MUCD: Unsupervised Point Cloud Change Detection via Masked Consistency

Yue Wu[1,2], Zhipeng Wang[1,2], Yongzhe Yuan[1,2], Maoguo Gong[1,3*], Hao Li[1,4], Mingyang Zhang[1,4], Wenping Ma[5], Qiguang Miao[1,2]

[1]MoE Key Lab of Collaborative Intelligence Systems, Xidian University
[2]School of Computer Science and Technology, Xidian University
[3]Academy of Artificial Intelligence, College of Mathematics Science, Inner Mongolia Normal University
[4]School of Electronic Engineering, Xidian University
[5]School of Artificial Intelligence, Xidian University
{ywu@, zpwang01@stu., yyz@stu., haoli@, myzhang@, wpma@mail., qgmiao@}xidian.edu.cn, gong@ieee.org

Abstract

3D Change Detection (3DCD) has gradually become another research hotspot after image change detection. Recent works focus on using artificial labels for supervised or weakly-supervised training of siamese networks to segment changed points. However, labeling every points of multi-temporal point clouds is very expensive and time-consuming. In addition, these works lack effective self-supervised signals, and existing self-supervised signals often fail to capture sufficiently rich change information. To solve this problem, we assume that the powerful representation of 3D objects should model the consistency information of unchanged regions and distinguish different objects. Based on this assumption, we propose a new unsupervised framework called MUCD to learn change information of multi-temporal point clouds through bidirectional optimization of change segmentor and feature extractor. The training of network is divided into two stages. We first design a foreknowledge point contrastive loss based on the characteristics of the 3DCD task to initialize the feature extractor, and then propose a masked consistency loss to further learn the shared geometric information of unchanged regions in the multi-temporal point clouds, utilizing it as a free and powerful supervised signal to train a change segmentor. In the inference stage, only the segmentor is used to take multi-temporal point clouds as input and produce change segmentation result. Extensive experiments are conducted on SLPCCD and Urb3DCD, two real-world datasets of streets and urban buildings, to verify that our proposed unsupervised method is highly competitive and even outperforms supervised methods in scenes where semantic information changes occur, exhibiting better performance in generalization ability and robustness.

Introduction

Change detection (CD) is one of the earliest and most widely used technologies in the field of remote sensing (Hussain et al. 2013; Zhang et al. 2021a; Benedek, Descombes, and Zerubia 2011; Wu, Du, and Zhang 2023; Wu et al. 2021b). The purpose of CD is to discover landscape changes from multi-temporal images observed at same location and different times. CD task has been widely applied in land-use/land-cover change analysis (Jin et al. 2017; Zhu and Woodcock 2014), urban study, environmental monitoring (Khan et al. 2017), and disaster assessment (Sakurada, Okatani, and Deguchi 2013).

In 2D image change detection, nuisance changes such as viewpoint and lighting differences are critical obstacles that have received widespread attention (Wang et al. 2021; Chen et al. 2020). Due to the rapid development of 3D acquisition devices such as LiDAR (Guo et al. 2020), a large amount of 3D data can be collected and recorded with a lower cost (Yuan et al. 2024a). Point cloud data, as a basic data representation for many 3D task applications, can capture rich geometric and appearance information of objects and scenes, and has the advantages of not being affected by perspective distortion and lighting changes. The above advantages avoid some tricky problems in 2D image CD and provide a more intuitive description of the positional relationship between objects and the surrounding environment. This fact has attracted more attention to point cloud data in 3DCD task, and the analysis of multi-temporal point clouds changes have become more urgent (Stilla and Xu 2023). In recent years, some researchers have successfully applied deep learning networks to 3DCD task and collect some available real-world scene datasets.

However, the success of all existing methods is based on fully supervised or weakly-supervised learning, requiring extensive point-level labels. Specifically, these methods use labels or clustering pseudo labels to train a siamese point cloud segmentation network to complete point cloud change detection task. In addition, due to the supervised training on specific datasets with labels, these networks exhibit limited generalization ability and are unable to effectively detect scenes involving semantic information changes.

In this paper, we attempt to abandon the method of training change segmentor with labels and use the feature consistency of unchanged regions as self-supervised signals to learn their shared shape information. Our assumption is that the shape information and contextual relationships of the unchanged regions in point cloud scenes at different times exhibit consistency, while the change regions have significant differences, which can guide a better change segmentation method. Based on this assumption, we design a bidirectional optimization of segmentor and feature extractor, where the feature extractor learns the shared shape information of unchanged regions to guide the segmentor, and

*Corresponding author.

the segmentor masks the changed regions to help the feature extractor learn better. The training process is divided into two stages. In the first stage, we initialize the feature extractor through the self-reconstruction task and the proposed foreknowledge point contrastive loss. We find that using only the self-reconstruction task as a self-supervised signal results in extracted feature differences of multi-temporal point clouds tending towards the mean, which is detrimental to distinguish changed and unchanged scenes. To address this problem, we propose a novel contrastive loss that takes the ground points in the same scene as positive samples and the non-ground points in different scenes as negative samples, which allows feature extractor to extract more robust features from multi-temporal point clouds. In the second stage, we propose a simple masked consistency objective as a new pretext task for point cloud change detection. Specifically, given the input point cloud, we use a siamese network and nearest neighbor feature fusion to obtain a segmentation map. The segmentation map is used to mask the impact of changed points on feature consistency task, allowing the feature extractor to focus more on shared features of unchanged regions. Finally, in the inference stage, we use the segmentor and threshold segmentation to obtain the final change segmentation result.

The main contributions are summarized as follows:

- To our knowledge, we are the first to propose an unsupervised point cloud change detection method, which trains a change segmentor through masked consistency. Our method is efficient, using a simple network structure and has high generalization ability.
- Our proposed foreknowledge point contrastive loss, as a new self-supervised signal, can effectively assist in feature extraction for point cloud change detection task at the beginning of training, and is more robust than self-reconstruction task.
- Our MUCD demonstrates that a simple architecture based solely on coordinate information can achieve highly competitive results, which indicates that the method can serve as a potential common framework for unsupervised point cloud change detection.

Related Work

Deep Learning on Point Cloud Change Detection

The main idea of point cloud change detection focuses on learning geometric change information from spatial location. Nagy *et al.* (Nagy, Kovács, and Benedek 2021) propose a 2D convolutional feature difference network using ChangeGAN, where the author project the point cloud onto the distance image for 2D convolution operations. Schauer *et al.* (Schauer and Nüchter 2018) voxelate point cloud and apply occupancy recognition to detect changed regions. Recent works attempt to directly process point cloud data using deep learning methods. For example, Ku *et al.* (Ku et al. 2021) propose a network SiamGCN based on graph convolutional networks to recognize scene level change classification. Wang *et al.* (Wang et al. 2023) propose a siamese network based on various popular point cloud encoders and improve the change segmentation effect using methods such as feature embedding. De Gélis *et al.* (de Gélis, Lefèvre, and Corpetti 2023) use deep clustering methods to train a siamese network for change segmentation. These works use various siamese networks to learn change information using labels or pseudo labels. Our method completes 3DCD task directly through the geometric similarity of multi-temporal point clouds without labels.

Deep Learning on Image Change Detection

In the past decade, image change detection methods based on deep learning have developed rapidly. With the sharing of some labeled change detection datasets, Daudt *et al.* (Daudt, Le Saux, and Boulch 2018) introduce supervised semantic segmentation networks into the field of change detection which are used to identify binary or target changes in high-resolution images through fully convolutional networks. The rich datasets enable change detection models to achieve increasingly high accuracy. However, labeling all the changed pixels of images is quite time-consuming and laborious. Wu *et al.* (Wu et al. 2021b) use autoencoder to extract common features of unchanged regions and use threshold segmentation for unsupervised image change detection. Similarly, Noh *et al.* (Noh et al. 2022) utilize image self-reconstruction loss for unsupervised image change detection. Wu *et al.* (Wu, Du, and Zhang 2023) propose a fully convolutional change detection framework based on generative adversarial networks, which unifies unsupervised, supervised, and weakly-supervised approaches, further advancing the field.

Self-supervised Learning on Point Cloud

Self-supervised learning methods for point cloud typically require well-designed pretext tasks to learn the intrinsic representations of the point cloud without labeled data (Yuan et al. 2024b). The recent works can be roughly summarized as contrastive and reconstructive methods (Wu et al. 2021a). Contrastive methods contrast the potential representations of different point cloud transformation views (e.g., rotation, jitter, scale, etc.), and design pretext tasks based on inter-data information such as similarity. Rao *et al.* (Rao, Lu, and Zhou 2022) force the correspondence between objects and their component point clouds. By its definition, contrastive learning methods can be well applied to multi-source or multi-temporal point clouds tasks, such as point cloud registration (Yuan et al. 2023). Self-reconstruction methods typically encode point cloud samples as representation vectors and decode them back to the original input data (Xiao et al. 2023). Yang *et al.* (Yang et al. 2018) propose a point cloud autoencoder that compresses and encodes point cloud objects into low dimensional embedding vectors, which are then decoded back into 3D space by the decoder and forced to be the same as the input point cloud.

Method

The core of 3D point cloud change detection is to learn discriminative and robust features that can capture the shape information in the multi-temporal point clouds and distinguish them. To achieve this goal in an unsupervised manner,

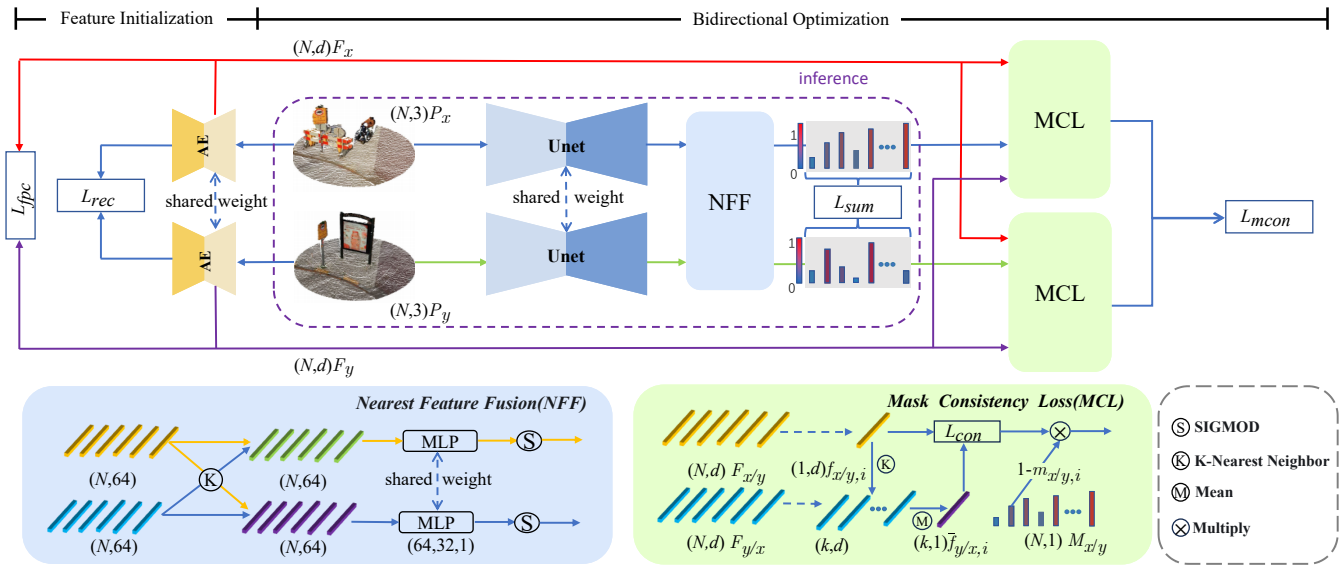

Figure 1: The overall framework of our unsupervised point cloud change detection approach. The training of network is divided into two stages. The first stage initializes features through self-reconstruction and the proposed foreknowledge point contrastive loss, and the second stage adds masked consistency loss for bidirectional optimization. In the inference stage, only the segmentor is used to obtain change detection results.

we propose to train a segmentor through the bidirectional optimization problem of shared weight feature extractor and segmentor, so as to segment the changed points. The overall framework of our method is shown in Figure 1.

Basic Module

The proposed framework consists of two basic modules: feature extractor and segmentor. In the current studies, many advanced segmentation models and techniques have been used for change detection, such as feature embedding (Zhang et al. 2021b), attention mechanism (Huang et al. 2019), etc. However, the focus of this paper is to provide a feasible unsupervised point cloud change detection training method. To decrease the complexity of the network, we only use the very basic structure for segmentor.

For the segmentor, we choose the basic Unet network structure PointNet++ (Qi et al. 2017), which has been used in many 3D point cloud tasks. Because the change detection task should fuse the features extracted from the two branches of the siamese network, we add a simple module Nearest Feature Fusion (NFF) based on the nearest neighbor in the last layer. For each point in a temporal point cloud, it takes the nearest neighbor from the rest of temporal point clouds for feature fusion. For the fused features, use Sigmoid as the final activation function to ensure that the output range is 0 to 1. $P_x \in \mathbb{R}^{N\times 3}$ and $P_y \in \mathbb{R}^{N\times 3}$ are multi-temporal point clouds as network inputs. Here, channel equal to 3 denotes that a point is represented by (x, y, z) coordinates. In some works (Li et al. 2024; de Gélis, Lefèvre, and Corpetti 2023), RGB, normal information or manual high-dimensional features of the point cloud is used as input, while we only use the most basic coordinate information. The output of segmentor can be written as:

$$M_x = SIGMOD(\phi(NN(PN(P_x), PN(P_y))), \quad (1)$$

where M_x is the probability that each point in P_x is a changed point, ϕ is implemented as a multi-layer perceptron (MLP) network, which maps features to a one-dimensional space. $PN(\cdot)$ is PointNet++ network, and $NN(\cdot)$ is the nearest neighbor operation. Similarly, M_y can be concluded. It is worth noting that in the inference stage, only the segmentor is retained to obtain the final change result through threshold segmentation.

For the feature extractor, we select the hierarchical point cloud feature learning network to extract the deep geometric features of the point cloud and obtain the point-level features by upsampling. Then the self-reconstruction task is performed on this basis. This feature extractor is described in detail in subsequent subsection.

Masked Consistency Loss

The basic assumption of 3D change detection is that for unchanged scenes in multi-temporal point clouds, they have certain consistency in geometric space. For the changed scenes, their features will be quite different. This assumption inspires us to exploit the feature relationship of unchanged points as free and abundant supervision signals to train the segmentor of point cloud change detection. This assumption can be similarly expressed as: for the unchanged scene in one point cloud, the adjacent area with similar features can be found in another point cloud, while the changed scene can not be found or features of the adjacent area are quite different. As a result, the goal of masked consistency loss (MCL) is to mine semantic knowledge shared by unchanged scenes in multi-temporal point clouds. The structure of MCL module is shown in Figure 1.

Since unchanged points and changed points always appear as a region, it is meaningless to consider only the feature of a point. In addition, different from pixel pairs in an image pair, point pairs in paired point clouds do not correspond one-to-one. Therefore, we propose to search for KNN points of any query point in $P_x = \{p_{x,1}, p_{x,2}, \ldots, p_{x,N}\}$ from $P_y = \{p_{y,1}, p_{y,2}, \ldots, p_{y,N}\}$. Then, for each point, average of the KNN point features is used to measure consistency, which is multiplied by change probability to calculate the masked consistency loss. In the following, we describe the details of masked consistency loss.

Region Feature. Let $F_x = \{f_{x,1}, f_{x,2}, \ldots, f_{x,N}\}$, $F_y = \{f_{y,1}, f_{y,2}, \ldots, f_{y,N}\}$, and $F_x, F_y \in \mathbb{R}^{N \times d}$ are the point-level features of P_x and P_y, respectively. For any point $p_{x,i}$ in P_x, we first search for its KNN points in P_y, then obtain features of these points in F_y and take the mean operation:

$$\overline{f}_{y,i} = \frac{1}{k} \sum\nolimits_{k=KNN(p_{x,i}, P_y)} f_{y,k}, \tag{2}$$

where $\overline{f}_{y,i}$ is feature of the neighboring region in P_y corresponding to point $p_{x,i}$. The symbol k is the number of neighboring points searched, which is set based on experience in the experiment.

Unsupervised Metric Learning. A straightforward method to optimizing feature relationship is to minimize the absolute difference between $f_{x,i}$ and $\overline{f}_{y,i}$, i.e., minimize $\sum_i \left\| f_{x,i} - \overline{f}_{y,i} \right\|$. However, this objective may not be the best choice, which imposes a linear penalty on the error of each feature dimension. Moreover in high-dimensional feature space, the presence of a large number of feature dimensions may lead to the influence of noise and redundant features. Therefore, we choose to supervise the relative quality of features and the quality of segmentation predictions through unsupervised metric learning tasks. Specifically, for the features $f_{x,i}$ of each point in P_x, we force them to approach the feature of neighboring regions in P_y, and filter out the impact of the changed points on feature consistency learning using the change probability obtained by the segmentation algorithm. The masked consistency loss can be written as:

$$L_{mxcon} = \frac{1}{N} \sum_i (1 - m_{x,i}) \log(1 + \exp(-(f_{x,i})^T \overline{f}_{y,i})), \tag{3}$$

where $m_{x,i}$ is change probability for each point. By minimizing this loss, we force the segmentor to maximize probability of changed points as much as possible, thereby segmenting them through a threshold. L_{mxcon} represents the masked consistency loss on point cloud P_x, and similarly, L_{mycon} on P_y can be obtained. We normalize the output of the feature extractor before calculating similarity and use dot product to obtain feature similarity.

It is necessary to note that when $m_{x,i}$ is all 1, the objective function will be 0. Thus, ℓ_1-norm constraint be used for segmentation graph to avoid full change output:

$$L_{sum} = \ell_1(M_x + M_y). \tag{4}$$

In summary, we arrive at the masked consistency loss term:

$$L_{mcon} = L_{mxcon} + L_{mycon} + \lambda L_{sum}, \tag{5}$$

where the weight λ is used to balance the impact of ℓ_1-norm on the optimization result. Masked consistency and ℓ_1-norm together enable us to achieve the optimization we are seeking for the segmentor, while also optimizing quality of the feature extractor.

Feature Extractor Initialization

Since discovering helpful knowledge for change detection from unlabeled data is usually quite difficulty, masked consistency loss may not necessarily lead to useful optimizations. Intuitively, quality of the feature extractor is crucial, because the masked consistency loss only supervises the segmentor to obtain points with similar features. That is, if the feature extractor is initialized well, it will offer decent supervision for the segmentor, thus creating a virtuous cycle for the learning of the segmentor and feature extractor. On the contrary, due to the poor initial state of the feature extractor, the learning process may lead to unpredictable results, and many unsupervised tasks have also pointed out this fact (Tschannen et al. 2019; Rao, Lu, and Zhou 2022). To avoid this issue, we propose auxiliary initialization tasks to supervise the network to jointly learn useful knowledge. Specifically, we employ two simple tasks, including self-reconstruction and foreknowledge point contrastive loss, as two self-supervised signals.

Self-Reconstruction. Self-reconstruction, or autoencoding, is a widely used unsupervised point cloud learning technique. In order to perform self-reconstruction, we use encoder E and decoder D from the point cloud autoencoder (Zhang et al. 2022) based on masked autoencoders to reconstruct the 3D coordinates of the point cloud on multi-temporal point clouds. The self-reconstruction loss is defined as the chamfer distance:

$$\begin{aligned} L_{rec} = &ChamferDist(D(E(P_x)), P_x) + \\ &ChamferDist(D(E(P_y)), P_y), \end{aligned} \tag{6}$$

where E is a hierarchical point cloud feature learning network that can extract deep geometric features of the point cloud. We extract features from different abstraction levels and upsample them hierarchically using three nearest neighborhoods interpolation following (Qi et al. 2017) to obtain point-level features.

Foreknowledge Point Contrastive Loss. Different from image CD, in 3DCD, some points can be known in advance whether they are changed. For example, ground points in the same scene and non-ground points in different scenes. By contrastive loss of these predicted points, the feature extractor can be initialized well. We treat ground points of different time periods in the current scene as positive samples, and use non-ground points of different scenes as negative samples. The contrastive loss of foreknowledge points can be written as:

$$L_{fpc} = -\frac{1}{N} \sum_i \log \frac{\exp(\mathbb{1}_{[i \in G_x]} \cdot (f^b_{x,i})^T (\overline{f}^b_{y,i}))}{\sum\limits_{b_j \neq b} \exp(\mathbb{1}_{[i \notin G_x]} \cdot (f^b_{x,i})^T (\overline{f}^{b_j}_{y,i}))}, \tag{7}$$

where G_x is the index set of ground points in P_x, obtained through coordinates in experiment. The symbol $\mathbb{1}_{[i \in G_x]} \in$

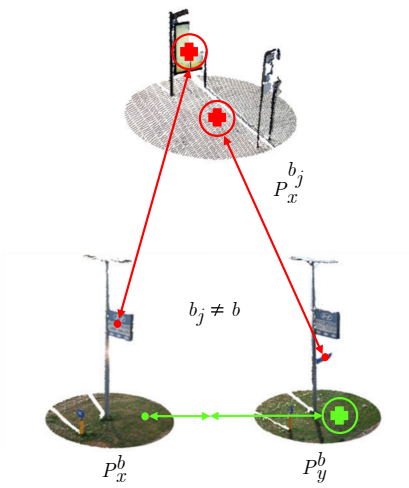

Figure 2: Feature Extractor initialization with foreknowledge point contrastive loss. Given a point cloud of a 3D scene, we encourage ground points to be similar to the neighborhood region of different temporal point clouds in the same scene (green lines), and non-ground points having dissimilar neighborhood regions with point clouds in different scenes (red line).

$\{0, 1\}$ is an indicator function that evaluates to 1 if and only if $P_{x,i}$ is a ground point, and similarly $\mathbb{1}_{[i \notin G_x]} \in \{0, 1\}$ is evaluated as 1 only if $P_{x,i}$ is a non-ground point. We use $\{b_j, j = 1, 2, 3, \ldots, m\}$ to represent different point sets in the mini-batch with batch size m and $f^b_{x,i}$ is the feature of $p_{x,i}$ in batch b. The overall framework of the foreknowledge point contrastive objective is illustrated in Figure 2.

Total Loss Function

Finally, combining the masked consistency loss and initialization of the feature extractor, we design the overall objective of this framework:

$$L_{mucd} = L_{mcon} + L_{rec} + L_{fpc}. \tag{8}$$

Through minimizing this loss function, we can optimize both the segmentor and feature extractor simultaneously. The extracted point-level features should better conform to the proposed assumption, and the segmentor should be able to obtain excellent point-level change probabilities.

Experiments

Experimental Setting

Datasets. In order to conduct 3D unsupervised change detection experiments, we chose the street point cloud dataset SLPCCD (Wang et al. 2023), which is generated by a public dataset called Change3D and used for the study of street scene change segmentation. Change3D contains colored point clouds obtained from real streets in the Netherlands in 2016 and 2020, collected by in car LiDAR sensors. This dataset is labeled in the time-1 point cloud for removed objects, and in the time-2 point cloud for added objects, witch can determine shape of the removed object. The objects of change marked in this dataset include pedestrians, vehicles, benches, signs, billboards, streetlights, bicycles, and other categories.

Another publicly available dataset is called Urb3DCD (de Gélis, Lefèvre, and Corpetti 2021), which is simulated by 3D aerial LiDAR to collect 3D models of cities. Urb3DCD is divided into 5 sub datasets each with different resolutions, noise, and sensors. Unlike Change3D, this dataset only scans the highest floors of buildings due to its use of aerial LiDAR to capture object descriptions with only height information. Change3D captures the overall shape information of objects more accurately through car LiDAR.

Overall, Urb3DCD belongs to the change detection task of large city 3D maps, while Change3D belongs to the change detection task of detailed object information in indoor or street small-scale scenes.

Implementation details. In all experiments, we sample 8192 points for each point cloud for training and testing, and only normalize coordinate information to zero mean and unit variance as input. The k in all regional features related to feature metric learning in the main experiment is set to 8. When training the network, we use the Adam optimizer, set the batch size to 4 and set the initial learning rate to 0.001 reducing it exponentially (with a decay rate of 0.7). In the inference stage, we use a threshold of 0.5 credibility to segment the change points.

In previous section, we have elaborated on the impact of feature quality for networks. In order to train the network normally, the epoch for feature initialization be set to 40, and the epoch for joint training should also is set to 40. In each epoch of the overall training process, we optimize parameters of the feature extractor using the objective L_{rec} and L_{fpc} to better guide segmentation with the extracted features. When current epoch exceeds the epoch of feature initialization, L_{mcon} is calculated to optimize parameters of the segmentor. In summary, the objective of initialization operation in the feature extractor makes it possible to learn discriminative features, while the loss of masked consistency forces the network to predict similar features in unchanged regions.

For evaluation indicators, we use the most commonly used methods in change detection, namely overall accuracy (OA) and mean intersection over union (mIoU), for evaluation. Where mIoU is the mean of the IoU values for three marker points: unchanged, removed, and added.

Comparison Experiments

We conduct a fair comparison with state-of-the-art methods, which are mainly divided into two categories: supervised methods and weakly-supervised methods. To our knowledge, there are not works to solve point cloud change detection through unsupervised methods.

Comparisons with Supervised Method. We first compare our method with existing supervised method and provide a comparison of 3DCDNet and its proposed various popular supervision baselines. The results are summarized in Table 1. We observe that the small PointNet++ basic model trained by our method can surpass most advanced

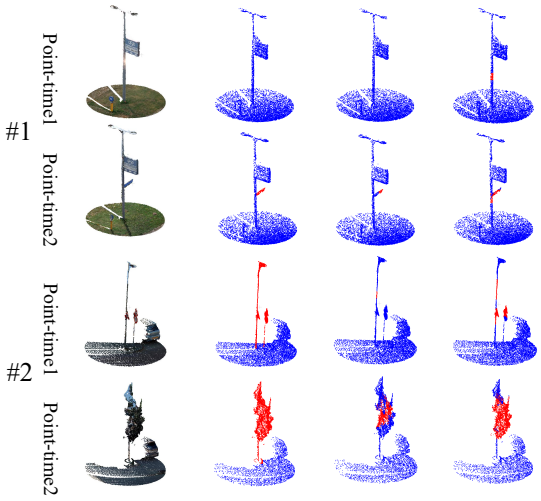

Figure 3: Visualization of the comparison results on SLPCCD. The changed points are marked in red and the unchanged points in blue.

methods. Compared with supervised methods, MUCD outperforms PointNet-based, PointMLP-based, and PointConv-based methods with 12.47%, 1.07%, and 18.32% mIoU terms, respectively. In terms of OA, our method outperforms any other method except 3DCDNet, achieving improvements of 6.08%, 0.66%, 3.43%, 4.74%, and 1.68% compared to the selected comparative method.

Method	Type	OA	mIoU
PointNet-Siamese	Sup.	89.67	48.01
PointNet2-Siamese	Sup.	95.09	67.55
PointMLP-Siamese	Sup.	92.32	59.41
PointConv-Siamese	Sup.	91.01	42.16
DGCNN-Siamese	Sup.	94.07	62.30
3DCDNet	Sup.	95.85	74.45
MUCD(Ours)	Unsup.	**95.75**	**60.48**

Table 1: Comparisons of our method aganist the state-of-the-art supervised point cloud models on SLPCCD.

We visualize the change graph in Figure 3. It can be seen that our method performs better in situations where there are changes in semantic information in the scene (such as road signs changed to trees), rather than just changes in spatial occupancy. This is because unlike supervised point cloud change detection that directly learns the content of interest from labels, our method focuses more on mining the shared intrinsic semantic and structural information contained in the unchanged regions of the point cloud itself.

Comparisons with Weakly-Supervised Method. More exciting, compared to existing weakly-supervised methods, our method also achieve competitive results in terms of training time, parameters and effectiveness. DC3DCD (de Gélis, Lefèvre, and Corpetti 2023) utilizes a siamese network to extract the point-level features of a multi-temporal point clouds, and then obtains change features through nearest neighbor fusion and clusters them. Use the obtained cluster centers to calculate pseudo labels for each point to guide backbone learning. In the inference stage, due to the need to map the clustering category k to the final segmentation category c using point level labels, De Gélis *et al.* (de Gélis, Lefèvre, and Corpetti 2023) classify this method as a weakly-supervised method. It is worth noting that DC3DCD needs to extract point level features twice in each epoch, respectively, to obtain the cluster center and train the network backbone, which also results in its training speed far inferior to our method. We compare the comparison results of the two methods in detail in Table 2.

Method	Type	mIoU	#Params	Time(S.)
DC3DCD	Weakly Sup.	47.27	39.9M	730
MUCD(Ours)	Unsup.	**60.48**	22.02M	69

Table 2: Comparisons of our method aganist the state-of-the-art weakly-supervised point cloud models on SLPCCD.

Transfer Experiment. In order to further explore the generalization ability of segmentor trained by our unsupervised method, we conduct direct transfer experiments on existing datasets. As shown in Table 3, we train the same segmentor using our method and labels on the SLPCCD training set, and test it on the Urb3DCD testing set. We see that unsupervised methods outperform supervised methods on five sub datasets with heterogeneous quality. This indicates that unsupervised learning has stronger transferability than supervised learning, and our model can be well extended to various invisible data because we learn from shape structures rather than labels. Figure 4 reports the visualization results of our method on the Urb3DCD testing set.

Dataset	Training Method	OA	mIoU
Urb3DCD subdataset-1 ALS Low Res	sup.	91.53	35.88
	ours	**96.43**	**46.53**
Urb3DCD subdataset-2 ALS High Res	sup.	93.96	35.33
	ours	**96.79**	**47.59**
Urb3DCD subdataset-3 ALS High Noise	sup.	91.14	36.04
	ours	**96.83**	**44.04**
Urb3DCD subdataset-4 Photogrammetry	sup.	92.13	35.67
	ours	**97.1**	**43.03**
Urb3DCD subdataset-5 Multi-Sensor	sup.	90.41	35.28
	ours	**95.6**	**42.46**

Table 3: Comparisons of our method aganist the supervised counterpart on Urb3DCD testing set.

Ablation Experiments

Ablation Experiments of Segmentation Effect. Firstly, we conduct a detailed ablation study based on the small PointNet++ network. The results are summarized in Table 4. Baseline model A can be seen as a variant of autoencoder (Zhang et al. 2022), which is trained only through self-reconstruction loss. In fact, the model has degenerated into a method of autoencoder plus threshold segmentation, which uses size of feature difference for segmentation in the inference stage. The segmentation accuracy is relatively low, with an mIoU of 44.45%. The baseline model B is trained

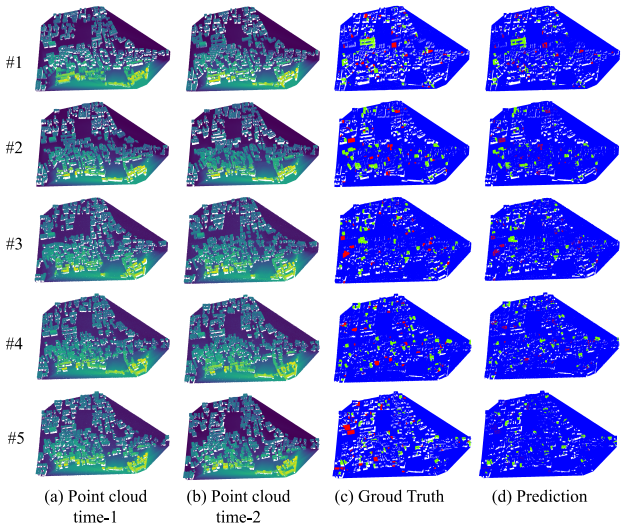

Figure 4: Visualization of results on Urb3DCD testing set. The added points are marked in green, removed points in red and the unchanged points in blue.

on a small PointNet++ segmentor using MCL, and the performance is only 41.79%. This is consistent with the issue that training the segmentor becomes exceptionally difficult due to feature quality. We can see that the model trained jointly with the proposed feature extractor and segmentation (Model C) can improve the baseline models of the autoencoder and segmentation by 2.65% and 5.31%, respectively. This convincingly confirms its effectiveness. After adding NFF modules to the splitters in models B and C, it can be seen that the performance has been improved. This confirms the necessity of feature fusion in handling change detection task, and this paper only proposes one of the most basic feature fusion methods. In addition to reconstruction loss, the predictive point comparison loss (Model F) proposed in this paper also significantly improved by 11.48% on the baseline. Then, when combining these two losses, the effect can be further improved to 60.48%.

Model	MCL(L_{mcon})	NFF	L_{rec}	L_{fpc}	OA	mIoU
A			✓		92.32	44.45
B	✓				92.9	41.79
C	✓		✓		93.9	47.1
D	✓	✓			93.01	44.24
E	✓	✓	✓		94.64	55.72
F	✓	✓		✓	94.31	54.48
G	✓	✓	✓	✓	**95.75**	**60.48**

Table 4: Ablation study on SLPCCD testing set.

For the proposed loss terms, the above ablation experiments achieve the best results when $k = 8$. The number of nearest neighbors is a crucial setting. In Table 5, we conduct ablation experiments on this hyperparameter, and the model achieve optimal performance when k is set to 8. It can be seen that before $k = 8$, the segmentation performance is significantly improved compared to $k = 1$ and $k = 4$, indicating that a larger receptive field can improve model performance. However, when k is large ($k = 16$), it will have a negative impact. This is because for small objects in unchanged regions, using a larger receptive field for feature consistency loss may cause their features to be closer to those of negative samples, contaminate features, and lead to a decrease in network performance.

K-NN	Background	Added	Removed	OA	mIoU
1	94.72	34.79	37.87	94.46	55.79
4	94.61	38.11	40.14	94.8	57.62
8	**95.61**	**42.53**	**43.32**	**95.75**	**60.48**
16	94.82	33.14	33.58	95.11	56.23

Table 5: Ablation study of the k of MCL on SLPCCD testing set.

Ablation Experiments of Quality of Feature Extractor. We conduct ablation experiments on the quality of extracted features to confirm the effectiveness of the method and the proposed assumption. We obtain a feature difference map by comparing the feature difference between each point and its nearest neighbor in another phase point cloud, and further obtain a feature difference heatmap. In order to clearly demonstrate the feature similarity of unchanged regions, we report the feature visualization results in Figure 5 for both autoencoder only and with the addition of our modules. It can be seen that feature difference only trained by self-reconstruction task (i.e. autoencoder) are concentrated in the middle value, which is not conducive to segmenting and obtaining the change map.

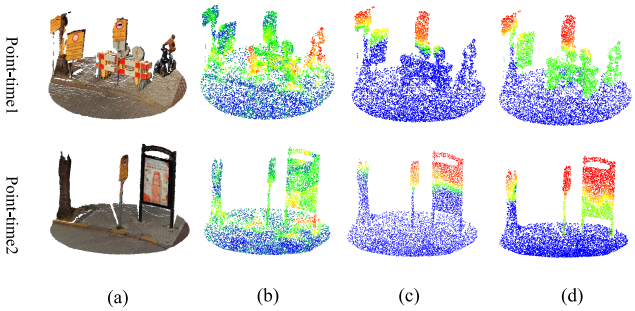

Figure 5: Feature different map visualization of the proposed method. (a): Point clouds, (b): L_{rec}, (c): $L_{rec} + L_{fpc}$, (d) :$L_{rec} + L_{fpc}$ + MCL.

Conclusion

We propose a novel unsupervised learning method to solve the task of point cloud change detection. The proposed MCL considers the feature similarity of unchanged scenes by utilizing shared information from point cloud of different time phases. We also enhance feature representation through excellent initialization tasks to reduce the adverse effects of feature pollution on training. Extensive experiments demonstrate the effectiveness of this method on two existing benchmark datasets for 3D change detection in real scenes. For the proposed MUCD, the segmentor and feature extractor networks can be replaced or improved, which will be explored in future.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (62276200, 62036006).

References

Benedek, C.; Descombes, X.; and Zerubia, J. 2011. Building development monitoring in multitemporal remotely sensed image pairs with stochastic birth-death dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1): 33–50.

Chen, J.; Yuan, Z.; Peng, J.; Chen, L.; Huang, H.; Zhu, J.; Liu, Y.; and Li, H. 2020. DASNet: Dual attentive fully convolutional Siamese networks for change detection in high-resolution satellite images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14: 1194–1206.

Daudt, R. C.; Le Saux, B.; and Boulch, A. 2018. Fully convolutional siamese networks for change detection. In *2018 25th IEEE international conference on image processing (ICIP)*, 4063–4067. IEEE.

de Gélis, I.; Lefèvre, S.; and Corpetti, T. 2021. Change detection in urban point clouds: An experimental comparison with simulated 3d datasets. *Remote Sensing*, 13(13): 2629.

de Gélis, I.; Lefèvre, S.; and Corpetti, T. 2023. DC3DCD: Unsupervised learning for multiclass 3D point cloud change detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 206: 168–183.

Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; and Bennamoun, M. 2020. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12): 4338–4364.

Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; and Liu, W. 2019. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 603–612.

Hussain, M.; Chen, D.; Cheng, A.; Wei, H.; and Stanley, D. 2013. Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of photogrammetry and remote sensing*, 80: 91–106.

Jin, S.; Yang, L.; Zhu, Z.; and Homer, C. 2017. A land cover change detection and classification protocol for updating Alaska NLCD 2001 to 2011. *Remote Sensing of Environment*, 195: 44–55.

Khan, S. H.; He, X.; Porikli, F.; and Bennamoun, M. 2017. Forest change detection in incomplete satellite images with deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(9): 5407–5423.

Ku, T.; Galanakis, S.; Boom, B.; Veltkamp, R. C.; Bangera, D.; Gangisetty, S.; Stagakis, N.; Arvanitis, G.; and Moustakas, K. 2021. SHREC 2021: 3D point cloud change detection for street scenes. *Comput. Graph.*, 99(C): 192–200.

Li, X.; Xu, Q.; Zhang, J.; Zhang, T.; Yu, Q.; Sheng, L.; and Xu, D. 2024. Multi-Modality Affinity Inference for Weakly Supervised 3D Semantic Segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3216–3224.

Nagy, B.; Kovács, L.; and Benedek, C. 2021. ChangeGAN: A deep network for change detection in coarsely registered point clouds. *IEEE Robotics and Automation Letters*, 6(4): 8277–8284.

Noh, H.; Ju, J.; Seo, M.; Park, J.; and Choi, D.-G. 2022. Unsupervised change detection based on image reconstruction loss. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1352–1361.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Rao, Y.; Lu, J.; and Zhou, J. 2022. PointGLR: Unsupervised structural representation learning of 3D point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2): 2193–2207.

Sakurada, K.; Okatani, T.; and Deguchi, K. 2013. Detecting changes in 3D structure of a scene from multi-view images captured by a vehicle-mounted camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 137–144.

Schauer, J.; and Nüchter, A. 2018. Removing non-static objects from 3D laser scan data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 143: 15–38.

Stilla, U.; and Xu, Y. 2023. Change detection of urban objects using 3D point clouds: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 197: 228–255.

Tschannen, M.; Djolonga, J.; Rubenstein, P. K.; Gelly, S.; and Lucic, M. 2019. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*.

Wang, Z.; Peng, C.; Zhang, Y.; Wang, N.; and Luo, L. 2021. Fully convolutional siamese networks based change detection for optical aerial images with focal contrastive loss. *Neurocomputing*, 457: 155–167.

Wang, Z.; Zhang, Y.; Luo, L.; Yang, K.; and Xie, L. 2023. An End-to-end Point-based Method and A New Dataset for Street Level Point Cloud Change Detection. *IEEE Transactions on Geoscience and Remote Sensing*.

Wu, C.; Du, B.; and Zhang, L. 2023. Fully convolutional change detection framework with generative adversarial network for unsupervised, weakly supervised and regional supervised change detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8): 9774–9788.

Wu, L.; Lin, H.; Tan, C.; Gao, Z.; and Li, S. Z. 2021a. Self-supervised learning on graphs: Contrastive, generative, or predictive. *IEEE Transactions on Knowledge and Data Engineering*, 35(4): 4216–4235.

Wu, Y.; Li, J.; Yuan, Y.; Qin, A. K.; Miao, Q.-G.; and Gong, M.-G. 2021b. Commonality autoencoder: Learning common features for change detection from heterogeneous images. *IEEE transactions on neural networks and learning systems*, 33(9): 4257–4270.

Xiao, A.; Huang, J.; Guan, D.; Zhang, X.; Lu, S.; and Shao, L. 2023. Unsupervised point cloud representation learning with deep neural networks: A survey. *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, 45(9): 11321–11339.

Yang, Y.; Feng, C.; Shen, Y.; and Tian, D. 2018. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 206–215.

Yuan, Y.; Wu, Y.; Fan, X.; Gong, M.; Ma, W.; and Miao, Q. 2023. EGST: Enhanced geometric structure transformer for point cloud registration. *IEEE transactions on visualization and computer graphics*.

Yuan, Y.; Wu, Y.; Fan, X.; Gong, M.; Miao, Q.; and Ma, W. 2024a. Inlier Confidence Calibration for Point Cloud Registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5312–5321.

Yuan, Y.; Wu, Y.; Gong, M.; Miao, Q.; and Qin, A. K. 2024b. One-nearest neighborhood guides inlier estimation for unsupervised point cloud registration. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhang, J.; Jia, X.; Hu, J.; and Tan, K. 2021a. Moving vehicle detection for remote sensing video surveillance with nonstationary satellite platform. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 5185–5198.

Zhang, R.; Guo, Z.; Gao, P.; Fang, R.; Zhao, B.; Wang, D.; Qiao, Y.; and Li, H. 2022. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pretraining. *Advances in neural information processing systems*, 35: 27061–27074.

Zhang, X.; Wei, Y.; Li, Z.; Yan, C.; and Yang, Y. 2021b. Rich embedding features for one-shot semantic segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11): 6484–6493.

Zhu, Z.; and Woodcock, C. E. 2014. Continuous change detection and classification of land cover using all available Landsat data. *Remote sensing of Environment*, 144: 152–171.