Joint Margin, Cograph, and Label Constraints for Semisupervised Scene Parsing From Point Clouds

Jie Mei, Liqiang Zhang, Yuebin Wang, Zidong Zhu, and Huiqian Ding

***Abstract*— To parse large-scale urban scenes using the supervised methods, a large amount of training data that can account for the vast visual and structural variance of urban environment is necessary. Unfortunately, such training data are mostly obtained by tedious and time-consuming manual work. To overcome the drawback, we propose a semisupervised learning framework that combines the margin, cograph, and label constraints into an objective function for point cloud parsing. Mathematically, the margin constraint is presented to learn a novel distance criterion that can effectively recognize points of different classes. The graph regularization is then employed to characterize the intrinsic geometry structure of the data manifold and explore relationships among points. The label consistency regularization is introduced to ensure the category consistency of the clustered points and single point. To classify the out-of-sample data, the framework successfully transforms the semisupervised classification results into the linear classifier by adopting a linear regression. An iterative algorithm is utilized to efficiently and effectively optimize the objective function with characteristics of multiple variables and highly nonlinear. The point clouds of four urban scenes are used to validate our method. The experimental results show that our method outperforms the state-of-the-art algorithms.**

***Index Terms*— Classification, feature, label consistency, manifold regularization, margin constraint, point cloud.**

I. Introduction

POINT cloud understanding plays a critical role in image processing and analysis [1]–[5]. The full-scene labeling is an important prerequisite for scene understanding and interpretation semantics of the point clouds. The scene parsing means that every point in the point cloud is labeled with the category of the object which it belongs to [6]. It is a challenge for accurately parsing a scene from the unorganized and unoriented 3-D point cloud corrupted with noise, outliers, nonuniformities, and partial data missing.

The current supervised approaches like [7]–[12] have obtained high-quality classification or segmentation results from point clouds. However, they usually need a large amount of training data to learn the features and classifiers, reducing the computational efficiency. Moreover, such training data are mostly obtained by tedious and time-consuming manual labeling, which inevitably become an obstacle to expanding these traditional parsing methods to the large scale [13].

In this paper, we propose a semisupervised learning method that integrates the margin constraint, cograph regularization, and label regularization for point cloud classification. Fig. 1 illustrates the proposed approach. First, the features of each point are generated. Then, the margin constraint is proposed to learn a novel distance criterion, which can maximize the boundaries among different objects. The cograph regularization is employed to discover the intrinsic geometry structure of the data manifold and explore relationships among points. The label consistency regularization is introduced to ensure the category consistency of the clustered points as well as the global and local consistencies of single points. In order to classify the out-of-sample data, our method adopts a linear regression to project the semisupervised classification results into the linear classifier. We present an iterative algorithm to optimize the variables of the objective function. Our method has been evaluated on the point clouds of three scenes with different complexities. The experimental results show that our method is superior to other state-of-the-art algorithms.

We explicitly state our original contributions as follows.

1) A semisupervised learning method that integrates the advantages of the margin constraint, cograph regularization, and label regularization is proposed to parse point cloud. The method considers the intraclass compactness and interclass separability of labeled points, as well as the label consistency on grouped points and single point of unlabeled data. The experimental results show that our method outperforms the state-of-the-art algorithms.
2) The constraints for the margins of different class points, features, spatial, and label consistency are combined into an objective function, which can effectively distinguish the same class points from the point clouds with noise and data missing. To solve the objective function, an effective and efficient optimization algorithm is developed. The convergence is fast.
3) The cograph is constructed by integrating the manifolds of the point features and spatial distributions to ensure the local consistency of neighboring points.

II. Related Work

In this section, we mainly discuss the feature representation of point clouds and the classification approaches including supervised and semisupervised methods.

Manuscript received April 6, 2017; revised December 11, 2017; accepted February 24, 2018. Date of publication March 28, 2018; date of current version June 22, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 41371324 and in part by China Postdoctoral Science Foundation under Grant 2016M600953. *(Corresponding author: Liqiang Zhang.)*

The authors are with the State Key Laboratory of Remote Sensing Science, Faculty of Geographical Science, Beijing Normal University, Beijing 100875, China (e-mail: meijie@mail.bnu.edu.cn; zhanglq@bnu.edu.cn; xxgcdxwyb@163.com; zhuzidong@mail.bnu.edu.cn; hqding163@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2018.2811748

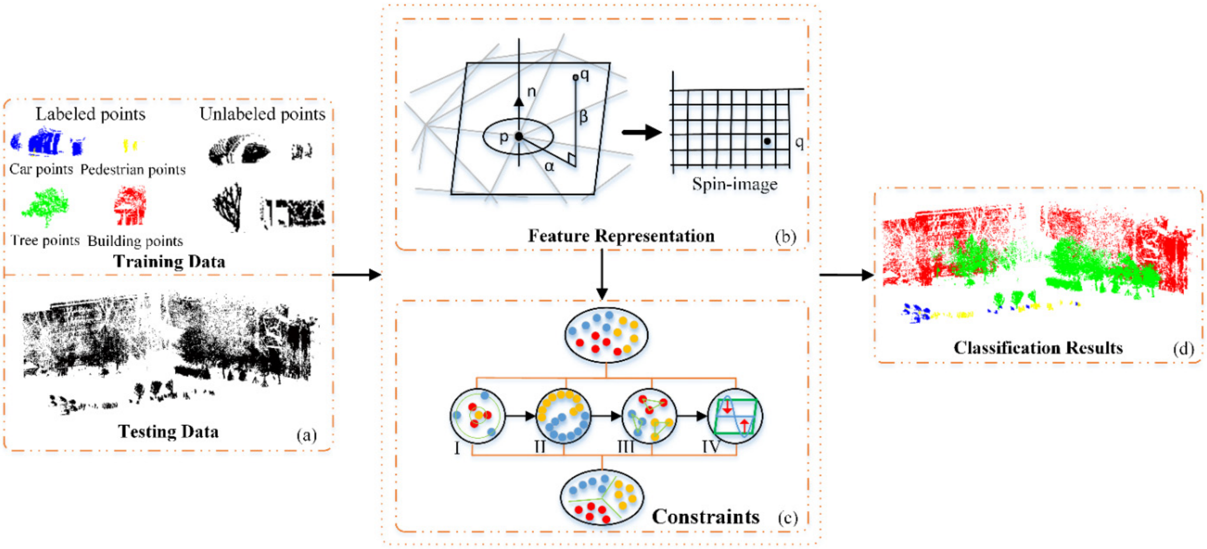


Fig. 1. Overview of the proposed approach. (a) Point clouds. (b) Feature extraction procedure for point clouds. (c) High-level feature learning. (d) Classification results. I, II, III, and IV correspond to the margin constraints, cograph regularization, label consistency constraint, and linear regression, respectively.

A. Feature Representation

Feature representation is very critical for achieving good scene parsing results [14], [15]. Lin *et al.* [16] used a weighted covariance matrix with a geometric median to extract local geometric features of point cloud. Each point is assigned with a weight to estimate the geometric median and represent its spatial contribution in the weighted principal component analysis. The eigenvector-based feature [17], [18] and the features derived from spin image (SI) [19]–[22] have been widely used in point cloud classification. For example, Martinovic *et al.* [23] introduced a set of feature descriptors for each 3-D point. The point-based feature descriptors are generally sensitive to the noise, data missing, and outliers. To overcome the drawbacks and obtain discriminative features, the point cluster-based features were proposed to extract and encode the shape features of the multilevel point clusters [24], [25]. Through fusing airborne lasers scanning point clouds and images, voxels' features, including geometric, textural, and multiple-level image features, were derived for follow-up classification [26]. Yang *et al.* [27] combined multiple aggregation levels of features and contextual features to recognize road facilities. Geometric, textural, and low-level and middle-level image features are assigned to laser points which are quantified into voxels. Frome *et al.* [28] introduced two shape descriptors of 3-D shape contexts and harmonic shape contexts, to obtain a high recognition rate on noisy scenes. In a rule-based hierarchical semantic classification scheme, Rau *et al.* [29] employed geometry-spectral information and topology-related features to 3-D point clouds, which were classified into five categories including road, roof, grass, tree, and facade.

The above hand-crafted features often fail to adequately utilize the consistency and complementary information between points. In other words, the abovementioned features are difficult to capture high-level semantic structures. Although the features learned from most of the current deep-learning methods [30]–[36] can generate high-quality point cloud classification results, these methods do not adequately recognize 3-D point clouds due to unorganized distribution, uneven point density, and time consuming.

B. Supervised Methods

In recent years, many supervised methods are proposed to classify point clouds [37]–[39]. Guo *et al.* [40] proposed a supervised classification method to recognize objects from point clouds using a JointBoost classifier, and the features are computed based on the echo and geometry information of a light detection and ranging (LiDAR) point cloud. Mallet *et al.* [41] utilized a multiclass support vector machine (SVM) based on the echo amplitude and radiometric features to classify point clouds decomposed from the full-waveform LiDAR. Negri *et al.* [42] adopted a local adjustment of the separating hyperplane based on contextual information, which is used to displace the separation hyperplane defined by the SVM. Lodha *et al.* [43] used the SVM to classify the aerial LiDAR point clouds into trees, grass, roads, and buildings based on the five features including height (HI), HI variation, normal variation, LiDAR intensity, and image intensity. In order to classify land use and land cover maps from high-resolution LiDAR data, García-Gutiérrez *et al.* [44] utilized a contextual classifier which combines the SVM and evolutionary majority voting. Niemeyer *et al.* [45] integrated a random forest classifier into the conditional random field to classify complex urban scenes. Lafarge and Mallet [46] employed an energy minimization optimized by graph-cut-based algorithm to classify the point set. To reduce the intensive labeling in supervised methods, Li *et al.* [3] and Zhang *et al.* [7] proposed approaches to automatically generate training data from the input data.

C. Semisupervised Methods

Compared with the supervised algorithms, the semisupervised methods need less labeled training data. The latter can utilize a large amount of unlabeled data to improve the performance of classification. Joachims [47] proposed transductive SMs to minimize the misclassifications. Blum and Mitchell [48] provided a probably approximately correct-style framework to learn from unlabeled data and labeled data. Cai *et al.* [49] proposed semisupervised discriminant analysis, which used the unlabeled data to estimate the intrinsic geometric structure and used the labeled data to maximize the separability of different classes.

Recently, semisupervised learning based on graph Laplacian has gained great research interest. In general, the graph regularization propagates the label information from the labeled data to the unlabeled data to improve the classification results of semisupervised learning methods [50]. Based on the graph regularization term, Belkin *et al.* [51] introduced Laplacian regularized least squares (LapRLS) and Laplacian SVMs that integrate the labeled and unlabeled data in a general-purpose learner. Nie *et al.* [52] proposed a flexible manifold embedding (FME) framework for semisupervised and unsupervised dimension reduction, which can effectively take advantage of the manifold structure from unlabeled and labeled data. A semisupervised feature analyzing framework for multimedia data understanding was proposed in [53], which can learn a classifier by selecting the discriminating features related to the semantic concepts. Fang *et al.* [54] proposed a novel nonnegative sparse graph (NNSG) learning method by unifying the graph structure learning and linear regression within the same framework.

III. Feature Construction

Terrestrial laser scanning (TLS) point cloud data are generally incomplete and noisy due to severe occlusions caused by both self-occlusion and other objects. However, the point-based features introduced in recent years are sensitive to the noise and computed inefficient. Since the SIs are robust to the point cloud with missing data and noise [55], we use the SIs to construct the feature of the region around a point.

The SI obtains the abundant shape features for surrounding area of a point in the 3-D scene, which represents the 3-D spatial information by 2-D histograms. Let $\mathbf{N}_p = \{q | q$ is one of the neighbor points of p, and $|p - q| \leq r\}$ be the support region of p. $\mathbf{N}_p$ is the point set within the sphere whose radius is r and the centroid is p. SI is generated by projecting the 3-D point clouds onto 2-D image defined in object-oriented coordinate system with a specific support point p and the associated normal. Equations (1) and (2) are used to compute the coordinate of p in the SI

$$x = \sqrt{\|\mathbf{q} - \mathbf{p}\|^2 - [\mathbf{n} \cdot (\mathbf{q} - \mathbf{p})]^2} \tag{1}$$

$$y = \mathbf{n} \cdot (\mathbf{q} - \mathbf{p}) \tag{2}$$

where x denotes the horizontal coordinate of p in the SI and y denotes the vertical coordinate of p in the SI. Parameter $\mathbf{n}$ denotes the normal vector of p. $\mathbf{p}$ denotes the 3-D coordinates of p, and $\mathbf{q}$ denotes the 3-D coordinates of q.

We create a 5 × 8 SI per point and construct three spheres with radii of 0.2, 0.8, and 1.4 m, respectively. Subsequently, a 5 × 8 × 3 SI feature can be generated from the three spheres. Apart from the **SI** descriptor, we also calculate a set of descriptors for each point: mean **RGB** colors of the points as seen in the images; the **LAB** values of the mean RGB; the normal vector $\mathbf{n}_i$ of each point; the **HI** of the point above the approximated ground plane, and its inverse **HI** ($\mathbf{HI}^{-1}$), defined as the distance from the highest point of the object facade in the gravity vector direction. The full 131-D features of each point is defined as

$$\underset{131\times1}{\mathbf{F}_i} = \left[\underset{3\times1}{\mathbf{RGB}_i^T} \quad \underset{3\times1}{\mathbf{LAB}_i^T} \quad \underset{3\times1}{\mathbf{n}_i^T} \quad \underset{120\times1}{\mathbf{SI}_i^T} \quad \underset{1\times1}{\mathbf{HI}_i^T} \quad \underset{1\times1}{\mathbf{HI}_i^{-1^T}}\right]. \tag{3}$$

IV. Methodology

In this section, we describe the proposed method for point cloud parsing, which contains four main parts: feature learning, cograph regularization, label consistency regularization, and linear regression. The intrinsic structure of points and label manifolds integrated by the graph is utilized to constrain the features. Then, the linear regression is utilized to obtain the linear classifier for parsing the out-of-sample data.

A. Margin Constraint

The label of an unknown point is usually the same as the labels of its nearest neighbors. In general, the meaning of the term nearest is defined with the notion of distance in data space. It is difficult to measure the similarity between points using the Euclidean distance due to the high dispersity and incompletion of point clouds. A suitable distance metric should reduce the distance between intraclass points and increase the distance between interclass points. The most frequently used distance is the Mahalanobis distance with a positive definite square matrix M.

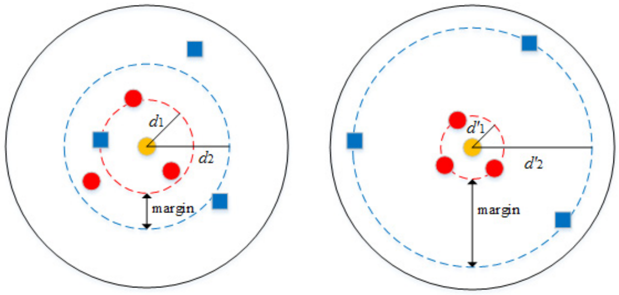
Given the training data matrix $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^{d\times n}$, where n is the number of the training data and d is the dimension of the feature, the distance between two features $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as

$$\begin{aligned} d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) &= \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{M}} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)} \\ &= \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{A}\mathbf{A}^T)(\mathbf{x}_i - \mathbf{x}_j)} \\ &= \|\mathbf{A}^T \mathbf{x}_i - \mathbf{A}^T \mathbf{x}_j\| \end{aligned} \tag{4}$$

where $\mathbf{M} = \mathbf{A}\mathbf{A}^T$ and $\mathbf{A} \subset \mathbb{R}^{d\times d'}$ $(d < d')$ is a feature transformation matrix.

From (4), it is noted that the learning distance metric matrix $\mathbf{M}$ is equivalent to learning the feature transformation matrix $\mathbf{A}$. Essentially, the metric learning algorithm aims to seek a separation between sets of data belonging to different classes. In this paper, we propose the margin constraint to learn a novel distance criterion that can provide a tradeoff between intraclass compactness and interclass separability. Specifically, we define (5) to form the margin constraint condition

$$\begin{aligned} \Theta_1(\mathbf{A}) = &\frac{1}{nk_1} \sum_{i=1}^{n} \sum_{j=1}^{k_1} \|\mathbf{A}^T \mathbf{x}_i - \mathbf{A}^T \mathbf{x}_j\|_2^2 \\ &- \alpha \frac{1}{nk_2} \sum_{i=1}^{n} \sum_{j=1}^{k_2} \|\mathbf{A}^T \mathbf{x}_i - \mathbf{A}^T \mathbf{x}_j\|_2^2 \end{aligned} \tag{5}$$



(a) (b)

Fig. 2. Illustration of the margin constraint. (a) Points in the original space. (b) Points in the transformed space. The yellow dots in the center represent $\mathbf{x}_i$, the red dots are the points belonging to the same class with $\mathbf{x}_i$, and the blue squares are the points that are different classes with $\mathbf{x}_i$. Distances d_1 and d_1' represent the average distances between points belonging to the same class before/after the feature transformation. Distances d_2 and d_2' represent the average distances between points belonging to different classes before/after the feature transformation.

where k_1 is the number of k_1-intraclass neighbor points of $\mathbf{x}_i$ and k_2 is the number of k_2-interclass neighbor points of $\mathbf{x}_i$. It is obvious that the first term of (5) measures the average distance between the pairwise points belonging to the same class, representing the compactness of the distribution of the same class in other words. The second term measures the average distance between the pairwise points belonging to the different classes, indicating the separability of the distribution of different classes. By minimizing the objective function in (5), the distances between points of the same class are decreased and the distances between points belonging to different classes are increased, achieving the maximum margin between different classes in the transformed space, as shown in Fig. 2.

B. Cograph Regularization

After the feature learning with the margin constraint, we get a new d'-dimensional feature $\mathbf{A}^T\mathbf{X}$ instead of the initial feature $\mathbf{X}$. While the margin constraint ignores the local manifold structure, we construct a cograph to ensure the local consistency of neighboring points by integrating the manifolds of the feature space and spatial distribution.

Motivated by the assumption that points with a close intrinsic geometry relationship in the data distribution usually have similar feature structure in the feature space [56], we construct the graph Laplacian $\mathbf{G}_F$ to express the relationships among the points. Then, we can build the graph with the ith node corresponding to the training data $\mathbf{x}_i$. Specifically, we select k_F-nearest neighbors of $\mathbf{x}_i$ according to the weight matrix $\mathbf{U}_{ij}$, which is defined as follows:

$$\mathbf{U}_{ij} = \begin{cases} \exp\left(-d_{ij}^2\right) & \mathbf{x}_i \in \mathbf{N}_{k_F}(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathbf{N}_{k_F}(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$d_{ij}^2 = \|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2 = (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{A}\mathbf{A}^T(\mathbf{x}_i - \mathbf{x}_j) \tag{7}$$

where d_{ij} is the distance between two features $\mathbf{A}^T\mathbf{x}_i$ and $\mathbf{A}^T\mathbf{x}_j$. $\mathbf{N}_{kF}(\mathbf{x}_i)$ is the set of the k_F-nearest neighbors of $\mathbf{x}_i$ according to d_{ij}.

After $\mathbf{G}_F$ is constructed, we encode the learned point features by the following objective function that simultaneously satisfies the manifold assumption and preserves the local visual similarity among different points [57]:

$$\begin{aligned}\Theta_{21}(\mathbf{A}) &= \frac{1}{2}\sum_{i,j=1}^{n} \mathbf{U}_{ij}\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2 \\ &= \mathrm{Tr}\left[(\mathbf{A}^T\mathbf{X})\mathbf{L}_F(\mathbf{A}^T\mathbf{X})^T\right]\end{aligned} \tag{8}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace operator. The Laplacian matrix $\mathbf{L}_F = \mathbf{D}_F - \mathbf{U}$, $\mathbf{D}_F$ is a diagonal matrix with the diagonal elements as $\mathbf{D}_{F_{ii}} = \sum_j \mathbf{U}_{ij}$.

Since the adjacent points generally have similar feature properties [49], [58], we design the graph $\mathbf{G}_S$ to model the spatial relationships of points according to the spatial information. Similar to the construction of graph $\mathbf{G}_F$, we select the k_s-nearest neighbors by the weight matrix $\mathbf{V}$ to construct the graph $\mathbf{G}_S$. The weight matrix $\mathbf{V}$ is defined as follows:

$$\mathbf{V}_{ij} = \begin{cases} \exp\left(-\dfrac{\|\mathbf{p}_i - \mathbf{p}_j\|_2^2}{\sigma}\right) & \mathbf{x}_i \in \mathbf{N}_{k_S}(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathbf{N}_{k_S}(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $\mathbf{N}_{ks}(\mathbf{x}_i)$ is the set of the k_S-nearest neighbors of point $\mathbf{x}_i$ according to the Euclidean distance $||\mathbf{p}_i - \mathbf{p}_j||$ and σ is a scale parameter. The corresponding objective function based on the manifold regularization is expressed as

$$\begin{aligned}\Theta_{22}(\mathbf{A}) &= \frac{1}{2}\sum_{i,j=1}^{n} \mathbf{V}_{ij}\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2 \\ &= \mathrm{Tr}\left[(\mathbf{A}^T\mathbf{X})\mathbf{L}_S(\mathbf{A}^T\mathbf{X})^T\right]\end{aligned} \tag{10}$$

where $\mathbf{L}_S = \mathbf{D}_S - \mathbf{V}$; $\mathbf{D}_S$ is a diagonal matrix of which the (i, i)th element equals to the sum of the ith row of $\mathbf{V}$.

Combining (8) and (10), the cograph regularization is formed to exploit the consistent information of features and the spatial for point clouds. The objective function is defined as follows:

$$\begin{aligned}\Theta_2(\mathbf{A}) &= \frac{1}{2}\sum_{i,j=1}^{n} (\mathbf{U}_{ij} + \beta\mathbf{V}_{ij})\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2 \\ &= \mathrm{Tr}\left[(\mathbf{A}^T\mathbf{X})(\mathbf{L}_F + \beta\mathbf{L}_S)(\mathbf{A}^T\mathbf{X})^T\right]\end{aligned} \tag{11}$$

where β is a tradeoff parameter.

C. Label Consistency Constraint on Grouped Points

The above margin constraint and cograph regularization are utilized to learn the feature and discover the intrinsic geometrical structure of the data manifold. However, there are no label constraints on intraclass points to deal with the data with multiclass. We further introduce the label consistency constraint.

The distance metric should consider the label consistency of the point clouds, that is, the intraclass points should keep the label consistency before/after the feature transformation. Moreover, the points of different categories should not be mixed together. To obtain the label consistency and the discriminative distance metric, we propose the group label matrix $\mathbf{Q}$. For example, assuming that $\mathbf{x}_1$ and $\mathbf{x}_2$ belong to class one, $\mathbf{x}_3$ and $\mathbf{x}_4$ belong to class two, and $\mathbf{x}_5$ and $\mathbf{x}_6$ belong to class three, then $\mathbf{Q}$ can be defined as

$$\mathbf{Q} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

The corresponding objective function can be defined as

$$\Theta_3(\mathbf{A}, \mathbf{G}) = \|\mathbf{Q} - \mathbf{G}\mathbf{A}^T\mathbf{X}\|_F^2 + \gamma \|\mathbf{G}\|_2^2 \tag{12}$$

where $\mathbf{G}$ is a weight matrix, γ is a parameter used to balance the weight of second term in the formula, and the second term is a penalty term to avoid overfitting.

D. Label Consistency Constraint on Single Point

When a point has similar feature to its surrounding points, there is a high possibility that the points belong to the same class. We adopt the semisupervised method based on the manifold smoothness with labels to regularize the local and global label consistencies, learn the predicted label matrix $\mathbf{F} \in \mathbb{R}^{n\times c}$, and further maintain the feature similarity between points. The feature transformation matrix $\mathbf{A}$ and predicted label matrix $\mathbf{F}$ are further regularized by the following objective function:

$$\Theta_4(\mathbf{F}, \mathbf{A}) = \sum_{i,j=1}^{n} (\mathbf{U}_{ij} + \beta\mathbf{V}_{ij})\|\mathbf{F}_i - \mathbf{F}_j\|_2^2 + \sum_{i=1}^{n} \|\mathbf{F}_i - \mathbf{Y}_i\|_2^2 \tag{13}$$

where $\mathbf{Y} \subset \mathbb{R}^{n\times c}$ is the label matrix used to make the point cloud classification results are consistent with the given labels and c is the class number. $\mathbf{Y}_{ij}$ indicates the jth data of $\mathbf{Y}_i$. $\mathbf{Y}_{ij} = 1$ if $\mathbf{x}_i$ belongs to the jth class; otherwise, $\mathbf{Y}_{ij} = 0$.

The first term of (13) is used to simultaneously optimize the relationships among the points and the classification results. In the second term, the classification results are learned with global and local consistencies, since the actual label is integrated into the term.

E. Linear Regression

It is noted that the predicted label matrix $\mathbf{F}$ is constrained by the semisupervised algorithm, and only the classification results of points that are vertices of the graphs can be obtained using $\mathbf{F}$. For the out of sample, that is, test data, it is inefficient and time consuming to add these points into the graphs. To classify the out-of-sample data effectively, we present a new linear transformation to convert $\mathbf{F}$ into the linear classifier, which learns the transformation matrix $\mathbf{H} \in \mathbb{R}^{d'\times c}$ and label matrix $\mathbf{F} \in \mathbb{R}^{n\times c}$ by the linear regression. The objective function can be defined as

$$\Theta_5(\mathbf{H}) = \|(\mathbf{A}^T\mathbf{X})^T\mathbf{H} - \mathbf{F}\|_F^2. \tag{14}$$

Combining the constraints of (5) and (11)–(14), the joint objective function of our method can be defined as follows:

$$\begin{aligned}
&\Theta(\mathbf{A}, \mathbf{G}, \mathbf{F}, \mathbf{H}) \\
&= \Theta_1 + \lambda_1\Theta_2 + \lambda_2\Theta_3 + \lambda_3\Theta_4 + \lambda_4\Theta_5 \\
&= \arg\min_{\mathbf{A},\mathbf{G},\mathbf{F},\mathbf{H}} \frac{1}{nk_1}\sum_{i=1}^{n}\sum_{j=1}^{k_1} \|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2 \\
&\quad - \alpha\frac{1}{nk_2}\sum_{i=1}^{n}\sum_{j=1}^{k_2} \|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2 \\
&\quad + \lambda_1 \mathrm{Tr}\,[(\mathbf{A}^T\mathbf{X})(\mathbf{L}_F + \beta\mathbf{L}_S)(\mathbf{A}^T\mathbf{X})^T] \\
&\quad + \lambda_2\big(\|\mathbf{Q} - \mathbf{G}\mathbf{A}^T\mathbf{X}\|_F^2 + \gamma\|\mathbf{G}\|_2^2\big) \\
&\quad + \lambda_3\left(\sum_{i,j=1}^{n} (\mathbf{U}_{ij} + \beta\mathbf{V}_{ij})\|\mathbf{F}_i - \mathbf{F}_j\|_2^2 + \sum_{i=1}^{n}\|\mathbf{F}_i - \mathbf{Y}_i\|_2^2\right) \\
&\quad + \lambda_4\|(\mathbf{A}^T\mathbf{X})^T\mathbf{H} - \mathbf{F}\|_F^2
\end{aligned} \tag{15}$$

where λ_1, λ_2, λ_3, and λ_4 are used to balance the weight of each term. $\| \bullet \|$ denotes the norm of a matrix, which is generally l_2 norm or Frobenius norm.

V. Optimization Algorithm

For the highly nonlinear optimization problem such as (15), it is impractical to solve the variables $\mathbf{H}$, $\mathbf{A}$, $\mathbf{G}$, and $\mathbf{F}$ using the gradient descent method or Newton's method. We adopt an iterative algorithm to optimize the variables, in which the variables can be optimized, respectively, in each iteration as other three variables are fixed. In this case, $\boldsymbol{\Theta}$ can be taken as the linear function of $\mathbf{H}$, $\mathbf{G}$, and $\mathbf{F}$, so we can obtain the minimum value without computing the nearest gradient. Although the optimization of $\mathbf{A}$ is still nonlinear, the difficulty of the optimization for $\mathbf{A}$ is significantly reduced compared with optimizing the four variables simultaneously by the gradient descent method. In practice, our algorithm can converge very quickly during the iteration process.

A. Update H

We rewrite (15) to the following equation as $\mathbf{A}$, $\mathbf{G}$, and $\mathbf{F}$ are fixed:

$$\Theta(\mathbf{H}) = \arg\min_{\mathbf{H}} \lambda_4\|(\mathbf{A}^T\mathbf{X})^T\mathbf{H} - \mathbf{F}\|_F^2. \tag{16}$$

Equation (16) is an unconstrained optimization; we compute its derivative

$$\frac{\partial\Theta(\mathbf{H})}{\partial\mathbf{H}} = 2\lambda_4\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A}\mathbf{H} - 2\lambda_4\mathbf{A}^T\mathbf{X}\mathbf{F}. \tag{17}$$

If $\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A}$ is a singular matrix, $\mathbf{B} = (\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A} + \mu\mathbf{I})^{-1}\mathbf{A}^T\mathbf{X}$; otherwise, $\mathbf{B} = (\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{X}$. μ is a positive constant and $\mathbf{I}$ is a unit matrix. By setting the derivative $\partial\Theta(\mathbf{H})/\partial\mathbf{H} = 0$, we obtain the following equation:

$$\mathbf{H} = \mathbf{B}\mathbf{F}. \tag{18}$$

B. Update G

As **H**, **A**, and **F** are fixed, we rewrite (15) as follows:

$$\Theta(\mathbf{G}) = \arg\min_{\mathbf{G}} \lambda_2\left(\|\mathbf{Q} - \mathbf{G}\mathbf{A}^T\mathbf{X}\|_F^2 + \gamma\|\mathbf{G}\|_2^2\right). \quad (19)$$

Equation (19) is also an unconstrained optimization. We compute its derivative

$$\frac{\partial\Theta(\mathbf{G})}{\partial\mathbf{G}} = 2\lambda_2(\mathbf{G}\mathbf{A}^T\mathbf{X} - \mathbf{Q})\mathbf{X}^T\mathbf{A} + 2\lambda_2\gamma\mathbf{G}. \quad (20)$$

By setting the derivative $\partial\Theta(\mathbf{G})/\partial\mathbf{G} = 0$, we obtain

$$\mathbf{G} = \mathbf{Q}\mathbf{X}^T\mathbf{A}(\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A} + \gamma\mathbf{I})^{-1}. \quad (21)$$

C. Update F

As **H**, **A**, and **G** are fixed, (15) can be rewritten as

$$\begin{aligned}\Theta(\mathbf{F}) &= \arg\min_{\mathbf{F}} \lambda_3\left(\sum_{i,j=1}^{n}(\mathbf{U}_{ij} + \beta\mathbf{V}_{ij})\|\mathbf{F}_i - \mathbf{F}_j\|_2^2 + \sum_{i=1}^{n}\|\mathbf{F}_i - \mathbf{Y}_i\|_2^2\right)\\ &\quad + \lambda_4\|(\mathbf{A}^T\mathbf{X})^T\mathbf{H} - \mathbf{F}\|_F^2\\ &= \arg\min_{\mathbf{F}} \lambda_3(\mathrm{Tr}\,(\mathbf{F}^T\mathbf{L}\mathbf{F}) + \mathrm{Tr}\,((\mathbf{F} - \mathbf{Y})^T\mathbf{S}(\mathbf{F} - \mathbf{Y})))\\ &\quad + \lambda_4\|(\mathbf{A}^T\mathbf{X})^T\mathbf{H} - \mathbf{F}\|_F^2 \end{aligned} \quad (22)$$

where $\mathbf{L} = \mathbf{L}_F + \beta\mathbf{L}_S$; $\mathbf{S} \in \mathbb{R}^{n\times n}$ is a diagonal matrix. If data in **X** are labeled, $\mathbf{S}_{ii} = 1$; otherwise, $\mathbf{S}_{ii} = 0$. We substitute **H** in (18) into (22) and compute the partial derivative of (22) with respect to **F**

$$\frac{\partial\Theta(\mathbf{F})}{\partial\mathbf{F}} = 2\lambda_3(\mathbf{L}\mathbf{F} + \mathbf{S}\mathbf{F} - \mathbf{S}\mathbf{Y}) + 2\lambda_4\mathbf{E}\mathbf{F} \quad (23)$$

where $\mathbf{E} = (\mathbf{X}^T\mathbf{A}\mathbf{B} - \mathbf{I})^T(\mathbf{X}^T\mathbf{A}\mathbf{B} - \mathbf{I})$; by setting the derivative $\partial\Theta(\mathbf{F})/\partial\mathbf{F} = 0$, we get

$$\mathbf{F} = \left(\mathbf{S} + \mathbf{L} + \frac{\lambda_4}{\lambda_3}\mathbf{E}\right)^{-1}\mathbf{S}\mathbf{Y}. \quad (24)$$

D. Update A

As **H**, **F**, and **G** are fixed, (15) can be rewritten as

$$\begin{aligned}\Theta(\mathbf{A}) &= \arg\min_{\mathbf{A}} \frac{1}{nk_1}\sum_{i=1}^{n}\sum_{j=1}^{k_1}\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2\\ &\quad - \alpha\frac{1}{nk_2}\sum_{i=1}^{n}\sum_{j=1}^{k_2}\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2\\ &\quad + \lambda_1\mathrm{Tr}\,[(\mathbf{A}^T\mathbf{X})(\mathbf{L}_F + \beta\mathbf{L}_S)(\mathbf{A}^T\mathbf{X})^T]\\ &\quad + \lambda_2\|\mathbf{Q} - \mathbf{G}\mathbf{A}^T\mathbf{X}\|_F^2\\ &\quad + \lambda_3\sum_{i,j=1}^{n}\mathbf{U}_{ij}\|\mathbf{F}_i - \mathbf{F}_j\|_2^2 + \lambda_4\|(\mathbf{A}^T\mathbf{X})^T\mathbf{H} - \mathbf{F}\|_F^2. \end{aligned} \quad (25)$$

We obtain its partial derivative with respect to **A**

$$\begin{aligned}\frac{\partial\Theta(\mathbf{A})}{\partial\mathbf{A}} &= \frac{1}{nk_1}\sum_{i=1}^{n}\sum_{j=1}^{k_1}2\mathbf{C}\mathbf{A} - \alpha\frac{1}{nk_2}\sum_{i=1}^{n}\sum_{j=1}^{k_2}2\mathbf{C}\mathbf{A}\\ &\quad + \lambda_1\left(-\sum_{i,j=1}^{n}\mathbf{U}_{ij}\mathbf{C}\mathbf{A}\mathbf{R} + \sum_{i,j=1}^{n}(\mathbf{U}_{ij} + \beta\mathbf{V}_{ij})\mathbf{C}\mathbf{A}\right)\\ &\quad + 2\lambda_2\mathbf{X}(\mathbf{X}^T\mathbf{A}\mathbf{G}^T - \mathbf{Q}^T)\mathbf{G}\\ &\quad + \lambda_3\left(-\sum_{i,j=1}^{n}\mathbf{U}_{ij}\mathbf{C}\mathbf{A}\|\mathbf{F}_i - \mathbf{F}_j\|_2^2\right)\\ &\quad + 2\lambda_4\mathbf{X}(\mathbf{X}^T\mathbf{A}\mathbf{H} - \mathbf{F})\mathbf{H}^T \end{aligned} \quad (26)$$

where

$$\mathbf{C} = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \quad (27)$$

$$\mathbf{R} = (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{A}\mathbf{A}^T(\mathbf{x}_i - \mathbf{x}_j). \quad (28)$$

In order to optimize **A**, we compute the function cost of (25) and the gradient utilizing (26). Then, **A** can be updated using the unconstrained optimizer.

E. Initialization of A, G, and H

We first define a weight matrix $\mathbf{W} \subset \mathbb{R}^{n\times n}$, of which the (i, j)th element is computed as $\mathbf{W}_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$, and n is the number of the training data. We compute the average value θ of all the elements in **W**, $\theta = \sum_i\sum_j \mathbf{W}_{ij}/n(n-1)$. Then, the feature transformation matrix **A** is initialized as the following form:

$$\mathbf{A} = \begin{bmatrix} \frac{1}{\theta} & \cdots & 0 & \cdots & 0\\ \vdots & \ddots & \vdots & \vdots & \vdots\\ 0 & 0 & \frac{1}{\theta} & \cdots & 0 \end{bmatrix}.$$

After the initialization of **A**, we initialize **H** and **G**, as denoted in (18) and (21), respectively. The experiments have verified that variables initialized in this way converge faster than the way initialized with random numbers.

As the output of our method is derived (see Algorithm 1), we use transformation matrices **A** and **H** to classify the point cloud

$$l_i = \arg\max_{j}\left(\left(\mathbf{x}_i^T\mathbf{A}\mathbf{H}\right)_{1\times j}\right)(1 \le j \le c). \quad (29)$$

VI. Experimental Results

To validate the performance of our method, we perform both qualitative and quantitative evaluations on the point clouds of four outdoor urban scenes.

A. Data Sets

1) The point cloud of the first scene (scene I) is collected by the TLS scanner in a single scan. Four-class objects, such as cars, trees, pedestrians, and buildings, appear in the urban scene. The scene contains a lot of trees

Algorithm 1 Implementation of the Proposed Method

```
Input: The training matrix $\mathbf{X}$, the group label matrix $\mathbf{Q}$, the label matrix $\mathbf{Y}$, parameters $\alpha$, $\beta$, $\lambda_1$, $\lambda_2$, $\lambda_3$, $\lambda_4$, error $\varepsilon$, the number of iterations $t$ and learning speed $\eta$.
Initialize: Initialize the matrix $\mathbf{A}$, $\mathbf{G}$ and $\mathbf{H}$.
  For number of iterations t.
  $\mathbf{F}_{t+1} = \left(\mathbf{S}_t + \mathbf{L}_t + \frac{\lambda_4}{\lambda_3}\mathbf{E}_t\right)^{-1} \mathbf{S}_t \mathbf{Y}_t$;
  $\mathbf{A}_{t+1} = \mathbf{A}_t - \eta \frac{\partial \Theta(\mathbf{A})}{\partial \mathbf{A}}$;
  $\mathbf{G}_{t+1} = \mathbf{Q}\mathbf{X}^T \mathbf{A}_t \left(\mathbf{A}_t^T \mathbf{X}\mathbf{X}^T \mathbf{A}_t + \gamma \mathbf{I}\right)^{-1}$;
  $\mathbf{H}_{t+1} = \mathbf{B}\mathbf{F}_{t+1}$;
  When convergence criterion satisfied.
  End for
Output: The feature transformation matrix $\mathbf{A}$, the transformation matrix $\mathbf{H}$ and predicted label matrix $\mathbf{F}$.
```

with different shapes, so the point cloud is generally incomplete and noisy because of the severe occlusions caused by the trees or self-occlusion. The point density of the point cloud is uneven and changes with respect to the distance between the objects and the scanner.

2) Semantic-8 data set (scene II) is captured by the Swiss Federal Institute of Technology Zurich using the TLS scanners. The density of the point cloud changes greatly, and it covers a range of diverse objects: terrain, trees, shrubs, buildings, hard scape, and cars, where hard scape is some artifacts.

3) The Oakland 3-D data set [59] (scene III) is scanned by using Navlab11 equipped with side looking SICK LMS laser scanners around CMU campus in Oakland, Pittsburgh, PA, USA.
There are seven-class objects such as cars, trees, ground, buildings, light poles, grass, and shrubs in the scene.

4) The point cloud of scene IV is obtained by the TLS scanner, which contains the same four categories as the point cloud of scene I including car, tree, pedestrian, and building. The data set is mainly used to validate the generalization ability of the proposed method.

The point clouds of scenes I, II, and IV are obtained by TLS scanners. There are huge terrain points in the TLS point clouds. To reduce the computational cost, we remove the terrain points of TLS point clouds by using the mathematical morphological filtering algorithm [60].

The class information of each data set is shown in Table I.

TABLE I
CLASS INFORMATION OF EACH DATA SET

Scenes	Scene I	Scene II	Scene III	Scene IV
Car points	31,026	404,061	39,874	72,835
Tree points	214,151	352,618	186,599	726,200
Pedestrian points	51,163	--	--	32,686
Building points	88,818	1,612,232	63,188	183,483
Shrub points	--	172,100	5,829	--
Hard scape points	--	46,140	--	--
Ground points	--	--	678,165	--
Light poles points	--	--	4,215	--
Grass points	--	--	11,319	--
Total points	385,158	2,587,151	989,189	1,015,204

B. Comparisons With Other Methods

Since our method is semisupervised, we compare it with the following four state-of-the-art semisupervised methods. And we also compare with one supervised method to enhance the essential value of the proposed semisupervised method.

1) *Linear Manifold Regularization for Large-Scale Semisupervised Learning (LapRLS) [51]:* It develops linear LapRLS as promising solutions to efficiently exploit the enormous unlabeled data for learning semisupervised method.

2) *FME:* A framework for semisupervised and unsupervised dimension reduction [52]. It proposes a unified manifold learning framework for dimension reduction by employing a simple but effective linear regression function to map the new data points.

3) *Discriminating Joint Feature Analysis for Multimedia Data Understanding (SFSS) [53]:* It combines $l_{2,1}$-norm regularized feature selection and manifold learning to learn discriminative features.

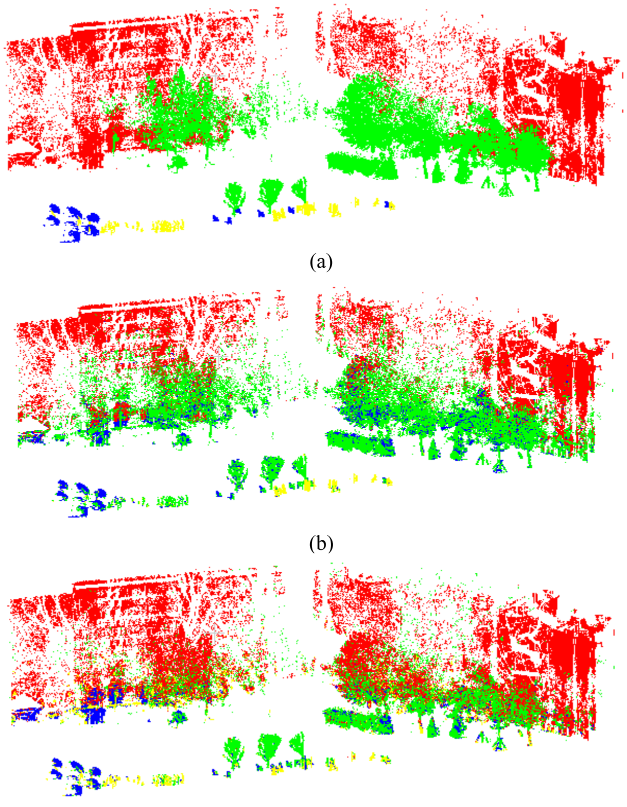
4) *Learning an NNSG for Linear Regression [54]:* In the NNSG, the linear regression and graph learning are simultaneously performed to guarantee an overall optimum.

5) The SVM [61], which is a supervised method.

For the above four semisupervised methods, the nearest neighbor algorithm is used for the semisupervised classification, so the graph Laplacian matrix $\mathbf{L}$ need to be calculated first. For the LapRLS and FME, the weight matrix is calculated using $\mathbf{U}_{ij} = \exp((-\|\mathbf{x}_i - \mathbf{x}_j\|^2)/\sigma)$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are the k-nearest neighbors; otherwise, $\mathbf{U}_{ij} = 0$, where σ and k are the heat kernel and the number of nearest neighbors parameters which are selected from the sets $\{10^{-9}, 10^{-8}, \ldots, 10^8, 10^9\}$ and $\{3, 4, 5, 6, 7, 8, 9, 10\}$, respectively. For SFSS, $\mathbf{U}_{ij} = 1$ if x_i and x_j are k-nearest neighbors. In the NNSG, the sample neighbors and weight matrix can be automatically determined by using a regularization term. In the LapRLS, FME, SFSS, NNSG, SVM, and our method, the parameters are, respectively, selected from the set $\{10^{-9}, 10^{-8}, \ldots, 10^8, 10^9\}$.

For each data set, we randomly select t (percent) of the points per class as the training data and the remaining points are taken as the test data. In the training data, l (percent) of the points per class is randomly selected as the labeled data and the remaining points are utilized as unlabeled data. The test data are used to test the classification performance, and the unlabeled data are utilized for semisupervised classification. For SVM, we only select t per class as the training data and the remaining points are taken as the test data. To validate the performance of our method, we compare the classification results of the six methods in terms of F_1 measure, precision/recall, accuracy, and Cohen's Kappa coefficient.

We set $t = 5\%$ and $l = 30\%$, 50%, and 70%, respectively, in scene I. The classification results of scene I are shown in Table II. Fig. 3 shows the classification results of our

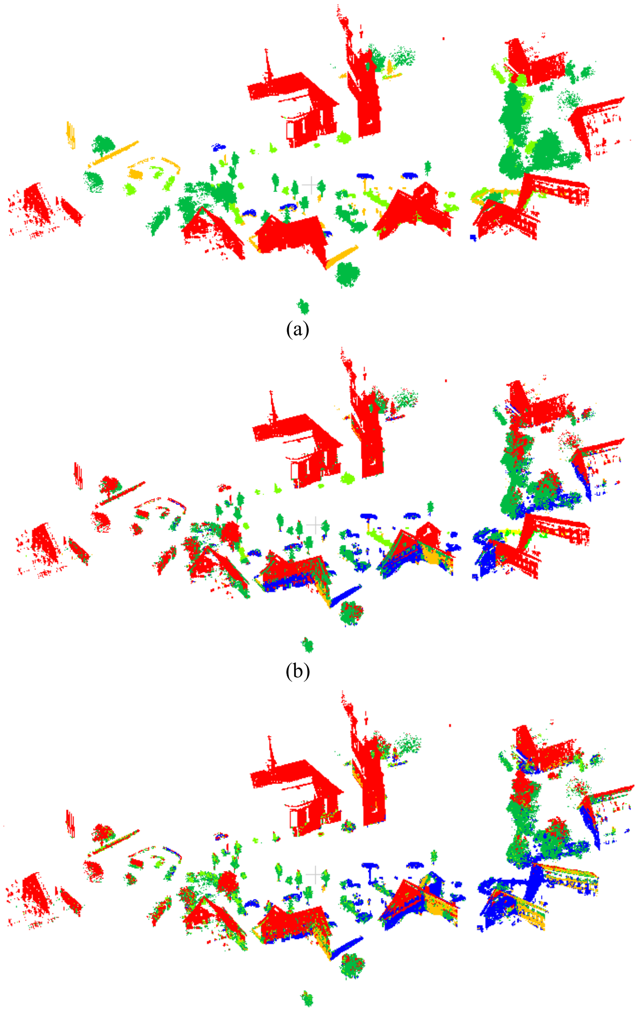


(a)

(b)

(c)

Fig. 3. Classification results of scene I. (a) Ground truth. (b) Our method. (c) FME. Green points represent trees, blue points represent cars, red points represent buildings, and yellow points represent pedestrians.



(a)

(b)

(c)

Fig. 4. Classification results of scene II. (a) Ground truth. (b) Our method. (c) SFSS. Light green points represent shrub, dark green points represent tree, blue points represent cars, red points represent buildings, and brown points represent hard scape.

method and FME in the setting of $t = 5\%$ and $l = 70\%$. It is noted that the classification accuracy of our method is higher than those of other methods on unlabeled data, which explains the graph structure learned by our method can more effectively transmit labels to unlabeled data. For the testing data, our method also obtains higher accuracies. As shown in Fig. 3(b), our method can distinguish these classes of objects very well.

Our method can be applied to the classification of scenes with a large number of points, such as scene II, which contains a total of 2 587 151 points including 404 061 car points, 1 612 232 building points, 352 618 tree points, 172 100 shrub points, and 46 140 hard scape points. We set $t = 1\%$ and $l = 50\%$ for parsing the point cloud of scene II. As shown in Table III, our method obtains higher performance on unlabeled data and testing data compared with other methods. For each class in the scene, our method has higher precisions and recalls in most categories. Compared with the SFSS, our method has a lower precision/recall in classifying hard scape on unlabeled data, but it has higher classification accuracies (AC) on both unlabeled data and testing data. Simultaneously, we notice that the precision/recall per class have the same trend, that is, the precisions and recalls of buildings and cars are higher because of the large number of points, while those of hard scape and shrub are lower. Fig. 4 shows the classification results of our method and the SFSS. It is noted that some hard scape and shrub points are misclassified into buildings due to the insufficient learning during the training stage.

In scene III with seven classes, we set $t = 2\%$ and $l = 30\%$, that is, we select 2% of the points per class as the training data and 30% of the training points per class as the labeled data. The classification results are shown in Table IV and Fig. 5. It is noted that our method also obtains higher accuracies on unlabeled data and test data compared with other methods. The precisions/recalls of the classes with less point in the scene, such as light poles, grass, and shrub, are also lower than other classes. Compared with the supervised method SVM, the classification results of the proposed method are significantly better than those of SVM, which indicates the superiority of the proposed semisupervised method.

C. Comparisons of Independent Constraint and the Joint Constraints

To validate the effectiveness of each constraint term, we compare the independent constraint (IC) and the joint

TABLE II
SEMISUPERVISED CLASSIFICATION ACCURACY (%) OF DIFFERENT METHODS ON SCENE I

Method	*l*=30%		*l*=50%		*l*=70%	
	Unlabeled Data	Test Data	Unlabeled Data	Test Data	Unlabeled Data	Test Data
LapRLS	93.2	68.5	93.7	70.0	94.5	69.9
FME	93.7	71.0	94.1	73.2	95.0	75.6
SFSS	92.6	68.8	93.2	70.1	93.5	71.1
NNSG	92.9	60.1	95.6	61.6	95.9	63.3
Our method	**94.8**	**85.6**	**97.7**	**87.1**	**98.0**	**88.2**

The best results are highlighted in bold.

TABLE III
SEMISUPERVISED CLASSIFICATION RESULTS (PRECISION/RECALL, F1, ACCURACY, KAPPA, AND %) OF DIFFERENT METHODS ON SCENE II

Class	SVM	LapRLS		FME		SFSS		NNSG		Our method	
	Test Data	Unlabeled Data	Test Data	Unlabeled Data	Test Data	Unlabeled Data	Test Data	Unlabeled Data	Test Data	Unlabeled Data	Test Data
Tree	51.1/0.2	86.2/85.9	66.2/52.3	71.2/85.6	66.1/67.2	86.2/86.5	68.4/54.8	90.7/81.7	65.8/47.2	92.0/88.3	70.0/82.9
	0.4	86.0	58.4	77.7	66.6	86.3	60.8	86.0	55.0	90.1	75.9
Shrub	100/0.3	77.6/29.6	29.6/9.6	78.6/34.6	21.5/12.9	77.8/73.6	27.2/16.6	71.9/85.7	21.3/16.6	80.1/86.3	67.1/25.4
	0.6	42.9	14.5	48.1	16.1	75.6	18.7	78.2	18.7	83.1	36.9
Buildings	69.3/99.9	92.7/95.6	93.9/75.4	95.2/92.0	97.5/66.6	97.8/98.2	94.4/75.1	96.8/95.3	95.4/69.9	98.6/99.2	89.9/96.1
	81.8	94.1	83.6	93.6	79.1	98.0	83.7	96.0	80.7	98.8	92.9
Hard scape	100/0.2	94.6/93.1	8.2/18.1	92.8/92.0	8.5/23.7	95.3/93.6	9.4/21.5	92.6/91.3	6.4/21.8	94.2/93.3	88.1/13.9
	0.4	93.8	11.3	92.4	12.5	94.4	13.1	91.9	9.9	93.7	24.0
Cars	99.9/64.0	94.5/95.0	46.9/98.9	97.3/96.6	41.2/97.8	95.3/94.9	46.7/99.0	96.5/93.6	45.6/97.8	97.5/97.1	87.4/80.8
	78.0	94.7	63.6	96.9	58.0	95.1	63.5	95.0	62.2	97.2	84.0
Accuracy	72.4	91.8	70.6	89.6	66.8	92.6	70.8	92.2	69.0	**94.8**	**85.8**
Kappa	0.347	0.815	0.525	0.792	0.499	0.900	0.537	0.868	0.485	**0.936**	**0.735**

The best results are highlighted in bold.

TABLE IV
SEMISUPERVISED CLASSIFICATION RESULTS (PRECISION/RECALL, F1, ACCURACY, KAPPA, AND %) OF DIFFERENT METHODS ON SCENE III

Class	SVM	LapRLS		FME		SFSS		NNSG		Our method	
	Test Data	Unlabeled Data	Test Data	Unlabeled Data	Test Data	Unlabeled Data	Test Data	Unlabeled Data	Test Data	Unlabeled Data	Test Data
Cars	99.8/1.4	88.1/79.2	37.1/61.0	85.9/78.8	36.0/61.0	87.2/79.6	37.4/62.0	86.7/81.4	27.6/66.0	91.0/82.6	99.9/61.9
	2.8	83.4	46.1	82.2	45.3	83.2	46.7	84.0	38.9	86.6	76.4
Tree	36.1/99.7	74.6/93.4	81.4/73.3	67.1/96.1	75.7/79.4	75.3/93.2	81.8/74.5	80.6/87.1	84.6/62.6	82.7/97.8	68.5/97.0
	53.0	82.9	77.1	79.0	77.5	83.3	78.0	83.7	72.0	89.6	80.3
Ground	99.9/69.6	98.6/96.8	95.8/85.4	98.1/95.3	95.5/86.1	96.7/97.5	96.9/85.5	98.9/93.4	96.1/69.9	96.1/98.2	96.5/98.8
	82.0	97.7	90.3	96.7	90.6	97.1	90.8	96.1	80.9	97.1	97.6
Buildings	63.2/0.4	97.3/92.4	81.2/37.5	98.0/90.1	84.9/38.1	97.7/93.1	81.3/37.9	96.8/94.2	79.0/35.0	93.9/90.7	88.8/42.0
	0.8	94.8	51.3	93.9	52.6	95.3	51.7	95.5	48.5	92.3	57.0
Light poles	3.2/2.9	87.4/75.8	26.4/55.3	83.0/66.6	24.8/51.7	86.6/78.0	26.1/55.9	78.5/82.9	19.6/56.4	89.8/79.6	42.3/59.1
	3.0	81.2	35.7	73.9	33.5	82.1	35.6	80.6	29.1	84.4	49.3
Grass	8.6/9.1	83.1/84.1	18.9/83.3	81.5/76.5	19.8/77.6	83.7/86.0	29.2/65.7	65.1/85.3	15.6/80.1	85.2/89.5	43.7/85.9
	8.8	83.6	30.8	78.9	31.5	84.8	40.4	73.8	26.1	87.3	57.9
Shrub	5.6/5.2	75.5/24.3	12.6/18.6	50.0/12.3	13.2/8.3	78.5/23.3	12.4/18.4	65.4/51.0	18.8/33.8	79.8/56.3	36.7/31.2
	5.4	36.8	15.0	19.7	10.2	35.9	14.8	57.3	24.2	66.0	33.7
Accuracy	66.7	88.9	78.8	86.4	79.8	89.2	79.0	88.2	75.3	**95.6**	**90.8**
Kappa	0.421	0.888	0.605	0.869	0.626	0.896	0.611	0.832	0.456	**0.928**	**0.842**

The best results are highlighted in bold.

constraints for point cloud classification. In the proposed method, the joint objective function can be divided into the following learning models.

1) *IC1:* IC1 is equivalent to the margin constraint, which is proposed to learn a novel distance criterion that can provide a tradeoff between intraclass compactness and

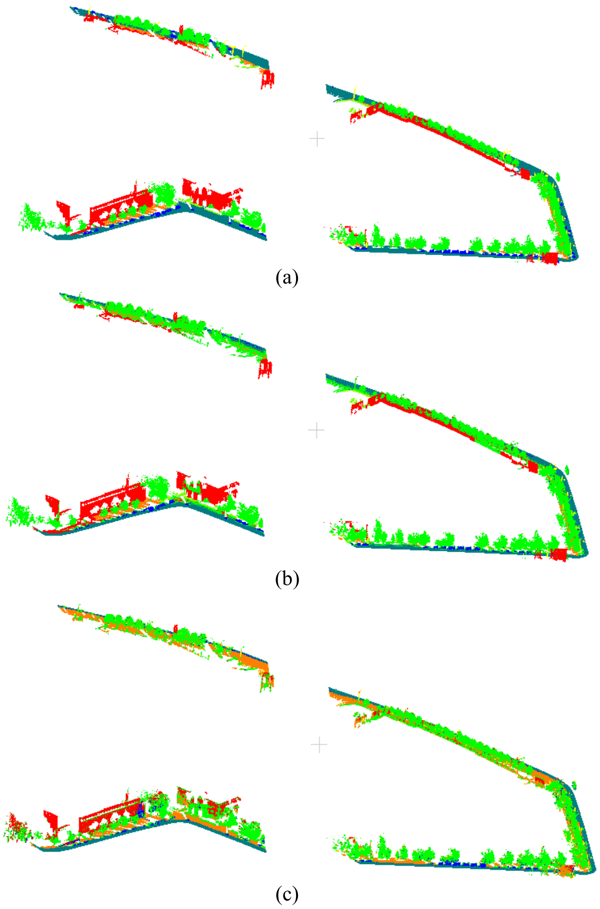


Fig. 5. Classification results of scene III. (a) Ground truth. (b) Our method. (c) FME. Light green points represent shrub, dark green points represent trees, dark cyan points represent ground, blue points represent cars, red points represent buildings, yellow points represent light poles, and brown points represent grass.

TABLE V
SEMISUPERVISED CLASSIFICATION ACCURACY (%) OF SCENE I

Method	Unlabeled Data	Test Data
IC1	91.1	80.6
IC2	92.9	82.8
IC3	94.6	83.5
Our method	**97.7**	**87.1**

The best results are highlighted in bold.

TABLE VI
SEMISUPERVISED CLASSIFICATION ACCURACY (%) OF SCENE II

Method	Unlabeled Data	Test Data
IC1	91.3	78.0
IC2	92.1	78.9
IC3	93.0	80.1
Our method	**94.8**	**85.8**

The best results are highlighted in bold.

TABLE VII
SEMISUPERVISED CLASSIFICATION ACCURACY (%) OF SCENE III

Method	Unlabeled Data	Test Data
IC1	90.9	85.6
IC2	92.6	86.1
IC3	93.8	87.5
Our method	**95.6**	**90.8**

The best results are highlighted in bold.

interclass separability. IC1 can obtain the discriminative features by learning the feature transformation matrix

$$\min_{\mathbf{A}} \frac{1}{nk_1}\sum_{i=1}^{n}\sum_{j=1}^{k_1}\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2 - \alpha\frac{1}{nk_2}\sum_{i=1}^{n}\sum_{j=1}^{k_2}\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2. \tag{30}$$

2) *IC2:* To ensure the local consistency of neighboring points, IC2 adds the cograph regularization term to the objective function based on IC1 by integrating the manifolds of the feature space and spatial distribution

$$\min_{\mathbf{A}} \frac{1}{nk_1}\sum_{i=1}^{n}\sum_{j=1}^{k_1}\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2 - \alpha\frac{1}{nk_2}\sum_{i=1}^{n}\sum_{j=1}^{k_2}\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2 + \frac{1}{2}\lambda_1\sum_{i,j=1}^{n}(\mathbf{U}_{ij} + \beta\mathbf{V}_{ij})\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2. \tag{31}$$

3) *IC3:* Based on the IC2, IC3 further adds label consistency constraint on grouped points to obtain the label consistency and the discriminative distance metric

$$\min_{\mathbf{A}} \frac{1}{nk_1}\sum_{i=1}^{n}\sum_{j=1}^{k_1}\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2 - \alpha\frac{1}{nk_2}\sum_{i=1}^{n}\sum_{j=1}^{k_2}\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2 + \frac{1}{2}\lambda_1\sum_{i,j=1}^{n}(\mathbf{U}_{ij} + \beta\mathbf{V}_{ij})\|\mathbf{A}^T\mathbf{x}_i - \mathbf{A}^T\mathbf{x}_j\|_2^2 + \lambda_2\left(\|\mathbf{Q} - \mathbf{G}\mathbf{A}^T\mathbf{X}\|_F^2 + \gamma\|\mathbf{G}\|_2^2\right). \tag{32}$$

For IC1, IC2, and IC3, the following classification function is learned separately from the feature learning process for semisupervised classification

$$\min_{\mathbf{F},\mathbf{H}} \sum_{i,j=1}^{n}(\mathbf{U}_{ij} + \beta\mathbf{V}_{ij})\|\mathbf{F}_i - \mathbf{F}_j\|_2^2 + \sum_{i=1}^{n}\|\mathbf{F}_i - \mathbf{Y}_i\|_2^2 + \lambda_4\|(\mathbf{A}^T\mathbf{X})^T\mathbf{H} - \mathbf{F}\|_F^2. \tag{33}$$

We set $t = 5\%$ and $l = 50\%$ for scene I, $t = 1\%$ and $l = 50\%$ for scene II, and $t = 2\%$ and $l = 30\%$ for scene III. The classification accuracy of unlabeled data and testing data is reported in Tables V–VII.

From Tables V–VII, we have the following observations.

1) The semisupervised classification results of IC2 are better than those of IC1. The main reason is that

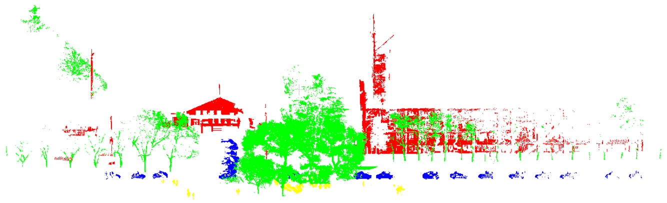


Fig. 6. Point cloud of scene IV.

TABLE VIII

SEMISUPERVISED CLASSIFICATION ACCURACY (%) OF DIFFERENT METHODS ON SCENE IV

Method	Scene IV
SVM	60.56
LapRLS	51.53
FME	24.30
SFSS	42.61
NNSG	24.20
Our method	**71.53**

The best results are highlighted in bold.

the cograph regularization in IC2 can ensure the local consistency and preserve the local geometric structure of points.

2) IC3 outperforms IC2 because the label consistency constraint in IC3 can maintain the same labels for the points with the same category before and after feature transformation.

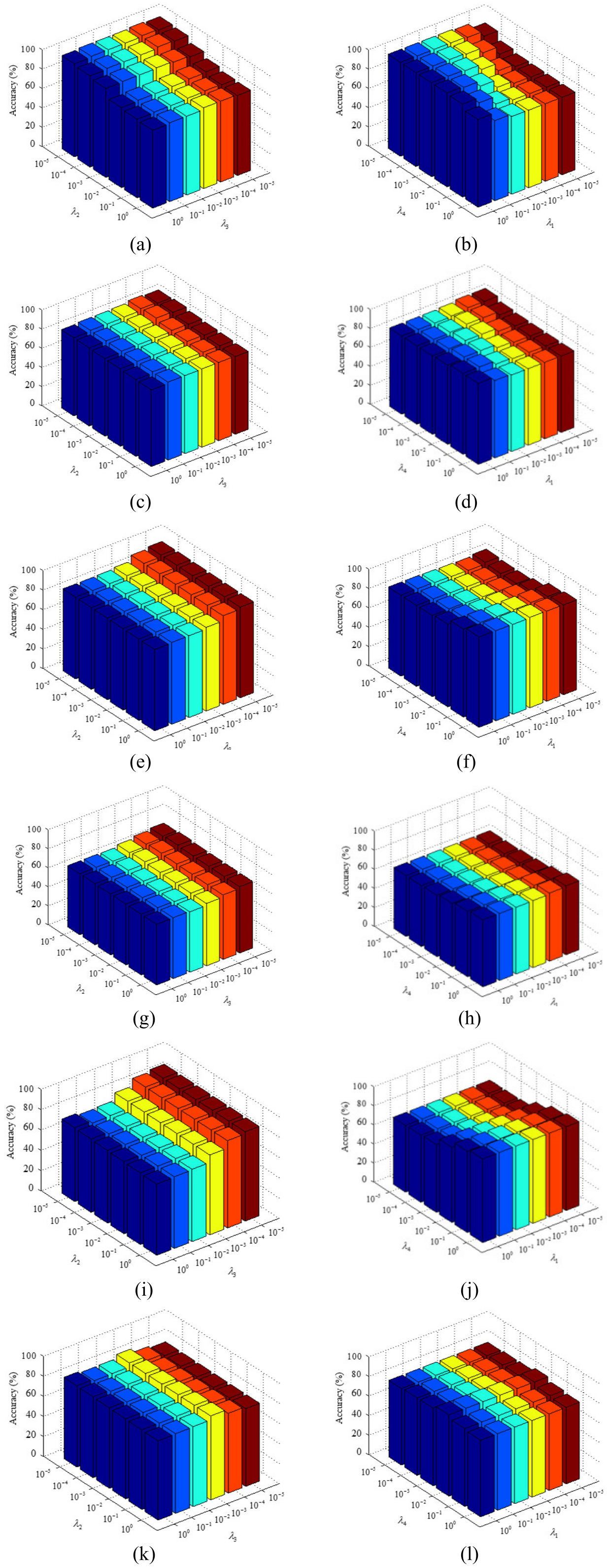
3) Compared with IC1, IC2 and IC3, our method obtains the best classification performance owing to the joint learning framework.

D. Generalization

To validate the generalization ability of the proposed method, we have performed the experiments on the point cloud data set of Scene IV as shown in Fig. 6. The methods are trained on the point cloud of Scene I with $t = 5\%$, $l = 50\%$ and are used to classify Scene IV without training again. The classification results are listed in Table VIII. It is noted that the proposed method achieves the highest classification results compared with other methods, which indicates that our method has better generalization ability and applicability.

E. Sensitivity of the Parameters

In our method, six parameters α, β, λ_1, λ_2, λ_3, and λ_4 need to be tuned in each data set. We set $\alpha = 0.3$ and $\beta = 10$ and mainly discuss the impact of different parameters λ_1, λ_2, λ_3, and λ_4 on the classification results. And we analyze λ_2 and λ_3 together to discuss the relationship of label consistency constraint on grouped points and single point. Fig. 7 illustrates that the AC vary with different parameters in the three data sets.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Fig. 7. AC variation of the parameters λ_1, λ_2, λ_3 and λ_4. (a) and (b) Unlabeled data on scene I. (c) and (d) Test data on scene I. (e) and (f) Unlabeled data on scene II. (g) and (h) Test data on scene II. (i) and (j) Unlabeled data on scene III. (k) and (l) Test data on scene III.

It is observed that the highest AC for unlabeled data is obtained with small λ_2 and λ_3 as other parameters are fixed. The classification performance for testing data changes little. Through adjusting the parameters λ_2 and λ_3, the best classification results for unlabeled data are achieved when the two parameters are close to 10^{-4}, which demonstrates that the

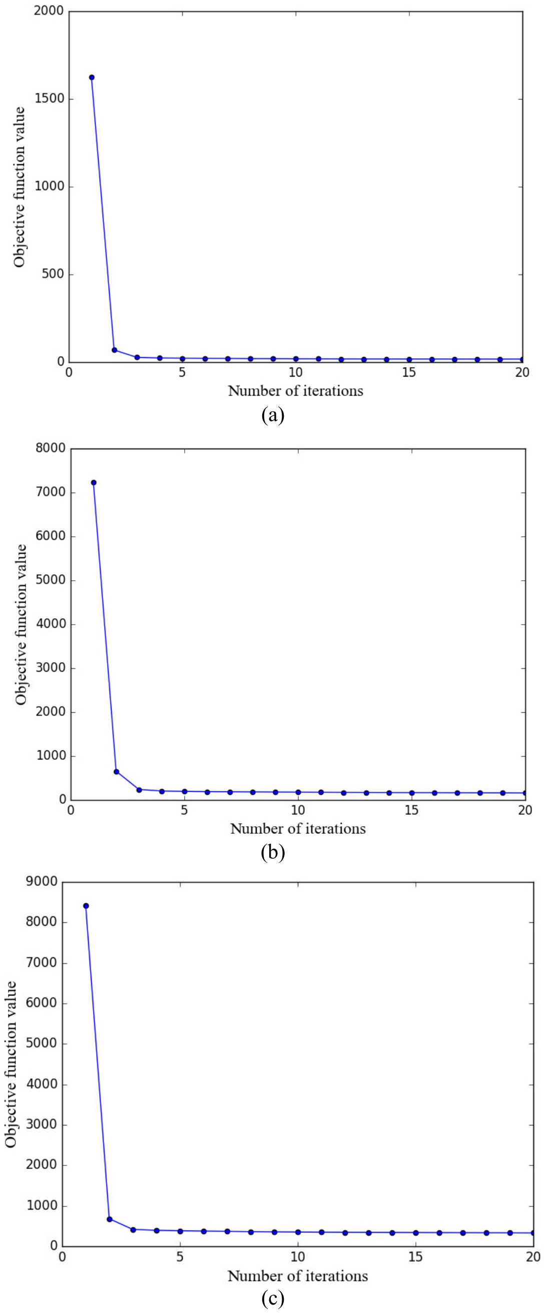


Fig. 8. Convergence processes of different data sets. (a) Scene I. (b) Scene II. (c) Scene III.

terms in (6) corresponding to λ_2 and λ_3 are less significant to learn the graph structure. It is noted that the classification results are best for unlabeled data and test data with other parameters are fixed when λ_1 is close to 1. It demonstrates that the second term corresponding to λ_1 in (6) is more significant to learn a model for fitting labels and classifying new test data.

F. Algorithmic Convergence

In the process of solving the variables $\mathbf{H}$, $\mathbf{A}$, $\mathbf{G}$, and $\mathbf{F}$, an iterative algorithm is adopted to optimize the variables due to the highly nonlinear nature of (15). Here, we prove the objective function can converge to a local optimum quickly by using the proposed update rule. The functions of optimizing $\mathbf{H}$, $\mathbf{G}$, and $\mathbf{F}$ are convex, so they are convergent. The process of optimizing $\mathbf{A}$ makes the objective function achieve a local minimum as other variables are fixed. The convergence process on different data sets is shown in Fig. 8. It can be seen that our algorithm converges within less than four iterations, which demonstrates that the proposed update rule is very effective.

VII. Conclusion

In this paper, we have proposed a semisupervised learning method that simultaneously joints the margin constraint, cograph constraint, and adaptive label consistency constraint for point cloud classification. The margin constraint can learn a novel distance criterion to effectively recognize points of different classes. The cograph is constructed by integrating the manifolds of the point features and spatial distributions to characterize the geometry structure and ensure the local consistency of neighboring points. The proposed method considers not only the intraclass compactness and interclass separability of labeled points, but also the label consistency on grouped points and single point of unlabeled data. To solve the objective function, an effective and efficient optimization algorithm is developed, and the convergence is very fast. Experiments performed on four scene point clouds clearly confirmed that our method outperforms the state-of-the-art algorithms.

In the future work, we will combine the distance metric learning and deep learning to learn a deep metric network for further enhancing the performance of point cloud classification.

References

[1] D. Gonzalez-Aguilera, E. Crespo-Matellan, D. Hernandez-Lopez, and P. Rodriguez-Gonzalvez, "Automated urban analysis based on LiDAR-derived building models," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 3, pp. 1844–1851, Mar. 2013.

[2] J. Mei, L. Zhang, S. Wu, Z. Wang, and L. Zhang, "3D tree modeling from incomplete point clouds via optimization and L_1-MST," *Int. J. Geograph. Inf. Sci.*, vol. 31, no. 5, pp. 999–1021, Nov. 2016.

[3] Z. Li, L. Zhang, R. Zhong, T. Fang, L. Zhang, and Z. Zhang, "Classification of urban point clouds: A robust supervised approach with automatically generating training data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 3, pp. 1207–1220, Mar. 2017.

[4] Q. Zhu, Y. Li, H. Hu, and B. Wu, "Robust point cloud classification based on multi-level semantic relationships for urban scenes," *ISPRS J. Photogramm. Remote Sens.*, vol. 129, pp. 86–102, Jul. 2017.

[5] Y. Liu and X. Li, "Domain adaptation for land use classification: A spatio-temporal knowledge reusing method," *ISPRS J. Photogramm. Remote Sens.*, vol. 98, pp. 133–144, Dec. 2014.

[6] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[7] H. Zhang, J. Wang, T. Fang, and L. Quan, "Joint segmentation of images and scanned point cloud in large-scale street scenes with low-annotation cost," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4763–4772, Nov. 2014.

[8] Z. Zhang, L. Zhang, X. Tong, B. Guo, L. Zhang, and X. Xing, "Discriminative dictionary learning-based multi-level point-cluster features for ALS point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7309–7322, Dec. 2016.

[9] L. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Hyperspectral remote sensing image subpixel target detection based on supervised metric learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4955–4965, Aug. 2014.

[10] C. Ma, X. Li, C. Notarnicola, S. Wang, and W. Wang, "Uncertainty quantification of soil moisture estimations based on a Bayesian probabilistic inversion," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3194–3207, Jun. 2017.

[11] X. Xu, X. Li, X. Liu, H. Shen, and Q. Shi, "Multimodal registration of remotely sensed images based on Jeffrey's divergence," *ISPRS J. Photogramm. Remote Sens.*, vol. 122, pp. 97–115, Dec. 2016.

[12] D. Chen, J. Peethambaran, and Z. Zhang, "A supervoxel-based vegetation classification via decomposition and modelling of full-waveform airborne laser scanning data," *Int. J. Remote Sens.*, vol. 39, no. 9, pp. 2937–2968, Jan. 2018.

[13] Z. Wang *et al.*, "A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2409–2425, May 2015.

[14] L. Zhang, C. Wu, and B. Du, "Automatic radiometric normalization for multi-temporal remote sensing imagery with iterative slow feature analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6141–6155, Oct. 2014.

[15] H. Hu, Q. Zhu, Z. Du, Y. Zhang, and Y. Ding, "Reliable spatial relationship constrained feature point matching of oblique aerial images," *Photogramm. Eng. Remote Sens.*, vol. 81, no. 1, pp. 49–58, Jan. 2015.

[16] C.-H. Lin, J.-Y. Chen, P.-L. Su, and C.-H. Chen, "Eigen-feature analysis of weighted covariance matrices for LiDAR point cloud classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 94, pp. 70–79, Aug. 2014.

[17] M. Weinmann, B. Jutzi, and C. Mallet, "Feature relevance assessment for the semantic interpretation of 3D point cloud data," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. II-5/W2, pp. 313–318, Oct. 2013.

[18] K. F. West, B. N. Webb, J. R. Lersch, S. Pothier, J. M. Triscari, and A. E. Iverson, "Context-driven automated target detection in 3D data," *Proc. SPIE*, vol. 5426, pp. 133–143, Sep. 2004.

[19] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.

[20] D. Huber, A. Kapuria, R. Donamukkala, and M. Hebert, "Parts-based 3D object classification," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jul. 2004, pp. II-82–II-89.

[21] A. Johnson, O. Carmichael, D. Huber, and M. Hebert, "Toward a general 3-d matching engine: Multiple models, complex scenes, and efficient data filtering," in *Proc. Robot. Inst.*, vol. 431. 1998, pp. 1097–1107.

[22] J. J. Caceres and K. C. Slatton, "Improved classification of building infrastructure from airborne Lidar data using spin images and fusion with ground-based Lidar," in *Proc. IEEE Urban Remote Sens. Joint Event*, Jun. 2007, pp. 1–7.

[23] A. Martinović, J. Knopp, H. Riemenschneider, and L. Van Gool, "3D all the way: Semantic segmentation of urban scenes from start to end in 3D," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4456–4465.

[24] Z. Wang *et al.*, "A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification," *IEEE Trans. Geosci. Remote Sens*, vol. 53, no. 5, pp. 2409–2425, May 2015.

[25] Z. Zhang *et al.*, "A multi-level point cluster-based discriminative feature for ALS point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3309–3321, Jan. 2016.

[26] M. Gerke and J. Xiao, "Fusion of airborne laserscanning point clouds and images for supervised and unsupervised scene classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 78–92, Jan. 2014.

[27] B. Yang, Z. Dong, Y. Liu, F. Liang, and Y. Wang, "Computing multiple aggregation levels and contextual features for road facilities recognition using mobile laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 126, pp. 180–194, Apr. 2017.

[28] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proc. Eur. Conf. Comput. Vis.*, vol. 3023. 2004, pp. 224–237.

[29] J. Rau, J. Jhan, and Y. Hsu, "Analysis of oblique aerial images for land cover and point cloud classification in an urban environment," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1304–1319, 2015.

[30] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1912–1920.

[31] H. Guan, Y. Yu, Z. Ji, J. Li, and Q. Zhang, "Deep learning-based tree classification using mobile LiDAR data," *Remote Sens. Lett.*, vol. 6, no. 11, pp. 864–873, Sep. 2015.

[32] D. Prokhorov, "A convolutional learning system for object classification in 3-D lidar data," *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 858–863, May 2010.

[33] D. Maturana and S. Scherer, "3D convolutional neural networks for landing zone detection from LiDAR," in *Proc. ICRA*, May 2015, pp. 3471–3478.

[34] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IROS*, Oct. 2015, pp. 922–928.

[35] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view cnns for object classification on 3D data," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5648–5656.

[36] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 77–85.

[37] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using minimum spanning forest grown from automatically selected markers," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 5, pp. 1267–1279, Oct. 2010.

[38] A. Golovinskiy, V. G. Kim, and T. Funkhouser, "Shape-based recognition of 3D point clouds in urban environments," in *Proc. IEEE Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep. 2009, pp. 2154–2161.

[39] E. H. Lim and D. Suter, "3D terrestrial LIDAR classifications with supervoxels and multi-scale conditional random fields," *Comput.-Aided Des.*, vol. 41, no. 10, pp. 701–710, 2009.

[40] B. Guo, X. Huang, F. Zhang, and G. Sohn, "Classification of airborne laser scanning data using JointBoost," *ISPRS J. Photogramm. Remote Sens.*, vol. 100, pp. 71–83, Feb. 2015.

[41] C. Mallet, F. Bretar, and U. Soergel, "Analysis of full-waveform lidar data for classification of urban areas," *Photogramm. Fernerkundung GeoInf.*, vol. 5, pp. 337–349, May 2008.

[42] R. G. Negri, L. V. Dutra, and S. J. S. Sant'Anna, "An innovative support vector machine based method for contextual image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 241–248, Jan. 2014.

[43] S. K. Lodha, E. J. Kreps, D. P. Helmbold, and D. Fitzpatrick, "Aerial LiDAR data classification using support vector machines (SVM)," in *Proc. 3rd Int. Symp. 3D Data Process., Vis., Transmiss.*, Jun. 2006, pp. 567–574.

[44] J. García-Gutiérrez, D. Mateos-García, M. Garcia, and J. C. Riquelme-Santos, "An evolutionary-weighted majority voting and support vector machines applied to contextual classification of LiDAR and imagery data fusion," *Neurocomputing*, vol. 163, no. 2, pp. 17–24, Sep. 2015.

[45] J. Niemeyer, F. Rottensteiner, and U. Soergel, "Contextual classification of lidar data and building object detection in urban areas," *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 152–165, Jan. 2014.

[46] F. Lafarge and C. Mallet, "Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation," *Int. J. Comput. Vis.*, vol. 99, no. 1, pp. 69–85, Aug. 2012.

[47] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. ICML*, 1999, pp. 200–209.

[48] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, 1998, pp. 92–100.

[49] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–7.

[50] X. Fang, Y. Xu, X. Li, Z. Fan, H. Liu, and Y. Chen, "Locality and similarity preserving embedding for feature selection," *Neurocomputing*, vol. 128, pp. 304–315, Mar. 2014.

[51] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.

[52] F. Nie, D. Xu, I. W. Tsang, and C. Zhang, "Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1921–1932, Jul. 2010.

[53] Z. Ma, F. Nie, Y. Yang, J. R. R. Uijlings, N. Sebe, and A. G. Hauptmann, "Discriminating joint feature analysis for multimedia data understanding," *IEEE Trans. Multimedia*, vol. 14, no. 6, pp. 1662–1672, Dec. 2012.

[54] X. Fang, Y. Xu, X. Li, Z. Lai, and W. K. Wong, "Learning a nonnegative sparse graph for linear regression," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2760–2771, Sep. 2015.

[55] X. Li *et al.*, "Toward an improved data stewardship and service for environmental and ecological science data in West China," *Int. J. Digit. Earth*, vol. 4, no. 4, pp. 347–359, Jul. 2011.

[56] L. Zhang, Q. Zhang, B. Du, X. Huang, Y. Y. Tang, and D. Tao, "Simultaneous spectral-spatial feature selection and extraction for hyperspectral images," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 16–28, Jan. 2018.

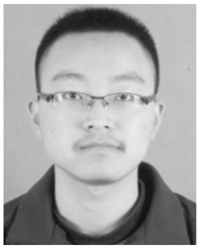
[57] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14. 2002, pp. 585–591.

[58] Y. Y. Tang, H. Yuan, and L. Li, "Manifold-based sparse representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 12, pp. 7606–7618, Dec. 2014.

[59] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin Markov networks," in *Proc. CVPR*, Jun. 2009, pp. 975–982.

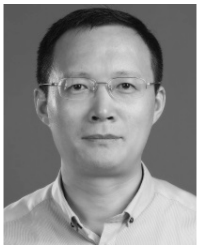
[60] D. Chen, L. Zhang, Z. Wang, and H. Deng, "A mathematical morphology-based multi-level filter of LiDAR data for generating DTMs," *Sci. China Inf. Sci.*, vol. 56, no. 10, pp. 1–14, Oct. 2013.

[61] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.



Jie Mei is currently pursuing the master's degree with the Faculty of Geographical Science, Beijing Normal University, Beijing, China.

His research interests include remote sensing image processing and image-based and LiDAR-based segmentation and reconstruction.

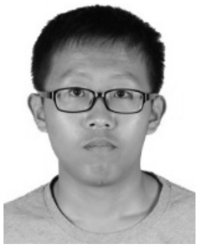


Liqiang Zhang received the Ph.D. degree in geoinformatics from the Institute of Remote Sensing Applications, Chinese Academy of Science, Beijing, China, in 2004.

He is currently a Professor with the Faculty of Geographical Science, Beijing Normal University, Beijing. His research interests include remote sensing image processing, 3-D urban reconstruction, and spatial object recognition.

Yuebin Wang received the Ph.D. degree from the School of Geography, Beijing Normal University, Beijing, China, in 2016.

He is currently a Post-Doctoral Researcher with the School of Mathematical Sciences, Beijing Normal University. His research interests include remote sensing imagery processing and 3-D urban modeling.



Zidong Zhu is currently pursuing the bachelor's degree with the Faculty of Geographical Science, Beijing Normal University, Beijing, China.

His research interests include remote sensing image processing and image-based segmentation.

Huiqian Ding is currently pursuing the master's degree with the Faculty of Geographical Science, Beijing Normal University, Beijing, China.

Her research interests include remote sensing image processing and LiDAR-based segmentation and reconstruction