

Article

GSANet: Geometric Structure-Aware Siamese Network for 3D Change Detection

Jiakang Chen ¹, Rongfang Wang ^{1,*} , Libin Sun ² and Changzhe Jiao ¹ 

¹ Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China; chenjiakang@stu.xidian.edu.cn (J.C.); cjiao@xidian.edu.cn (C.J.)

² School of Computer Science and Technology, Xidian University, Xi'an 710071, China; 23171214463@stu.xidian.edu.cn

* Correspondence: rfwang@xidian.edu.cn

Highlights

What are the main findings?

- A Geometric Structure-Aware Siamese Network (GSANet) is proposed for 3D point cloud change detection, which explicitly integrates boundary and edge priors via Boundary-Aware Subsampling (BAS), Edge-Aware Adaptive Graph Convolution, and a Difference Enhancement Module (DEM).
- GSANet achieves state-of-the-art performance on the SLPCCD and HKCD datasets, with an mIoU of 82.87% and OA of 97.30%, outperforming existing methods including 3DCDNet, PGN3DCD, and Ms-DANet.

What are the implications of the main findings?

- The proposed framework enhances feature representation robustness by preserving critical geometric cues, effectively suppressing false positives and improving boundary localization in complex urban environments affected by noise and viewpoint variations.
- It offers a practical and reliable solution for real-world remote sensing applications such as urban monitoring, disaster assessment, and environmental analysis, particularly in scenarios with sparse or occluded point cloud data.

Abstract

Three-dimensional (3D) point cloud change detection is essential for urban monitoring and environmental analysis, yet existing methods mainly rely on point-wise semantic differences and overlook change–unchange boundaries and object edges—key geometric cues for precise localization, especially for subtle or gradual changes. To address this, we propose GSANet, a Geometric Structure-Aware Siamese Network that explicitly integrates boundary and edge priors into sampling and feature learning. First, a Boundary-Aware Subsampling (BAS) strategy preserves key points near change boundaries while reducing redundancy, and a Boundary-Aware Binary Cross-Entropy (BA-BCE) loss assigns higher supervision weights to boundary points, enhancing learning in ambiguous regions. Second, an Edge-Aware Siamese Network captures robust local shapes by embedding edge priors into feature extraction, incorporating Edge-Aware Adaptive Graph Convolution, Edge-Aware Downsampling, and Cross-Attention Upsampling to maintain structural consistency across temporal branches. Additionally, a Difference Enhancement Module (DEM) amplifies feature discrepancies between bitemporal point clouds, improving sensitivity to subtle changes. Extensive experiments on a street-level dataset and urban dataset show our method outperforms state-of-the-art approaches.



Academic Editor: John Trinder

Received: 26 June 2026

Revised: 29 July 2026

Accepted: 3 August 2026

Published: 7 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: 3D change detection; deep learning; point clouds; Geometric Structure-Aware; difference enhancement

1. Introduction

Change detection (CD) is an essential task in the fields of remote sensing and computer vision, playing an essential role. By accurately identifying structural, surface, or object changes within a given area, change detection provides reliable data support for urban planning, disaster monitoring, environmental assessment [1] and so on, making it irreplaceable in real-world applications. In recent years, with the rapid development of deep learning and computer vision technologies, research on two-dimensional change detection (2D CD) has made significant progress. A large number of high-precision, end-to-end detection methods have emerged, demonstrating outstanding performance across various application scenarios [2]. However, 2D CD inherently suffers from projection distortion and a lack of depth perception, making it difficult to accurately capture real three-dimensional structural variations, especially in complex urban environments [3].

To overcome the limitations of traditional 2D change detection methods, researchers have increasingly focused on three-dimensional change detection (3D CD), which directly analyzes geometric and structural differences in 3D space [4]. Point clouds, which represent real-world scenes through discrete spatial coordinates (x , y , z), offer a highly accurate representation of geometric shape, scale, and spatial distribution, without the distortion or artifacts inherent in projection-based methods [5]. With the growing availability of advanced technologies such as LiDAR, mobile laser scanning (MLS), airborne laser scanning (ALS), and RGB-Depth sensors, the acquisition of large-scale, high-resolution point cloud data has become more accessible [6,7]. This has greatly facilitated the development of 3D CD, enabling precise geometric modeling of buildings, roads, vegetation, and other natural structures. Although 2D change detection methods have achieved remarkable success, their direct application to 3D point cloud change detection remains challenging. This is mainly because 2D images are represented as regular pixel grids with explicit spatial adjacency, whereas point clouds consist of irregularly distributed points without fixed neighborhood structures. Therefore, conventional image-based operations cannot be directly applied to point clouds. Moreover, multi-temporal point clouds usually exhibit variations in point density, incomplete observations caused by occlusion, and residual geometric misalignment caused by registration errors, which also introduce uncertainty into spatial correspondence. Traditional 3D change detection (3D CD) methods typically rely on measuring distances between point clouds in Euclidean space to detect changes. These methods analyze geometric differences between point clouds, including point-based, voxel-based, and object-based approaches. Point-based methods detect changes by calculating spatial position differences between corresponding points [8,9]. Voxel-based methods rasterize point clouds into voxel grids and detect changes by comparing voxel occupancy [10]. Object-based methods segment point clouds into objects and detect changes by analyzing geometric differences between these objects [11]. While these methods can effectively detect changes, they face challenges in handling noise, sparse sampling, and occlusions, especially when detecting subtle changes in complex environments.

In recent years, with the rapid development of deep learning technologies, researchers have attempted to use neural networks to automatically extract spatial and geometric features, enabling more robust and efficient change detection [12,13]. By learning hierarchical representations and capturing spatial context in an end-to-end manner, these methods have

significantly improved both accuracy and generalization in complex environments, making deep learning the dominant approach for 3D CD.

Despite this progress, several challenges remain. First, at the data-processing level, the vast majority of methods [14] rely on random subsampling to reduce data redundancy during training. These approaches fail to prioritize retaining the ambiguous yet critical points along the boundaries between changed and unchanged regions [15]. Second, at the feature extraction level, conventional networks often do not explicitly leverage strong geometric priors inherent in the data, such as the structure of object edges. Lacking both critical input samples and a model architecture guided by geometric structure, the resulting feature representations struggle to learn robust, temporally consistent features, which are essential for accurate change detection.

Moreover, many existing methods fail to fully exploit the differential information between bitemporal point clouds, resulting in limited sensitivity to subtle or small-scale geometric variations. In contrast, in 2D CD, many methods have effectively addressed boundary localization and the utilization of differential information [16–18], successfully handling noise and detecting subtle changes. However, in 3D CD, these issues have not been sufficiently explored, which limits the sensitivity and accuracy of existing methods when detecting fine or small-scale geometric variations.

Our model is guided by geometric features from change-unchange boundaries and object edges. By explicitly incorporating these geometric structures into the sampling and feature learning processes, GSANet is designed to be aware of both change-related transitions and intrinsic object geometries. These geometric cues provide informative guidance for feature learning, enabling the network to learn robust spatial representations that are less sensitive to noise and viewpoint variations. By anchoring the feature-learning process on these reliable geometric cues, our approach naturally enhances cross-temporal consistency and ultimately achieves superior accuracy in 3D point cloud change detection. Extensive experimental results on the Street-Level Point Cloud Change Detection dataset and a urban dataset, together with comprehensive ablation studies, demonstrate that our method achieves superior performance against state-of-the-art approaches.

Our contributions can be summarized as follows:

- (1) To leverage boundary priors between changed and unchanged regions, we propose a dual mechanism of Boundary-Aware Subsampling and Boundary-Aware BCE Loss. The former efficiently reduces point cloud redundancy while preferentially preserving boundary points. The latter amplifies the learning signal from these boundary points by assigning them greater weight during optimization.
- (2) We propose a novel Edge-Aware Siamese Network architecture that systematically embeds geometric edge priors. The network integrates Edge-Aware Adaptive Graph Convolution and sampling to extract structurally robust features, and is introduced with a Difference Enhancement Module to specifically amplify temporal feature contrast, boosting sensitivity to subtle changes.

2. Related Work

2.1. Point Cloud Sampling

Point cloud sampling is a crucial step in point cloud processing, with the core goal being to extract a representative subset from dense point clouds while preserving key geometric information and significantly reducing computational complexity. To satisfy constraints regarding GPU memory and computational efficiency, point clouds are typically subsampled before being fed into neural networks. In this research area, traditional non-learning-based methods have been widely adopted due to their simplicity and efficiency. Among them, Farthest Point Sampling (FPS) [19] iteratively selects the points farthest from

the already selected subset, effectively ensuring spatial coverage and uniformity of the sampled subset. As a result, FPS has been incorporated as a downsampling module in many classic networks, such as PointNet++ [20] and PointConv [21]. Random Sampling (RS), on the other hand, is highly computationally efficient and has become the preferred choice for processing large-scale point clouds, widely used in methods like RandLA-Net [22]. FPS and RS are also widely employed for downsampling within network architectures.

With the advancement of deep learning, recent research has shifted towards task-driven, learning-based downsampling methods, which aim to adaptively evaluate point importance based on specific task objectives. S-Net [23] processes the original point cloud with an MLP and directly outputs a simplified point set, which is optimized through backpropagation to suit the task requirements. SampleNet [24] builds on this by introducing a projection step, which constrains the generated points to the k-nearest neighbors in the original point cloud to enhance their representativeness. LightTN [25] incorporates attention mechanisms to improve sampling performance, while APES [26] extracts edge points by calculating local attention values within the point cloud.

However, these approaches are mainly designed for common tasks like classification or segmentation and rarely emphasize information of change-unchange boundaries and object edges, which are particularly critical for change detection.

2.2. Structure-Based Point Cloud Methods

In the field of 3D point cloud analysis, leveraging structural information in point clouds has been demonstrated to significantly enhance performance across a range of core tasks. For instance, PointSCNet [27] employs space-filling curves to guide sampling and fuses structure and correlation information, effectively capturing local geometric structures in point clouds and achieving superior performance, indicating that explicitly modeling structural cues leads to more discriminative representations. GPSFormer [28] integrates global perception with local structure fitting mechanisms in a Transformer framework, encoding both global context and local geometric relationships. Adaptive spatial structure graph transformer [29] dynamically adjusts structural weights between points to better capture edge and contextual information, resulting in significant improvements in complex scene segmentation. SAPCNet [30] explicitly incorporates global structural constraints to more accurately recover missing regions, highlighting the key role of structure preservation in geometric reconstruction.

Although these studies make full use of structural information to enhance feature learning in classification, segmentation, detection, and reconstruction tasks, existing point cloud change detection methods predominantly focus on comparing geometric differences between bi-temporal point clouds and lack systematic modeling of structural information. This limitation results in performance bottlenecks when identifying complex scene changes or subtle structural alterations.

2.3. Deep Learning-Based 3D CD Methods

In the early research, some of the methods were used to detect changes by converting the point clouds into 2D images with the help of convolutional neural networks (CNN). Zhang et al. [31] combined a multi-modal point cloud with an orthophoto image into a CNN network to extract the change information. However, this method loses the spatial structure in the original point cloud, which limits the ability of three-dimensional representation.

In order to more fully utilize the structural information of point clouds, a large number of studies in recent years have focused on point-level deep learning models, such as PointNet [32], PointNet++ [20], PointMLP [33], PointConv [21], DGCNN [34], and KP-Conv [35]. Therefore, some researchers have also started to focus on point-level deep

learning 3DCD models, which capture the changes of 3D scenes by directly processing point cloud data. For example, De Gélis et al. [13] proposed a KPConv-based Siamese network that detects multi-class semantic changes by analyzing feature differences between neighboring point pairs. Wang et al. [14] introduced a local feature aggregation module to achieve end-to-end point-level change detection in street-scale point clouds. Zhan et al. [36] proposed a Prior Knowledge-Guided 3D Change Detection Network, which explicitly incorporates prior knowledge to enhance the model's ability to discriminate real changes, addressing real-world challenges such as registration errors and semantic class imbalance in multi-temporal point clouds. To further tackle these issues, Zhang et al. [37] introduced a Multi-task Enhanced Cross-temporal Point Transformer Network, which simultaneously performs semantic segmentation and change detection to improve robustness and accuracy. Unlike the methods described above, our model utilizes geometric features from change boundaries and object edges to guide feature learning, generating more robust spatial representations. This feature-learning process, anchored in reliable geometric cues, enhances cross-temporal consistency and the accuracy of 3D point cloud change detection.

3. Method

The framework of our proposed GSANet method is shown in Figure 1. GSANet is an end-to-end two-branch siamese network, which consists of four main components: (1) Boundary-Aware Subsampling (BAS) strategy; (2) Edge-Aware Siamese Network; (3) Difference Enhancement Module (DEM); and (4) Boundary-Aware Binary Cross-Entropy (BA-BCE) loss. Specifically, BAS first preserves informative points located near change-unchange transitions during input preparation, preventing critical boundary information from being discarded. Then, the Edge-Aware Siamese Network exploits intrinsic object geometric structures to learn robust spatial representations. During hierarchical feature extraction, EA-DS and CA-US maintain structural consistency while reducing computational redundancy. Finally, DEM explicitly enhances temporal feature discrepancies, improving the discrimination of subtle changes. Through this sequential process, GSANet integrates geometric priors from input selection, feature learning, and temporal comparison. The details of each part are described in the following.

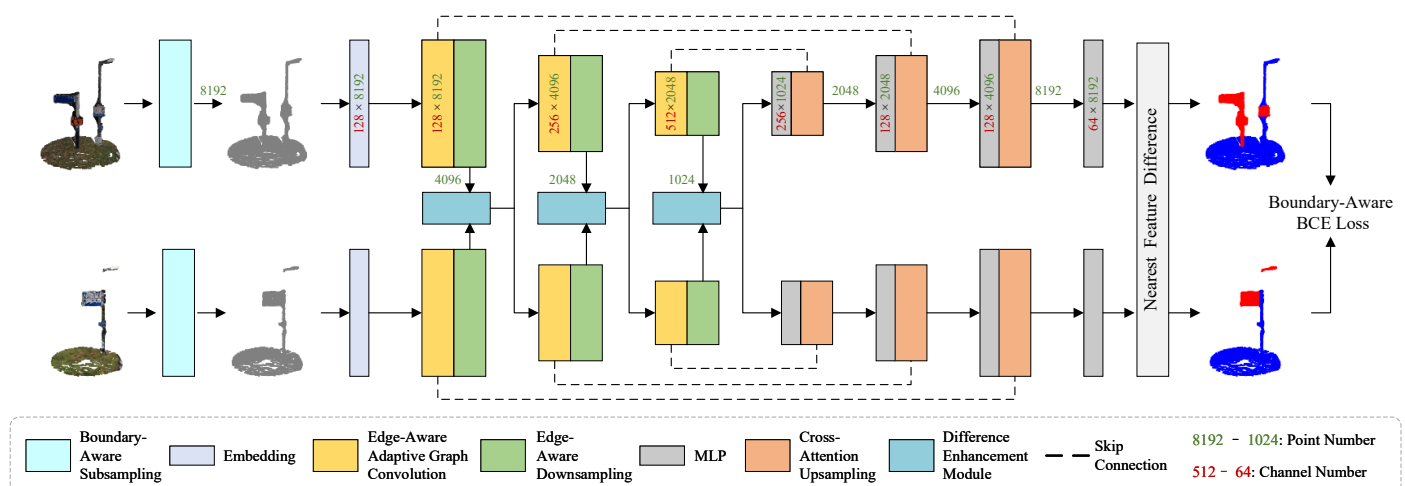


Figure 1. The architecture of GSANet, which employs a two-branch architecture with shared weight.

3.1. Boundary-Aware Subsampling

In this work, boundary refers to the transition regions between changed and unchanged points, where neighboring points have different change states and the localization of changes becomes ambiguous. Such regions contain important semantic information for

change detection but are easily discarded during point cloud sampling due to their limited spatial proportion. To address the issues of blurred boundaries and imprecise localization between changed and unchanged regions, we propose the Boundary-Aware Subsampling (BAS) strategy, which reduces redundancy in the original point cloud data while preferentially preserving points located near change-unchange transition regions. The first step involves utilizing ground truth to divide all points of each sample into two sets: the boundary set Ω_B and the non-boundary set Ω_{NB} , using the KNN algorithm. Specifically, each point in the point cloud data is taken as the center point p_i , and the KNN algorithm is used to find its K_B nearest neighbors. If the label of the center point p_i and the labels of all its nearest neighbors p_{ij} are the same, then p_i is assigned to the non-boundary set Ω_{NB} . Otherwise, if any of the nearest neighbors has a label different from that of the center point, p_i is assigned to the boundary set Ω_B . An illustration is shown in Figure 2. Here, for KNN, K_B is set to 16, and further experimental analysis of this parameter is presented in Section 4.

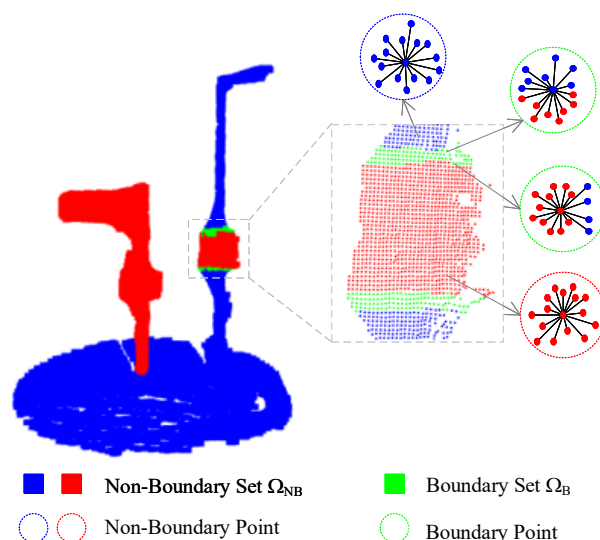


Figure 2. Examples of boundary and non-boundary points.

Next, we proceed to select boundary and non-boundary points. Given a subsampling number $SN = 8192$ and selection ratio $\eta = 0.2$, we randomly select 8192 points from each sample, 20% of which are randomly selected from boundary points and the remaining 80% are randomly selected from non-boundary points. This selection process can be represented by Equations (1) and (2).

$$X_B = \mathcal{R}(\Omega_B, \min(|\Omega_B|, \eta \times SN)) \quad (1)$$

$$X_{NB} = \mathcal{R}(\Omega_{NB}, SN - |X_B|) \quad (2)$$

where $\mathcal{R}(\Omega, s)$ represents randomly selecting s points from the set Ω and $|\cdot|$ denotes the number of elements in the set.

The logical explanation of $\min(\cdot)$ operation is that we first select at most $\lfloor \eta \times SN \rfloor$ points from the boundary set Ω_B . If $|\Omega_B|$ is less than $\lfloor \eta \times SN \rfloor$, select all boundary points. Then, supplement the remaining points $SN - |X_B|$ from the non-boundary set Ω_{NB} to ensure that the total point number of each sample is SN . The SN is set to 8192 based on the method [14]. Through ablation analysis, the author verified that selecting 8192 points per sample is the optimal balance between performance and resource usage.

3.2. Edge-Aware Siamese Network

Unlike boundary information that describes semantic transitions between changed and unchanged regions, the edge represents intrinsic geometric discontinuities within point clouds, such as abrupt variations in object shapes, surface orientations, or local spatial distributions. As shown in Figure 3, the proposed Edge-Aware Siamese Network mainly consists of three components: Edge-Aware Adaptive Graph Convolution (EA-AGConv), Edge-Aware Downsampling (EA-DS), and the Cross-Attention Upsampling (CA-US) module. Specifically, the EA-AGConv module is responsible for adaptively capturing local geometric structures and enhancing edge-aware feature learning by integrating edge-aware attention into graph convolution. The EA-DS module leverages the learned structural correlations to guide the downsampling process, preserving geometrically significant (edge) points and enabling multi-scale feature representation. Finally, the CA-US module reconstructs detailed non-edge geometric features that were lost during downsampling and progressively restores the original point cloud size via a cross-attention mechanism, ensuring structural completeness and feature consistency between the two temporal branches.

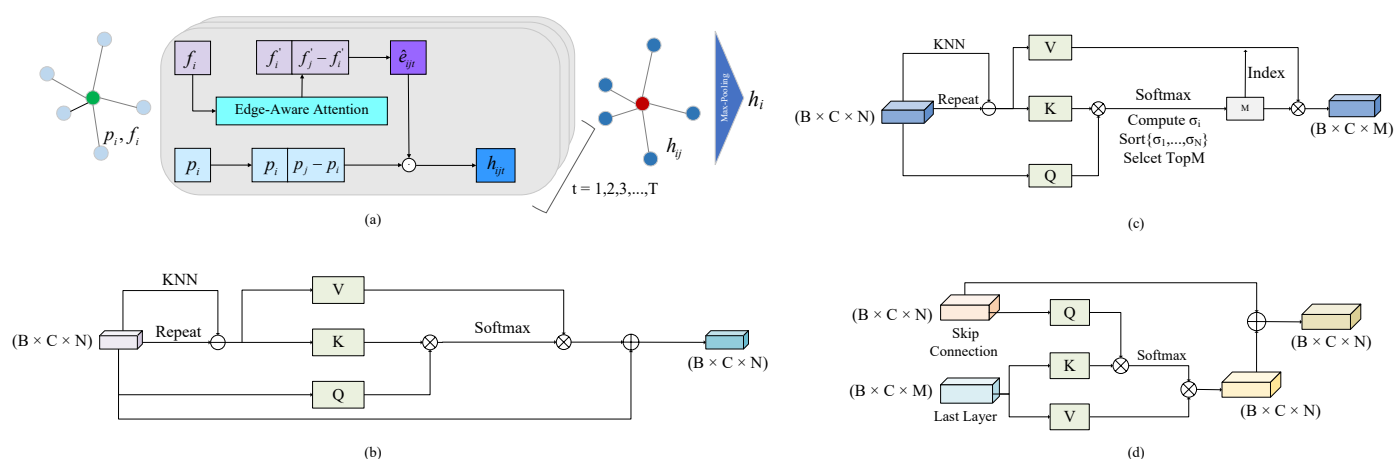


Figure 3. Details of the Edge-Aware Siamese Network. (a) Edge-Aware Adaptive Graph Convolution. (b) Edge-Aware attention. (c) Edge-Aware Downsampling. (d) Cross-Attention Upsampling.

3.2.1. Edge-Aware Adaptive Graph Convolution

To enhance the local structure modeling capability and geometric edge perception of point clouds, based on the Adaptive Graph Convolution (AGConv) [38], we propose Edge-Aware AGConv, which integrates an edge-aware attention mechanism with AGConv, enabling the network to adaptively aggregate local geometric structural information during feature extraction. Specifically, for an input point cloud $\mathbf{P} = \{\mathbf{p}_i \mid i = 1, 2, \dots, N\} \in \mathbb{R}^{N \times 3}$, we first use the KNN algorithm to find K_N local neighbors. For each center point $\mathbf{p}_i$, the correlation between it and its neighbors $\mathbf{p}_{ij}$ is computed as follows:

$$\mathbf{m}_i = \text{softmax} \left(\frac{\mathbf{Q}(\mathbf{f}_i) \mathbf{K}(\mathbf{f}_{ij} - \mathbf{f}_i)^T}{\sqrt{d}} \right) \quad (3)$$

where $\mathbf{Q}(\cdot)$ and $\mathbf{K}(\cdot)$ are linear layers, and d is the feature dimension. The correlation map $\mathbf{m}_i$ measures the consistency between the center point and its neighbor point. Points in structural transition regions tend to show larger feature variations and thus receive higher attention weights.

$$\mathbf{f}_i^{\text{str}} = \mathbf{m}_i \mathbf{V}(\mathbf{f}_{ij} - \mathbf{f}_i) \quad (4)$$

$$\mathbf{f}_i' = \mathbf{f}_i + \mathbf{f}_i^{\text{str}} \quad (5)$$

where $\mathbf{Q}(\cdot)$ is a linear layer. The edge-aware attention explicitly captures local geometric variations. The enhanced feature $\mathbf{f}'_i$ is then processed by the Adaptive Graph Convolution. Unlike fixed kernels, the convolution weights are dynamically generated based on feature and spatial differences:

$$\Delta \mathbf{f}_{ij} = [\mathbf{f}'_i; \mathbf{f}'_{ij} - \mathbf{f}'_i] \quad (6)$$

$$\hat{\mathbf{e}}_{ijt} = g_t(\Delta \mathbf{f}_{ij}) \quad (7)$$

where $[\cdot; \cdot]$ is the concatenation operation and $t = 1, 2, \dots, T$ indicates one of the T output dimensions; $g_t(\cdot)$ denotes a MLP layer used to produce adaptive convolution kernels. The kernel interacts with the spatial offset $\Delta \mathbf{p}_{ij} = [\mathbf{p}_i, \mathbf{p}_{ij} - \mathbf{p}_i]$ to generate edge-adaptive features:

$$\mathbf{h}_{ijt} = \text{RELU}(\langle \hat{\mathbf{e}}_{ijt}, \Delta \mathbf{p}_{ij} \rangle) \quad (8)$$

Finally, we use max-pooling to aggregate the neighborhood edge features to obtain the output features $\mathbf{h}_i$ of the center point $\mathbf{p}_i$.

3.2.2. Edge-Aware Downsampling

After the Edge-Aware AGConv, we obtain the feature $\mathbf{h}_i$ for each center point $\mathbf{p}_i$; it is then downsampled to construct a hierarchical representation and improve computational efficiency. This operation enables the network to capture multi-scale contextual information while reducing redundancy. Traditional random point sampling or farthest point sampling tends to lose critical edge points. To address this, we exploit the correlation map $\mathbf{m}_i$ from the Edge-Aware attention to quantify the geometric significance of each point. Edge points typically exhibit larger standard deviations $\sigma_i = \text{std}(\mathbf{m}_i)$ in their correlation distributions, indicating more intense local feature variation [26]. During downsampling, we prioritize points with higher σ_i :

$$\mathbf{P}_{\text{down}} = \text{Top}_M(\sigma_i \mid i = 1, 2, \dots, N) \quad (9)$$

where $\mathbf{P}_{\text{down}}$ is the set of points after downsampling containing M points, and Top_M refers to the selection of the top M points with the larger standard deviation σ_i .

3.2.3. Cross-Attention Upsampling

The Cross-Attention Upsampling Module is designed to restore the downsampled point clouds to its original number. Since the downsampling process primarily retains edge points, the upsampling layer must effectively reconstruct the information of non-edge points that were discarded during downsampling. We employ a cross-attention mechanism $\text{Att}_c(\cdot)$ to guide the recovery process based on the relationship between the downsampled features and the residual features:

$$\mathbf{P}_{\text{up}} = \text{Att}_c(\mathbf{Q}_{\text{res}}, \mathbf{K}_{\text{down}}, \mathbf{V}_{\text{down}}) \quad (10)$$

where $\mathbf{K}_{\text{down}}$ and $\mathbf{V}_{\text{down}}$ are features of the last layer, and $\mathbf{Q}_{\text{res}}$ is the residual features of the downsampling process. This design enables the network to learn the mapping between preserved and missing points, effectively restoring geometric details.

3.3. Difference Enhancement Module

The details of the Difference Enhancement Module (DEM) are shown in Figure 4. DEM is designed to enhance the difference information between bi-temporal point clouds, thus improving the discrimination of the change regions.

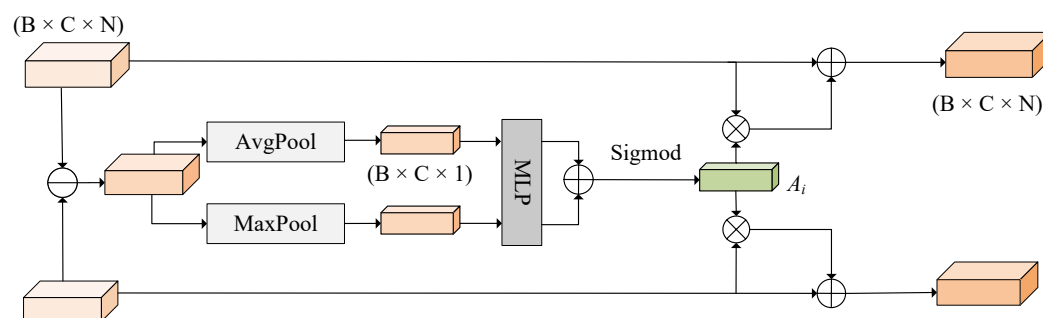


Figure 4. Details of the Difference Enhancement Module.

Firstly, the bi-temporal feature vectors $\mathbf{f}_i^{t1}$ and $\mathbf{f}_i^{t2}$ of point $\mathbf{p}_i$ are fed into the DEM to obtain the feature difference map $\mathbf{D}_i$ as:

$$\mathbf{D}_i = |\mathbf{f}_i^{t1} - \mathbf{f}_i^{t2}| \quad (11)$$

Next, we extract the channel context using global average pooling and maximum pooling. The pooled values are passed through the shared MLP layer to generate the channel attention weights, which are calculated using the Sigmoid activation function. This process is described as follows:

$$\mathbf{A}_i = \sigma(\text{MLP}(\text{Pool}_{\text{Avg}}(\mathbf{D}_i)) + \text{MLP}(\text{Pool}_{\text{Max}}(\mathbf{D}_i))) \quad (12)$$

where $\mathbf{A}_i$ denotes the obtained attention map after the channel attention operation. $\text{Pool}_{\text{Avg}}(\cdot)$ and $\text{Pool}_{\text{Max}}(\cdot)$ refer to the average-pooling and max-pooling operations, respectively. $\sigma(\cdot)$ stands for the Sigmoid function. Afterwards, we enhance the original features of the bi-temporal point clouds using the channel attention weights as follows:

$$\hat{\mathbf{f}}_i^{t1} = \mathbf{A} \cdot \mathbf{f}_i^{t1} + \mathbf{f}_i^{t1}, \quad \hat{\mathbf{f}}_i^{t2} = \mathbf{A} \cdot \mathbf{f}_i^{t2} + \mathbf{f}_i^{t2} \quad (13)$$

Finally, the enhanced bi-temporal features $\hat{\mathbf{f}}_i^{t1}$ and $\hat{\mathbf{f}}_i^{t2}$ with the improved discrimination of the changing regions are provided to the subsequent modules.

3.4. Boundary-Aware BCE Loss

In 3D point cloud change detection tasks, the boundary between changed and unchanged regions is often difficult to identify but is crucial. To further address this, we utilize the boundary points Ω_B identified in the Boundary-Aware Subsampling stage and design a Boundary-Aware Binary Cross-Entropy (BCE) Loss. By assigning higher weights to these boundary points, the network is encouraged to focus more on the boundary regions during training, helping the model concentrate on the subtle differences between changed and unchanged areas, thereby improving the accuracy of boundary detection. The Boundary-Aware BCE Loss is defined as Equations (14)–(16)

$$\mathcal{L}_{\text{BCE}} = -[y_i \log(p(\hat{y}_i)) + (1 - y_i) \log(1 - p(\hat{y}_i))] \quad (14)$$

$$w_i = \begin{cases} \lambda & \text{if } \mathbf{p}_i \in \Omega_B \\ 1 & \text{if } \mathbf{p}_i \notin \Omega_B \end{cases} \quad (15)$$

$$\mathcal{L}_{\text{BABCE}} = \mathbf{W}^{t1} \mathcal{L}_{\text{BCE}}(y_i^{t1}, \hat{y}_i^{t1}) + \mathbf{W}^{t2} \mathcal{L}_{\text{BCE}}(y_i^{t2}, \hat{y}_i^{t2}) \quad (16)$$

where y_i is the true label, 0 indicates no change, and 1 indicates change. $p(\hat{y}_i)$ is the probability of the predicted class by the model, calculated through the Sigmoid function.

$\mathbf{W} = \{\omega_i \mid i = 1, 2, \dots, SN\}$, SN denotes the subsampling number of each sample. If the contribution of the boundary points to the loss is too large, the model may tend to over-adjust the output in these regions. To prevent the model from overfitting the details of the boundary regions, we set $\lambda = 2$. The superscripts t_1 and t_2 represent the two branches at temporal one and two, respectively.

4. Experiment

4.1. Dataset

4.1.1. SLPCCD Dataset

The Street-Level Point Cloud Change Detection (SLPCCD) dataset [14] is established for the detection of changes in point clouds at the street level, including colored point clouds acquired in 2016 and 2020 using vehicle-mounted LiDAR sensors, with each point represented by a six-dimensional vector $[x, y, z, r, g, b]$. Following [14], each original point input contains only a three-dimensional vector $[x, y, z]$ of coordinates. The SLPCCD dataset classifies the changed objects into two types: ‘Removed’ objects that only exist in the earlier point clouds and ‘Added’ objects that are newly included in the later scans. The dataset provides 621 pairs of labeled samples, among which 398 pairs are used for training, 95 pairs for validation, and 128 pairs for testing.

4.1.2. HKCD Dataset

The HKCD dataset is constructed from multi-temporal photogrammetric point clouds in the Kowloon area of Hong Kong, covering approximately 8.1 km² with a total of about 128 million points, featuring high spatial resolution and strong realism. The dataset takes bi-temporal point cloud pairs as input and adopts a point-wise binary labeling scheme, where each point is annotated as changed or unchanged, covering various urban objects such as buildings, terrain, vegetation, vehicles, and street facilities. The annotations are generated through a combination of point cloud registration, automatic screening, and manual refinement, with unchanged points accounting for nearly 90% of the dataset.

4.2. Implementation Details

For our method, we set the epochs as 40 and the batch size as 8. The initial learning rate $lr = 1 \times 10^{-3}$, and then lr is decayed to $0.95 \times lr$. The model is trained with the Adam optimizer with momentum 0.9 and weight decay 5×10^{-3} . All experiments are conducted on an Ubuntu system with Python 3.7, PyTorch 1.11.0, CUDA 11.7, and an NVIDIA RTX3090Ti GPU with 24 GB of memory. We use KNN to find the neighborhood points; the K_N value is set to 4.

Metrics

Three widely used measures, Intersection over Union (IoU), mean IoU (mIoU), and Overall Accuracy (OA), are adopted as criteria to evaluate the performance of the different methods.

$$OA = \frac{TP + TN}{TP + FP + TN + FN} \quad (17)$$

$$IoU = \frac{A \cap B}{A \cup B} \quad (18)$$

$$mIoU = \frac{1}{n} \sum_{i=1}^n IoU_i \quad (19)$$

In addition to IoU, mIoU and OA to evaluate the performance, we measure the complexity of different models using the number of parameters (Params) and floating-point operations (FLOPs).

4.3. Comparison with SOTA Methods

4.3.1. Comparison Methods

To verify the effectiveness of our proposed method, we selected nine 3D CD methods for comparison. These include three methods specifically designed for 3D change detection (3DCDNet [14], PGN3DCD [36], and Ms-DANet [12]), and six methods adapted from other tasks. The first five (PointNet [32], PointNet++ [20], PointMLP [33], PointConv [21], and DGCNN [34]) are general-purpose point cloud feature learning networks. The proposers of 3DCDNet modified the original methods and applied them to the 3D CD task [14]. For the sixth, APES [26], whose edge sampling inspired our work, we modified the original semantic segmentation into a Siamese architecture with an added Nearest Feature Difference (NFD) module. It should be noted that the proposed BAS strategy is only employed during the training stage to guide the network to focus on boundary regions. Because there was no change map label, we replace BAS with the RS during the test stage.

The performance metrics for each comparison method were obtained as follows. Because the source code was not provided, PointNet, PointNet++, PointMLP, PointConv, and DGCNN are cited directly from the 3DCDNet paper [14]; Ms-DANet is taken from its publication [12]. We generated results for 3DCDNet using its official pre-trained weights. The results for PGN3DCD and our adapted version of APES were produced by reproducing their work from the official source code.

4.3.2. Quantitative Analysis

As shown in Table 1, our proposed GSANet outperforms all nine compared methods across key metrics, including mIoU, OA, and class-specific IoU. A detailed analysis is provided from the perspectives of sampling strategy, overall performance, and computational complexity.

Table 1. Comparison of different methods on the SLPCCD dataset.

Method	Subsampling	Downsampling	Upsampling	IoU (%)			mIoU (%)	OA (%)	Flops (G)	Params (M)
				Unchanged	Remove	Add				
PointNet [32]	RS	RS	Interpolation	89.26	25.13	29.64	18.10	89.67	19.21	3.54
PointNet++ [20]	RS	RS	Interpolation	94.80	47.86	60.00	67.55	95.09	3.82	1.89
PointMLP [33]	RS	RS	Interpolation	91.85	46.74	39.64	59.41	92.32	15.05	16.74
PointConv [21]	RS	RS	Interpolation	90.93	16.13	19.51	42.16	91.01	20.67	7.50
DGCNN [34]	RS	RS	Interpolation	93.76	43.73	49.40	62.30	94.07	26.20	1.00
APES [26]	RS	EA-DS	CA-US	96.15	65.58	68.93	76.89	96.43	58.54	8.26
3DCDNet [14]	RS	RS	Interpolation	95.52	63.46	64.37	74.75	95.85	91.76	6.30
PGN3DCD [36]	RS	RS	Interpolation	96.67	74.58	75.25	82.17	96.97	/	/
Ms-DANet [12]	RS	RS	Interpolation	96.88	73.54	75.50	81.97	97.14	106.31	6.37
GSANet (Ours)	BAS	EA-DS	CA-US	97.06	75.47	76.08	82.87	97.30	63.15	8.94

Bold values represent the best-performing results among all compared methods.

- Analysis of Sampling and Processing Strategies.** The comparison highlights fundamental differences in data processing strategies. Our GSANet is the only method to employ Boundary-Aware Subsampling (BAS), which is designed to preferentially retain critical points along the boundaries of changed regions. In contrast, all other methods rely on conventional Random Subsampling (RS), which risks losing these vital details. Furthermore, GSANet and APES both utilize advanced Edge-Aware Downsampling (EA-DS) and Cross-Attention Upsampling (CA-US) to better preserve and reconstruct structural information. This contrasts with the simpler random downsampling and interpolation-based upsampling used by most other baselines, providing a stronger foundation for feature learning.

- Performance Comparison.** Our GSANet achieves the highest mIoU of 82.87% and the best OA of 97.30%. Compared with methods specifically designed for 3D change detection, GSANet shows significant improvements. It outperforms 3DCDNet, PGN3DCD, and Ms-DANet by 8.12%, 0.70%, and 0.90% in mIoU, respectively. The improvements in OA are 1.45%, 0.33%, and 0.16% over the same methods. The results also reveal the limitations of adapting generic point cloud networks for 3D CD. The first five baselines (PointNet to DGCNN) yield relatively lower performance. Notably, APES, although not a dedicated 3D CD method, achieves a strong mIoU of 76.89% by leveraging edge-structure information. Its performance significantly exceeds the other five non-dedicated methods and is even superior to the purpose-built 3DCDNet. The most direct comparison is with APES, which shares our mechanism of utilizing edge information and employs the same advanced downsampling/upsampling modules. Despite these similarities, our GSANet achieves a remarkable 5.98% higher mIoU. This substantial gain can be attributed to several key innovations in our architecture: the Boundary-Aware Subsampling strategy (vs. RS in APES), the more powerful Edge-Aware Adaptive Graph Convolution, the integration of a Difference Enhancement Module for explicit feature interaction, and the targeted Boundary-Aware BCE Loss function.

Table 2 reports quantitative results of all comparison methods on the HKCD urban dataset. GSANet (Ours) achieves the optimal mIoU and OA. Our GSANet achieves the highest OA of 90.42% and the best mIoU of 57.45% on the HKCD urban dataset, surpassing all nine competing state-of-the-art approaches. Compared with specialized 3D point cloud change detection algorithms, GSANet delivers consistent and notable performance gains. It outperforms 3DCDNet, APES, and PGN3DCD by 4.25%, 3.01%, and 1.98% in OA, respectively. For the mIoU metric, our method exceeds 3DCDNet by 8.12%, APES by 5.12%, and PGN3DCD by 0.34%. Class-wise IoU results further demonstrate the superiority of our approach: GSANet obtains the highest IoU for the Add category (89.73%), and competitive IoU scores of 41.33% for Unchanged and 41.28% for Remove classes, balancing detection accuracy across all three semantic categories.

- Complexity Analysis.** In terms of computational complexity, GSANet strikes an effective balance between performance and efficiency. With 63.15 GFlops and 8.94 M parameters, our model is significantly more efficient than top-performing competitors such as 3DCDNet (91.76 GFlops) and Ms-DANet (106.31 GFlops). While it is more computationally intensive than lightweight models such as PointNet++, the substantial improvement in accuracy justifies the moderate increase in complexity. The complexity of PGN3DCD is not reported, as its authors present it as a modular enhancement applicable to various backbones and provide no specific complexity analysis in the original or subsequent work.

Table 2. Comparison of different methods on HKCD dataset.

Method	Subsampling	Downsampling	Upsampling	IoU (%)			OA (%)	mIoU (%)
				Unchanged	Remove	Add		
PointNet [32]	RS	RS	Interpolation	15.48	10.47	78.54	79.20	34.83
PointNet++ [20]	RS	RS	Interpolation	21.23	28.10	86.24	86.93	45.22
DGCNN [34]	RS	RS	Interpolation	25.40	15.14	80.21	81.14	40.25
APES [26]	RS	EA-DS	CA-US	36.17	34.35	86.49	87.41	52.33
3DCDNet [14]	RS	RS	Interpolation	30.82	31.93	85.24	86.17	49.33
PGN3DCD [36]	RS	RS	Interpolation	41.23	41.51	88.59	88.44	57.11
GSANet (Ours)	BAS	EA-DS	CA-US	41.33	41.28	89.73	90.42	57.45

Bold values represent the best-performing results among all compared methods.

4.3.3. Qualitative Analysis

To provide insights beyond quantitative metrics, Figure 5 presents a qualitative comparison on the SLPCCD dataset. We visualize the results of our GSANet and three competitors (APES, 3DCDNet, PGN 3DCD), whose predictions can be reliably generated using their publicly available code or pre-trained models.

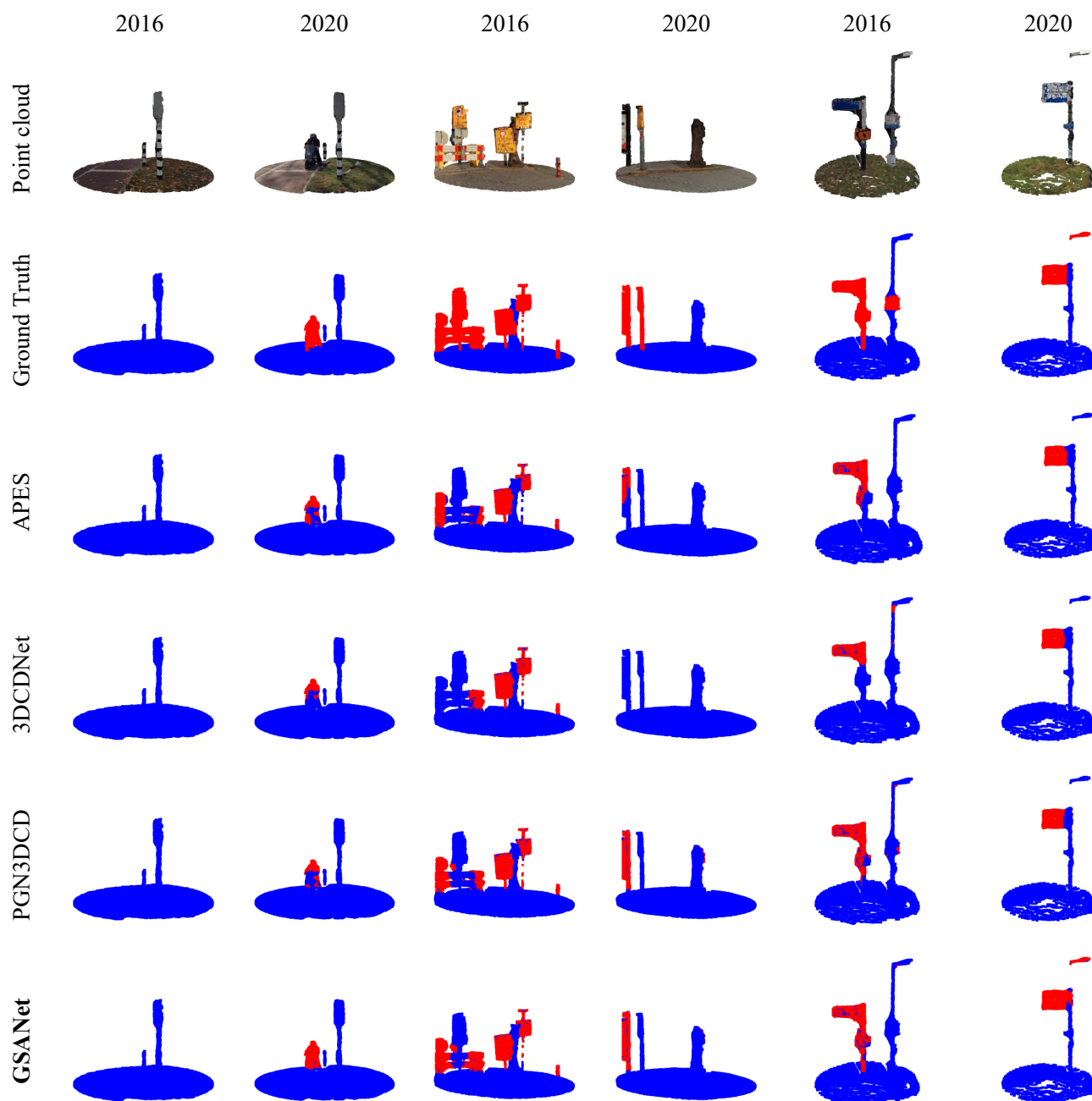


Figure 5. Qualitative comparison of SLPCCD. All of the changed points in the 2016 columns represent point clouds that were “removed” in 2016. All of the changed points in the 2020 columns represent point clouds that were “added” in 2020. Red points represent changed regions, whereas blue points represent unchanged regions.

Across scenarios, our method consistently produces change maps that are cleaner, more complete, and more structurally coherent than those of competitors. For instance, 3DCDNet struggles with both sensitivity and completeness, as it entirely misses the small added object (left column) and produces a highly fragmented detection for the large removed structure (middle column). While APES and PGN3DCD are more effective at

detecting changes, they frequently exhibit a high rate of false positives. This is evident in their results, in which significant portions of unchanged structures, such as the ground or existing flagpoles, are incorrectly flagged as changed, resulting in noisy and imprecise boundaries. In contrast, our GSANet demonstrates a superior ability to precisely localize changes—whether small additions or large removals—while effectively suppressing noise. This visual evidence highlights the benefits of our boundary-aware and edge-aware design, which yields a more robust and accurate feature representation that excels in complex, real-world scenes.

Figures 6 and 7 provide qualitative comparisons on the HKCD urban dataset, visualizing predictions of GSANet, 3DCDNet, APES and PGN3DCD reproduced with official open-source codes. Across local urban patches and large panoramic city scenes, our GSANet generates cleaner, more complete change regions with precise structural contours than all competitors. 3DCDNet suffers from incomplete detection and low sensitivity: it misses large parts of added/removed structures in all scenes, producing fragmented change boundaries. APES yields severe false negatives, leaving obvious hollow areas within real continuous change zones, which explains its lower per-class IoU values. PGN3DCD roughly captures the general range of changes yet introduces massive background false positives, leading to noisy, jagged segmentation edges on unchanged roads and building facades. In contrast, GSANet accurately localizes all scales of urban changes without evident missing detection, while effectively suppressing background noise. Its predicted change boundaries are smooth and highly consistent with ground truth. These visual observations validate the superiority of our boundary-aware design, which builds robust geometric feature representations for complex dense urban point clouds and achieves finer segmentation in diverse city scenarios.

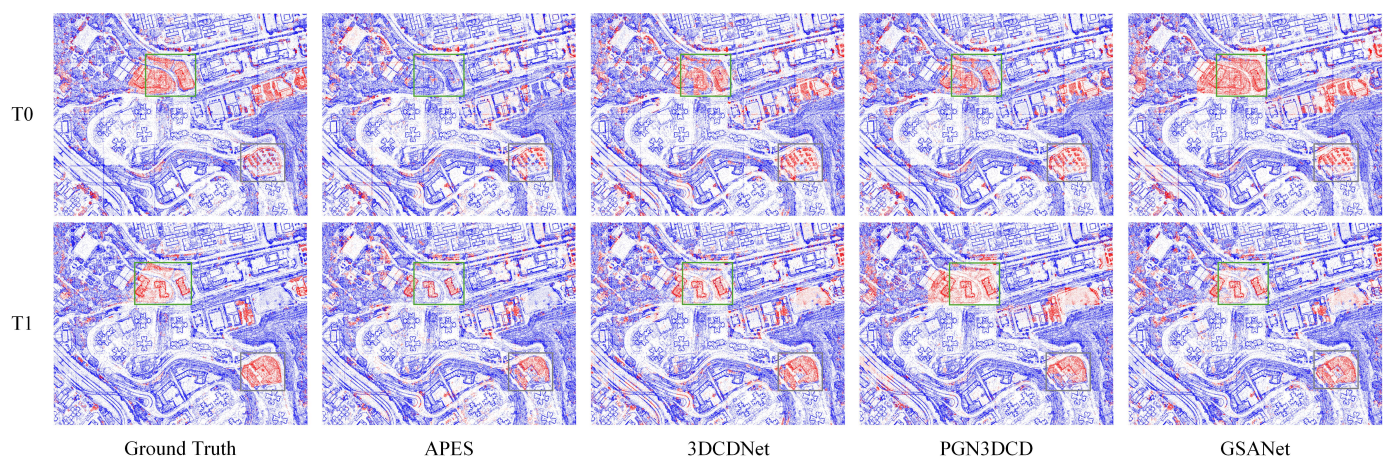


Figure 6. Visual comparison on the large-scale panoramic urban area of the HKCD dataset.

4.4. Ablation Studies

4.4.1. Model Analysis

To explore the influence of different factors on model performance, we conduct ablation studies on the SLPCCD dataset and discuss these factors below.

- **Baseline.** We adopt random sample for downsampling, Nearest Neighbor Interpolation for upsampling and conventional Adaptive Graph Convolution (AGConv) [38] for feature extraction. The encoder–decoder structure and the Nearest Feature Difference (NFD) module remain unchanged.
- **AS1: The effect of Edge-Aware AGConv.** On the basis of the baseline, we replace the AGConv with Edge-Aware AGConv, while keeping the subsampling, upsampling, and downsampling modules unchanged.

- **AS2: The effect of Edge-Aware Downsampling.** Based on AS1, we further introduce the Edge-Aware downsampling module, whereas the up-sampling strategy and other network components remain unchanged.
- **AS3: The effect of Cross-Attention Upsampling.** On the basis of AS2, we additionally incorporate the Cross-Attention upsampling module to enhance feature propagation during decoding.
- **AS4: The effect of the Difference Enhancement module.** Based on AS3, we further add the DEM module, while keeping the rest of the network unchanged.
- **AS5: The effect of Boundary-Aware Subsampling (BAS).** On the basis of AS4, we replace the random subsampling strategy with the proposed BAS to evaluate its contribution to boundary information preservation.
- **AS6: The effect of BA-BCE Loss.** Finally, based on AS5, we further introduce the BA-BCE Loss to improve boundary-sensitive supervision.

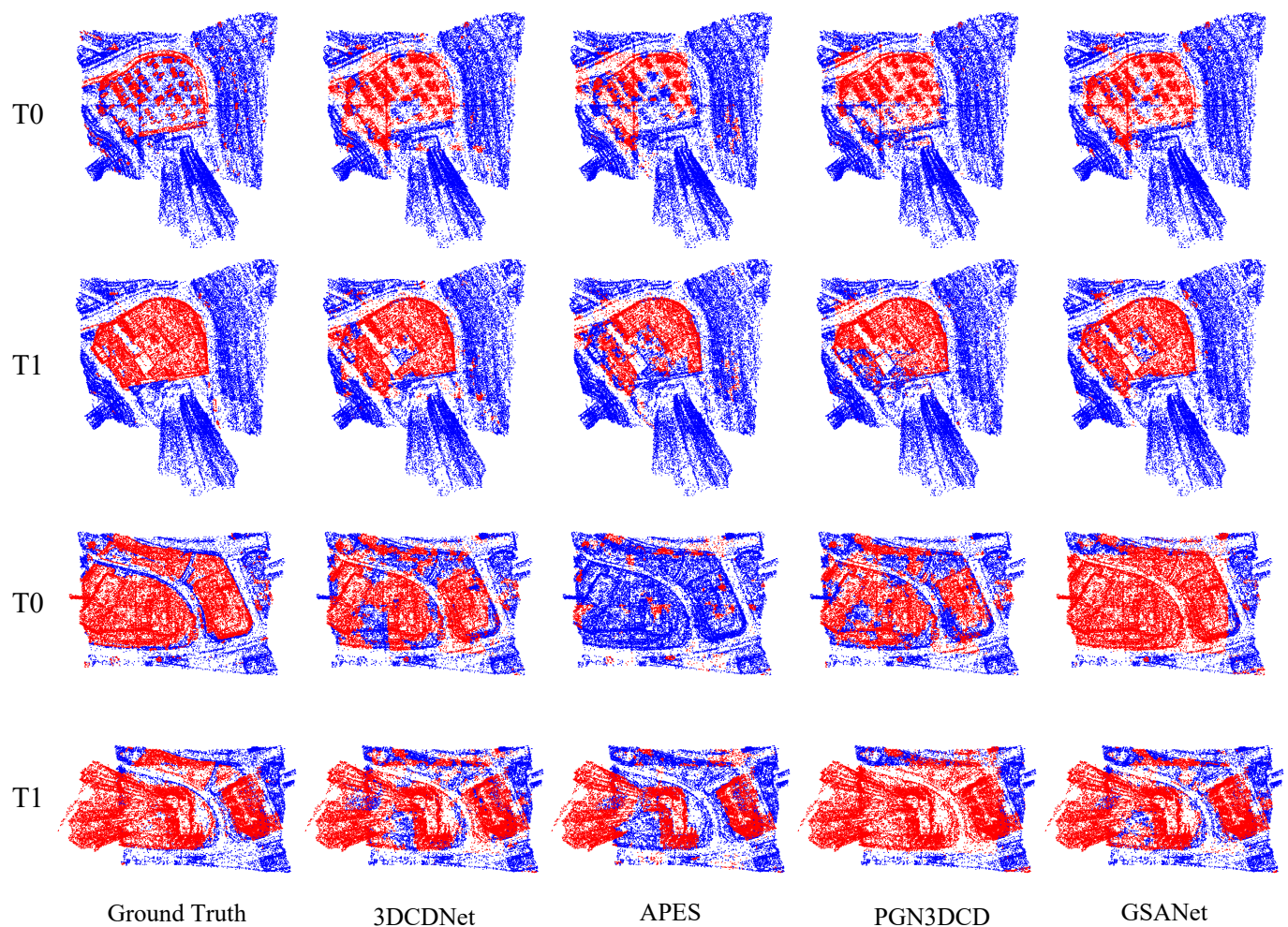


Figure 7. Qualitative comparison of change detection results on local scenes of the HKCD dataset. Upper: corresponds to the gray-boxed region in Figure 6. Lower: corresponds to the green-boxed region in Figure 6. Red regions denote changed areas, and blue regions denote unchanged areas.

The quantitative results of the above ablation studies are reported in Table 3. The Baseline model achieves mIoU of 68.78% and OA of 94.76%. In AS1, we introduce the Edge-Aware AGConv while keeping the rest of the network unchanged. This modification leads to a noticeable performance improvement, increasing the mIoU to 70.11% and the OA to 94.92%, indicating that Edge-Aware feature extraction benefits local geometric representation. AS2 further incorporates the Edge-Aware Downsampling module on top

of AS1. By explicitly preserving boundary-related structures during the downsampling process, the mIoU is improved to 71.72%, with OA rising to 95.19%, demonstrating the effectiveness of boundary-aware sampling in mitigating information loss. In AS3, we additionally introduce the Cross-Attention Upsampling module. This enables more effective feature fusion during decoding, resulting in a significant performance gain, with mIoU reaching 77.02% and OA increasing to 96.54%. Building upon AS3, AS4 incorporates the Difference Enhancement Module. By strengthening cross-temporal feature contrast, DEM further boosts the detection capability, raising the mIoU to 80.72% and OA to 96.83%, which confirms its effectiveness in highlighting real change regions. In AS5, we replace random subsampling with the proposed Boundary-Aware Subsampling, achieving an mIoU of 82.00% and an OA of 97.13%, validating the importance of preserving boundary points for accurate change detection. Finally, AS6 integrates the Boundary-Aware Binary Cross-Entropy loss. With boundary-sensitive supervision, the model attains the best overall performance, achieving an mIoU of 82.87% and an OA of 97.30%. This demonstrates that the BA-BCE loss further enhances the network's focus on boundary regions, resulting in more accurate and reliable change detection.

Table 3. Quantitative results of the ablation study.

Ablation Study	BAS	Edge-Aware Siamese Network			DEM	BA-BCE Loss	mIoU (%)	OA (%)
		EA-AGConv	EA-DS	CA-US				
Baseline	RS	AGConv	RS	Interpolation	Without	BCE	68.78	94.76
AS1	RS	✓	RS	Interpolation	Without	BCE	70.11	94.92
AS2	RS	✓	✓	Interpolation	Without	BCE	71.72	95.19
AS3	RS	✓	✓	✓	Without	BCE	77.02	96.54
AS4	RS	✓	✓	✓	✓	BCE	80.72	96.83
AS5	✓	✓	✓	✓	✓	BCE	82.00	97.13
AS6	✓	✓	✓	✓	✓	✓	82.87	97.30

The symbol ✓ indicates that the corresponding proposed module is adopted in the model, Bold values indicate the best performance among all compared methods.

4.4.2. Edge Hyperparameters K_E Analysis

In this paper, we design the Edge-Aware Adaptive Graph Convolution for point cloud feature extraction. This module employs the KNN algorithm to identify neighboring points for each point in the cloud, making the choice of the K_E value a critical factor affecting model performance. As shown in Table 4, the model achieves the best performance when K_E is set to 4. When K_E is too small, the limited number of neighboring points makes it difficult to construct complete local structural information. Conversely, when K_E is too large, too many irrelevant points may be included in the neighborhood, which reduces the discriminability of edge features and negatively impacts feature extraction.

Table 4. Effect of the Edge Hyperparameter K_E .

K_E	mIoU (%)	OA (%)
1	79.28	96.29
4	82.87	97.30
8	82.63	97.28
12	81.39	96.94
16	80.15	96.76

Bold values represent the best results among all competing methods.

4.5. Boundary-Aware Subsampling Analysis

4.5.1. Boundary Hyperparameter K_B Analysis

In the boundary-aware subsampling strategy during training, we use the ground-truth map (label 1 for changed, 0 for unchanged) to extract boundary points using the KNN algorithm. If any nearest neighbor of each point $\mathbf{p}_i$ has a different label, $\mathbf{p}_i$ is treated as a boundary point. The hyperparameter K_B , which controls the neighborhood size, is critical for the effectiveness of boundary detection between changed and unchanged regions. As shown in Table 5, the model achieves optimal performance when $K_B = 16$, with performance degrading as K_B deviates from this value. If K_B is too small, the boundary is confined to an overly narrow local area, potentially failing to capture the structural transitions between regions and thus omitting critical boundary points. Conversely, when K_B is too large, the neighborhood incorporates too many distant, irrelevant points. This influx of noise not only reduces the precision of boundary identification but also impedes the model's ability to learn discriminative boundary features.

Table 5. Effect of the Boundary Hyperparameter K_B .

K_B	mIoU (%)	OA (%)
1	80.74	96.78
4	80.89	96.89
8	82.06	97.23
12	82.17	96.97
16	82.87	97.30
20	78.52	96.72

Bold values represent the best results among all competing methods.

4.5.2. Boundary Point Proportion Analysis

Subsampling in point cloud change detection presents a critical trade-off between computational efficiency and information fidelity. To further evaluate the effectiveness of our proposed Boundary-Aware Subsampling (BAS) strategy, especially under varying point densities, we conducted a comparative analysis against conventional Random Subsampling (RS). We established five levels of point cloud density by setting the number of sampled points (SN) to 8192, 6144, 4096, 2048, and 1024. Our method, GSANet, employs the BAS strategy, while the baseline method, 3DCDNet, utilizes RS.

The results are presented in Figure 8, which comprises two key analyses. The left panel illustrates the proportion of boundary points successfully retained by each subsampling strategy. The right panel shows the direct impact of these strategies on the final change detection performance, measured in mIoU (%). This dual analysis compellingly demonstrates the superiority of our approach.

Analysis of Boundary Point Proportion highlights the stability and superiority of our BAS strategy. Across all sampling densities, BAS consistently and significantly preserves a higher proportion of boundary points compared to its random counterpart. Even in the most sparse scenario with only 1024 points, BAS retains a substantial 3.3% of boundary points. This robust performance stems from our strategy's explicit prioritization of sampling from the boundary set, thereby preserving this critical information. In contrast, RS showed that the boundary point ratio drops sharply with the total point count, plummeting from 2% at 8192 points to 0.5% at 1024 points. This confirms that purely random subsampling disproportionately discards the vital boundary points as the point cloud density decreases.

The disparity in boundary point retention directly translates into a significant gap in final change detection performance, as shown in the right panel. Our GSANet (with BAS) substantially outperforms the 3DCDNet baseline (with RS) across all tested condi-

tions. Crucially, this performance gap widens as the point cloud becomes sparser. In the sparsest 1024-point setting, our method maintains a strong 68.47% mIoU, while 3DCDNet's performance degrades sharply to 50.75%. This widens the performance margin to 17.72%, demonstrating that our model is more robust, especially in sparse point clouds. This compellingly validates our central hypothesis: by retaining crucial boundary information, BAS provides a higher-quality input that enables the network to learn more discriminative features, leading to more accurate and reliable change detection.

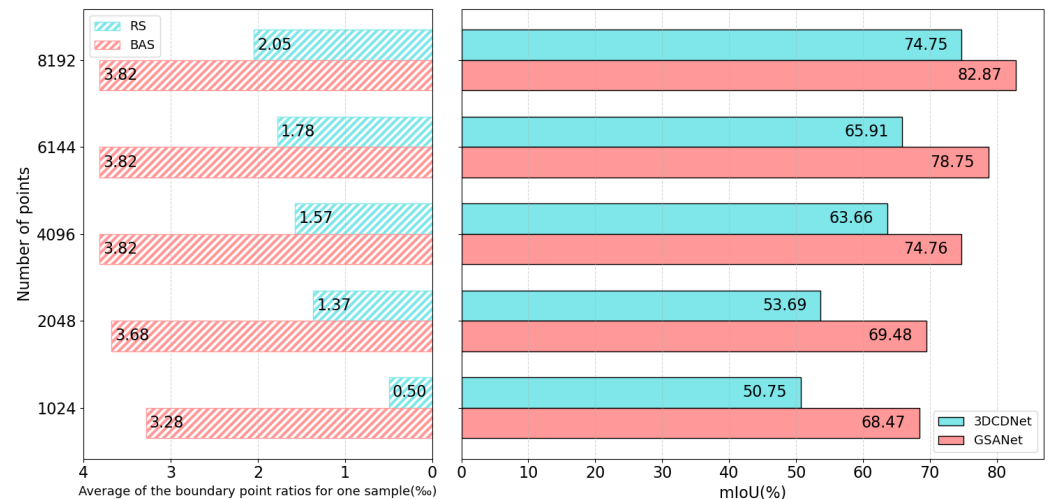


Figure 8. Boundary-point retention ratio and change-detection performance under different sampling point numbers.

5. Discussion

The experimental results substantiate the effectiveness of explicitly embedding geometric priors into the 3D point cloud change detection framework. In contrast to conventional detectors that rely on random subsampling, GSANet introduces a Boundary-Aware Subsampling (BAS) strategy alongside an Edge-Aware Siamese Network, jointly guiding the model to preferentially retain points proximal to change–unchange boundaries while reinforcing local shape structure encoding. This design proves particularly advantageous for handling sparse point clouds, complex urban environments, and subtle geometric deformations.

From the perspective of computational efficiency, GSANet introduces additional geometric modeling operations compared with relatively simple point-based networks. However, these operations provide a favorable balance between detection accuracy and computational cost. GSANet achieves higher detection performance while maintaining a moderate model complexity, with 8.94M parameters and 63.15 GFLOPs. Compared with some high-performing methods with larger computational requirements, GSANet does not rely on substantially increased model size or expensive hardware configurations to achieve performance improvements. Therefore, the proposed framework provides an effective solution by improving geometric representation capability without introducing excessive computational overhead.

Our analysis further highlights the critical role of the BAS strategy under varying point densities. As the sampling budget decreases from 8192 to 1024 points, random subsampling leads to a precipitous drop in boundary point retention from 2% to 0.5%. In contrast, BAS consistently preserves over 3.3% boundary points across all tested densities. Consequently, GSANet maintains a robust 68.47% mIoU under extreme sparsity, substantially surpassing 3DCDNet's 50.75%. These results collectively confirm that explicit geometric prior modeling effectively counteracts information degradation due to data sparsity, offering a viable pathway for reliable change detection in low-density point cloud scenarios.

Despite these advantages, GSANet still has several limitations. First, although edge-aware modules improve geometric feature representation, the additional geometric modeling operations may increase computational costs, which requires further optimization for large-scale and real-time applications. Second, the current framework is mainly evaluated on benchmark datasets, and further validation under more diverse real-world scenarios with varying point densities, acquisition conditions, and registration qualities is still needed. In future work, we will investigate more adaptive and lightweight geometric-aware strategies to improve computational efficiency while maintaining detection accuracy. We also plan to explore broader real-world applications and more flexible geometric structure learning approaches to further enhance the practical applicability of GSANet.

6. Conclusions

In this paper, we propose GSANet: a Geometric Structure-Aware Siamese Network for 3D Change Detection, designed to address insufficient modeling of structural information and inadequate exploitation of difference cues. Specifically, it employs boundary-aware sampling and a boundary-weighted loss to focus learning on the transition regions between changed and unchanged areas. Meanwhile, an edge-aware Siamese architecture (including Edge-Aware Adaptive Graph Convolution, Edge-Aware Downsampling, and Cross-Attention Upsampling) enhances geometric feature representation and cross-temporal consistency. In addition, a difference enhancement module explicitly amplifies bi-temporal feature discrepancies. The proposed framework demonstrates the effectiveness of integrating geometric structures into 3D change detection and provides a more robust solution for capturing subtle changes in irregular point clouds. Extensive experiments and ablation studies demonstrate that each component contributes to performance improvements and that GSANet achieves competitive performance compared with existing state-of-the-art approaches. A key finding is that our method's robustness is especially evident in limited point clouds. Most existing methods employ random subsampling, which leads to a significant loss of critical boundary information, particularly in sparse point clouds. Our BAS strategy directly counters this by intelligently retaining these points, providing a more informative input to the network. This enables GSANet to maintain high performance even with a limited number of points. In future work, we will investigate more adaptive BAS strategies to further improve robustness under extremely sparse or noisy point cloud settings. We also plan to evaluate GSANet in broader real-world scenarios to further assess its practical applicability.

Author Contributions: Conceptualization, R.W. and J.C.; methodology, R.W. and J.C.; software, R.W. and J.C.; validation, R.W. and J.C.; formal analysis, L.S.; investigation, R.W.; data curation, C.J.; writing—original draft preparation, J.C.; writing—review and editing, R.W. and J.C.; visualization, J.C.; supervision, L.S.; project administration, C.J.; funding acquisition, R.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Industry Innovation Chain Project of Shaanxi under Grant No. 2024NCZDCYL-05-04.

Data Availability Statement: The datasets (SLPCCD, HKCD) analyzed in this study are publicly available. The code, and models are available at <https://github.com/Code-RFGroup/GSANet> (accessed on 26 June 2026).

Acknowledgments: During the preparation of this manuscript, the authors used GPT-5.5 Thinking model (OpenAI, San Francisco, CA, USA) for language polishing, grammar correction, and improvement of textual clarity and coherence. The authors have reviewed and edited the output and take full responsibility for the content of this publication. The authors thank the editors and anonymous reviewers for their constructive feedback.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zheng, H.; Zhang, M.; Gong, M.; Qin, A.K.; Liu, T.; Jiang, F. Multi-scale hierarchical feature fusion network for change detection. *Pattern Recognit.* **2025**, *161*, 111266. [\[CrossRef\]](#)
2. Zhao, J.; Xie, J.; Zhou, Y.; Du, W.L.; Yao, R.; El Saddik, A. ST-mamba: Spatio-temporal synergistic model for remote sensing change detection. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*, 4412413. [\[CrossRef\]](#)
3. Zhao, J.; Xie, J.; Zhou, Y.; Du, W.L.; Zhu, H.; Yao, R. Causal decoupling domain generalization for remote sensing change detection. *Proc. AAAI Conf. Artif. Intell.* **2026**, *40*, 13135–13143. [\[CrossRef\]](#)
4. Tian, D.; Gong, M.; Li, J.; Shi, J. 3D point cloud classification network with hybrid sampling enhancement and point energy attention. *Pattern Recognit.* **2025**, *172*, 112791.
5. Bi, J.; Wu, Q.; Qian, J.; Luo, L.; Yang, J. Structure-aware spherical density steered cross-domain learning for effective point cloud understanding. *Pattern Recognit.* **2025**, *172*, 112527.
6. Gao, F.; Jin, Y.; Liu, Y.; Li, S.; Zheng, H.; Yu, J. Bidirectional Cross-Modal Image-Guided Point Cloud Completion with Multi-Scale Progressive Refinement. *Pattern Recognit.* **2026**, *175*, 113123. [\[CrossRef\]](#)
7. Wen, C.; Long, J.; Yu, B.; Tao, D. PointWavelet: Learning in spectral domain for 3-D point cloud analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *36*, 4400–4412.
8. Girardeau-Montaut, D.; Roux, M.; Marc, R.; Thibault, G. Change detection on points cloud data acquired with a ground laser scanner. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2005**, *36*, W19.
9. Lague, D.; Brodu, N.; Leroux, J. Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (NZ). *ISPRS J. Photogramm. Remote Sens.* **2013**, *82*, 10–26. [\[CrossRef\]](#)
10. Liu, K.; Boehm, J.; Alis, C. Change detection of mobile LIDAR data using cloud computing. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *41*, 309–313. [\[CrossRef\]](#)
11. Tran, T.H.G.; Ressel, C.; Pfeifer, N. Integrated change detection and classification in urban areas based on airborne laser scanning point clouds. *Sensors* **2018**, *18*, 448. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Lu, J.; Dai, C.; Zhang, Z.; Liu, X.; Zhou, R.; Ji, S.; Zhang, Y.; Guan, H.; Wang, H. Ms-DANet: Multi-scale Difference-aware Network for 3D Point Cloud Change Detection. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*, 5702113. [\[CrossRef\]](#)
13. de Gélis, I.; Lefèvre, S.; Corpetti, T. Siamese KPConv: 3D multiple change detection from raw point clouds using deep learning. *ISPRS J. Photogramm. Remote Sens.* **2023**, *197*, 274–291. [\[CrossRef\]](#)
14. Wang, Z.; Zhang, Y.; Luo, L.; Yang, K.; Xie, L. An end-to-end point-based method and a new dataset for street-level point cloud change detection. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5703015. [\[CrossRef\]](#)
15. Wang, R.; Wang, L.; Wei, X.; Chen, J.W.; Jiao, L. Dynamic graph-level neural network for SAR image change detection. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 4501005. [\[CrossRef\]](#)
16. Wang, R.; Wang, W.; Dong, P.; Haojiang, W.; Jiao, L.; Chen, J.W. SAR image change detection via a few-shot learning-based neural network. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 5287–5290.
17. Wang, R.; Liu, C.; Wang, L.; Sun, L.; Dara Ly, V.; Jiao, C. Fine-grained meta-learning with semantic augmentation for synthetic aperture radar change detection. *J. Appl. Remote Sens.* **2025**, *19*, 046502. [\[CrossRef\]](#)
18. Chen, Z.; Zhou, Y.; Wang, B.; Xu, X.; He, N.; Jin, S.; Jin, S. EGDE-Net: A building change detection method for high-resolution remote sensing imagery based on edge guidance and differential enhancement. *ISPRS J. Photogramm. Remote Sens.* **2022**, *191*, 203–222. [\[CrossRef\]](#)
19. Eldar, Y.; Lindenbaum, M.; Porat, M.; Zeevi, Y.Y. The farthest point strategy for progressive image sampling. *IEEE Trans. Image Process.* **1997**, *6*, 1305–1315. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017; Volume 30.
21. Wu, W.; Qi, Z.; Fuxin, L. Pointconv: Deep convolutional networks on 3d point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9621–9630.
22. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigi, N.; Markham, A. Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11108–11117.
23. Dovrat, O.; Lang, I.; Avidan, S. Learning to sample. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 2760–2769.
24. Lang, I.; Manor, A.; Avidan, S. Samplenet: Differentiable point cloud sampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 7578–7588.
25. Wang, X.; Jin, Y.; Cen, Y.; Wang, T.; Tang, B.; Li, Y. Lightn: Light-weight transformer network for performance-overhead tradeoff in point cloud downsampling. *IEEE Trans. Multimed.* **2023**, *27*, 832–847. [\[CrossRef\]](#)

26. Wu, C.; Zheng, J.; Pfroimmer, J.; Beyerer, J. Attention-based point cloud edge sampling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023.
27. Chen, X.; Wu, Y.; Xu, W.; Li, J.; Dong, H.; Chen, Y. Pointscnet: Point cloud structure and correlation learning based on space-filling curve-guided sampling. *Symmetry* **2021**, *14*, 8. [[CrossRef](#)]
28. Wang, C.; Wu, M.; Lam, S.K.; Ning, X.; Yu, S.; Wang, R.; Li, W.; Srikanthan, T. Gpsformer: A global perception and local structure fitting-based transformer for point cloud understanding. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 75–92.
29. Han, T.; Chen, Y.; Ma, J.; Liu, X.; Zhang, W.; Zhang, X.; Wang, H. Point cloud semantic segmentation with adaptive spatial structure graph transformer. *Int. J. Appl. Earth Obs. Geoinf.* **2024**, *133*, 104105. [[CrossRef](#)]
30. Xue, Y.; Wang, G.; Fan, X.; Yu, L.; Tian, S.; Zhang, H. SAPCNet: Symmetry-aware point cloud completion network. *J. Electron. Imaging* **2024**, *33*, 053031. [[CrossRef](#)]
31. Zhang, Z.; Vosselman, G.; Gerke, M.; Persello, C.; Tuia, D.; Yang, M. Change detection between digital surface models from airborne laser scanning and dense image matching using convolutional neural networks. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *4*, 453–460. [[CrossRef](#)]
32. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
33. Ma, X.; Qin, C.; You, H.; Ran, H.; Fu, Y. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. *arXiv* **2022**, arXiv:2202.07123.
34. Phan, A.V.; Le Nguyen, M.; Nguyen, Y.L.H.; Bui, L.T. Dgcnn: A convolutional neural network over large-scale labeled graphs. *Neural Netw.* **2018**, *108*, 533–543. [[CrossRef](#)] [[PubMed](#)]
35. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6411–6420.
36. Zhan, W.; Cheng, R.; Chen, J. PGN3DCD: Prior-Knowledge-Guided Network for Urban 3D Point Cloud Change Detection. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5705415. [[CrossRef](#)]
37. Zhang, L.; Wang, H.; Liu, C.; Dong, Z.; Yang, B. ME-CPT: Multi-Task Enhanced Cross-Temporal Point Transformer for Urban 3D Change Detection. *arXiv* **2025**, arXiv:2501.14004.
38. Wei, M.; Wei, Z.; Zhou, H.; Hu, F.; Si, H.; Chen, Z.; Zhu, Z.; Qiu, J.; Yan, X.; Guo, Y.; et al. AGConv: Adaptive graph convolution on 3D point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 9374–9392. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.