Mining Spatio-Temporal Relations via Self-Paced Graph Contrastive Learning

Rongfan Li*, Ting Zhong*, Xinke Jiang*
University of Electronic Science and Technology of China

Goce Trajcevski
Iowa State University
USA

Jin Wu, Fan Zhou†
University of Electronic Science and Technology of China

ABSTRACT

Modeling complex spatial and temporal dependencies are indispensable for location-bound time series learning. Existing methods, typically relying on graph neural networks (GNNs) and temporal learning modules based on recurrent neural networks, have achieved significant performance improvements. However, their representation capabilities and prediction results are limited when pre-defined graphs are unavailable. Unlike spatio-temporal GNNs focusing on designing complex architectures, we propose a novel adaptive graph construction strategy: Self-Paced Graph Contrastive Learning (SPGCL). It learns informative relations by maximizing the distinguishing margin between positive and negative neighbors and generates an optimal graph with a self-paced strategy. Specifically, the existing neighborhoods iteratively absorb more reliable nodes with the highest affinity scores as new neighbors to generate the next-round neighborhoods, and augmentations are applied to improve the transferability and robustness. As the adaptively self-paced graph approaches the optimized graph for prediction, the mutual information between nodes and the corresponding neighbors is maximized. Our work provides a new perspective of addressing spatio-temporal learning problems beyond information aggregation in Euclidean space and can be generalized to different tasks. Extensive experiments conducted on two typical spatio-temporal learning tasks (traffic forecasting and land displacement prediction) demonstrate the superior performance of SPGCL against the state-of-the-art.

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; • **Computing methodologies** → **Unsupervised learning**; **Learning latent representations**; **Neural networks**.

KEYWORDS

Spatio-Temporal Learning; Graph Neural Networks; Contrastive Learning; Self-Paced Learning

*Equal contribution.
†Corresponding author: Fan Zhou (fan.zhou@uestc.edu.cn).

KDD '22, August 14–18, 2022, Washington, DC, USA.

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539422

ACM Reference Format:
Rongfan Li*, Ting Zhong*, Xinke Jiang*, Goce Trajcevski, and Jin Wu, Fan Zhou†. 2022. Mining Spatio-Temporal Relations via Self-Paced Graph Contrastive Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3534678.3539422

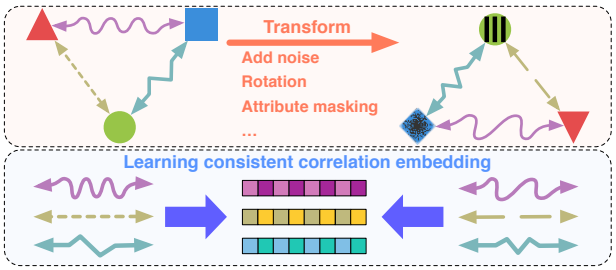
1 INTRODUCTION

Graph Neural Networks (GNNs) for spatio-temporal data forecasting has attracted tremendous research attention in recent years. Plethora of sophisticated GNN structures and massage passing mechanisms have been proposed to capture the intra-dependencies (i.e., temporal correlations within nodes) and inter-dependencies (i.e., spatial correlations among nodes) – sequentially [15, 48] or simultaneously [13, 16, 26]. Enabled by the expressive graph structures, spatio-temporal forecasting has been widely studied in range of applications, such as traffic forecasting [26, 41], landslide prediction [48], climate forecasting [20], etc.

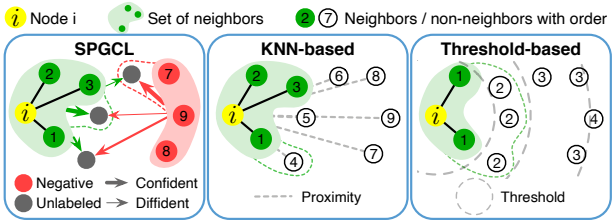
Motivations: Despite the significant breakthroughs achieved in spatio-temporal GNNs (STGNNs), there are certain observations regarding the state-of-the-arts which motivate our work.

O1: Most existing methods focus on node embedding learning, modeling structural and attribute similarity, using 1-dimensional adjacency and Laplacian matrices to guide feature aggregation. However, informative correlations between nodes are neglected and can not be sufficiently modeled solely using adjacency and Laplacian matrices. A semantic correlation embedding learning function is desired, which is also expected to make the model transferable and robust, e.g., the model can be transferred across graphs with performance guarantees. Current STGNNs are usually optimized end-to-end, ignoring transferability and robustness, which are hard to quantify and commonly not observed. To this end, in this work, we propose to learn transferable and robust correlation embeddings that are invariant to transformations, i.e., the output of embedding function is consistent with transformations (cf. Fig 1a for an example).

O2: The existing models are constrained by requiring *explicitly pre-defined* graph structures (i.e., adjacency matrix $\mathbf{A}$). In the case that the well-defined graph is not available, they utilize a variety of metrics to measure the pair-wise proximity, such as geographic distance [20, 28, 45], distance in manifold space [48], transportation connectivity [30], point of interests (POIs) [15], dynamic time wrapping (DTW) [13, 26], etc. These, however, incur serious problems of manually selecting suitable metrics and corresponding thresholds, and introduce extra inductive bias. In addition to threshold-based



(a) Learning invariances to different transformations.



(b) Neighbor selection strategy comparison.

Figure 1: Invariance learning and neighbor selection.

methods, there are also some KNN-based approaches [9] and iterative ones [6], which are still inflexible as they require specific metrics, and the global optimal hyperparameters may not be optimal for all nodes especially on dynamic graphs. Some works have proposed to reduce the dependence on pre-defined graphs and emphasize the capability of learning the latent [46], incomplete [2, 37] and even implicit graphs (i.e., graph without explicitly defined adjacency or highly sparse network) [6, 14, 47]. However, evaluating the quality of graph embedding relies on the downstream forecasting tasks or prior knowledge of graph structure. Moreover, most methods model spatial and temporal patterns separately, neglecting the spatio-temporal interactions [13, 26, 41, 48] – which also restricts the forecasting performance. Fig. 1b illustrates how our proposed model (denoted SPGCL) considers spatio-temporal features and adaptively adjusts the neighborhood until the optimal adjacency relationships (in comparison with the other approaches).

O3: Contrastive learning (CL) has recently become one of the most popular self-supervised approaches for capturing the invariance between similar (positive) data pairs and learning generalizable, transferrable, and robust representations via maximizing the mutual information (MI) [18, 32, 39]. CL methods have achieved great success in computer vision (CV) and natural language processing (NLP), and even outperformed supervised methods in some tasks [4, 10, 22, 29]. In contrast, graph CL is still under-explored due to the non-Euclidean properties and the rich structured information and node attributes. Various graph augmentations, encoding architectures, and contrastive objectives have been investigated in the literature [31, 33, 43, 44, 49]. They mainly focus on learning node embeddings on discrete graphs and labeled graphs (e.g., biochemical molecules, social networks [44], reference networks, and co-purchase networks [49]). However, it remains unclear whether CL is suitable for implicit graphs with undetermined correlations among the nodes (e.g., the traffic sensor data or point clouds data).

Present work: To address the issues observed above, we propose a novel Self-paced Graph Contrastive Learning framework (SPGCL). It learns the transferable joint spatio-temporal correlations on implicit graphs without manually designing the threshold of adjacency or constructing graphs with fixed neighborhoods, while capturing both spatial and temporal invariances. More specifically, the distinct features of SPGCL and our main contributions are:

- We theoretically prove that the global CL objective can be optimized via minimizing local CL loss of multiple steps. Specifically, we iteratively learn from the current context to discriminate reliable positive neighbor nodes incorporated in the following context so that the MI between a node and corresponding neighbors is maximized.
- We analyze the growth of MI and propose a self-paced labeling strategy to facilitate the model convergence on the implicit graphs under positive-unlabeled learning (PUL) paradigm [5, 25]. To our knowledge, this is the first work to train CL using PUL and calculate the contrastive loss by self-paced labeling.
- Inspired by [36], we learn the mapping from different proximities to the inter- and intra- relationships between nodes directly, instead of manually setting thresholds [13, 26] or learning node embedding first and then generating adjacency [2, 21, 24]. Therefore, the CL task is discriminating the positive relations from others which is significantly distinct from existing augmentation-based graph CL methods [31, 33, 43, 44, 49].

We conducted comprehensive evaluations on two typical spatio-temporal tasks, including traffic forecasting and land displacement prediction. The experimental results verify the advantages of SPGCL over the state-of-the-art baselines.

2 RELATED WORK

There is a large body of works tackling various GNN aspects [42]. In this section, we position our work focusing on two main categories:
–*Contrastive Learning*: The main idea of contrastive learning is to discriminate positive and negative data and learn invariances among similar data by special sampling strategies. The contrastive loss is derived from MI maximizing between similar (positive) data pairs X and Y (cf. [18, 32, 39]) as

$$I(X;Y) = H(X) - H(X|Y) = \mathbb{E}_{p(x,y)}\left[\log \frac{p(x|y)}{p(x)}\right], \qquad (1)$$

where $H(X) = -\sum p(x)\log p(x)$ is the entropy. In CV, positive data can be generated from augmentations (e.g., random flip, cropping, resizing) of the same image, and negative samples are augmented from other images. By learning the invariance to image augmentations, CL methods even outperformed supervised methods in certain tasks [4, 10, 22, 29]. Inspired by the success of CL in CV and NLP domains, a number of graph CL approaches have been proposed to explore effective augmentations, sample strategies, objectives, etc. For example, GCC [33] captures the local universal structural patterns among different graphs via subgraph sampling. GraphCL [44] consists of four types of data augmentations from the perspective of enforcing perturbation invariance. GMI [31] derives two contrastive objectives for direct optimization without graph augmentations. GCA [49] generates graph views via adaptive augmentation strategies. However, most (if not all) existing

methods focus on learning node embedding from the structural domain and/or the attribute domain, neglecting the correlation (edge) embeddings between two nodes. Inspired by [36], we assume a dense adjacency on implicit graphs, mark the edges that have a facilitating effect as positive, and consider the opposite and irrelevant edges as negative. We design an iterative training framework for learning the spatio-temporal interactions from contrasting different types of reliable edges.

–*Positive-unlabeled learning*: Certain works aim to classify positive data from unlabeled sets, which has been increasingly studied in recent years, along with the massive unlabeled realistic data produced by the rapid development of social networks, recommendation systems, etc. Early works relied on heuristic semi-supervised methods [27]. Subsequently, reweighting methods were investigated, which regard unlabeled data as weighted positive and negative data simultaneously [12], with unbiased [11] and non-negative [25] risk estimators. More recently, a self-paced learning framework was proposed [5] for classifying reliable positive examples progressively. In this work, we design an extended version of positive-unlabeled-negative (PUN) instead of a binary classifier combined with CL.

3 METHODOLOGY: SPGCL

We now provide the basic formalism and then proceed with the details of SPGCL.

–*Problem definition*: The *spatial* component of the data, denoted as a matrix $\mathbf{V} \in \mathbb{R}^{N\times d}$, corresponds to a collection of N monitored locations – each with a unique d dimensional spatial coordinates. In addition, each location is associated with a *temporal* sequence of length T, consisting of F observations per time instant, denoted as $\mathbf{X} \in \mathbb{R}^{N\times T\times F}$. The spatio-temporal observations at t-th timestamp are denoted as $\mathbf{X}^t = (\mathbf{x}_1^t, \mathbf{x}_2^t, \cdots, \mathbf{x}_N^t) \in \mathbb{R}^{N\times F}$. The main goal of **spatio-temporal forecasting** is to predict the $\widehat{\mathbf{X}} \in \mathbb{R}^{N\times T'\times F}$ of all N locations in future T' timestamps:

$$\left[\mathbf{X}^{t-T+1}, \mathbf{X}^{t-T+2}, \cdots, \mathbf{X}^{t}\right] \xrightarrow{f} \left[\mathbf{X}^{t+1}, \mathbf{X}^{t+2}, \cdots, \mathbf{X}^{t+T'}\right]. \quad (2)$$

–*Self-paced graph construction*: For predictions, we employ the message passing on graph structure $\mathcal{G} = (V, E, \mathbf{A})$, where V is a set of nodes, E is a set of edges and $\mathbf{A}$ is an adjacency matrix. Since E is generally not explicitly defined, we first provide an initial adjacency $\mathbf{A}^{t,0}$ and then infer the true adjacent relationship via the score function g, together with the observations $\mathbf{X}^t$ and locations $\mathbf{V}$. To better distinguish among the edges, we split all potential edges into three parts: E^P – labeled positive set (i.e., edges in E); E^N – labeled negative set; and E^U – unlabeled set (i.e., $E = E^P \cup E^N \cup E^U$). Potential edges in E^U with confident scores greater than δ^+ are added to E^P, while scores smaller than δ^- are put in E^N, i.e., $\mathbf{X}^t, \mathbf{V}, \mathbf{A}^{t,0} \xrightarrow{g} \hat{\mathbf{A}}^{t,1}$. We do it repeatedly until $E^U = \emptyset$, and finally obtain $\mathbf{A}^{t,K}$.

The basic framework of SPGCL is shown in Fig. 2, illustrating the different transitions from E^U into E^P and E^N, respectively.

3.1 Iterative Mutual Information Maximization

Constructing an informative graph is a fundamental step of spatio-temporal forecasting tasks [13, 26]. However, the representation learning problem on data with implicit correlations (or even without edges) remains unresolved. Inspired by the recent success of

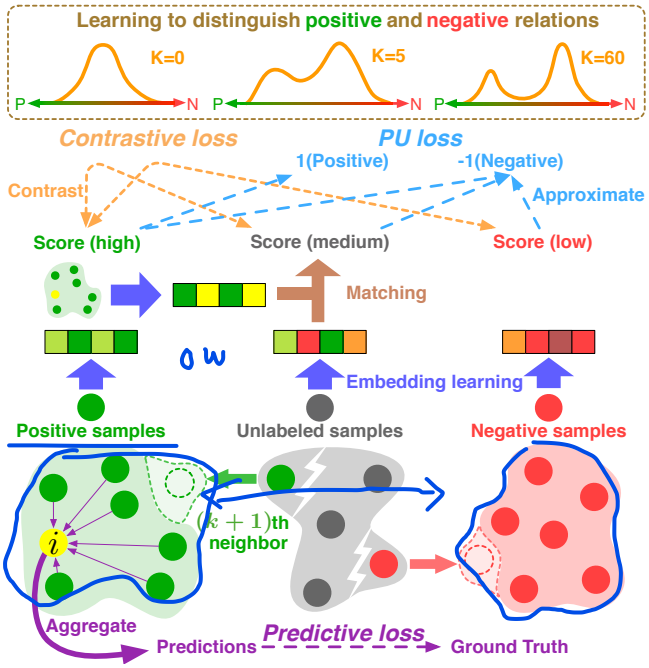


Figure 2: The framework of SPGCL. The $(k+1)$th neighbor is in E^U with the highest score.

contrastive learning [4, 33, 44, 49], we propose to learn the spatio-temporal correlations via maximizing mutual information (MI) between nodes and their corresponding neighbors.

Learning adjacency $\mathbf{A}$ from nodes (e.g., locations) $\mathbf{V}$ and observations $\mathbf{X}$ is also considered as link prediction or graph generation [17, 21, 24], and can be formally defined as computing $p(\mathbf{A}|\mathbf{V}, \mathbf{X})$. Optimizing the target likelihood is equivalent to finding all the K true neighbors of each node. Let $\mathbf{w}_i \in \mathbb{R}^{F'}$ denote the spatio-temporal representation of node i and $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N\} \in \mathbb{R}^{N\times F'}$. The combination of K arbitrary nodes, denoted as C_m^K and numbered by m, is $|C^K| = \frac{N!}{K!(N-K)!}$, which is significantly large especially for larger value of N and K. The task can be alternatively described from the perspective of MI as finding the best C_m^K maximizing the MI with every nodes:

$$\begin{aligned}\max \log p(\mathbf{A}|\mathbf{V}, \mathbf{X}) &:= \max \mathbb{E}_{p(C_m^K, \mathbf{w}_i)}[\log p(C_m^K|\mathbf{w}_i)] \\ &= \max I(\mathbf{W}; C^K) + \mathbb{E}_{p(C_m^K)}[\log p(C_m^K)], \quad (3)\end{aligned}$$

which can be further derived from the view of global and local MI:

$$\begin{aligned}I(\mathbf{W}; C^K) &= \mathbb{E}_{p(C_m^K, \mathbf{w}_i)}[\log p(C_m^K|\mathbf{w}_i)] + H(C_m^K) \\ &= \sum_{i=1}^{N}\sum_{m=1}^{|C^K|} p(C_m^K, \mathbf{w}_i) \log p(C_m^K|\mathbf{w}_i) + H(C_m^K) \\ &\geq \sum_{i=1}^{N}\sum_{r=1}^{R} p(C_m^{r,K}, \mathbf{w}_i) \log p(C_m^{r,K}|\mathbf{w}_i) + H(C_m^K), \quad (4)\end{aligned}$$

where $C_m^{r,K}$ denotes the permutation of neighbors representing a route to C_m^K in a given order and there are $R = \frac{N!}{(N-K)!}$ routes in total. The equality in Eq. (4) holds *iff* $p(C_m^K|\mathbf{w}_i) = p(C_m^{r,K}|\mathbf{w}_i)$ – i.e.,

there is one deterministic route for a given start and destination. However, the search space of R is huge and even prohibitive. We will later present an equivalent task in heuristic style, maximizing the MI between the community and the next neighbor iteratively.

Without loss of generality, we simplify the joint distribution $p(C_m^K, \mathbf{w}_i)$ as $p(C_{i,m}^K)$, and use $b^{r,k}$ to denote the k-th neighbor on the route, i.e., $p(C_m^{r,k}) = p(C_m^{r,k-1}, b^{r,k})$. Moreover, we assume that all nodes are equally selected as neighbors given an arbitrary start, i.e., $p(b^{r,k}) = \sum_{i=1}^{N} p(b^{r,k}|\mathbf{w}_i)p(\mathbf{w}_i) = \frac{1}{N-k+1}$. We can derive the log term in Eq. (4) as follows:

$$\log p(C_m^{r,K}|\mathbf{w}_i) = \sum_{k=1}^{K} \log \frac{p\left(b^{r,k}|C_{i,m}^{r,k-1}\right)}{p(b^{r,k})} + \sum_{k=1}^{K} \log p(b^{r,k}). \quad (5)$$

We split the joint distribution in Eq. (4) at an arbitrary k as

$$p(C_m^{r,K}, \mathbf{w}_i) \simeq p(C_{i,m}^{r,k}) \exp\left((K-k) \mathop{\mathbb{E}}_{b^k}\left[\log p(b^k|C_{i,m}^{k-1})\right]\right). \quad (6)$$

With Eq. (5) and (6), we can rewrite Eq. (4) as

$$I(\mathbf{W}; C^K) \geq H(C_m^K) + \xi(N,K) + \sum_{i=1}^{N}\sum_{k=1}^{K}\left(I\left(b^k; C_{i,m}^{k-1}\right) e^{(K-k)\,\mathbb{E}_{b^k}\left[\log sim\left(b^k, C_{i,m}^{k-1}\right)\right]} (N-k)^{(K-k)}\right), \quad (7)$$

where $\xi(N,K)$ is a constant determined by N and K, $I\left(b^k; C_{i,m}^{k-1}\right)$ is the MI between $(k-1)$-th context and k-th neighbor. According to [39], it has the following lower bound:

$$I\left(b^k; C_{i,m}^{k-1}\right) \geq \log N - \mathcal{L}_N^{i,k},$$

$$\text{where} \quad \mathcal{L}_N^{i,k} = - \mathop{\mathbb{E}}_{(b,c)}\left[\log \frac{\exp\left(sim\left(b^k, C_{i,m}^{k-1}\right)/\tau\right)}{\sum_{\epsilon \in E} \exp\left(sim\left(\epsilon, C_{i,m}^{k-1}\right)/\tau\right)}\right], \quad (8)$$

where $\mathcal{L}_N^{i,k}$ is the well-known InfoNCE loss widely used in contrastive learning [1, 4], τ is the temperature parameter, and ϵ denotes the negative samples. Besides, $sim\left(b^k, C_{i,m}^{k-1}\right) \propto \frac{p\left(b^k|C_{i,m}^{k-1}\right)}{p(b^k)}$ is a matching function measuring the affinity of b^k for $C_{i,m}^{k-1}$ – the higher the score, the higher likelihood of a positive context and a new neighbor. It usually contains three parts: two embedding functions that transform the original representations of one node and a group of nodes into contrastive embeddings, respectively, i.e., $\mathbf{R} = f(\mathbf{W}; \phi)$ and $\mathbf{R}_c = f(C^K; \Phi)$; and a scoring function that outputs the similarity between the given two embeddings, i.e., $s_{i,j} = f(\mathbf{r}_{i,c}, \mathbf{r}_j)$. For simplicity, we omit all the implementation details here and present them in Sec. 3.3.

Overall, given $(k-1)$-th neighbors and $sim()$, the optimization of $p_\theta(\mathbf{A}|\mathbf{V}, \mathbf{X})$ and $I(\mathbf{W}; C^K)$ can be achieved via maximizing the MI from $k = 1$ to K iteratively. This also means that the optimal neighbors w.r.t. to each nodes will be found while the global MI is maximized. The corresponding details of the derivation are in Appendix A, and we also prove in Appendix B that InfoNCE-based CPC model is a special case of our framework when $K = 1$.

Recent works have revealed the great impact of true negative samples on CL [7, 35]. To derive an unbiased version of Eq. (8), we sample noise data as follows:

$$\Gamma \sum_{\epsilon \in E^U} \exp\left(sim\left(\epsilon, C_{i,m}^{k-1}\right)/\tau\right) + (1-\Gamma) \sum_{\epsilon \in E^N} \exp\left(sim\left(\epsilon, C_{i,m}^{k-1}\right)/\tau\right), \quad (9)$$

where $\Gamma \in (0, 0.5)$ is a weighting parameter – i.e., we draw more true negative samples from E^N than E^U and reject the false negative samples from E^P.

3.2 Initialization and Training

There are two vital problems to be solved: how to define the positive edges at the beginning and when to stop training.

First, note that $sim()$ in Eq. (8) is optimized via distinguishing positive neighbors from the other nodes – i.e., the positive edges are indispensable to calculate a non-zero contrastive loss. The methods of selecting positive data pairs have been well studied in other domains. In computer vision [4] and graph learning [44], the positive data pairs are two augmentation views from the same image and graph structure. In NLP, words from the same sentence are all positive pairs [39]. Though arbitrary edges can be regarded as negative, the definition of positive edges is still a problem in an implicit graph where deterministic correlations are not available.

Second, the optimization of Eq. (7) obtains the neighbors that maximize the MI w.r.t. node i – but deciding the optimal parameters K for all the nodes is nontrivial, since different nodes have different receptive fields and the growth of MI is boundless with more neighbors, as shown in Lemma 3.1 (cf. Appendix C for proof).

Lemma 3.1. *Given a node* $\mathbf{w}$*, its existing neighbors* C *and the next neighbor* b*, we have:*

$$I(C, b; \mathbf{w}) \geq I(C; \mathbf{w}). \quad (10)$$

To address the above two problems, we design a self-paced graph contrastive learning framework that starts from reliable neighbors and searches all K optimal neighbors automatically.

If the initial reliable neighbors are not available, we first generate reliable positive and negative edges via KNN, which has low recall but high precision for a small $k \ll N$. Then we initialize the context and the next neighbor relative to node i – i.e., $C_{i,m}^{k-1}$ and b^k. The intuition is twofold: (1) the pair-wise proximity is insufficient to discriminate all relationships, but still serves as reasonable knowledge for reliable neighbors; and (2) the nodes near the discriminative margin are difficult to classify, but we are confident to find k reliable positive neighbors via unsupervised methods as long as k is small enough.

Next, given a context $C_{i,m}^{k-1}$ and the matching score function $sim()$, we modify the existing non-negative PUL loss – *a.k.a* unbiased empirical risk estimator [5, 25] – to a positive-negative-unlabeled version:

$$\mathcal{L}_{PU}^{i,k} = \frac{\eta}{|E_i^P|} \sum_{b \in E_i^P} \ell\left(g(b), 1\right) + \frac{1-\eta}{|E_i^N|} \sum_{b \in E_i^N} \ell\left(g(b), -1\right) + \max\left(0, \frac{1}{|E_i^U|} \sum_{b \in E_i^U} \ell\left(g(b), -1\right) - \frac{\eta}{|E_i^P|} \sum_{b \in E_i^P} \ell\left(g(b), -1\right)\right), \quad (11)$$

where $\eta \in (0,1)$ is the assumed prior probability of positive edges approximated by $|E^P|/|E|$. Edges in E_i^P are positive w.r.t. to node i, and $g(b) = sim(b, C_{i,m}^{k-1})$ is the matching score of b for a given context. In addition, $\ell : \mathbb{R} \times \{\pm 1\} \to \mathbb{R}$ is the classifier loss implemented as a Sigmoid loss here, i.e., $\ell(x,y) = 1/(1+\exp(xy))$.

After SPGCL and $sim()$ at k step have converged, we sort the scores of the unlabeled edges and select the edges with the most confident scores to label, which generates E at $k+1$ step. For brevity, we only discuss the positive ones. Ideally, we could select the edge with the highest score from E^U and add it to E^P; however, this process is extremely slow in practice. Therefore, we choose α edges whose scores are larger than δ^+ at each step. Edges whose scores are smaller than δ^+ in λ epochs are removed from E^P. Such strict access and elimination rules ensure that the positive edges in E^P are reliable and the training process is stable. For the negative edges, the hyperparameters are δ^- and β. In summary, the choices of $\delta^\pm$, α and β are empirically balanced between training time and stability, which, however, are not directly related to the optimal model.

Note that the coefficients are larger when k is smaller in Eq. (7), which means the nearest neighbors contribute more to global MI. Therefore, we have a high threshold at the beginning of training and with a decaying factor to ensure that only the most reliable nodes are selected when k is small. The details are described in Algorithm 2 in Appendix D.

3.3 Architecture of SPGCL

Instead of learning node embeddings first and then the correlations between nodes as most previous works [2, 21, 22, 24], we learn the spatio-temporal interactions directly via mapping relative representations to relation embeddings, which may also benefit to model transferability [36]. The whole training process of SPGCL is summarized in Algorithm 1 in Appendix D.

Without loss of generality, we first establish relative spatio-temporal coordinate systems with node i as the origin, and propose four relationship measurements between node i and its neighbors:

$$\begin{aligned} w_{i,j}^{dis} &= \|\mathbf{V}_i - \mathbf{V}_j\|_2, \\ w_{i,j}^{ang} &= \sin(\theta_{i,j}) = \frac{\mathbf{V}_i[2] - \mathbf{V}_j[2]}{w_{i,j}^{dis}}, \\ w_{i,j}^{seq} &= \mathrm{DTW}(\mathbf{x}_i^t, \mathbf{x}_j^t), \\ w_{i,j}^{t} &= \mathbf{x}_i^t - \mathbf{x}_j^t, \end{aligned} \tag{12}$$

where $w_{i,j}^{dis}$ and $w_{i,j}^{ang}$ are the spatial polar coordinates of node j relative to i; $w_{i,j}^{seq}$ is the similarity between temporal sequences $\mathbf{X}_i^t$ and $\mathbf{X}_j^t$ calculated by DTW distance [3] that are widely used for measuring the similarity between sequential data [13, 26]); and $w_{i,j}^t$ is the sequential distance from $\mathbf{x}_i^t$ to $\mathbf{x}_j^t$. $\|\cdot\|_2$ is a L2 norm.

Next, we can concatenate the representation of j relative to i $\mathbf{w}_{i,j}^t = \left[w_{i,j}^{dis}, w_{i,j}^{ang}, w_{i,j}^{seq}, w_{i,j}^t\right]$ if all measurements are available, where the first two and the last two dimensions measure spatial and temporal proximity respectively. Denote $\mathbf{W}_i^t = \{\mathbf{w}_{i,1}^t, \mathbf{w}_{i,2}^t, \cdots, \mathbf{w}_{i,N}^t\} \in \mathbb{R}^{N \times F'}$, where $\mathbf{w}_{i,j}^t \in \mathbb{R}^{F'}$ and $\mathbf{w}_{i,i}^t = \mathbf{0}$. Finally, we apply normalization on $\mathbf{W}^t$ to avoid scalar imbalance.

Before data is fed into SPGCL, augmentations are applied to improve the robustness and transferability. In our model, we used three types of data augmentations as suggested by [44], including node sampling, edge perturbation and attribute masking.

Now we introduce the implementation details of graph attention based SPGCL, where the learned relations are attentional coefficients. Given E at k step, we first get the embeddings of spatio-temporal relationships:

$$\mathbf{r}_{i,j}^t = f(\mathbf{w}_{i,j}^t; \phi) = \mathrm{MLP}\left(\mathbf{w}_{i,j}^t\right), \tag{13}$$

where $\mathbf{r}_{i,j}^t \in \mathbb{R}^{F'}$ is the spatio-temporal embedding and MLP is a multilayer perceptron. We then aggregate features from neighbors to get the embedding of context relative to node i:

$$\mathbf{r}_{i,c}^t = f(C_i^k; \Phi) = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \left(\mathbf{r}_{i,j}^t \odot \mathbf{w}_{i,j}^t\right), \tag{14}$$

where $\odot$ is Hadamard product, and $\mathcal{N}_i = \{j | e_{i,j} \in E_i^P, j = 1, 2, \cdots, N\}$ is the neighbors of node i. The matching score is calculated as

$$s_{i,j}^t = f(\mathbf{r}_{i,c}^t, \mathbf{r}_{i,j}^t) = -\|\mathbf{r}_{i,c}^t - \mathbf{r}_{i,j}^t\|_2, \tag{15}$$

where we further apply *min-max* normalization to rescale scores so that $s_{i,j}^t \in [-1, 1]$. The similarity function $sim()$ is therefore determined by computing Eq. (13), (14) and (15). Note that here we choose negative Euclidean distance to measure the proximity instead of the widely used cosine similarity in contrastive learning, because we contrast the representations of edge samples and aggregated neighbors rather than positive and negative samples as common settings. The superiority of negative Euclidean distance is also verified in our practice. Subsequently, we make predictions via aggregating features from neighbors and calculate the predictive loss as

$$\begin{aligned} \widehat{\mathbf{X}}_i &= \frac{1}{H|\mathcal{N}_i|} \sum_{h=1}^{H} \sum_{j \in \mathcal{N}_i} \left(\mathbf{M}^h \left(\rho\left(\mathbf{R}_{i,j}\mathbf{U}^h\right) \odot \mathbf{X}_j\right)\right), \\ \mathcal{L}_{mse}^{i,k} &= \|\mathbf{X}_i - \widehat{\mathbf{X}}_i\|_2^2, \end{aligned} \tag{16}$$

where $\mathbf{R}_{i,j} \in \mathbb{R}^{T \times F'}$ is the relationship matrix; $\mathbf{X}_j \in \mathbb{R}^{T \times F}$ is node features; $\mathbf{U} \in \mathbb{R}^{F' \times F}$ and $\mathbf{M} \in \mathbb{R}^{T' \times T}$ represent the linear transformations; H is the number of multi-head aggregation; $\rho(x) = \mathrm{LeakyReLU}(x)$ is the nonlinear activation.

In addition to the predictive loss in Eq. (16), we sample edges from E^P, E^U and E^N to calculate contrastive loss $\mathcal{L}_N^{i,k}$ and PU loss $\mathcal{L}_{PU}^{i,k}$ via Eq. (8) and (11). Overall, we train model via minimizing the hybrid loss at k-th step:

$$\mathcal{L}^k = \frac{1}{N} \sum_{i=1}^{N} \left(\mathcal{L}_N^{i,k} + \gamma_1 \mathcal{L}_{PU}^{i,k} + \gamma_2 \mathcal{L}_{mse}^{i,k}\right), \tag{17}$$

where γ is the reweighting hyperparameter. When testing, E^P and the optimal relations are obtained by Algorithm 1 quickly, without calculating loss and gradient backpropagation, and predictions are made via Eq. (16), respectively.

4 EXPERIMENTS

We now describe the details of our experimental evaluations.

4.1 Experimental Settings

Datasets. To evaluate the performance of SPGCL, we used four real-world datasets from two distinct spatio-temporal learning domains. The first two are INSAR-measured land deformation records of slopes on both sides of a large-scale hydropower dam – HZY-East and HZY-West collected by [48]. The other two are traffic flow data from two highways (PEMS03 and PEMS04) released by [38]. Table 1 summarizes the statistics of the datasets, and we note that the PEMS datasets have much fewer locations but longer observations.

Table 1: Dataset statistics.

	Nodes	Edges	Frequency	Time range
HZY-East	2,164	0	2 weeks	11/30/2018 - 9/8/2019
HZY-West	4,569	0	2 weeks	11/30/2018 - 9/8/2019
PEMS03	358	547	5 minutes	9/1/2018 - 11/30/2018
PEMS04	307	340	5 minutes	1/1/2018 - 2/28/2018

Baselines. We compare SPGCL with following baseline models:

- ARIMA: Auto-regressive Integrated Moving combines auto-regressive and moving average for time series prediction.
- SVR: Support Vector Regression uses a linear support vector machine for regression tasks.
- GRU [8]: Gated Recurrent Unit network, which captures long-short term dependency on time series.
- STGCN [45]: Spatial-Temporal Graph Convolutional Network uses spatial-graph convolution and temporal-gated convolution to capture the spatial and temporal dependencies, respectively.
- ASTGCN [16]: The attention-based spatio-temporal graph convolutional network consists of a spatial attention network and a temporal attention network. Besides, MSTGCN [16] is also introduced as a degraded version, which does not have the spatio-temporal attention mechanism.
- SA-GNN [48]: Slope-Aware Graph Neural Network proposes a weighted locally linear embedding to learn surface manifold through modeling the spatial dependency among different monitored locations.
- STGODE [13]: Spatial-Temporal Graph Ordinary Differential Equation Networks, which captures spatial-temporal dynamics through a tensor-based ordinary differential equation and utilize spatial-temporal features simultaneously.
- AGCRN [2]: Adaptive Graph Convolutional Recurrent Network, which adopts a learnable graph rather than a static one to infer the inter-dependencies among different traffic series automatically.
- IDGL [6]: Iterative Deep Graph Learning, which is an end-to-end graph learning framework for jointly and iteratively learning the graph structure and graph embedding. As a variant, IDGL-ANCH [6] uses an extra regularization loss on the anchor graph.

Settings. We split all the datasets with a 5:3:2 ratio into training sets, validation sets, and testing sets. For traffic data, we use the past 12 time steps to predict the future 12 time steps, and the results are evaluated via the root mean squared errors (RMSE), mean absolute errors (MAE), and mean absolute percentage errors (MAPE), following the setting in [2, 13]. For land deformation data, we use past 3 time steps to predict the future 1 time steps, and the results are evaluated via RMSE, MAE, accuracy with threshold 1 (ACC), coefficient of determination (R^2), and explained variance score (EVS), following the setting in [48]. While there are inherent (sparse) edges in PEMS03 and PEMS04, the adjacency matrices on HZY-East and HZY-West are generated by KNN with $K = 200$ and 250, respectively, except for methods with adaptive adjacency construction. All deep learning models, including ours, are optimized via Adam optimizer with an initial learning rate $3e^{-4}$, which decays with the rate of 0.9 every 50 epochs. Besides, early stopping is triggered when validation loss has not declined for 20 consecutive epochs. The hyperparameters of SPGCL are tuned as follows: the cross entropy temperature $\tau = 0.1$; prior probability $\eta = 0.1$; and the weighting parameters $\Gamma = 0.3$, $\gamma_1 = 10$, and $\gamma_2 = 20$. We have 2 layers of aggregation with $H = 3$ multi-heads. For HZY datasets, embedding dimensions $F' = 12$, patience $\lambda = 10$, the positive and negative threshold is $\delta^+ = 0.9$ and $\delta^- = 0.4$ respectively with $\alpha = \beta = 20$, the numbers of positive, unlabeled and negative neighbor samples of node i are $\min(|E_i^P|, 200)$, $\min(|E_i^U|, 2000)$ and $\min(|E_i^N|, 2000)$ respectively. For PEMS datasets with less nodes but longer time series, the hyperparameters are $F' = 24$, $\lambda = 5$, $\delta^+ = 0.9$, $\delta^- = 0.2$, the numbers of positive, unlabeled and negative edge samples are $\min(|E_i^P|, 30)$, $\min(|E_i^U|, 300)$ and $\min(|E_i^N|, 300)$, respectively.

All experiments are conducted on a NVIDIA GeForce RTX 3090 GPU, and the reported results are the best of 20 runs for all models.

4.2 Overall Comparisons

The overall performance comparisons are summarized in Table 2. Note that SA-GNN performs well on HZY, but the specially designed embedding algorithm requires spatial coordinates, which are missed on PEMS datasets. According to the results, deep learning methods achieved better results than statistical methods such as ARIMA and SVR – lacking spatial dependencies modeling. MSTGCN and STGCN model the spatial and temporal dependencies separately, which leads to poor performance. Though STGODE and ASTGCN consider the spatio-temporal dependencies simultaneously, they still rely on static graphs with manually set hyperparameters (distance thresholds or K neighbors). Their receptive fields are either oversized or undersized when nodes are densely or sparsely distributed with static global hyperparameters. Moreover, from a temporal perspective, the relations between nodes are also dynamically changing. SPGCL and AGCRN outperforms other baselines significantly, indicating the superiority of adaptive graph construction over static ones. However, AGCRN still relies on the inner vector product to infer the Laplacian matrix and the corresponding graph convolution to make forecasts, rather than taking full advantage of the learned informative pair-wise relationships. Though IDGL has used adaptive graphs, it is designed for classification tasks and therefore performs poorly on spatio-temporal forecasting.

It is noteworthy that the superiority of SPGCL is more significant on HZY than on PEMS, which can be attributed to three aspects. First, there is no predefined adjacency matrix in the HZY dataset. Note that KNN is utilized for other models except the adaptive methods (i.e., IDGL, AGCRN, and our SPGCL), which may inevitably introduce bias. Second, spatial coordinates are missed on PEMS, causing only two measurements in Eq. (12) to be available, and thus the expressiveness of SPGCL is limited. Finally, data changes rapidly and periodically on PEMS, but slowly and has a noticeable trend (land deformation) on HZY. Finally, SPGCL focuses on designing

Table 2: Overall performance comparisons of different approaches on two types of spatio-temporal forecasting. We use the paired t-test with significance level at 0.05 on all reported results.

	HZY-East					HZY-West					PEMS03			PEMS04		
Method	**RMSE**	**MAE**	**ACC**	**R^2**	**EVS**	**RMSE**	**MAE**	**ACC**	**R^2**	**EVS**	**RMSE**	**MAE**	**MAPE(%)**	**RMSE**	**MAE**	**MAPE(%)**
ARIMA	6.704	3.693	0.021	0.171	0.204	4.888	4.088	0.045	0.105	0.215	47.59	35.41	33.78	73.19	37.21	29.38
SVR	4.750	3.693	0.029	0.135	0.262	3.956	3.209	0.022	0.108	0.274	36.92	22.95	22.62	44.55	28.73	19.20
GRU	0.277	0.235	0.278	0.197	0.253	0.250	0.204	0.373	0.223	0.315	35.51	21.73	23.86	41.78	27.68	19.71
IDGL	0.197	0.170	0.328	0.207	0.290	0.169	0.164	0.591	0.286	0.293	33.46	20.21	21.77	39.29	25.59	18.35
IDGL-ANCH	0.184	0.188	0.467	0.288	0.286	0.150	0.158	0.598	0.295	0.301	33.02	19.84	20.93	37.69	24.66	17.24
STGCN	0.152	0.141	0.571	0.376	0.396	0.142	0.148	0.607	0.315	0.315	30.22	17.40	**17.08**	34.92	21.23	14.04
MSTGCN	0.143	0.149	0.575	0.389	0.409	0.146	0.147	0.621	0.356	0.356	32.70	19.80	22.03	38.39	25.15	19.02
ASTGCN	0.127	0.097	0.677	0.624	0.636	0.137	0.127	0.664	0.493	0.494	33.22	19.38	19.10	35.22	22.93	16.56
SA-GNN	0.124	0.104	0.723	0.646	0.646	0.103	0.098	0.709	0.641	0.655	-	-	-	-	-	-
STGODE	0.106	0.087	0.769	0.729	0.730	0.115	0.117	0.698	0.583	0.588	28.74	16.58	17.59	33.02	20.97	13.84
AGCRN	0.098	0.069	0.839	0.751	0.751	0.100	0.103	0.732	0.669	0.669	28.12	16.92	17.98	32.11	19.67	**13.37**
SPGCL	**0.093**	**0.062**	**0.852**	**0.766**	**0.766**	**0.091**	**0.096**	**0.789**	**0.685**	**0.687**	**27.79**	**16.52**	17.91	**31.85**	**19.41**	15.50

a general neighbor selection and adjacency construction method. Unlike other STGNNs, it did not pay attention to the periodicity trend of specific data. Without finetuning on the attributed data, SPGCL may suffer from the oversmoothing issue when aggregating features, which also explains why the performance of SPGCL is not the best in some cases.

4.3 Ablation Study

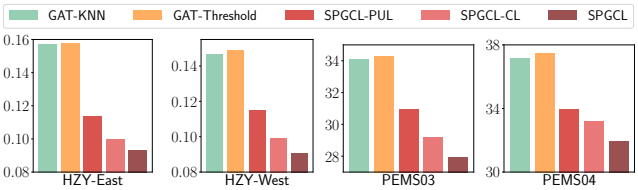


Figure 3: RMSE results of ablation experiments.

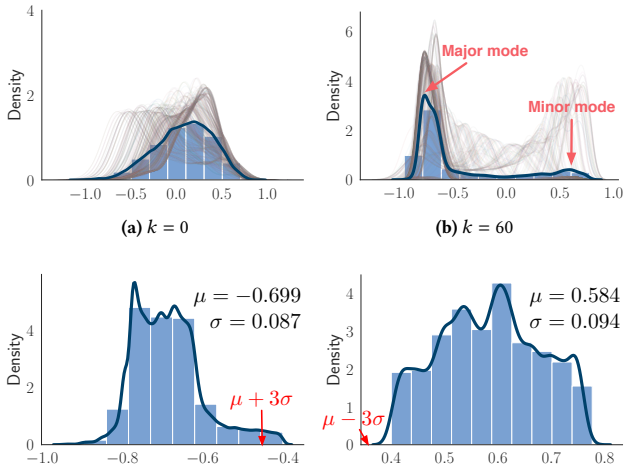
To further verify the effectiveness of the proposed hybrid loss in Eq. (17) and investigate different adjacency construction strategies, we conduct the following ablation experiments:

- GAT-KNN: The adjacency matrix is constructed by KNN with $k = 200, 250, 60, 50$ on four datasets. A 3-layers GAT [40] with 3 multi-heads is utilized to make predictions.
- GAT-Threshold: The adjacency matrix is constructed by connecting node pairs whose DTW distance is smaller than 0.5. The same GAT structure with GAT-KNN is applied.
- SPGCL-PUL: The contrastive loss (cf. Eq. (8)) is removed, but the predicting module remains the same.
- SPGCL-CL: The Positive-negative-unlabeled loss (cf. Eq. (11)) is removed and the other settings are the same.

The RMSE results shown in Fig. 3 indicate that adaptive methods significantly outperform static ones because: (1) the manually set hyperparameters are not optimal for all nodes; and (2) adjacent relations may change along with time. This phenomenon is more significant on HZY than PEMS, since there are more nodes in HZY data and the land deformation varies slowly, i.e., it is easier to model this trending pattern than the traffic pattern.

Though the empirical risk estimator of PUL contributes to forecasts and the adaptive labeling strategy, SPGCL-CL obtains better performance than SPGCL-PUL. This result indicates that the risk estimator is less effective than contrastive loss and the great impact of learning from contrasting positive and non-positive representations. Besides, SPGCL-CL performs better on HZY, as there are enough negative samples in the HZY dataset, which has been proved critical in contrastive learning [4, 39, 44].

4.4 Visualizations of SPGCL



(a) $k = 0$ **(b)** $k = 60$

(c) Negative scores distribution **(d) Positive scores distribution**

Figure 4: The distributions of similarity scores.

Visualization of similarity score distribution. We first visualize the distributions of similarity scores on HZY-east at 0th step and 60th step in Fig. 4a and 4b respectively. The histogram shows the distribution of all scores, and the N grey lines represent the probability density of scores associated with each of N nodes. Clearly, scores follow a symmetrical unimodal distribution when $k = 0$, i.e., normal distribution, which is intuitive since parameters are

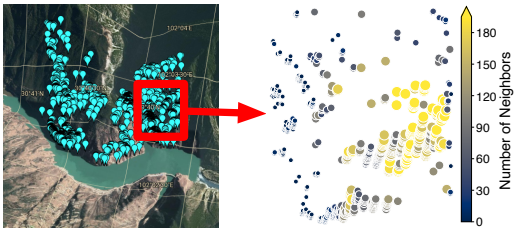


Figure 5: Neighbors of each nodes on the cropped area, where the radius and color indicate the number of neighbors.

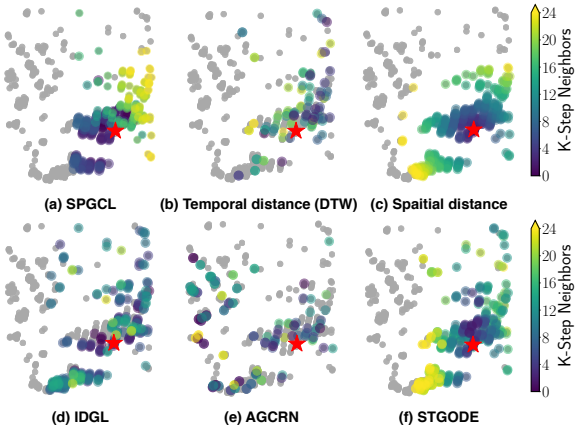


Figure 6: Selected neighbors of center node at different steps.

randomly initialized. The distribution at $k = 60$ is an asymmetrical bimodal distribution which means that the optimized SPGCL is capable of discriminating positive and negative edges. The observed major mode is much larger than the minor mode since the assumed positive prior $\eta = 0.1$ is low. We further draw the distributions of positive and negative scores in Fig. 4c and 4d, which are selected based on $\delta^{-}_{\text{opt}} = -0.4$ and $\delta^{+}_{\text{opt}} = 0.4$. The ratio of obtained positive edges is 0.5M / 4.6M $\simeq 0.12 \simeq \eta$. Considering the three-sigma rule of thumb, we are confident that most positive and negative edges are included in E^P and E^N respectively, and the rest edges remain in E^U.

Visualization of neighbor selection. Notable, some nodes take many other nodes as neighbors – i.e., the grey lines have large major modes. We further investigate those nodes with large major mode and draw scatters on the cropped area for brevity, as shown in Fig. 5, where the radius represents the number of neighbors selected by SPGCL. We found an apparent clustering effect of larger circles (the yellow ones). In other words, the nodes in some monitored areas exhibit high spatio-temporal proximity. To validate this observation, in Fig. 6 we show the neighbors of the node 425 (red star) chosen by different methods. For intuitive comparison, we compare SPGCL with three adaptive methods and two basic metrics, i.e., DTW and spatial Euclidean distance with manual threshold, representing the temporal and spatial proximity, respectively. Among all methods, SPGCL orders neighbors by affinity and shows a convincing neighbor selecting strategy considering spatio-temporal proximity, which is emphasized in spatio-temporal forecasting tasks [19, 48]. In contrast, AGCRN shows disorganized selection since it learns task-specific and forecasting-oriented node embeddings, neglecting the spatio-temporal dependencies. IDGL introduces iterative learning and graph similarity metric learning, obtaining more reasonable results than AGCRN, but it is still unable to generate interpretable embeddings. STGODE considers both spatial and DTW distances with manually designed thresholds, which obtains competitive performance but is less flexible than SPGCL's adaptive adjustment based on spatio-temporal proximity.

4.5 Transferability & Robustness

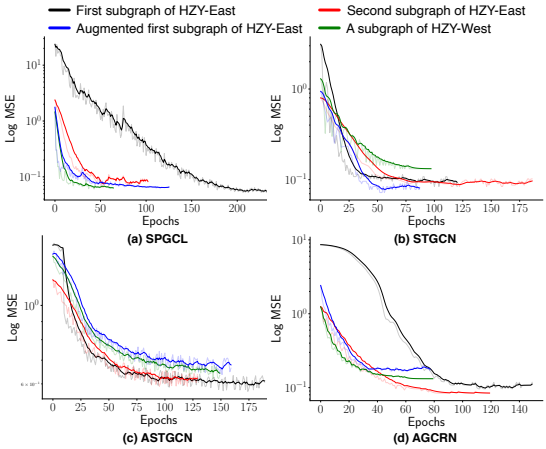


Figure 7: Predictive loss curves on 4 datasets.

One major advantage of our SPGCL over other STGNN models lies in its transferability and robustness. To validate this argument, we conduct a transfer learning experiment on HZY data. Specifically, we pre-train the models on a subgraph sampled from HZY-East and use the learned models to evaluate their performance on the target datasets – i.e., the augmented sample subgraph, another subgraph from HZY-East, and a subgraph from HZY-West. All models are pre-trained on the sampled subgraph until convergence and then directly transferred to the testing subgraphs having different node distributions. The training processes of all models are plotted in Fig. 7 where the light color lines and solid lines are the original and smoothed MSE loss, respectively. It shows that SPGCL converges within 50 epochs on the testing datasets with high stability, validating the superiority of SPGCL in terms of transferability and robustness. We also find slow convergence of STGCN and ASTGCN on testing subgraphs, which shows no improvement compared to the pre-training model. In contrast, AGCRN and SPGCL benefit a lot from pre-training and converge fast on target subgraphs. Besides, we note that SPGCL achieves the best performance within 50 epochs – significantly faster than other models. Overall, these results validate the high transferability and model robustness of SPGCL as it can quickly and stably generalize to different datasets.

5 CONCLUSION

We presented SPGCL, a general model for spatio-temporal learning which emphasizes the semantic relationships between monitored locations, and caters to graph construction when the deterministic adjacency is not available. SPGCL learns spatial and temporal

dependencies simultaneously and maps the pair-wise proximity to informative relation embeddings. The optimal node adjacency is obtained via a self-paced paradigm, selecting the new neighbors by maximizing the mutual information. While the graph approaches the optimized structure, the mutual information between nodes and their neighborhoods is also maximized. SPGCL is a general adjacency construction algorithm and can be explored by different downstream tasks. The comprehensive experiments conducted on two spatio-temporal learning tasks (with two datasets each) verified the effectiveness of our method. In the future, we attempt to extend SPGCL to other graph learning tasks where adjacency relations are not provided such as graph-based text classification, anomaly detection and recommendation. In addition, designing a better message passing mechanism or information aggregation strategy based on the proposed SPGCL is another direction of our future work.

ACKNOWLEDGEMENTS

This work was supported in part by National Natural Science Foundation of China (Grant No.62176043 and No.62072077), Sichuan Science and Technology Program (No.2022YFSY0006), and National Science Foundation SWIFT (Grant No.2030249).

REFERENCES

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. 2019. Learning Representations by Maximizing Mutual Information across Views. In *NeurIPS*. 15535–15545.
[2] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. In *NeurIPS*. 17804–17815.
[3] Donald J. Berndt and James Clifford. 1994. Using Dynamic Time Warping to Find Patterns in Time Series. In *SIGKDD*. 359–370.
[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. 1597–1607.
[5] Xuxi Chen, Wuyang Chen, Tianlong Chen, Ye Yuan, Chen Gong, Kewei Chen, and Zhangyang Wang. 2020. Self-pu: Self boosted and calibrated positive-unlabeled training. In *ICML*. 1510–1519.
[6] Yu Chen, Lingfei Wu, and Mohammed Zaki. 2020. Iterative Deep Graph Learning for Graph Neural Networks: Better and Robust Node Embeddings. In *NeurIPS*. 19314–19326.
[7] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased Contrastive Learning. In *NeurIPS*. 8765–8775.
[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:1412.3555
[9] Pádraig Cunningham and Sarah Jane Delany. 2021. k-Nearest Neighbour Classifiers - A Tutorial. *Comput. Surveys* 54, 6 (Jul 2021), 1–25.
[10] Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljacic. 2022. Equivariant Self-Supervised Learning: Encouraging Equivariance in Representations. In *ICLR*.
[11] Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. 2014. Analysis of learning from positive and unlabeled data. In *NeurIPS*. 703–711.
[12] Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *SIGKDD*. 213–220.
[13] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. 2021. Spatial-temporal graph ode networks for traffic flow forecasting. In *SIGKDD*. 364–373.
[14] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. 2019. Learning discrete structures for graph neural networks. In *ICML*. 1972–1982.
[15] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting. In *AAAI*. 3656–3663.
[16] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI*. 922–929.
[17] Xiaojie Guo, Yuanqi Du, and Liang Zhao. 2021. Deep Generative Models for Spatial Networks. In *SIGKDD*. 505–515.
[18] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*. 297–304.
[19] Ainaz Hajimoradlou, Gioachino Roberti, and David Poole. 2020. Predicting Landslides Using Locally Aligned Convolutional Neural Networks. In *IJCAI*. 3342–3348.
[20] Jindong Han, Hao Liu, Hengshu Zhu, Hui Xiong, and Dejing Dou. 2021. Joint Air Quality and Weather Prediction Based on Multi-Adversarial Spatiotemporal Networks. In *AAAI*. 4081–4089.
[21] Arman Hasanzadeh, Ehsan Hajiramezanali, Krishna Narayanan, Nick Duffield, Mingyuan Zhou, and Xiaoning Qian. 2019. Semi-Implicit Graph Variational Auto-Encoders. In *NeurIPS*. 10711–10722.
[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*. 9729–9738.
[23] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.
[24] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. arXiv:1611.07308
[25] Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. 2017. Positive-unlabeled learning with non-negative risk estimator. In *NeurIPS*. 1674–1684.
[26] Mengzhang Li and Zhanxing Zhu. 2021. Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting. In *AAAI*. 4189–4196.
[27] Xiaoli Li and Bing Liu. 2003. Learning to classify texts using positive and unlabeled data. In *IJCAI*. 587–592.
[28] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*.
[29] Ishan Misra and Laurens van der Maaten. 2020. Self-supervised learning of pretext-invariant representations. In *CVPR*. 6707–6717.
[30] Cheonbok Park, Chunggi Lee, Hyojin Bahng, Yunwon Tae, Seungmin Jin, Kihwan Kim, Sungahn Ko, and Jaegul Choo. 2020. ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed. In *CIKM*. 1215–1224.
[31] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph representation learning via graphical mutual information maximization. In *WWW*. 259–270.
[32] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. 2019. On variational bounds of mutual information. In *ICML*. 5171–5180.
[33] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *SIGKDD*. 1150–1160.
[34] Herbert Robbins. 1955. A remark on Stirling's formula. *The American mathematical monthly* 62, 1 (1955), 26–29.
[35] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive Learning with Hard Negative Samples. In *ICLR*.
[36] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. 2021. E (n) equivariant graph neural networks. In *ICML*. 9323–9332.
[37] Chao Shang, Jie Chen, and Jinbo Bi. 2020. Discrete Graph Structure Learning for Forecasting Multiple Time Series. In *ICLR*.
[38] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *AAAI*. 914–921.
[39] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation Learning with Contrastive Predictive Coding. arXiv:1807.03748
[40] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.
[41] 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 *WWW*. 1082–1092.
[42] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *TNNLS* 32, 1 (2020), 4–24.
[43] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. 2021. InfoGCL: Information-Aware Graph Contrastive Learning. In *NeurIPS*.
[44] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph Contrastive Learning with Augmentations. In *NeurIPS*. 5812–5823.
[45] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *IJCAI*. 3634–3640.
[46] Qi Zhang, Jianlong Chang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. 2020. Spatio-temporal graph structure learning for traffic forecasting. In *AAAI*. 1177–1185.
[47] Cheng Zheng, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Haifeng Chen, and Wei Wang. 2020. Robust Graph Representation Learning via Neural Sparsification. In *ICML*. 11458–11468.
[48] Fan Zhou, Rongfan Li, Kunpeng Zhang, and Goce Trajcevski. 2021. Land Deformation Prediction via Slope-Aware Graph Neural Networks. In *AAAI*. 15033–15040.
[49] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *WWW*. 2069–2080.

A DERIVATION OF EQUATIONS

A.1 MI Bound – Route Entropy Inequality

Here we derive the inequality in Eq. (4). Given node $\mathbf{w}_i$, assuming that there are $R_m^i = K!$ routes to the m-th combination of neighbors, i.e., $p(C_m^K|\mathbf{w}_i) = \sum_{r=1}^{R_m^i} p(C_m^{r,K}|\mathbf{w}_i)$. And we have following derivation:

$$
\begin{aligned}
&\sum_{m=1}^{|C^K|} p(C_m^K, \mathbf{w}_i) \log p(C_m^K|\mathbf{w}_i) \\
=&\sum_{m=1}^{|C^K|} p(\mathbf{w}_i) \left(\sum_{r=1}^{R_m^i} p(C_m^{r,K}|\mathbf{w}_i) \right) \log p(C_m^K|\mathbf{w}_i) \\
=&\sum_{m=1}^{|C^K|} p(\mathbf{w}_i) \left(\sum_{r=1}^{R_m^i} p(C_m^{r,K}|\mathbf{w}_i) \log p(C_m^K|\mathbf{w}_i) \right) \\
=&\sum_{m=1}^{|C^K|} p(\mathbf{w}_i) \left(\sum_{r=1}^{R_m^i} p(C_m^{r,K}|\mathbf{w}_i) \log \left(p(C_m^{r,K}|\mathbf{w}_i) + \sum_{j\neq r} p(C_m^{j,K}|\mathbf{w}_i) \right) \right) \\
\geq&\sum_{m=1}^{|C^K|} p(\mathbf{w}_i) \left(\sum_{r=1}^{R_m^i} p(C_m^{r,K}|\mathbf{w}_i) \log p(C_m^{r,K}|\mathbf{w}_i) \right) \\
=&\sum_{m=1}^{|C^K|} \sum_{r=1}^{R_m^i} p(C_m^{r,K}, \mathbf{w}_i) \log p(C_m^{r,K}|\mathbf{w}_i) \\
=&\sum_{r=1}^{R} p(C_m^{r,K}, \mathbf{w}_i) \log p(C_m^{r,K}|\mathbf{w}_i),
\end{aligned} \tag{18}
$$

where $R = \frac{N!}{(N-K)!} = R_m^i \times |C^K|$, the equal holds if and only if $p(C_m^K|\mathbf{w}_i) = p(C_m^{r,K}|\mathbf{w}_i)$ and $\sum_{j\neq r} p(C_m^{j,K}|\mathbf{w}_i) = 0$. This conclusion is intuitive because the entropy is smaller when the probability of the optimal route is larger.

A.2 Derivation of Eq. (5) and (6)

Eq. (5) is derived as follows:

$$
\begin{aligned}
&\log p(C_m^{r,K}|\mathbf{w}_i) \\
=&\log \left[p(b^{r,K}|C_{i,m}^{r,K-1}) p(b^{r,K-1}|C_{i,m}^{r,K-2}) \cdots p(b^{r,1}|\mathbf{w}_i) \right] \\
=&\sum_{k=1}^{K} \log p\left(b^{r,k}|C_{i,m}^{r,k-1} \right) \\
=&\sum_{k=1}^{K} \log \frac{p\left(b^{r,k}|C_{i,m}^{r,k-1} \right)}{p(b^{r,k})} + \sum_{k=1}^{K} \log p(b^{r,k}),
\end{aligned} \tag{19}
$$

where we further denote $\xi(N,K) = \sum_{k=1}^{K} \log p(b^{r,k})$ for simplicity, which can be calculated as follows:

$$
\xi(N,K) \simeq K + \log \frac{(N-K)^{(N-K+\frac{1}{2})}}{N^{(N+\frac{1}{2})}}, \tag{20}
$$

where the approximation is more accurate when $N \gg K$, which is derived from Stirling's formula [34]. Besides, it has a more roughly lower bound $-K \log N$ for computation convenience.

Eq. (6) is derived as follows:

$$
\begin{aligned}
&p(C_m^{r,K}, \mathbf{w}_i) = p(C_{i,m}^{r,k}, b^{r,k+1}, ..., b^{r,K}) \\
=&p(C_{i,m}^{r,k}) \prod_{k'=k+1}^{K} p(b^{r,k'}|C_{i,m}^{r,k'-1}) \\
=&p(C_{i,m}^{r,k}) \exp\left(\sum_{k'=k+1}^{K} \log p(b^{r,k'}|C_{i,m}^{r,k'-1}) \right) \\
\simeq&p(C_{i,m}^{r,k}) \exp\left((K-k) \mathop{\mathbb{E}}_{b^k} \left[\log p(b^k|C_{i,m}^{k-1}) \right] \right),
\end{aligned} \tag{21}
$$

where the sum is estimated by Monte Carlo.

A.3 Relationship of Local and Global MI

Based on Eq. (19) and (21), Eq. (4) can be derived as follows:

$$
\begin{aligned}
&I(\mathbf{W}; C^K) \geq \sum_{i=1}^{N} \sum_{r=1}^{R} p(C_m^{r,K}, \mathbf{w}_i) \log p(C_m^{r,K}|\mathbf{w}_i) + H(C_m^K) \\
\geq&\sum_{i=1}^{N} \sum_{r=1}^{R} \left(\sum_{k=1}^{K} p(C_m^{r,K}, \mathbf{w}_i) \log \frac{p\left(b^{r,k}|C_{i,m}^{r,k-1} \right)}{p(b^{r,k})} \right) + \xi(N,K) + H(C_m^K) \\
\simeq&H(C_m^K) + \xi(N,K) + \sum_{i=1}^{N} \sum_{r=1}^{R} \sum_{k=1}^{K} \Bigg(e^{(K-k) \mathbb{E}_{b^k} \left[\log p(b^k|C_{i,m}^{k-1}) \right]} \\
&\left(p(C_{i,m}^{r,k}) \log \frac{p(b^{r,k}|C_{i,m}^{r,k-1})}{p(b^{r,k})} \right) \Bigg) \\
=&H(C_m^K) + \xi(N,K) + \sum_{i=1}^{N} \sum_{k=1}^{K} \Bigg(e^{(K-k) \mathbb{E}_{b^k} \left[\log p(b^k|C_{i,m}^{k-1}) \right]} \\
&\sum_{r=1}^{R} \left(p(C_{i,m}^{r,k}) \log \frac{p(b^{r,k}|C_{i,m}^{r,k-1})}{p(b^{r,k})} \right) \Bigg) \\
=&H(C_m^K) + \xi(N,K) + \sum_{i=1}^{N} \sum_{k=1}^{K} \Bigg(e^{(K-k) \mathbb{E}_{b^k} \left[\log p(b^k|C_{i,m}^{k-1}) \right]} \\
&\sum_{b^{r,k}} \sum_{C_{i,m}^{r,k-1}} \left(p(b^{r,k}, C_{i,m}^{r,k-1}) \log \frac{p(b^{r,k}|C_{i,m}^{r,k-1})}{p(b^{r,k})} \right) \Bigg) \\
=&\sum_{i=1}^{N} \sum_{k=1}^{K} \left(e^{(K-k) \mathbb{E}_{b^k} \left[\log p(b^k|C_{i,m}^{k-1}) \right]} I\left(b^k; C_{i,m}^{k-1} \right) \right) \\
&+ H(C_m^K) + \xi(N,K) \\
=&\sum_{i=1}^{N} \sum_{k=1}^{K} \left(e^{(K-k)\log(N-k) + (K-k) \mathbb{E}_{b^k} \left[\log \frac{p(b^k|C_{i,m}^{k-1})}{p(b^k)} \right]} I\left(b^k; C_{i,m}^{k-1} \right) \right) \\
&+ H(C_m^K) + \xi(N,K) \\
=&\sum_{i=1}^{N} \sum_{k=1}^{K} \left(I\left(b^k; C_{i,m}^{k-1} \right) e^{(K-k) \mathbb{E}_{b^k} \left[\log \frac{p(b^k|C_{i,m}^{k-1})}{p(b^k)} \right]} (N-k)^{(K-k)} \right) \\
&+ H(C_m^K) + \xi(N,K)
\end{aligned} \tag{22}
$$

B RELATIONSHIP BETWEEN SPGCL AND CPC

The original problem described in section 3 is to find K neighbors for N nodes, and we simplify it to find one neighbor for one nodes, i.e., $K = 1$ and $\mathbf{W}' = \{\mathbf{w}\}$. Besides, we denote $b = b^k$ for simplicity and Eq. (22) can be rewritten as follows:

$$
\begin{aligned}
&I(\mathbf{W}'; C^K) \\
&= \sum_{i=1}^{|\mathbf{W}'|} \sum_{k=1}^{1} \left(I\left(b; \mathbf{w}_i\right) e^{(K-k)\, \mathbb{E}_b \left[\log \frac{p(b|\mathbf{w}_i)}{p(b)}\right]} (N-k)^{(K-k)} \right) \\
&\quad + H(C_m^1) + \log p(b) \\
&= \sum_{i=1}^{|\mathbf{W}'|} I\left(b; \mathbf{w}_i\right) - \sum_{j=1}^{|\mathbf{W}|} p(b_j) \log p(b_j) + \log p(b) \\
&= I\left(b; \mathbf{w}\right) + \log N - \log N \\
&= I\left(b; \mathbf{w}\right),
\end{aligned} \tag{23}
$$

where $p(\mathbf{w}) = 1$ and $p(b) = \frac{1}{N}$. The first equal holds when $K = 1$ since the route from $\mathbf{w}$ to b is determined. Therefore we can conclude that, SPGCL degrades to CPC and InfoNCE loss [39] under some assumptions and constraints.

C PROOF OF LEMMA 3.1

$$
\begin{aligned}
&I(C, b; \mathbf{w}) - I(C; \mathbf{w}) \\
&= \sum_{C,b,\mathbf{w}} p(C, b, \mathbf{w}) \log \frac{p(C, b, \mathbf{w})}{p(C, b) p(\mathbf{w})} - \sum_{C,\mathbf{w}} p(C, \mathbf{w}) \log \frac{p(C, \mathbf{w})}{p(C) p(\mathbf{w})} \\
&= \sum_{C,b,\mathbf{w}} p(C, b, \mathbf{w}) \log \frac{p(C, b, \mathbf{w})}{p(C, b) p(\mathbf{w})} - \sum_{C,b,\mathbf{w}} p(C, b, \mathbf{w}) \log \frac{p(C, \mathbf{w})}{p(C) p(\mathbf{w})} \\
&= \sum_{C,b,\mathbf{w}} p(C, b, \mathbf{w}) \log \frac{p(C, b, \mathbf{w})}{p(C, b)} \cdot \frac{p(C)}{p(C, \mathbf{w})} \\
&= \sum_{C,b,\mathbf{w}} p(C, b, \mathbf{w}) \log \frac{p(C, b, \mathbf{w})}{p(b \mid C) p(C, \mathbf{w})} \\
&= \sum_{C,b,\mathbf{w}} p(b, \mathbf{w} \mid C) p(C) \log \frac{p(b, \mathbf{w} \mid C)}{p(b \mid C) p(\mathbf{w} \mid C)} \\
&= I(b; \mathbf{w} \mid C) \geq 0,
\end{aligned} \tag{24}
$$

where $\sum_{C,b,\mathbf{w}} = \sum_C \sum_b \sum_{\mathbf{w}}$.

D TRAINING AND ADAPTIVE LABELING ALGORITHMS

We train SPGCL according to Algorithm 1, and at the end of each epoch, edge labels are update via Algorithm 2, including label edges with high confidence and unlabel edges with low confidence. In practice, some edges are hard to classify, resulting in $E^U \neq \emptyset$. Next, we analyse the time complexity of SPGCL as follows:

- In Algorithm 1, learning embedding via Eq. (13) and (14) costs $O(NF'^2 + NF')$, where N is number of nodes and F' is the length of embeddings. The aggregation requires $O(NH(TFF' + TF + TFT'))$ time, where H is number of multi-heads, T and T' are the length of input and output sequence respectively. Furthermore, the pair-wise similarity requires $O(N^2F')$. Both $\mathcal{L}_N$ and $\mathcal{L}_{PU}$ need $O(N^2)$ computation, and $\mathcal{L}_{mse}$ costs $O(NTF^2)$. At the end of each epoch, Algorithm 2 costs $O(N^2)$, which is introduced below. Since $N \gg F' \geq F$ in practice, the overall complexity of Algorithm 1 is $O(NHTFF' + N^2F')$.
- The complexities of KNN and DTW are $O(NF)$ and $O(N^2)$ respectively when calculating $\mathbf{W}$ in Eq. (12). Once the data is prepared, it can be reused during training and testing.
- The cost of Algorithm 2 is expected to be $O(N^2)$, which mainly comes from Hadamard product and calling *topn* on the matrix.

Algorithm 1 The training of SPGCL.

Require: Locations $\mathbf{V} \in \mathbb{R}^{N \times d}$, deformations $\mathbf{X} \in \mathbb{R}^{N \times (T+T') \times F}$, initial neighbor volume k.

function $sim(\mathbf{W}, E^P)$: ▷ Matching score function
  Obtain $\mathbf{R}$ and $\mathbf{R}_c$ via Eq. (13) and (14) respectively;
  Calculate matching score $\mathbf{S}$ via Eq. (15);
  return $\mathbf{S}$
end function
Calculate $\mathbf{W}$ via Eq. (12);
Initialize E^P and E^N via KNN with k, the mistakenly-labeled counter mask matrix E^M with zeros;
Minimizing $\mathcal{L}_N$ and $\mathcal{L}_{PU}$ via Eq. (8) and (11) to warm sim up;
while SPGCL is not converged **do**
  Apply augmentations to $\mathbf{W}$ and E;
  Obtain matching score $\mathbf{S} = sim(\mathbf{W}, E^P)$;
  Calculate $\mathcal{L}_N$, $\mathcal{L}_{PU}$ and $\mathcal{L}_{mse}$ via Eq. (8) (11) and (16) respectively;
  Minimize hybrid loss $\mathcal{L}$ via Adam optimizer [23];
  Update labels via Algorithm 2;
end while

Algorithm 2 The pseudo-code of updating labels.

Require: Positive edges mask E^P and the initialized $E^P_{k=0}$, unlabeled edges mask E^U and the initialized $E^N_{k=0}$, negative edges mask E^N, mistakenly-labeled counter matrix E^M, score matrix $\mathbf{S}$, numbers of candidates α and β, threshold $\delta^{\pm}$, patience λ. Note that masks are all 0-1 matrixes, and $\Delta = 5e^{-4}$ is predefined parameter.

function $topn(\mathbf{S}, n, \text{largest})$:
  if largest == True **then**
    Set scores to 0, except the largest n for each node;
  else
    Set scores to 0, except the smallest n for each node;
  end if
  return $\mathbf{S}$
end function
$\mathbf{S}^P = topn(\mathbf{S} \odot E^U, \alpha, \text{True})$ ▷ Positive candidates
$E^P = E^P + [\mathbf{S}^P \geq \delta^+]$;
$\mathbf{S}^N = topn(\mathbf{S} \odot E^U, \beta, \text{False})$ ▷ Negative candidates
$E^N = E^N + [\mathbf{S}^N \leq \delta^-]$;
$E^M_e = [\mathbf{S} \odot E^P < \delta^+] + [\mathbf{S} \odot E^N > \delta^-]$; ▷ Count if the edge is labeled mistakenly
$E^M = E^M \odot E^M_e + E^M_e$;
$E^P = [E^P \odot E^M < \lambda] \bigcup E^P_{k=0}$; ▷ Unlabel edge if the count exceeds the patience, but always remember the initial labels
$E^N = [E^N \odot E^M < \lambda] \bigcup E^N_{k=0}$;
$E^U = \mathbf{1} - E^P - E^N$; ▷ Update the E^U
$E^M = E^M \odot E^U$; ▷ Reset the counter
$\delta^+ = \delta^+ - \Delta$ and $\delta^- = \delta^- + \Delta$; ▷ Update threshold