STPC-Net: Learn Massive Geo-Sensory Data as Spatio-Temporal Point Clouds

Chuanpan Zheng, Cheng Wang, *Senior Member, IEEE*, Xiaoliang Fan, *Senior Member, IEEE*, Jianzhong Qi, *Member, IEEE*, and Xu Yan

***Abstract*— Nowadays, a large number of sensors are equipped on mobile or stationary platforms, which continuously generate geo-tagged and time-stamped readings (i.e., geo-sensory data) that contain rich information about the surrounding environment. These data have irregular space and time coordinates. To represent geo-sensory data, there have been extensive research efforts using time sequences, grid-like images, and graph signals. However, there still lacks a proper representation that can describe both the mobile and stationary geo-sensory data without the information-losing discretization in spatial and temporal dimensions. In this paper, we propose to represent massive geo-sensory data as *spatio-temporal point clouds* (STPC), and present *STPC-Net*, a novel deep neural network for processing STPC. STPC leverages the original irregular space-time coordinates, and STPC-Net captures intra-sensor and inter-sensor correlations from STPC. In this way, STPC-Net learns the key information of STPC, and overcomes challenges in data irregularity. Experiments using real-world datasets show that STPC-Net achieves state-of-the-art performance in different tasks on both mobile and stationary geo-sensory data. The source code is available at https://github.com/zhengchuanpan/STPC-Net.**

***Index Terms*— Geo-sensory data, spatio-temporal point clouds, STPC-Net, deep learning.**

I. Introduction

IN MODERN intelligent transportation systems (ITS), a large number of sensors are deployed to monitor their surrounding environment [1], [2]. These sensors are associated with geo-spatial locations (e.g., latitude and longitude), and they continuously generate time-stamped readings [3]. Modeling such geo-sensory data offers a critical opportunity to measure, infer, and understand our living environment [4].

According to the attached platform, the geo-sensory data can be divided into mobile and stationary categories. For example, as shown in Figure 1(a), the GPS devices equipped on vehicles collect mobile geo-sensory data, where the locations are changing over time. Figure 1(b), on the other hand, shows stationary geo-sensory data collected from fixed stations (e.g., loop detectors), which are multiple time-series readings with unchanged locations.

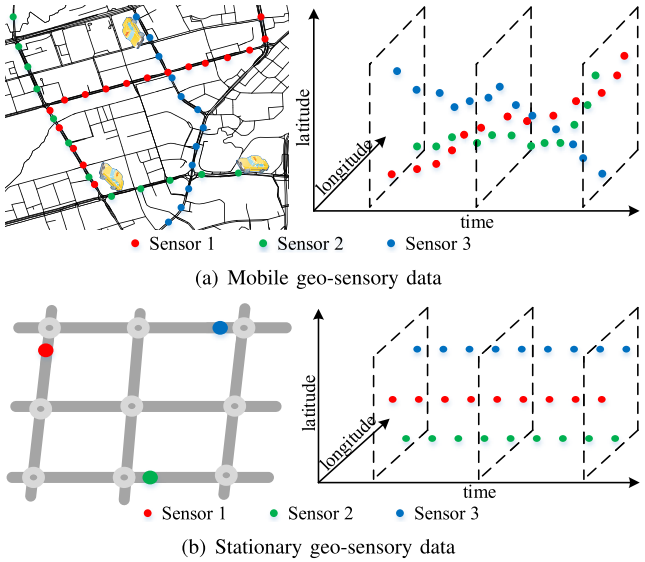


Fig. 1. Illustration of geo-sensory data. Each reading is associated with the space (e.g., latitude, longitude) and time coordinates, which may not locate in regular grids. (a) The sensors are equipped on mobile vehicles; (b) the sensors are deployed at stationary locations.

Due to the variety of applications, geo-sensory readings may distribute at arbitrary space-time coordinates, and not locate in rasterized uniform spatio-temporal grids (Figure 1). This irregular distribution makes it difficult to apply representation learning techniques such as deep learning [5] to model the geo-sensory data. To address this issue, earlier studies consider each individual sensor's readings as a time sequence and apply time series methods for sequence learning, e.g., long short-term memory (LSTM) [6], [7]. This time series representation ignores the interaction information between different sensors. To take advantages of the existing convolutional neural networks (CNN) [8], some studies convert the data into regular grid-like images [9], [10]. This transformation introduces quantization errors that obscure the detailed spatio-temporal information in the data. Furthermore, it is nontrivial to apply these methods to fine-grained tasks such as trajectory point classification [11] and sensor readings prediction [12]

Manuscript received January 31, 2021; revised June 23, 2021; accepted August 1, 2021. This work was supported in part by the Natural Science Foundation of China under Grant 61872306, in part by the Xiamen Science and Technology Bureau under Grant 3502Z20193017, and in part by the Fundamental Research Funds for the Central Universities under Grant 20720200031. The Associate Editor for this article was D. F. Wolf. *(Corresponding author: Cheng Wang.)*

Chuanpan Zheng, Cheng Wang, Xiaoliang Fan, and Xu Yan are with Fujian Key Laboratory of Sensing and Computing for Smart Cities, Digital Fujian Institute of Urban Traffic Big Data Research, Xiamen University, Xiamen 361005, China, and also with the School of Informatics, Xiamen University, Xiamen 361005, China (e-mail: zhengchuanpan@stu.xmu.edu.cn; cwang@xmu.edu.cn; fanxiaoliang@xmu.edu.cn; yanxu97@stu.xmu.edu.cn).

Jianzhong Qi is with the School of Computing and Information Systems, The University of Melbourne, Melbourne, VIC 3010, Australia (e-mail: jianzhong.qi@unimelb.edu.au).

Digital Object Identifier 10.1109/TITS.2021.3102747

as different sensors may locate in the same grid. Recent studies [13], [14] treat geo-sensory data as graph signals on a fixed sensor graph and apply graph neural networks (GNN) [15] for learning. However, such approaches can only be applied on stationary geo-sensory data (Figure 1(b)) as the sensors need to be fixed for constructing a sensor graph. In addition, how to pre-define an effective sensor graph remains a challenge as the weights between sensors are probably unseen [16]. In general, there lacks a representation that can describe both mobile and stationary geo-sensory data without the information-losing discretization in spatial and temporal dimensions.

We observe the following essential characteristics of geo-sensory data. 1) *Point data*. Sensors have small sizes such that their readings are often modeled as points. 2) *Irregularity*. The sensor readings may distribute at arbitrary space-time coordinates, and not locate in rasterized uniform spatio-temporal grids. 3) *Interdependency*. The sensor readings are spatially and temporally correlated to each other. 4) *Sparseness*. The distribution of the sensors is usually sparse due to limitations such as cost. According to these observations, a geo-sensory reading can be viewed as an informative point in space-time coordinates, and massive geo-sensory data can be represented as a *spatio-temporal point cloud* (STPC) with complex correlations in it. Under this definition, both mobile and stationary geo-sensory data (Figure 1) can be represented as STPC, which leverages the irregular space-time coordinates.

The motivation of this work is to learn massive geo-sensory data as a unified STPC representation. Recently, numerous deep learning methods have been successfully applied on 3D point clouds to solve various 3D vision problems [17], [18]. These methods are effective in extracting 3D geometric information from a single 3D point could [19], [20]. Some studies attempt to learn spatio-temporal features from 3D point cloud sequences [21], [22]. However, they are not suitable for processing the STPC due to the following reasons. First, the STPC is a set of points with both space and time coordinates, but it cannot be represented as a point cloud sequence because every point could be with totally different timestamps. Second, the STPC is generated by multiple sensors, and it inherently has complex intra-sensor and inter-sensor correlations, which are not existed in 3D point clouds.

We thus propose *STPC-Net*, a new deep neural network for processing the STPC. The key components of STPC-Net include convolution and combination modules. The convolution module contains a conv-intra operation to extract sequential features at each sensor (intra-sensor correlations), a conv-inter operation to learn interactional features between different sensors (inter-sensor correlations), and a gated fusion mechanism to adaptively fuse these two features. In the combination module, we aggregate all point features into a global representation that has a global view of the entire point cloud, and then concatenate local and global features to obtain combined point features. At last, we apply a multi-layer perceptron network upon the combined features to produce the final output. We conduct experiments on both mobile and stationary geo-sensory data and demonstrate our STPC-Net achieves state-of-the-art performance.

The main contributions of this work are summarized as follows:

- To the best of our knowledge, we are the first to propose a unified representation (i.e., spatio-temporal point clouds) that can describe both mobile and stationary geo-sensory data without the information-losing discretization in spatial and temporal dimensions.
- We design a novel deep network architecture (i.e., STPC-Net) suitable for processing spatio-temporal point clouds. STPC-Net is able to learn both local intra-sensor, inter-sensor correlations and global information in massive geo-sensory data.
- Extensive experiments are carried out on both mobile and stationary geo-sensory data, which demonstrate STPC-Net achieves state-of-the-art performance.

The rest of this paper is organized as follows. Section II reviews the deep learning studies on geo-sensory data and 3D point clouds. Section III presents some preliminaries of this study. Section IV details the method of STPC-Net. Section V compares STPC-Net with state-of-the-art methods on both mobile and stationary geo-sensory data. Section VI provides the discussions of STPC versus 3D point clouds and STPC-Net versus PointNet [19]. Finally, Section VII concludes this paper and draws future work.

II. Related Work

A. Deep Learning on Geo-Sensory Data

Recent years have witnessed the rapid growth of geo-sensory applications, along with a large number of sensors being equipped on different platforms to monitor their surrounding environment [23]. The distribution of these geo-sensory data is not in a regular format. Researchers have conducted various data representations to process the data. Some of these studies consider each sensor's readings as a time series for sequence modeling using recurrent neural networks (RNN) [24]–[26], etc. Such representations ignore the interaction information between different sensors.

To take advantages of existing CNN architectures, other studies convert the geo-sensory data into 2D images [9], [10], [27]. They partition the investigated area into regular grids, and map the sensor readings to the corresponding grids according to the latitude and longitude. Then, a well-engineered CNN (e.g., ResNet [28]) is applied to the images for features learning. This transformation may introduce quantization error and lose detailed spatio-temporal information in the data. In addition, these approaches cannot be applied to some fine-grained tasks such as sensor readings prediction, as a grid may contain multiple sensors.

Recent studies learn geo-sensory data as graph signals on a fixed sensor graph [13], [29], [30]. Each static sensor is considered as a node and multiple sensors form a sensor graph. Then, graph neural networks [15] are applied for spatio-temporal graph modeling. This transformation requires a well-defined sensor graph. Thus, it can only be applied to process stationary geo-sensory data, as the sensors need to be fixed for constructing the sensor graph.

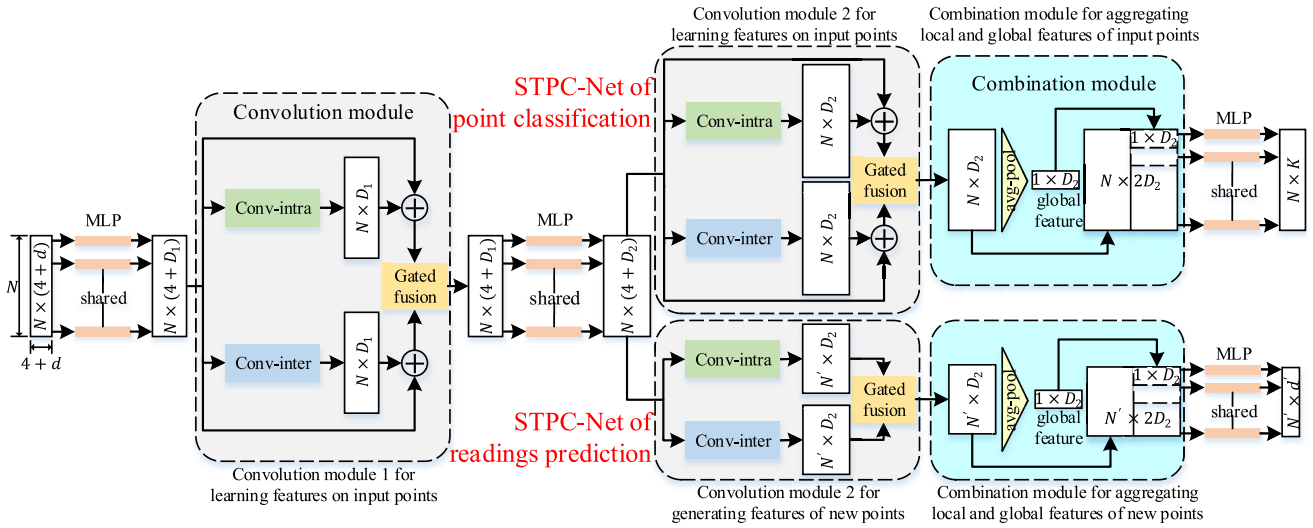


Fig. 2. STPC-Net architecture. STPC-Net of point classification contains several multi-layer perceptron (MLP) networks for features projection, two convolution modules for local features learning on the input points, and a combination module for local and global information aggregation. STPC-Net of readings prediction has a similar architecture, while the second convolution module is used to generate features for new points. At the last stage, both nets use an MLP network to produce the final output.

In general, there still lacks a unified representation that can describe both mobile and stationary geo-sensory data without the information-losing discretization in spatial and temporal dimensions. In this paper, we firstly represent these data as unified spatio-temporal point clouds (STPC).

B. Deep Learning on 3D Point Clouds

A 3D point cloud is a set of points defined in 3D metric space, and it is one of the most significant data format for 3D representation [18]. Recently, deep learning on 3D point clouds is thriving [17]. Among this, PointNet [19] is a pioneering and successful method of dealing with 3D point clouds. It learns features independently for each point, which means it does not model local dependencies among points. The subsequent studies [20], [31], [32] further capture local structural information from the neighborhoods of each point. These methods achieve state-of-the-art performance in many 3D vision tasks such as classification and segmentation. These methods mainly focus on learning features in geometric space.

A handful of studies have started to learn spatio-temporal information from 3D point cloud sequences. Researchers in [33] represent the 3D point cloud sequence as a set of time-varying plane curves, and apply CNNs for classification. In [34], the 3D point cloud sequences are voxelized into 4D occupancy grids, and then the sparse 4D convolution is used. Recently, several works attempt to extract dynamic features directly from point cloud sequences. FlowNet3D [21] learns features from two consecutive frames. MeteorNet [22] is able to handle multiple frames of point cloud sequences. Some studies [35], [36] apply recurrent models to aggregate features from different frames.

However, the models designed for 3D point cloud sequences are not suitable for processing the STPC. First, the STPC cannot be represented as a point cloud sequence, as every point could be with totally different timestamps. Second, the STPC contains complex intra-sensor and inter-sensor correlations, which are not existed in 3D point clouds. Thus, in this paper, we design a new deep neural network (i.e., STPC-Net) suitable for processing the STPC.

III. Preliminary

A. Notations

A spatio-temporal point cloud is a set of N points $\mathcal{X} = \{P_i \in \mathbb{R}^{4+d} | i = 1, 2, \ldots, N\}$, where each point $P_i = (s_i, C_i, X_i)$. Herein, s_i denotes the corresponding sensor id; $C_i = (lat_i, lng_i, t_i)$ is the space-time coordinate of point P_i, i.e., latitude, longitude, and timestamp; $X_i \in \mathbb{R}^d$ represents a d-dimensional feature vector of point P_i, consisting of the sensor readings and other associated features if available, such as the time feature.

B. Problem Statement

We study two problems on spatio-temporal point clouds.

1) Point Classification: Given N input points $\mathcal{X} \in \mathbb{R}^{N\times(4+d)}$, this problem aims to output $N \times K$ scores that represent the probability of each point belonging to each of K pre-defined point classes, represented as $\hat{Y} \in \mathbb{R}^{N\times K}$.

2) Readings Prediction: Given M sensors and their readings in Q consecutive timestamps (i.e., given $N = MQ$ input points $\mathcal{X} \in \mathbb{R}^{N\times(4+d)}$), this problem aims to predict a series of readings of the M sensors in the next Q' consecutive timestamps, i.e., to generate $N' = MQ'$ new points $\hat{Y} \in \mathbb{R}^{N'\times d'}$, where the sensor ids and space-time coordinates are pre-known, d' represents the number of kinds of readings to be predicted.

IV. STPC-Net

A. Network Architecture

Figure 2 depicts the network architectures of STPC-Net for point classification and readings prediction. The input of STPC-Net is a spatio-temporal point cloud. First, a point-wise multi-layer perceptron (MLP) network projects the input

features into D_1 dimensions. The sensor ids and space-time coordinates are concatenated back to the projected features for further processing. Then, a convolution module is proposed to model both intra-sensor and inter-sensor correlations on the input points, which will be detailed in Section IV-B. The residual connections [28] are applied. The sensor ids and space-time coordinates of the input points are also carried forward into the output features. Next, we use another MLP network to lift the features into higher dimensional ($D_2 > D_1$) and more abstract representations. In the second convolution module, there are some differences between point classification and readings prediction tasks.

In STPC-Net of point classification, the second convolution module is same as the first one to abstract higher level local features on the input points. After that, the combination module aggregates the local and global features, which will be detailed in Section IV-C. Finally, an MLP network outputs K scores for each of the N input points.

In STPC-Net of readings prediction, the second convolution module is used to generate features for new points. The residual connections are not applied, as the input and output represent different points. In the combination module, the local and global features of the new points are combined. At last, we apply an MLP network upon the combined features to produce the d'-dimensional prediction result for each of the N' new points.

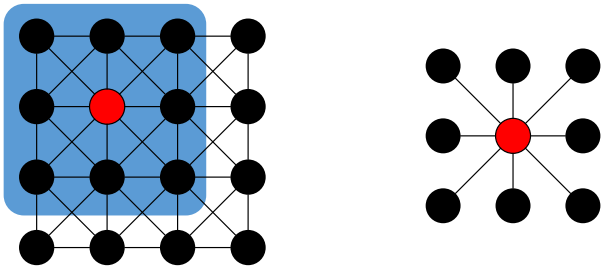
B. Convolution Module

A spatio-temporal point cloud (STPC) contains points generated by multiple sensors in a period of time, which may have complex intra-sensor and inter-sensor correlations. We design a convolution module to capture such correlations, which consists of a *conv-intra* operation abstracts the sequence features at each sensor, a *conv-inter* operation learns the interaction information between different sensors, and a *gated fusion* mechanism adaptively fuses them.

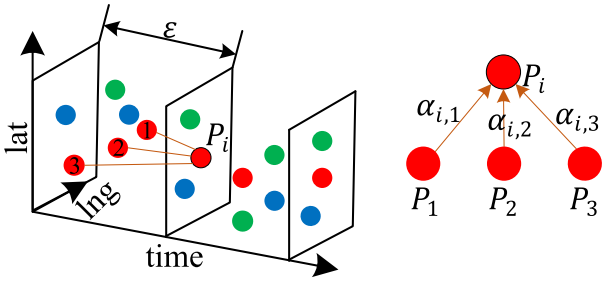
We denote the input of the convolution module as $\mathcal{X}$. As described in Section IV-A, STPC-Net includes two types of convolution module, one for leaning features on the input points and the other to generate features for new points. In both types, the sensor ids and space-time coordinates of the target output points are pre-known, represented as $\{P_i = (s_i, C_i)\}$, where $C_i = (lat_i, lng_i, t_i)$. The computations of these two types are similar. The only difference is that whether the output represents the same points as the input or not.

In the following discussion, we first briefly review the standard 2D convolution operation in section IV-B.1, and then detail the conv-intra and conv-inter operations in sections IV-B.2 and IV-B.3, respectively. At last, section IV-B.4 introduces the gated fusion mechanism.

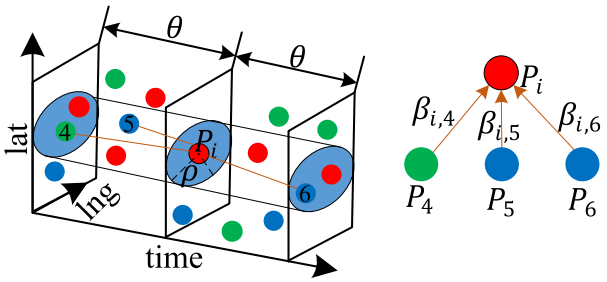
1) Standard 2D Convolution Operation: As shown in Figure 3(a), each pixel has the same number of neighborhoods, which are spatial ordered (e.g., from left to right and top to bottom). Within each local region, the relative positions between different pixels are always fixed. Thus, the features of neighborhoods could be projected by a fixed size of convolution kernel (e.g., 3×3), and then aggregated into the centroid pixel via average summation.



(a) Standard 2D convolution operation. For each centroid pixel (e.g., the red one), the number of neighborhoods is same. In each local region (e.g., 3×3), the neighborhoods are ordered and the relative positions are fixed. The feature of the centroid pixel is updated as the average sum of neighborhoods' features.



(b) Conv-intra operation. For each centroid point (e.g., P_i), the neighborhoods are defined as the points generated by the same sensor at earlier adjacent timestamps. In different local region, the number of neighborhoods is different, the order and the relative positions can be diverse. The feature of the centroid point is calculated as a weighted sum of neighborhoods' features. The points in different colors are generated by different sensors.



(c) Conv-inter operation. The neighborhoods of each centroid point (e.g., P_i) are searched as points generated by nearby sensors at recent timestamps. Different point has different number of neighborhoods, which are unordered and with different relative positions within the local region. The features of neighboring points are weighted aggregated into the centroid point. The points in different colors represent readings generated by different sensors.

Fig. 3. The comparison of stand 2D convolution with conv-intra and conv-inter operations. In the illustration of each operation, the figure on the left side illustrate the neighborhoods searching approach, and the right figure show the features aggregation method.

2) Conv-Intra Operation: A sensor generates successive readings in a period of time, where each reading may be affected by its previous readings. We propose a conv-intra operation to model such intra-sensor correlations, as illustrated in Figure 3(b) and Algorithm 1.

Similar to the standard 2D convolution operation, the conv-intra operation abstracts features in the local region. For each target output point P_i, we examine its neighboring points are generated by the same sensor at earlier adjacent timestamps:

$$\mathcal{N}_{intra}(P_i) = \{P_j | s_i = s_j, 0 \le t_i - t_j \le \epsilon, P_j \in \mathcal{X}\}, \quad (1)$$

Algorithm 1 Conv-Intra Operation

Input: $\mathcal{X}$, $P_i = (s_i, C_i)$, ϵ
Output: X_i

1: Search the intra-sensor neighboring points of point P_i in $\mathcal{X}$ as $\mathcal{N}_{intra}(P_i)$ using Equation 1
2: **for** each $P_j \in \mathcal{N}_{intra}(P_i)$ **do**
3: $C_j \leftarrow MLP_{intra,1}(C_j - C_i)$
4: $X_j \leftarrow MLP_{intra,2}([C_j, X_j])$
5: $\alpha_{i,j} \leftarrow MLP_{intra,3}(t_i - t_j)$
6: **end for**
7: $X_i \leftarrow \sum_{P_j \in \mathcal{N}_{intra}(P_i)} \alpha_{i,j} \cdot X_j$
8: **return** X_i

Algorithm 2 Conv-Inter Operation

Input: $\mathcal{X}$, $P_i = (s_i, C_i)$, θ, ρ, $dis(\cdot, \cdot)$
Output: X_i

1: Search the inter-sensor neighboring points of point P_i in $\mathcal{X}$ as $\mathcal{N}_{inter}(P_i)$ using Equation 4
2: **for** each $P_k \in \mathcal{N}_{inter}(P_i)$ **do**
3: $C_k \leftarrow MLP_{inter,1}(C_k - C_i)$
4: $X_k \leftarrow MLP_{inter,2}([C_k, X_k])$
5: $\beta_{i,k} \leftarrow MLP_{inter,3}(dis(P_i, P_k))$
6: **end for**
7: $X_i \leftarrow \sum_{P_k \in \mathcal{N}_{inter}(P_i)} \beta_{i,k} \cdot X_k$
8: **return** X_i

where ϵ is the threshold to control the size of the local region. As the points in a STPC are not located in regular grids but can take arbitrary values, the neighboring points in $\mathcal{N}_{intra}(P_i)$ are unordered, with diverse relative positions and variable sizes at different centroid point P_i. Thus, the standard discretized convolution kernels on raster images cannot be applied on the STPC to project neighborhoods' features.

We introduce the features projection approach as follows. First, in a local region $\mathcal{N}_{intra}(P_i)$, the relative position of a neighboring point P_j to the centroid point P_i is a discriminative feature. Thus, we position local coordinate system at the centroid point P_i. For each neighboring point $P_j \in \mathcal{N}_{intra}(P_i)$, we take its local coordinate $(C_j - C_i)$, and project it into a higher dimensional representation using an MLP network (line 3 in Algorithm 1). Then, we concatenate this local coordinate representation with the associated feature, and apply another MLP network to output the projected feature of point P_j (line 4 in Algorithm 1).

In the features aggregation stage, all neighboring points' features are aggregated into the target output point P_i. As the time differences vary across neighboring points, they contribute differently to the target point. Thus, different to the standard 2D convolution using average aggregation, we design a weighted approach, where the weight between two points is defined as a function of the time difference:

$$\alpha_{i,j} = f(t_i - t_j), \tag{2}$$

where f could be learned using an MLP network (line 5 in Algorithm 1). The weights are then normalized into $[0, 1]$ at each output point P_i. Based on the weights, the output feature of point P_i is computed as a weighted sum of its neighboring points' features (line 7 in Algorithm 1):

$$X_i = \sum\nolimits_{P_j \in \mathcal{N}_{intra}(P_i)} \alpha_{i,j} \cdot X_j, \tag{3}$$

where X_j is the projected feature of the neighboring point P_j (lines 3-4 in Algorithm 1). The learnable parameters of the MLP networks in the conv-intra operation are shared across all the points to solve the unordered problem.

3) Conv-Inter Operation: A sensory reading may also be affected by nearby sensors' readings in recent time period. As shown in Figure 3(c), we propose a conv-inter operation to model such inter-sensor correlations.

As detailed in Algorithm 2, the computation of the conv-inter operation is similar to that of conv-intra. Now, the neighboring points of point P_i are defined as those from nearby sensors at recent timestamps:

$$\mathcal{N}_{inter}(P_i) = \{P_k | s_i \neq s_k, |t_i - t_k| \leq \theta, \\ dis(P_i, P_k) \leq \rho, P_k \in \mathcal{X}\}, \tag{4}$$

where θ and ρ are thresholds of time difference and distance, $dis(P_i, P_k)$ denotes the geographical distance (e.g., Euclidean) between points P_i and P_k. Then, the features of neighboring points are projected in the same manner as in the conv-intra operation (lines 3-4 in Algorithm 2). In the weighted aggregation approach, the weight is conditioned on the distance between two points:

$$\beta_{i,k} = h(dis(P_i, P_k)), \tag{5}$$

where h is learned using an MLP network (line 5 in Algorithm 2). Finally, the output feature of point P_i is aggregated as (line 7 in Algorithm 2):

$$X_i = \sum\nolimits_{P_k \in \mathcal{N}_{inter}(P_i)} \beta_{i,k} \cdot X_k, \tag{6}$$

where X_k is the projected feature of the neighboring point P_k (lines 3-4 in Algorithm 2). The parameters of the MLP networks in the conv-inter operation are shared across all the points.

4) Gated Fusion: We denote the outputs of the conv-intra and conv-inter operations as X_{intra} and X_{inter}, which contain intra-sensor and inter-sensor information respectively. To fuse them in a data-dependent way, we apply a gated fusion mechanism [30]:

$$X = z \odot X_{intra} + (1 - z) \odot X_{inter}, \tag{7}$$

where $\odot$ denotes the element-wise product operation. The gate z is computed as:

$$z = \sigma(MLP_{g,1}(X_{intra}) + MLP_{g,2}(X_{inter})), \tag{8}$$

where $MLP_{g,1}$ and $MLP_{g,2}$ represent two different MLP networks, $\sigma(\cdot)$ is the sigmoid activation to normalize the output into $[0, 1]$. This gated fusion mechanism could adaptively control the importance of two features at point-wise and channel-wise level, according to the input data.

C. Combination Module

As the convolution modules only extract features in the local region, it is essential to inject the global information into the learning process. To this end, we add a combination module upon the output of the second convolution module, before it is fed into the last MLP network to produce the final output, as shown in Figure 2.

Inspired by PointNet [19], we apply an average pooling layer to aggregate all point features into a discriminative global signature, which has a global view of the entire point cloud. Such a global feature enables the network to take into account readings over a long time period and at distant sensors, which is beneficial to model the long-term patterns and the long-range dependencies. We then concatenate the global feature with each local point feature to obtain combined point features, which encode both local and global information. At the last stage, we use an MLP network upon the combined features to produce the final output.

V. Experiments

We evaluate our method using both mobile and stationary geo-sensory data on two different types of tasks, i.e., point classification and readings prediction.

A. Point Classification

STPC-Net of point classification is evaluated on the GeoLife dataset [11] for transportation mode identification. This dataset contains GPS trajectories of 69 users with labeled transportation modes (e.g., walk, bike). Each GPS point is associated with the user (sensor) id, latitude, longitude, timestamp, etc. The problem is to identify the transportation mode of each GPS point given the sensor ids and space-time coordinates, which is a point classification problem.

As the sensors are carried by users (not stationary), these data are actually mobile geo-sensory data (Figure 1(a)). Previous studies on such data for transportation mode identification generally follow a two-step approach. They first detect the point where the transportation mode changes, and divide the trajectory into segments accordingly, expecting that each segment contains only one transportation mode. Then, a classification model is applied to classify each segment. In this setting, the geo-sensory data are considered as a set of isolated time sequences, and the correlations between different users are ignored.

We learn these mobile geo-sensory data as spatio-temporal point clouds. Our STPC-Net could capture both intra-sensor and inter-sensor correlations, and predict the transportation mode of each point end-to-end.

1) Data Preprocessing: We adopt the same data preprocessing procedures as in [37] and obtain 4.7 million valid points in total. These points are divided into 9,362 samples, where each sample contains 512 points (i.e., $N = 512$). We randomly select 70% of the samples for training, 10% for validation, and the rest 20% for testing.

Following [37], each point is associated with a 6-dimensional feature vector ($d = 6$), which are the motion features including the distance and time interval between two consecutive points, the speed, acceleration, jerk, and heading change rate of each point. Five transportation modes (i.e., $K = 5$) that constitute the majority of the dataset are considered, i.e., walk, bike, bus, drive, and train. More detailed information about the data preprocessing procedure could refer to [37].

TABLE I

Performance Comparison of Different Methods on the GeoLife Dataset for Transportation Mode Identification

Method	Accuracy	Precision	Recall	F1-score
Zheng *et al.* [11]	76.2%	76.9%	76.4%	74.8%
Endo *et al.* [39]	67.9%	-	-	-
Wang *et al.* [41]	74.1%	-	-	-
TrajectoryNet [42]	78.0%	78.4%	77.2%	77.5%
SECA [37]	76.8%	79.8%	75.8%	77.0%
STPC-Net	**82.9%**	**83.8%**	**82.2%**	**82.9%**

2) Experimental Settings: As described above, the number of input points is $N = 512$, each with $4 + d$ ($d = 6$) dimensions, and STPC-Net outputs $K = 5$ scores for each point. In the conv-intra operation, the threshold of time difference is set as $\epsilon = 20$ minutes. In the conv-inter operation, the distance between two points is the Euclidean distance ($dis(\cdot, \cdot)$ in Algorithm 2). The thresholds of time difference and distance are set as $\theta = 20$ minutes and $\rho = 1$ kilometers, respectively. We will investigate the influence of these hyperparameters in section V-A.6. The output dimensions of two convolution modules are $D_1 = 64$ and $D_2 = 128$, respectively.

The objective function is the cross entropy loss, which is optimized using the Adam optimizer [38] with an initial learning rate of 0.001. The evaluation metrics include accuracy, average precision, average recall, and average F1-score.

3) Baseline Methods: We compare STPC-Net with the following baseline methods.

- **Zheng *et al.* [11]** that abstracts a set of hand-crafted features and applies Decision Tree for classification.
- **Endo *et al.* [39]** that uses Stacked Denoising Auto-encoders (SDA) [40] to extract deep features from GPS trajectories and applies logistic regression LR) for classification.
- **Wang *et al.* [41]** that uses a sparse auto-encoder to extract deep features from hand-crafted features and applies deep fully-connected neural networks for classification.
- **TrajectoryNet** [42] is a bi-directional gated recurrent unit (GRU) network with maxout activations [43], and uses point-and-segment-based features to detect transportation modes.
- **SECA** [37] is a deep SEmi-supervised Convolutional Autoencoder architecture that integrates a convolutional-deconvolutional autoencoder and a convolutional neural network.

4) Experimental Results: Table I shows the comparison of our method with recent studies for transportation mode identification. STPC-Net achieves the best performance in terms of all metrics. Our advantages are two-fold. First, STPC-Net enables end-to-end learning while previous studies

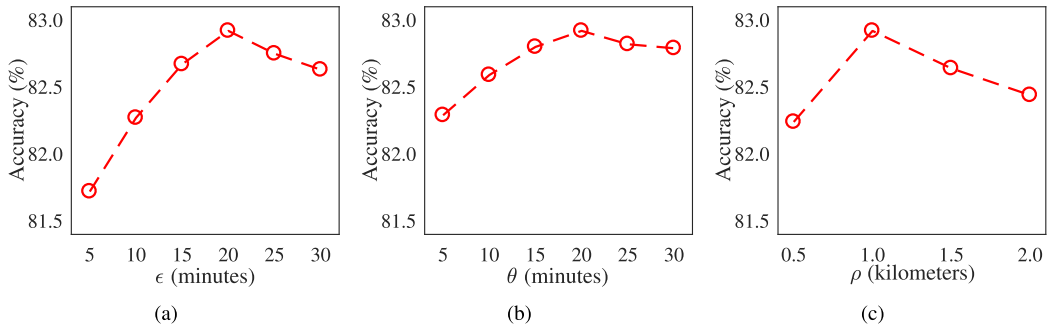


Fig. 4. The impact of hyperparameters in STPC-Net. (a) Accuracy with respect to the threshold of time difference in conv-intra operation. (b) Accuracy with respect to the threshold of time difference in conv-inter operation. (c) Accuracy with respect to the threshold of distance in conv-inter operation.

need two steps, i.e., trajectory segmentation and segment classification. Second, previous studies learn the data as isolated time sequences that ignore the interactions among users, while STPC-Net well respects the data and captures both intra-sensor and inter-sensor correlations.

There is another method [44] that has a slightly higher accuracy (84.8%) than ours (82.9%), but it assumes perfect trajectory segmentation. This method also follows the two-step approach, while it uses true change points to segment trajectories. In real applications, the change points are unseen and the trajectories may not be segmented precisely. Thus, the performance of [44] may be of less practical value. Our STPC-Net only requires sensor ids and space-time coordinates (always available), and could achieve state-of-the-art performance in an end-to-end way.

5) Ablation Studies: To further investigate the effect of each component in our model, we compare STPC-Net with its variants as follows.

- **STPC-Net w/o conv-intra**: STPC-Net without the conv-intra operations.
- **STPC-Net w/o conv-inter**: STPC-Net without the conv-inter operations.
- **STPC-Net w/o wa**: STPC-Net without weighted aggregation approaches in conv-intra and conv-inter operations. We aggregate neighborhoods' features with average sum.
- **STPC-Net w/o gf**: STPC-Net without the gated fusion mechanism. We replace it by directly adding the intra-sensor and inter-sensor features.
- **STPC-Net w/o combination**: STPC-Net without the combination module.

Table II shows the results of each model. STPC-Net performs better than STPC-Net w/o conv-intra and STPC-Net w/o conv-inter, which demonstrate the effectiveness of intra-sensor and inter-sensor correlations. Especially, the introduction of the conv-intra operation significantly improves the results, indicating that the temporal patterns at each sensor are essential in the geo-sensory data. By further modeling the inter-sensor correlations, STPC-Net consistently improves the performance, showing the importance of the interaction information.

By removing the weighted aggregation approach, the performance of STPC-Net w/o wa degrades obviously, pointing out that the weighted aggregation is a key factor to the success of the conv-intra and conv-inter operations. It helps the network to focus on the most correlated neighboring points to update the point features.

TABLE II
ABLATION STUDIES ON THE GEOLIFE DATASET

Method	Accuracy	Precision	Recall	F1-score
STPC-Net w/o conv-intra	76.3%	77.2%	75.2%	76.0%
STPC-Net w/o conv-inter	80.3%	81.8%	78.7%	79.7%
STPC-Net w/o wa	81.4%	81.8%	80.9%	81.3%
STPC-Net w/o gf	81.6%	82.8%	80.7%	81.5%
STPC-Net w/o combination	79.7%	80.9%	78.6%	79.5%
STPC-Net	**82.9%**	**83.8%**	**82.2%**	**82.9%**

STPC-Net performs better than STPC-Net w/o gf, showing the effectiveness of the gated fusion mechanism that adaptively fuses two features in a data-dependent way.

STPC-Net outperforms STPC-Net w/o combination by a large margin, which indicates that the global information of the entire point cloud (e.g., the long-term pattern and long-range dependencies) is of great importance for learning point features.

6) Impact of Hyperparameters: We further analyze the impact of hyperparameters in STPC-Net, including the threshold of time difference in conv-intra operation ϵ, the threshold of time difference in conv-inter operation θ, and the threshold of distance in conv-inter operation ρ.

As shown in Figure 4(a), as ϵ is larger, the accuracy of STPC-Net first increases and then decreases, demonstrating that more temporal information could yields better performance. However, when the length of time window is very long, it would introduce unrelated or useless information, and the learning becomes more difficult, which hinders the performance. Similar results could be observed from Figures 4(b) and 4(c).

B. Readings Prediction

We evaluate STPC-Net of readings prediction on the PeMSD8 dataset [45] for traffic flow prediction. This dataset contains three kinds of traffic condition readings (traffic flow, speed, and occupancy) of 170 sensors in California. These sensors are fixed, and thus generate stationary geo-sensory data (Figure 1(b)). The readings are pre-aggregated into every 5-minute interval. The problem is to predict the traffic flow readings at all sensors in the next 12 time steps (1 hour) given all the three traffic condition readings of previous 12 time steps [45].

Earlier works for sensor readings prediction consider each sensor's readings as a time sequence and apply time series methods for prediction. These methods ignore the correlations between different sensors. Recent studies formulate readings prediction as a graph modeling problem. They construct a fixed sensor graph according to the locations of sensors and transform the sensory data into graph signals on that graph. By utilizing graph neural networks [15], these methods have achieved significant progress in the literature.

We learn such stationary geo-sensory data using a new data representation, i.e., spatio-temporal point clouds. In this setting, the input is $N = 170 \times 12 = 2040$ points, and the network needs to generate $N' = 2040$ new points, where the sensor ids and space-time coordinates are pre-known.

1) Data Preprocessing: Each point is associated with the sensor id, space-time coordinate, and a 3-dimensional ($d = 3$) feature vector consisting of the traffic flow, speed, and occupancy. These features are normalized via the Z-Score method. Following [46], [47], the data is split in chronological order with 60% for training, 20% for validation, and 20% for testing.

2) Experimental Settings: As described above, the input contains $N = 2040$ points, each with $4+d$ ($d = 3$) dimensions, and STPC-Net needs to output $N' = 2040$ new points, each with the predicted traffic flow value ($d' = 1$). In the first convolution module for features learning on input points, the threshold of time difference is set as $\epsilon_1 = 30$ minutes (six time steps) in the conv-intra operation. In the conv-inter operation, as the locations of sensors are fixed, we could pre-calculate the pair-wise road network distance between any two sensors. Thus, the computation of distance ($dis(\cdot,\cdot)$ in Algorithm 2) could be realized as a look-up table operation. With these non-Euclidean distances, we could inject the road network information into the learning process. The thresholds of time difference and distance in the conv-inter operation are set as $\theta_1 = 5$ minutes (one time step) and $\rho_1 = 1000$ miles, respectively. In the second convolution module for new points generation, the threshold of time difference is set as $\epsilon_2 = 120$ minutes in the conv-intra operation. In the conv-inter operation, the thresholds of time difference and distance are $\theta_2 = 90$ minutes and $\rho_2 = 1000$ miles, respectively. We will discuss the impact of these hyperparameters in section V-B.6. The output dimensions of two convolution modules are $D_1 = 64$ and $D_2 = 128$, respectively.

The loss function is the mean absolute error (MAE), which is optimized using the Adam optimizer [38] with an initial learning rate of 0.001. The evaluation metrics include root mean squared error (RMSE), MAE, and mean absolute percentage error (MAPE).

3) Baseline Methods: We compare STPC-Net with the following baseline methods.

- **SVR** [48]: Support Vector Regression that uses a linear support vector machine for regression tasks.
- **LSTM** [6]: Long Short-Term Memory is special kind of recurrent neural networks (RNN).
- **STGCN** [14]: Spatio-Temporal Graph Convolution Network that combines graph convolutional layers and convolutional sequence learning layers to capture spatial and temporal dependencies.
- **DCRNN** [13]: Diffusion Convolutional Recurrent Neural Network that integrates diffusion convolution with sequence-to-sequence architecture to encode spatial and temporal information.
- **Graph WaveNet** [29] that combines graph convolution with dilated casual convolution to capture spatio-temporal dependencies.
- **STG2Seq** [49]: Spatio-Temporal Graph to Sequence Model that uses multiple gated graph convolutional module and seq2seq architecture with attention mechanisms to make multi-step prediction.
- **ASTGCN** [45]: Attention Based Spatio-Temporal Graph Convolutional Networks that designs spatial attention and temporal attention mechanisms to model spatial and temporal dynamics, respectively.
- **STSGCN** [46]: Spatio-Temporal Synchronous Graph Convolutional Networks that synchronously captures the complex spatio-temporal correlations and takes the heterogeneity into account.
- **APTN** [47]: Attention-based Periodic-Temporal neural Network that incorporates attention mechanisms and LSTM into an encoder-decoder framework.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON THE PEMSD8 DATASET FOR TRAFFIC FLOW PREDICTION

Method	RMSE	MAE	MAPE (%)
SVR [48]	36.16±0.02	23.25±0.01	14.64±0.11
LSTM [6]	34.06±0.32	22.20±0.18	14.20±0.59
STGCN [14]	27.83±0.20	18.02±0.14	11.40±0.10
DCRNN [13]	27.83±0.05	17.86±0.03	11.45±0.03
Graph WaveNet [29]	31.05±0.07	19.13±0.08	12.68±0.57
STG2Seq [49]	30.71±0.61	20.17±0.49	17.32±1.14
ASTGCN [45]	28.16±0.48	18.61±0.40	13.08±1.00
STSGCN [46]	26.80±0.18	17.13±0.09	10.96±0.07
APTN [47]	**24.76±0.11**	15.63±0.08	9.91±0.06
STPC-Net	24.82±0.10	**15.26±0.14**	**9.77±0.14**

4) Experimental Results: Table III presents the performance of STPC-Net as compared to baseline methods. The results of baseline methods except for APTN are provided by [46], where STSGCN is compared to others. For APTN, we use the default settings in its original proposal [47]. We repeat the experiment 10 times and report the average of MAE, RMSE, MAPE with a standard deviation in Table III. We observe that even though we are working on a new data representation (i.e., spatio-temporal point clouds), we are able to achieve on-par performance with state-of-the-art.

Among the baseline methods, SVR and LSTM are time series methods and achieve poor performance because they ignore the interactions among sensors. Other baseline methods transform the data into graph signals and specially design suitable models for spatio-temporal graph modeling. Compared to the time series algorithms, these graph-based methods further capture the dependencies between different sensors via well-designed graph neural networks. Although these methods achieve promising performance, they can only be applied on

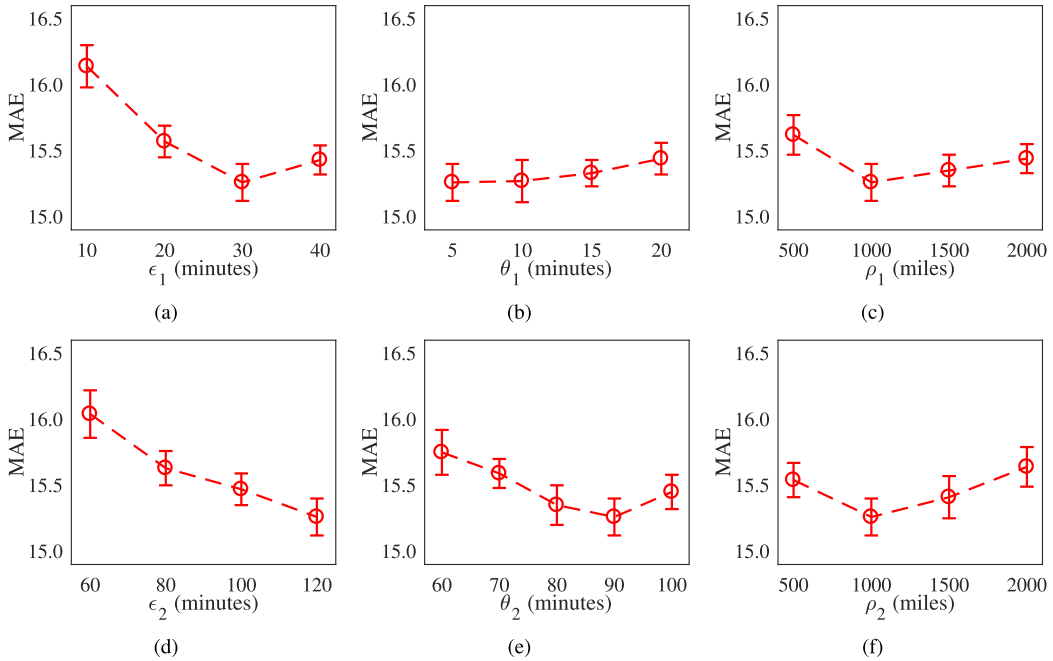


Fig. 5. The impact of hyperparameters in STPC-Net. (a) MAE with respect to the threshold of time difference in conv-intra operation in the first convolution module. (b) MAE with respect to the threshold of time difference in conv-inter operation in the first convolution module. (c) MAE with respect to the threshold of distance in conv-inter operation in the first convolution module. (d) MAE with respect to the threshold of time difference in conv-intra operation in the second convolution module. (e) MAE with respect to the threshold of time difference in conv-inter operation in the second convolution module. (f) MAE with respect to the threshold of distance in conv-inter operation in the second convolution module.

TABLE IV
ABLATION STUDIES ON THE PEMSD8 DATASET

Method	RMSE	MAE	MAPE (%)
STPC-Net w/o conv-intra	26.29±0.29	16.64±0.07	10.92±0.05
STPC-Net w/o conv-inter	25.22±0.13	15.47±0.09	10.00±0.10
STPC-Net w/o wa	25.10±0.15	15.53±0.15	10.04±0.12
STPC-Net w/o gf	24.98±0.11	15.46±0.08	9.94±0.12
STPC-Net w/o combination	25.81±0.23	16.11±0.12	10.44±0.15
STPC-Net	**24.82±0.10**	**15.26±0.14**	**9.77±0.14**

stationary geo-sensory data as they need the sensors to be stationary to pre-define the sensor graph. In other words, these graph-based models cannot deal with the mobile geo-sensory data, e.g., transportation mode identification on the GeoLife dataset in Section V-A. Our STPC-Net learns both mobile and stationary geo-sensory data as a unified spatio-temporal point clouds representation, and could achieve state-of-the-art performance on different types of tasks.

5) Ablation Studies: To further study the effect of each component in our model, we compare STPC-Net with its five variants, which are defined in section V-A.5. As shown in Table IV, STPC-Net performs much better than STPC-Net w/o conv-intra and STPC-Net w/o conv-inter, demonstrating the effectiveness of the conv-intra and conv-inter operations. In addition, the weighted aggregation approach improves the performance (STPC-Net v.s. STPC-Net w/o wa) as it enables the model to select the useful information at each convolution operation. The effect of gated fusion is evident as well (STPC-Net v.s. STPC-Net w/o gf), showing that the gated fusion mechanism is helpful to flexibly control the importance of intra-sensor and inter-sensor features. By aggregating the global feature, STPC-Net significantly improves the performance as compared to STPC-Net w/o combination, which validates the importance of the global information.

6) Impact of Hyperparameters: We present the impact of hyperparameters in Figure 5. Unlike STPC-Net of point classification, where two convolution modules are both used for abstracting features from input points (Figure 2) and adopt the same hyperparameters, in STPC-Net of readings prediction, the second convolution module is used for generating new points from the input points, which is different from the first one. Thus, here we apply different hyperparameters in two convolution modules.

As shown in Figure 5, in general, as the size of neighborhood becomes larger (ϵ, θ, or ρ increases), the model could achieve better performance because more spatio-temporal information is considered. However, when the size of neighborhood is very large, it would introduce useless or unrelated information into the learning process and thus degrades the performance.

VI. DISCUSSION

In this paper, we extend the representation of point clouds to represent massive geo-sensory data as spatio-temporal point clouds (STPC). Inspired by PointNet [19], which is the first neural network directly consumes 3D point clouds, we further propose a new deep network (i.e., STPC-Net) that suitable for processing STPC. In this section, we discuss the similarities and differences between STPC and 3D point clouds, STPC-Net and PointNet, respectively.

A. Spatio-Temporal Point Clouds Versus 3D Point Clouds

1) Similarities: Both the STPC and 3D point clouds are sets of points, which are not in regular formats. The points in both

point clouds are usually sparse, and the neighboring points are correlated with each other.

2) Differences: The differences between STPC and 3D point clouds mainly lie in two aspects. First, in a 3D point cloud, each point is associated with space coordinates. While, in the STPC, each point is associated with both space and time coordinates. Several recent works [21], [22], [35], [36] attempt to study on 3D point cloud sequences. While, they are also different to STPC, which is not a point cloud sequence, but a set of points with different timestamps. Second, the points in STPC are generated by multiple sensors. It inherently has complex intra-sensor and inter-sensor correlations in the STPC. While, these correlations are not existed in 3D point clouds.

B. STPC-Net Versus PointNet

1) Similarities: Both the STPC-Net and PointNet extract information from irregular point sets. Both of them apply a pooling layer to form a global feature to model the correlations among distant points.

2) Differences: The differences between STPC-Net and PointNet are two-fold. The main difference lies in the features learning process. PointNet uses multi-layer perception (MLP) networks to learn each point's feature independently. By this design, PointNet does not capture local information. The improved version, i.e., PointNet++ [20] models the local correlations among neighboring points. It focuses on learning features in the spatial dimension. While, the STPC has complex intra-sensor and inter-sensor correlations in both spatial and temporal dimensions. We carefully design the conv-intra and conv-inter operations to model these correlations in STPC-Net. Moreover, it applies a gated fusion mechanism to adaptively fuse them. The other difference is due to the learning task. STPC-Net is able to generate new points to predict future sensor readings. While, there is no similar design in PointNet.

VII. Conclusion

We proposed a unified representation to describe both mobile and stationary geo-sensory data without information-losing discretization in spatial and temporal dimensions, i.e., spatio-temporal point clouds (STPC). We further designed a deep network (STPC-Net) suitable for processing the STPC that collectively models the local intra-sensor and inter-sensor correlations, as well as the global information. When evaluated on two types of data in different tasks (i.e., point classification on mobile geo-sensory data and readings prediction on stationary geo-sensory data), STPC-Net consistently achieves state-of-the-art performance. This study extends the representation of point clouds to a new filed, i.e., representing massive geo-sensory data, and provides a new way to handle the irregular geo-sensory data.

In practical, the mobile sensors could travel all over the city to monitor the environment, while they may not monitor a location at all times. The stationary sensors can provide continuous measurements at specific locations, while their high maintenance cost allows only a limited number of installations. It is important to fuse both the mobile and stationary geo-sensory data to provide a wide spatial range of continuous measurements. The proposed unified representation (i.e., STPC) that can describe both mobile and stationary geo-sensory may potentially benefit to the fusion. We plan to investigate this in future work.

References

[1] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[2] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[3] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: Multi-level attention networks for geo-sensory time series prediction," in *Proc. IJCAI*, 2018, pp. 3428–3434.

[4] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[6] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.

[7] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, "When will you arrive estimating travel time based on deep neural networks," in *Proc. AAAI*, 2018, pp. 2500–2507.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[9] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI*, 2017, pp. 1655–1661.

[10] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proc. AAAI*, 2019, pp. 5668–5675.

[11] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on GPS data," in *Proc. UbiComp*, 2008, pp. 312–321.

[12] T. H. Do *et al.*, "Graph-deep-learning-based inference of fine-grained air quality from mobile IoT sensors," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8943–8955, Sep. 2020.

[13] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. ICLR*, 2018, pp. 1–16.

[14] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3634–3640.

[15] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[16] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Oct. 2020.

[17] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jun. 29, 2020, doi: 10.1109/TPAMI.2020.3005434.

[18] S. A. Bello, S. Yu, C. Wang, J. M. Adam, and J. Li, "Review: Deep learning on 3D point clouds," *Remote Sens.*, vol. 12, no. 11, p. 1729, May 2020.

[19] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proc. CVPR*, 2017, pp. 652–660.

[20] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. NeurPIS*, 2017, pp. 5099–5108.

[21] X. Liu, C. R. Qi, and L. J. Guibas, "FlowNet3D: Learning scene flow in 3D point clouds," in *Proc. CVPR*, 2019, pp. 529–537.

[22] X. Liu, M. Yan, and J. Bohg, "Meteornet: Deep learning on dynamic 3D point cloud sequences," in *Proc. ICCV*, 2019, pp. 9245–9254.

[23] C. Zhang, Y. Zheng, X. Ma, and J. Han, "Assembler: Efficient discovery of spatial co-evolving patterns in massive geo-sensory data," in *Proc. KDD*, 2015, pp. 1415–1424.

[24] Q. Gao, F. Zhou, K. Zhang, G. Trajcevski, X. Luo, and F. Zhang, "Identifying human mobility via trajectory embeddings," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1689–1695.

[25] Y. Wang *et al.*, "Unlicensed taxis detection service based on large-scale vehicles mobility data," in *Proc. ICWS*, 2017, pp. 857–861.

[26] H. Wu, Z. Chen, W. Sun, B. Zheng, and W. Wang, "Modeling trajectories with recurrent neural networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3083–3090.

[27] C. Zheng, X. Fan, C. Wen, L. Chen, C. Wang, and J. Li, "DeepSTD: Mining spatio-temporal disturbances of multiple context factors for citywide traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3744–3755, Sep. 2020.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[29] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proc. IJCAI*, 2019, pp. 1907–1913.

[30] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, pp. 1234–1241, Apr. 2020.

[31] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on *X*-transformed points," in *Proc. NeurPIS*, 2018, pp. 820–830.

[32] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9621–9630.

[33] Z. Rozsa and T. Sziranyi, "Object detection from a few LIDAR scanning planes," *IEEE Trans. Intell. Veh.*, vol. 4, no. 4, pp. 548–560, Dec. 2019.

[34] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal convnets: Minkowski convolutional neural networks," in *Proc. CVPR*, 2019, pp. 3075–3084.

[35] H. Fan and Y. Yang, "PointRNN: Point recurrent neural network for moving point cloud processing," 2019, *arXiv:1910.08287*. [Online]. Available: http://arxiv.org/abs/1910.08287

[36] Y. Min, Y. Zhang, X. Chai, and X. Chen, "An efficient pointLSTM for point clouds based gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5760–5769.

[37] S. Dabiri, C. Lu, K. Heaslip, and C. K. Reddy, "Semi-supervised deep learning approach for transportation mode identification using GPS trajectory data," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 5, pp. 1010–1023, May 2020.

[38] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.

[39] Y. Endo, H. Toda, K. Nishida, and A. Kawanobe, "Deep feature extraction from trajectories for transportation mode estimation," in *Proc. PAKDD*, 2016, pp. 54–66.

[40] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[41] H. Wang, G. Liu, J. Duan, and L. Zhang, "Detecting transportation modes using deep neural network," *IEICE Trans. Inf. Syst.*, vol. 100, no. 5, pp. 1132–1135, May 2017.

[42] X. Jiang, E. N. de Souza, A. Pesaranghader, B. Hu, D. L. Silver, and S. Matwin, "Trajectorynet: An embedded GPS trajectory representation for point-based classification using recurrent neural networks," in *Proc. CASCON*, 2017, pp. 192–200.

[43] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. ICML*, May 2013, pp. 1319–1327.

[44] S. Dabiri and K. Heaslip, "Inferring transportation modes from GPS trajectories using a convolutional neural network," *Transp. Res. C, Emerg. Technol.*, vol. 86, pp. 360–371, Jan. 2018.

[45] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI*, 2019, pp. 922–929.

[46] C. Song, Y. Lin, S. Guo, and HuaiyuWan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proc. AAAI*, 2020, pp. 914–921.

[47] X. Shi, H. Qi, Y. Shen, G. Wu, and B. Yin, "A spatial-temporal attention approach for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 13, 2020, doi: 10.1109/TITS.2020.2983651.

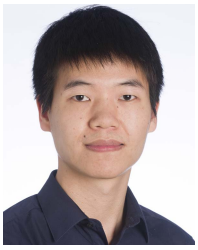
[48] C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004.

[49] L. Bai, L. Yao, S. S. Kanhere, X. Wang, and Q. Z. Sheng, "STG2Seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1981–1987.

Chuanpan Zheng received the B.Sc. degree in applied physics from Shandong University, Jinan, China, in 2012. He is currently pursuing the Ph.D. degree in computer science and technology with Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, China. His research interests include spatio-temporal data representation learning and graph neural networks.

Cheng Wang (Senior Member, IEEE) received the Ph.D. degree in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2002. He is currently a Professor and an Associate Dean with the School of Informatics, Xiamen University, China, where he is an Executive Director of Fujian Key Laboratory of Sensing and Computing for Smart Cities. He has coauthored over 150 articles in referred journals and top conferences including IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), *PR*, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (TITS), AAAI, CVPR, IJCAI, and *ISPRS-JPRS*. His research interests include remote sensing image processing, mobile LiDAR data analysis, and multi-sensor fusion.

Xiaoliang Fan (Senior Member, IEEE) received the Ph.D. degree from University Pierre and Marie CURIE, France, in 2012. He is currently a Senior Research Specialist with Fujian Key Laboratory of Sensing and Computing for Smart Cites, Computer Science and Technology Department, Xiamen University. He has published more than 60 journals and conference papers in these areas. His research interests include spatio-temporal data mining and privacy-aware computing. He is a Senior Member of China Computer Federation (CCF).



Jianzhong Qi (Member, IEEE) received the Ph.D. degree from The University of Melbourne in 2014. He is currently a Senior Lecturer with the School of Computing and Information Systems, The University of Melbourne. His research interests include machine learning and data management and analytics, with a focus on spatial, temporal, and textual data.

Xu Yan received the B.Sc. degree in computer science and technology from Xiamen University, Xiamen, China, in 2019, where she is currently pursuing the M.S. degree in computer science and technology with Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics.