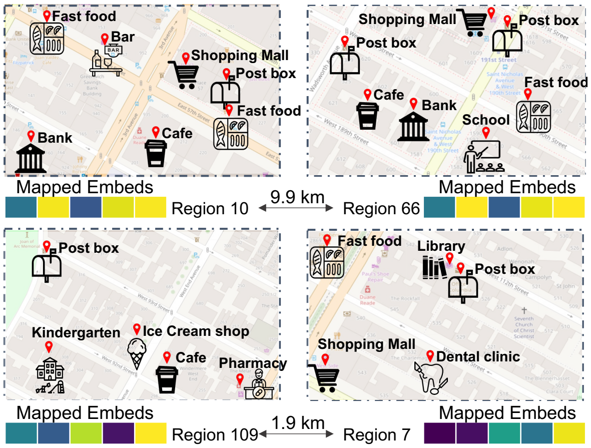


Figure 5. Case study of GraphST on New York City. Our GraphST can alleviate the connection bias in spatial-temporal graphs by i) refining the salient relational signals between far away regions (region 10 & region 66) ii) denoising connections between adjacent regions with weak dependence (region 7 & region 109).

Table 4. Computational cost (seconds) of our GraphST and SOTA spatial-temporal region representation methods. MAE and MAPE are estimated for NYC crime prediction results.

Models	HDGE	ZE-Mob	MV-PN	CGAL	MVURE	MGFN	Ours
Training	303.7	82.7	33.4	4144.8	240.7	852.3	275.1
MAE	4.2021	4.3560	4.2342	4.1782	3.7683	3.5971	2.1060
MAPE	0.7821	0.8012	0.7791	0.7506	0.7318	0.7065	0.5203

work embedding techniques to map region networks into latent representations. Motivated by the effectiveness of graph neural network for relational learning, many GNN-based models are introduced to perform the message passing over the generated region graph structures for region embedding refinement by considering the correlations between connected regions (Zhang et al., 2021a; 2019; Fu et al., 2019; Wu et al., 2022). However, most of the existing models are vulnerable to the quality of the constructed region graphs. In response to this limitation, this work proposes a self-supervised learning paradigm that explores data augmentation adaptive to spatial-temporal region dependencies.

5.2. Graph Neural Networks for Spatial-Temporal Data

In recent years, Graph Neural Networks (GNNs) have been successfully applied to representation learning on various spatial-temporal data. One notable example is traffic modeling, where GNNs are used to propagate information among neighboring geographical regions for traffic pattern representations (Li et al., 2018; Zhang et al., 2021b; Li & Zhu, 2021). While some attempts propose to discriminate the region-wise message passing with graph attention networks (Zheng et al., 2020; Fang et al., 2021), the noisy graph structures can still mislead the derivation of attentive weights. To encode high-order region correlations, GNN-enhanced crime predictive models are developed to encode crime dynamics (Xia et al.; Wang et al., 2022). Additionally, the location recommender systems are improved by graph neural models to jointly encode information from both temporal and spatial dimensions (Yang et al., 2022; Lim et al., 2022). The use of graph contrastive learning in embedding spatial-temporal data has shown promising results in recent attempts (Zhang et al., 2023). Our new framework takes a step further by explicitly incorporating adversarial SSL and cross-view CL paradigms to enhance spatial-temporal graph learning. With our framework, we provide a powerful and flexible solution for embedding spatial-temporal data.

5.3. Graph Contrastive Learning

Recent studies have shown the benefits of contrastive learning in graph representations to address the data sparsity and noise challenges (Zhu et al., 2021; Xia et al., 2023; Lin et al., 2022; Yin et al., 2022) by generating contrastive views with various techniques. These studies generate contrastive views using various techniques, such as stochastic node/edge masking strategies (You et al., 2020). Follow-up models have been proposed to improve the representation power of graph contrastive learning, including rationale-aware contrastive augmentation (Li et al., 2022a), exploring information transformation in GCL (Xu et al., 2021), and latent factor disentanglement (Li et al., 2021). Motivated by those approaches, the proposed GraphST method has the potential to offer significant benefits for spatial-temporal graph learning tasks by providing effective SSL signal distillation.

6. Conclusion

In this paper, we introduce the GraphST model for spatial-temporal graph learning, which enables self-supervised augmentation over spatial-temporal graphs. Our model designs an adversarial contrastive learning paradigm with generative autoencoder. We also propose a cross-view contrastive learning method that takes into account the heterogeneity of region relation to improve inter-dependency modeling. Our experimental results on various spatial-temporal prediction tasks demonstrate the superior performance of our method. Our distillation of essential self-supervised information also reduces the need for manual feature engineering.

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. We also thank the Department of Computer Science and the Musketeers Foundation Institute of Data Science at HKU for their support in this work.

References

Boyd, S., Boyd, S. P., and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Fang, M., Tang, L., Yang, X., Chen, Y., Li, C., and Li, Q. Ftpg: A fine-grained traffic prediction method with graph attention network using big trace data. *Transactions on Intelligent Transportation Systems (TITS)*, 2021.

Feng, J., Zhang, M., Wang, H., Yang, Z., Zhang, C., Li, Y., and Jin, D. Dplink: User identity linkage via deep neural network from heterogeneous mobility data. In *The Web Conference (WWW)*, pp. 459–469, 2019.

Feng, S., Jing, B., Zhu, Y., and Tong, H. Adversarial graph contrastive learning with information regularization. In *The Web Conference (WWW)*, pp. 1362–1371, 2022.

Fu, Y., Wang, P., Du, J., Wu, L., and Li, X. Efficient region embedding with multi-view spatial networks: A perspective of locality-constrained spatial autocorrelations. In *International Conference on Artificial Intelligence (AAAI)*, volume 33, pp. 906–913, 2019.

Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 855–864, 2016.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *International Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Jin, W., Ma, Y., Liu, X., Tang, X., Wang, S., and Tang, J. Graph structure learning for robust graph neural networks. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 66–74, 2020.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Li, H., Wang, X., Zhang, Z., Yuan, Z., Li, H., and Zhu, W. Disentangled contrastive learning on graphs. In *International Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 21872–21884, 2021.

Li, M. and Zhu, Z. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *International Conference on Artificial Intelligence (AAAI)*, pp. 4189–4196, 2021.

Li, S., Wang, X., Zhang, A., Wu, Y., He, X., and Chua, T.-S. Let invariant rationale discovery inspire graph contrastive learning. In *International Conference on Machine Learning (ICML)*, pp. 13052–13065. PMLR, 2022a.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR)*, 2018.

Li, Z., Huang, C., Xia, L., Xu, Y., and Pei, J. Spatial-temporal hypergraph self-supervised learning for crime prediction. In *International Conference on Data Engineering (ICDE)*, pp. 2984–2996. IEEE, 2022b.

Lim, N., Hooi, B., Ng, S.-K., Goh, Y. L., Weng, R., and Tan, R. Hierarchical multi-task graph recurrent network for next poi recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, 2022.

Lin, S., Liu, C., Zhou, P., Hu, Z.-Y., Wang, S., Zhao, R., Zheng, Y., Lin, L., Xing, E., and Liang, X. Prototypical graph contrastive learning. *Transactions on Neural Networks and Learning Systems (TNNLS)*, 2022.

Pan, Z., Liang, Y., Wang, W., Yu, Y., Zheng, Y., and Zhang, J. Urban traffic prediction from spatio-temporal data using deep meta learning. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1720–1730, 2019.

Rahmani, H. A., Aliannejadi, M., Mirzaei Zadeh, R., Baratchi, M., Afsharchi, M., and Crestani, F. Category-aware location embedding for point-of-interest recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 173–176, 2019.

Rusak, E., Schott, L., Zimmermann, R. S., Bitterwolf, J., Bringmann, O., Bethge, M., and Brendel, W. A simple way to make neural networks robust against diverse image corruptions. In *European Conference on Computer Vision (ECCV)*, pp. 53–69. Springer, 2020.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. What makes for good views for contrastive learning? In *International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 6827–6839, 2020.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Wang, C., Lin, Z., Yang, X., Sun, J., Yue, M., and Shahabi, C. Hagen: Homophily-aware graph convolutional recurrent network for crime forecasting. In *International Conference on Artificial Intelligence (AAAI)*, volume 36, pp. 4193–4200, 2022.

Wang, H. and Li, Z. Region representation learning via mobility flow. In *Conference on Information and Knowledge Management (CIKM)*, pp. 237–246, 2017.

Wang, J., Zhang, T., Liu, S., Chen, P.-Y., Xu, J., Fardad, M., and Li, B. Towards a unified min-max framework for adversarial exploration and robustness. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Wu, S., Yan, X., Fan, X., Pan, S., Zhu, S., Zheng, C., Cheng, M., and Wang, C. Multi-graph fusion networks for urban region embedding. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.

Xia, L., Huang, C., Xu, Y., Dai, P., Bo, L., Zhang, X., and Chen, T. Spatial-temporal sequential hypergraph network for crime prediction with dynamic multiplex relation learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1631–1637.

Xia, L., Huang, C., Shi, J., and Xu, Y. Graph-less collaborative filtering. In *The Web Conference (WWW)*, pp. 17–27, 2023.

Xu, D., Cheng, W., Luo, D., Chen, H., and Zhang, X. Infogcl: Information-aware graph contrastive learning. *International Conference on Neural Information Processing Systems (NeurIPS)*, 34:30414–30425, 2021.

Xu, K., Chen, H., Liu, S., Chen, P.-Y., Weng, T.-W., Hong, M., and Lin, X. Topology attack and defense for graph neural networks: An optimization perspective. *arXiv preprint arXiv:1906.04214*, 2019.

Yang, S., Liu, J., and Zhao, K. Getnext: trajectory flow map enhanced transformer for next poi recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 1144–1153, 2022.

Yao, Z., Fu, Y., Liu, B., Hu, W., and Xiong, H. Representing urban functions through zone embedding with human mobility patterns. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

Yi, X., Zheng, Y., Zhang, J., and Li, T. St-mvl: filling missing values in geo-sensory time series data. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

Yin, Y., Wang, Q., Huang, S., Xiong, H., and Zhang, X. Autogcl: Automated graph contrastive learning via learnable view generators. In *International Conference on Artificial Intelligence (AAAI)*, volume 36, pp. 8892–8900, 2022.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. *International Conference on Neural Information Processing Systems (NeurIPS)*, 33:5812–5823, 2020.

Yu, B., Yin, H., and Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

Zhang, M., Li, T., Li, Y., and Hui, P. Multi-view joint graph representation learning for urban region embedding. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4431–4437, 2021a.

Zhang, Q., Chang, J., Meng, G., Xiang, S., and Pan, C. Spatio-temporal graph structure learning for traffic forecasting. In *International Conference on Artificial Intelligence (AAAI)*, volume 34, pp. 1177–1185, 2020.

Zhang, Q., Huang, C., Xia, L., Wang, Z., Li, Z., and Yiu, S. Automated spatio-temporal graph contrastive learning. In *Proceedings of the ACM Web Conference 2023*, pp. 295–305, 2023.

Zhang, X., Huang, C., Xu, Y., Xia, L., Dai, P., Bo, L., Zhang, J., and Zheng, Y. Traffic flow forecasting with spatial-temporal graph diffusion network. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 15008–15015, 2021b.

Zhang, Y., Fu, Y., Wang, P., Li, X., and Zheng, Y. Unifying inter-region autocorrelation and intra-region structures for spatial embedding via collective adversarial learning. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1700–1708, 2019.

Zheng, C., Fan, X., Wang, C., and Qi, J. Gman: A graph multi-attention network for traffic prediction. In *International Conference on Artificial Intelligence (AAAI)*, volume 34, pp. 1234–1241, 2020.

Zhou, X., Mascolo, C., and Zhao, Z. Topic-enhanced memory networks for personalised point-of-interest recommendation. In *Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 3018–3028, 2019.

Zhou, Z., Wang, Y., Xie, X., Chen, L., and Liu, H. Riskoracle: a minute-level citywide traffic accident forecasting framework. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pp. 1258–1265, 2020.

Zhou, Z., Lin, G., Yang, K., BAI, L., Wang, Y., et al. Greto: Remedying dynamic graph topology-task discordance via target homophily. In *International Conference on Learning Representations (ICLR)*, 2023.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., and Wang, L. Graph contrastive learning with adaptive augmentation. In *The Web Conference (WWW)*, pp. 2069–2080, 2021.

A. Appendix

A.1. Description of Baselines

To provide a comprehensive evaluation of our GraphST model, we compare it with many baseline methods from three different research lines. The first research line includes graph representation approaches, which are commonly used for embedding graph-structured data. The second research line includes graph contrastive learning models, which have shown promising results in learning discriminative representations for graphs. The third research line includes spatial-temporal region representation methods, which are specifically designed for embedding spatial-temporal data.

Network Embedding/GNN Approaches. To evaluate the effectiveness of our GraphST model, we compare it with several representative network embedding and graph neural network models. We apply these models on our region graph $\mathcal{G}$ to generate region embeddings. The details of each baseline is described as follows: **Node2vec** (Grover & Leskovec, 2016): It encodes graph structural information via random walk-based Skip-gram. **GCN** (Kipf & Welling, 2017): It performs the convolution-based message passing between neighbor nodes along the edges for embedding refinement. **GraphSage** (Hamilton et al., 2017): It is a graph neural architecture that enables information aggregation from the sampled sub-graph structures. **GAE** (Kipf & Welling, 2016): Graph Auto-encoder maps nodes into a latent embedding space with the input reconstruction objective over the graph structures. **GAT** (Veličković et al., 2018): Graph Attention Network enhances the discrimination ability of GNNs by differentiating the relevance degrees among neighboring nodes.

Graph Contrastive Learning Methods. In addition to the above-mentioned graph representation and GNN-based models, we also compare our GraphST model with two graph contrastive learning models, namely, **GraphCL** (You et al., 2020): This method creates multiple contrastive views for augmentation based on mutual information maximization. The embedding consistency is aimed to be achieved among different correlated views. **RGCL** (Li et al., 2022a): This is a state-of-the-art graph contrastive learning approach that performs data augmentation based on the designed rationale generator.

Spatial-Temporal Region Representation Models. we also compare it with state-of-the-art spatial-temporal representation methods for region embedding. These methods are as follows: **POI**: It utilizes POI attributes to represent spatial regions via TF-IDF tokenization given POI matrix. **HDGE** (Wang & Li, 2017): It uses human trajectories to generate a crowd flow graph and embeds regions into latent vectors to preserve graph structural information. **ZE-Mob** (Yao et al., 2018): In this method, region correlations are captured with the consideration of human mobility and taxi moving traces for producing embeddings. **MV-PN** (Fu et al., 2019): It is an encoder-decoder network to model intra-region and inter-region correlations. **CGAL** (Zhang et al., 2019): A graph-regularized adversarial learning method that considers graph-structured pairwise relations to embed regions into latent space. **MVURE** (Zhang et al., 2021a): It leverages the graph attention mechanism to model region correlations with inherent region attributes and human mobility data. **MGFN** (Wu et al., 2022): It encodes region embeddings with multi-level cross-attention to aggregate information for both intra-pattern and inter-pattern.

Table 5. 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
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
GraphCL	0.9994	0.8992	1.0578	0.6174	1.2938	0.5683	1.2457	0.6181	4.5047	0.7288	0.3834	0.3091	1.0498	0.5961	0.7724	0.6929
RGCL	0.9377	0.8779	1.0356	0.5865	1.2764	0.5457	1.2336	0.5975	4.4712	0.7064	0.3652	0.2832	0.9987	0.5769	0.7512	0.6743
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
GraphST	**0.9107**	**0.8424**	**0.9969**	**0.5618**	**1.2068**	**0.4944**	**1.1056**	**0.5438**	**4.3095**	**0.6504**	**0.2958**	**0.1924**	**0.8809**	**0.3902**	**0.7501**	**0.6510**

A.2. Learning Process of GraphST

As shown in Algorithm 1, our GraphST model first constructs the multi-view region-wise graph using the three data views (POI, mobility, and distance). Then it learns urban region embeddings with the space-time message passing paradigm. The variational graph autoencoder is employed for data augmentations, based on which GraphST conducts adversarial learning against hard samples, cross-view contrastive learning, and information regulation.

Algorithm 1 Learning Process of GraphST

```
Input: Region Point-of-Interest matrix $\mathcal{P}$, mobility trajectories $\mathcal{M}$, region geographical positions $\mathcal{D}$, learning rate $\eta$, training
    epochs $E$.
Output: Regional embeddings $\tilde{\mathbf{H}}$
Initialize model parameters in each module
Initialize POI-based region embeddings through $\bar{\mathbf{E}} = \text{MLP}(\text{Skip-gram}(\mathcal{P}))$
Capture region-wise correlations with self-attention to acquire the initial embeddings $\mathbf{E}$
Construct the multi-view region graph $\mathcal{G}$
for $e = 1$ to $E$ do
    Conduct spatial-temporal message passing to get low-dimensional region embeddings $\mathbf{H}$;
    Apply variational graph augmentation in the latent space twice for $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{H}}'$;
    Apply minimax adversarial learning in the latent space $\tilde{\mathbf{H}}$ and obtain the hard sample $\mathbf{H}_{\text{adv}}$ and conduct adversarial
      contrastive learning;
    Conduct cross-view contrastive learning between pairs of data views $\tilde{\mathbf{H}}_p$ and $\tilde{\mathbf{H}}_m$, $\tilde{\mathbf{H}}_p$ and $\tilde{\mathbf{H}}_s$, and $\tilde{\mathbf{H}}_m$ and $\tilde{\mathbf{H}}_s$ following
      Eq 8 and Eq 9;
    Apply contrative learning between $\tilde{\mathbf{H}}$ and $\tilde{\mathbf{H}}'$; Apply on inforamtion regularation on the results of contrastive learning
      following Eq 11;
    Calculate the loss of GraphST folloing the Eq 12;
end
for each parameter $\theta$ in $\mathbf{\Theta}$ do
    $\theta = \theta - \eta \cdot \frac{\partial \mathcal{L}}{\partial \theta}$;
end
return all parameters $\mathbf{\Theta}$
```

A.3. In-Depth Analysis of GraphST

Projection in Adversarial Contrasting. In this section, we aim to provide further discussion on the projection operation $\prod_{\mathcal{K}_{\hat{\mathbf{A}}'}}(\mathbf{Y})$ in projected gradient descent (PGD) of our adversarial contrasting learning. In general, our adversarial contrasting module is designed to enable our GraphST to be robust by distilling hard samples for improving model optimization. Inspired by the works in (Xu et al., 2019; Feng et al., 2022), the projection operation generates similar-form solutions as projected gradient descent. We first transform the optimization objective from the PGD attack to the projection operation as:

$$\begin{aligned} \underset{\mathcal{K}_{\hat{\mathbf{A}}'}}{\text{minimize}} \quad & \frac{1}{2}||\mathcal{K}_{\hat{\mathbf{A}}'} - \mathbf{Y}||_2^2 + \mathcal{N}_{[0,1]}(\mathcal{K}_{\hat{\mathbf{A}}'}) \\ \text{subject to} \quad & \mathbf{1}^{\mathrm{T}}\mathcal{K}_{\hat{\mathbf{A}}'} \leq \triangle_{\hat{\mathbf{A}}'} \end{aligned} \tag{12}$$

where when $\mathcal{K}_{\hat{\mathbf{A}}'} \in [0,1]^n$, $\mathcal{N}_{[0,1]}(\mathcal{K}_{\hat{\mathbf{A}}'}) = 0$, otherwise $\mathcal{N}_{[0,1]}(\mathcal{K}_{\hat{\mathbf{A}}'}) = \infty$. Thus, the Lagrangian function of Eq 12 is expressed as follows (Boyd et al., 2004):

$$\underset{\mathcal{K}_{\hat{\mathbf{A}}'}}{\text{minimize}} \quad \frac{1}{2}||\mathcal{K}_{\hat{\mathbf{A}}'} - \mathbf{Y}||_2^2 + \mathcal{N}_{[0,1]}(\mathcal{K}_{\hat{\mathbf{A}}'}) + \vartheta(\mathbf{1}^{\mathrm{T}}\mathcal{K}_{\hat{\mathbf{A}}'} - \triangle_{\hat{\mathbf{A}}'}) \tag{13}$$

where ϑ is the dual variable and $\vartheta \geq 0$. Thus the minimization of Eq 13 is $\mathcal{K}_{\hat{\mathbf{A}}'} = P_{[0,1]}(\mathbf{Y} - \vartheta\mathbf{1})$, which provides solutions for PGD attack problem in our adversarial contrastive training process.

Robust Adversarial Training with PGD Attack. In this part, we aim to clarify the rationale of adopting PGD Attack as the solution for the adversarial training ($\mathcal{L}_{\text{adv}}$ in Eq. 11) of spatial-temporal graph neural networks, so as to achieve model robustness. To this end, we first define our PGD attack problem for our spatial-temporal graph neural network as follows:

$$\underset{\mathbf{k}\in\mathcal{K}}{\text{minimize}}\,\underset{\mathbf{W}}{\text{maximize}}\,-g(\mathbf{k},\mathbf{W}) \tag{14}$$

where Eq. 14 is the min-max form of the PGD attack problem, where g is a cross-entropy-based loss function that measures the discrepancy between the model prediction and the true label. Here, we present how to approximate the PGD attack problem as our target adversarial training objective. By doing so, the PGD attack can serve as an effective solution for our adversarial augmentation with hard sample mining over our generated multi-view spatial-temporal graph $\mathcal{G}$.

Proof: Given $\mathcal{K}$ and $\mathbf{W}$, we need to optimize the following formula in adversarial training:

$$\underset{\mathbf{W}}{\text{minimize}}\,\underset{\mathbf{k}\in\mathcal{K}}{\text{maximize}}\,-g(\mathbf{k},\mathbf{W}) \tag{15}$$

where $\mathbf{W}$ is the weight matrix of the spatial-temporal graph neural network. Given a general loss function g of the attack operation on the spatial-temporal neural networks, Eq. 15 is equivalent to $\underset{\mathbf{W}}{\text{maximize}}\,\underset{\mathbf{k}\in\mathcal{K}}{\text{minimize}}\,g(\mathbf{k},\mathbf{W})$, which can be proved as follows. We introduce an epigraph variable h (Boyd et al., 2004) to rewrite Eq. 15 as:

$$\begin{aligned} \underset{\mathbf{W},h}{\text{minimize}} \quad & h \\ \text{subject to} \quad & -g(\mathbf{k},\mathbf{W}) \le h, \forall \mathbf{k}\in\mathcal{K} \end{aligned} \tag{16}$$

We set $d := -h$, Eq 16 is expressed as:

$$\begin{aligned} \underset{\mathbf{W},d}{\text{maximize}} \quad & d \\ \text{subject to} \quad & g(\mathbf{k},\mathbf{W}) \ge d, \forall \mathbf{k}\in\mathcal{K} \end{aligned} \tag{17}$$

Then removing the epigraph variable d, Eq 17 is equal to $\underset{\mathbf{W}}{\text{maximize}}\,\underset{\mathbf{k}\in\mathcal{K}}{\text{minimize}}\,g(\mathbf{k},\mathbf{W})$. According max-min inequality (Boyd et al., 2004), we can proof that:

$$\underset{\mathbf{W}}{\text{maximize}}\,\underset{\mathbf{k}\in\mathcal{K}}{\text{minimize}}\,g(\mathbf{k},\mathbf{W}) \le \underset{\mathbf{k}\in\mathcal{K}}{\text{minimize}}\,\underset{\mathbf{W}}{\text{maximize}}\,-g(\mathbf{k},\mathbf{W}) = \mathcal{L}_{\text{adv}} \tag{18}$$

Based on the discussion above, it can be inferred that the PGD attack generation problem is a suitable approximation for adversarial training when using the min-max optimization paradigm. This approach, coupled with projection operations, can enhance spatial-temporal adversarial contrastive learning, allowing for the identification of hard samples for self-supervision.

A.4. Category-Specific Crime Prediction Results

In the supplementary materials, we present detailed evaluation results for different crime types in Chicago and New York City. We evaluate different methods in terms of MAE and MAPE, with ST-SHN as the backbone method for all methods. Our GraphST framework consistently achieves the best results across all crime categories for both cities, as shown in Table 5. These results demonstrate the significant benefits of our spatial-temporal graph learning framework. We attribute the superiority of our approach to the factors, including the graph encoding on the multi-view region graph, which effectively extracts useful regional features for region representation. Additionally, the various contrastive learning tasks, including autoencoder-based adaptive contrastive learning, adversarial contrastive learning with hard negative generation, and cross-view contrastive learning, further enhancing the performance of our approach.