PGN3DCD: Prior-Knowledge-Guided Network for Urban 3-D Point Cloud Change Detection

Wenxiao Zhan, Ruozhen Cheng, and Jing Chen

***Abstract*— Efficient and intelligent 3-D change detection is a fundamental problem in urban data updating, management, and planning. However, this paradigm faces some challenges: 1) there is a lack of publicly available datasets of realistic urban 3-D change detection and 2) the long-tailed distribution of 3-D variation is intractable for existing 3-D change detection methods. To address these issues, we present the first urban-scale realistic point cloud change detection dataset from Hong Kong with nearly 128 million binary-change annotated points, called the Hong Kong change detection (HKCD) dataset, which is the largest realistic point cloud change detection dataset. This dataset consists of photogrammetric point clouds from Hong Kong, covering about 8.1 km^2 of the city landscape. Moreover, we propose a novel prior-knowledge-guided network for 3-D point cloud change detection (PGN3DCD). Specifically, we build a nonparametric and generalizable 3-D change detection prior knowledge, which guides the neural network to focus on the changes, thus more accurately characterizing the variations. Experimental results on several 3-D change detection benchmarks show a significant improvement, demonstrating the innovation and sophistication of PGN3DCD for 3-D change detection.**

***Index Terms*— 3-D change detection, dataset, deep learning, point cloud, prior knowledge.**

I. Introduction

THREE-DIMENSIONAL change detection is the technique of observing the same 3-D space several times in different periods and obtaining the differences and changes therein through comparative analysis [1]. With the improvement of 3-D data accuracy and the shortening of the data acquisition cycle, 3-D change detection is beginning to be gradually popularized and widely used in various fields, such as resource monitoring [2], urban planning [3], and land management [4]. Due to the simple data structure, high data accuracy, and easy accessibility, the point cloud is the mainstream data for 3-D change detection [5]. However, the internal and external factors of each data acquisition are different, resulting in inconsistent density and occlusions [6], which will produce a large number of false positive samples, thus confusing the detection of true positive change objects. It brings great challenges to 3-D change detection.

Existing 3-D change detection methods can be roughly divided into two categories: traditional methods and deep learning. Traditional methods are mainly algebra-based (height difference [7], distance difference [8]), classification-based (random forest [9], support vector machine (SVM) [10]), and others (Markov random field [11], octree [12]). Most of these methods rely on hand-crafted features that cannot comprehensively represent the point cloud and are less generalizable and portable, which will lead to a decrease in accuracy. Deep learning-based methods always establish a Siamese structure from two feature extraction backbone networks [13], [14], and acquire the feature differences between different temporal point clouds to detect the change, which has prominent learning and generalization capabilities [15], [16]. An outstanding feature difference calculation method is an important basis for realizing efficient 3-D change detection, including direct difference methods [17], multiscale difference methods [18], and feature fusion methods [19]. These methods can easily access the differences between point clouds by subtracting the corresponding features or fusing multiple temporal features but suffer from the following problems: 1) insufficient change features. Adequate change features can effectively improve the performance of the model, but the existing methods can only implicitly extract the change features and scantily express the change-related features and 2) inadequate attention to variation. Change detection is a typical long-tail problem, but existing change detection methods are overly focused on unchanged information, inhibiting the learning of change information. These problems will greatly limit the performance of 3-D change detection methods.

Datasets are necessary conditions for the construction of 3-D change detection models. The 3-D change detection dataset is mainly divided into realistic data and synthetic data. Existing realistic 3-D change detection datasets are mainly acquired by the mobile LiDAR system (MLS) [17], [20], [21], [22], which focuses on changes in the streetscape, e.g., vehicle movement, street furniture construction, etc. Synthetic datasets manually create changes through a simulator [23], [24]. However, due to the limitations of scenarios and authenticity, both street realistic data and urban synthetic data are unable to accurately describe changes in urban realistic scenarios, and there is no publicly available urban realistic dataset for 3-D change detection, which greatly restricts the implementation of 3-D change detection in urban.

Manuscript received 9 February 2024; revised 6 June 2024 and 11 July 2024; accepted 27 July 2024. Date of publication 1 August 2024; date of current version 12 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 42371417. *(Corresponding author: Jing Chen.)*

Wenxiao Zhan and Jing Chen are with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China (e-mail: zhanwenxiao@whu.edu.cn; jchen@whu.edu.cn).

Ruozhen Cheng is with Changjiang Survey, Planning, Design and Research Company Ltd., Wuhan 430010, China (e-mail: chengruozhen@cjwsjy.com.cn).

Digital Object Identifier 10.1109/TGRS.2024.3436854

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

To tackle the above challenges, we build a large-scale urban realistic 3-D change detection dataset, termed the Hong Kong change detection (HKCD) dataset, which is composed of photogrammetry 3-D point clouds from Hong Kong. HKCD dataset is richly annotated with binary-change of trees, buildings, vehicles, terrain, and street furniture in the city, covering about 8.1 km^2 of the city landscape. Then, we propose a novel prior-knowledge-guided 3-D change detection approach. This approach introduces a nonparametric and generalizable 3-D change detection prior knowledge, which effectively guides the neural network to pay full attention to the differences of the multitemporal point clouds, and strengthens the learning capability of variations in the city as shown in Fig. 1. Experiments on benchmarks demonstrate the superiority of the proposed methods. The contributions are as follows.

1) A publicly available urban-scale photogrammetric 3-D point cloud change detection dataset is established, richly annotated by binary change in urban, which is the largest urban 3-D change detection dataset. The dataset is available at https://github.com/zhanwenxiao/HKCD.
2) The nonparametric 3-D change detection prior knowledge incorporated with the deep learning method is proposed, which adequately describes the prior variations from both geometry and texture perspectives.
3) A novel prior-knowledge-guided 3-D change detection deep learning method is proposed, which employs the proposed 3-D change detection prior knowledge to reinforce the characterization of variations, achieving efficient and interpretable urban 3-D change detection. The code is available at https://github.com/zhanwenxiao/PGN3DCD.
4) Experiments on several benchmarks demonstrate the superiority of the proposed method on scene and instance 3-D change detection.

II. Related Work

A. Three-Dimensional Point Clouds Change Detection

Three-dimensional change detection is an important method for detecting variation in 3-D scenes. Compared to 2-D image change detection, 3-D change detection is able to detect vertical changes, which is more in line with the needs of the actual physical world. Three-dimensional point cloud change detection can be categorized into algebra-based methods, machine learning-based methods, and deep learning-based methods.

The algebra-based methods determine whether point clouds have changed by calculating the differences in height, distance, or texture over multiple temporal point clouds and comparing them to the threshold [26], [27], [28]. Among them, the most widely used is to calculate the spatial Euclidean distance between point clouds, and realize the change detection of the point clouds by judging whether the distance exceeds the threshold or not [29]. These methods are simple and easy to implement, but there is a strong subjectivity in the setting of the threshold, different scenes need to match different thresholds, and the robustness and stability are poor, which is not able to effectively deal with the 3-D change detection of complex scenes.

The machine learning-based approaches detect variations in point clouds by machine learning methods such as SVM [10], [30], Random Forest [9], [31], conditional random field (CRF) [11], [22], etc. SVM focuses on binary changes, Random Forest focuses on multiclass changes, and CRF allows for the incorporation of more influences in the model to improve the accuracy of change detection [31]. These methods rely on meticulous mathematical paradigms to achieve accurate 3-D change detection, but they need hand-crafted change features, which are highly specialized and proprietary, unable to effectively respond to various urban scenarios, with poor generalizability and portability.

The deep learning-based methods use neural networks to extract point cloud features and perform change detection based on these features. These methods usually use classic feature extraction backbone to capture the point features, such as PointNet [32], PointNet++ [33], PointMLP [34], PointConv [35], DGCNN [36], and KPConv [18], and subtracts these features to obtain the change features. However, a simple subtraction of backbone features does not provide a comprehensive representation of variations. To make change features more adequate, de Gélis et al. [25] organically couple the change decoder with the feature encoder and propose a series of change models, such as OneConvFusion, Triplet KPConv [25], Encoder Fusion SKPconv [25], etc. Wang et al. [17] propose a 3DCDNet, which constructs a more efficient semantic feature extraction backbone network and algebraically differentiates the point cloud features at the end of the network. These methods are highly accurate and adaptable, but only calculating differences between different temporal point clouds through algebraic differences will lead to a deficient representation of change features, as well as insufficient attention to variations.

B. Three-Dimensional Change Detection Dataset

Three-dimensional change detection datasets are the necessary basis for deep learning methods for 3-D change detection. Different from datasets for downstream tasks such as semantic segmentation, 3-D change detection datasets require multiple acquisitions of the same region at different times, which greatly increases the difficulty of production, resulting in the fact that there are currently only a relatively small number of 3-D change detection datasets available. Currently, to the best of our knowledge, there are only five publicly available datasets for 3-D change detection, namely URB3DCD [23], SLPCCD [17], Change3D [20], ZTAKIBudapest [22], and BD dataset [21]. URB3DCD is the synthetic airborne laser scanning (ALS) point cloud dataset, whose variations are generated by a simulator. It has rich annotations, but the simulated data lacks realism and is not a substitute for the representation of a real scene. SLPCCD, Change3D, ZTAKIBudapest, and BD dataset are all MLS streetscape point clouds with rich annotation and large footprint but differ in that SLPCCD is for variations in points of interest (POIs) in the street, which is essentially the detection change of instance rather than the

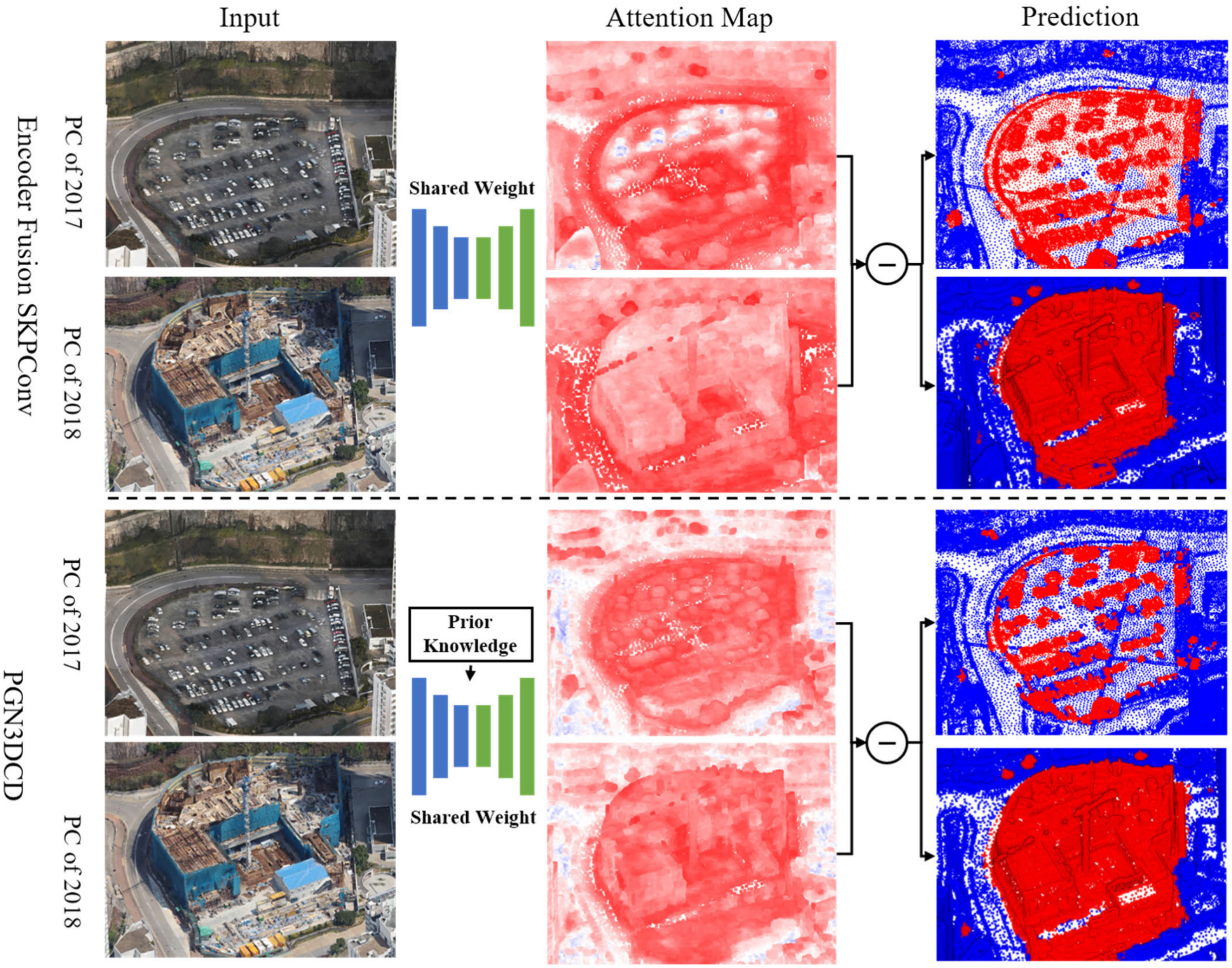


Fig. 1. Illustration of the major difference of pipeline between previous state-of-the-art work Encoder Fusion SKPConv [25] and the proposed PGN3DCD. The left column is the input realistic point clouds. The middle column is the attention of the method. A redder color means that the model pays more attention to this region, and a lighter color means that the model pays less attention, which demonstrates that PGN3DCD is more focused on learning in variations and weakening the unchanged. The right column is the prediction of methods. Red means change and blue means unchanged.

scene, whereas Change3D, ZTAKIBudapest, and BD dataset are for street scenes. Nevertheless, both the streetscape point clouds and the synthetic point clouds have certain limitations, neither of which can comprehensively and accurately describe the variations in urban, and there is currently no publicly available dataset for realistic urban 3-D change detection, which will greatly limit the implementation of urban-scale 3-D change detection.

C. Deep Learning Incorporating Prior Knowledge

Prior-knowledge-based deep learning methods can direct the model to focus on proprietary features of downstream tasks by incorporating relevant domain expertise into the model and adding a certain degree of interpretability to the model, thus improving model performance [37], [38], [39], [40]. Qiao et al. [41] establish the Kullback–Leibler divergence of crop yields through prior knowledge of agriculture. Chakraborty et al. [42] incorporate empirical knowledge of metallurgy into the neural network through the genetic algorithm. It is apparent that prior knowledge can indeed bring performance improvements, and the key issue is how to integrate the prior knowledge into deep learning. As for 3-D change detection, de Gélis et al. [25] incorporated some prior knowledge into the network to learn change-related features, which got significant performance improvements, but it only considered the geometric and was less able to deal with the changes about replacements where the number of neighborhood points remains stable. However, its success demonstrates prime importance for incorporating prior knowledge into deep learning-based 3-D change detection, which is the departure point for our method.

III. HKCD Dataset

A. Dataset Establishment

The production process of the HKCD dataset mainly consists of three steps: data collection, data preprocessing, and data annotation.

1) Data Collection: The data of HKCD are derived from publicly available 3-D data products released by Hong Kong's

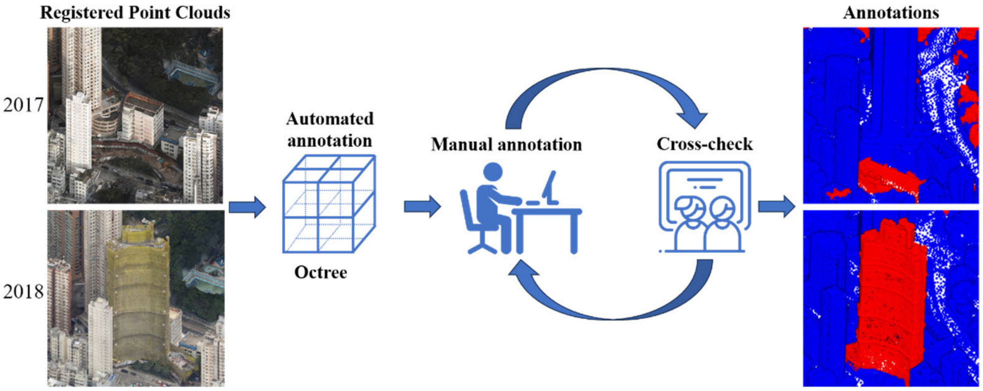


Fig. 2. Data annotation. The left column is the registered bi-temporal point clouds. The right column is the annotated labels. Red for change and blue for unchanged.

common spatial data infrastructure, 3-D Visualization Map[1] in 2017, and the 3-D Photo-realistic Model[1] in 2018. Both products contain photogrammetric 3-D models of the Kowloon area of Hong Kong with the EPSG 2326 coordinate system. These bi-temporal point clouds of the same area precisely meet the temporal and spatial requirements of 3-D change detection. Therefore, we employ them as the basis for the production of the subsequent change detection datasets.

2) Data Preprocessing: Data preprocessing mainly includes two steps: data cleaning and data registration. First, statistical filtering is used to clean the point cloud to remove the noise and outliers, reducing the influence of noisy data, and improving the quality and accuracy of the data. Then, the bi-temporal point clouds are registered through two steps: coarse registration and fine registration. Coarse registration is achieved by manually selecting the key points, and then the Iterative Closest Point algorithm [43] is used to achieve the fine registration of the bi-temporal point cloud to achieve the registration of point clouds.

3) Data Annotation: HKCD implements annotation of binary changes in buildings, terrain, street furniture, trees, and vehicles within the city, i.e., labeling whether the change has occurred or not. We mainly follow the two principles: 1) there must be explicit variations in the corresponding position of the point cloud in different periods, and the changes should be in line with the requirements of social and economic activities, such as construction management, tree monitoring, etc., and 2) the dataset focuses on three types of changes, namely additions, subtractions, and replacements. Additions are changes from nothing to something, e.g., the addition of a house; subtractions are changes from something to nothing, e.g., the dying of a tree; and replacements are changes in the category of objects in the same location, e.g., from a tree to a house. The labeling process is mainly divided into automatic coarse annotation, manual fine annotation, and data verification, the specific process is shown in Fig. 2. First, the octree [44] is used to carry out preliminary automatic screening of the changes, but the results are relatively coarse, with a lot of misclassifications. Then, the coarse classification results are refined manually in accordance with the above principles. Finally, the annotated results are cross-checked to ensure consistency and accuracy. It takes around three months and three people to label the entire dataset. Fig. 3 shows some examples of HKCD.

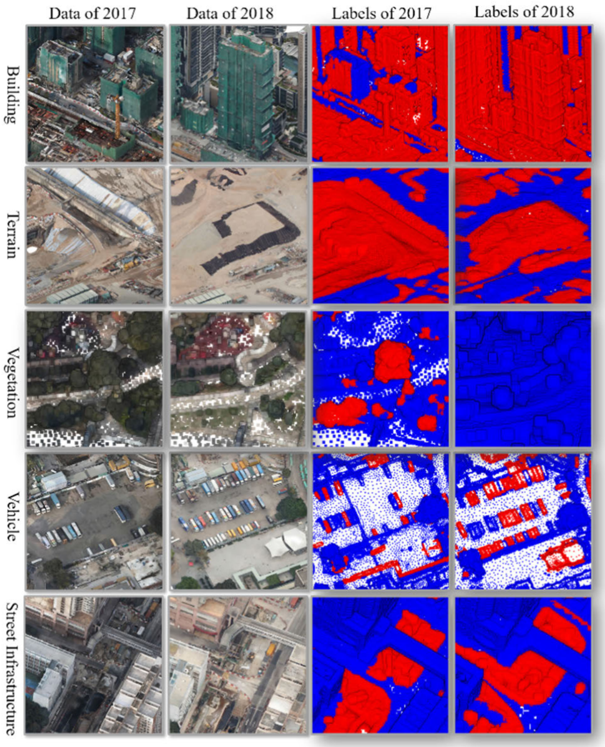


Fig. 3. Examples of HKCD. The two columns on the left are the photogrammetric original data. The two columns on the right are the labels of the point clouds. Red for change and blue for unchanged.

B. Dataset Statistics

We compare the HKCD with other mainstream 3-D change detection datasets as shown in Table I. HKCD selects a total of 18 data blocks in the Kowloon area of Hong Kong, each with a size of 750 × 600 m, covering about 8.1 km^2 and containing about 1.28×10^8 points, which is the largest urban 3-D change detection dataset. Each point contains a coordinate field, color field, and binary change field, where

[1] Hong Kong CSDI Portal.

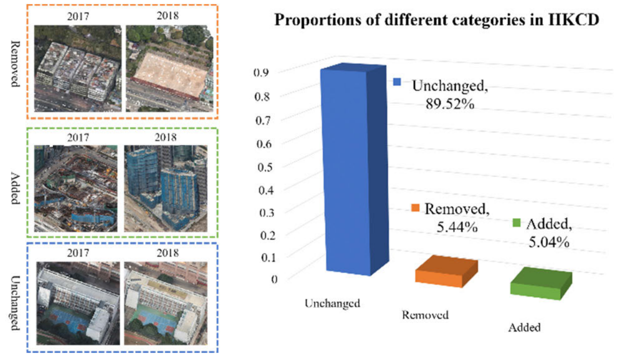


Fig. 4. Proportions of different categories in HKCD. Unchanged makes up the majority of the dataset. It is a typical long-tailed distribution.

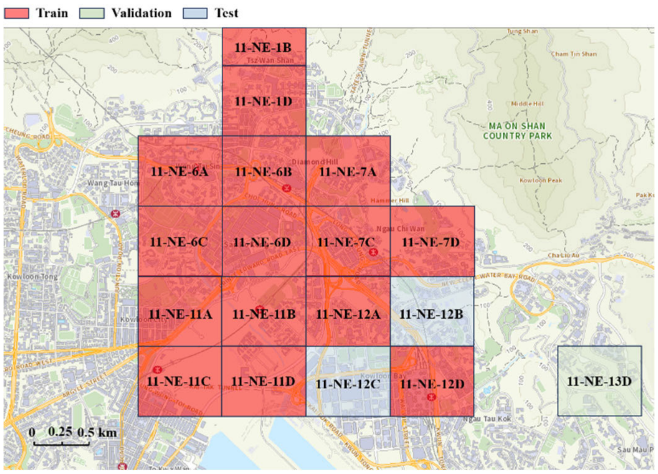


Fig. 5. Distribution of training, validation, and test sets in HKCD dataset.

the binary change field has attribute values of 0 and 1, with 0 meaning that the point is unchanged and 1 meaning that the point is changed. According to the chronological order, points that have changed in the old time-series point cloud can be considered to have undergone the “remove,” while points that have changed in the new time-series point cloud can be considered to have undergone the “add.” The proportions of different categories of variations in HKCD are shown in Fig. 4, with 5.44% of removed points, 5.04% of added points, and 89.52% of unchanged points, which forms a typical long-tail problem. In HKCD, “11-NE-13D” is selected as the validation set, “11-NE-12B” and “11- NE-12C” are selected as the test set, and the rest of the 15 blocks are the training set as shown in Fig. 5.

In summary, HKCD is a new publicly available urban 3-D change detection dataset, which annotates the binary changes of buildings, terrains, street furniture, trees, and vehicles within the city. Compared with other datasets, this dataset has more data and richer change samples, and fully describes the main changes in the urban, and can effectively support the implementation of urban 3-D change detection.

IV. Method

A. Overview

The long-tailed distribution of 3-D change detection leads to existing methods overly focusing on the background unchanged and weakens the learning of foreground variation, which inevitably affects the performance of these methods. Therefore, the objective of this method is: *Can we guide the model to focus on the learning of variation and weaken the influence of unchanged objects?*

Motivated by this, we propose an efficient, end-to-end prior-knowledge-guided network for 3-D point cloud change detection, named PGN3DCD. As illustrated in Fig. 6, PGN3DCD consists of two backbones with shared weights to form the Siamese structure, including feature encoders, feature decoders, 3-D change prior knowledge, and change-aware difference attention modules. Three-dimensional change prior knowledge exploits the coordinates, and colors of bi-temporal point clouds to calculate the change prior knowledge, including prior change and change mask, which serves as the prior bi-temporal variation perceptron. Different from the method [25], which implements the feature encoder to obtain the implicit prior knowledge, the proposed method uses an algebraic approach to calculate the explicit prior knowledge, which enables unequivocal guidance in the early stages of model training to improve performance. The differences between bi-temporal point clouds are captured by the change-aware difference attention module, in which the prior knowledge guides the module to focus on the variations and weaken the influence of the unchanged. At last, the refined difference features are dealt with by the feature decoders to get the variations. In summary, PGN3DCD takes full advantage of the speculative and predictability of 3-D variation to construct a prior knowledge of 3-D change detection, which adequately expresses the variations from geometry and texture perspectives and effectively guides the neural network to learn the features of the variations.

B. Prior Knowledge of 3-D Change Detection

Prior knowledge can incorporate domain knowledge into deep learning methods, thus effectively guiding the methods to purposefully learn task-specific features and improve the performance of the approaches. In this article, we introduce prior knowledge for 3-D change detection and integrate it with neural networks to improve the performance of 3-D change detection.

3-D change detection differs from change detection in 2-D images, which have more pronounced differences in geometry and texture, and such differences are predictable and speculative. We define the prior knowledge of 3-D change detection as *there are significant differences in the geometry or texture in areas where there are variations.* Based on this definition, the variations present in multitemporal point clouds can be extracted beforehand in terms of geometry and texture, which is implemented as shown below.

1) Confirmation of the Corresponding Position: In order to compare the multitemporal point clouds, it is necessary to confirm the correspondence between the different temporal points. There is spatial proximity of the corresponding positions, so this article adopts the shortest Euclidean distance as

TABLE I

ACQUISITION CONFIGURATIONS FOR 3-D CHANGE DETECTION DATASET

	#Type	# Interval	#Size	#Classes	#Points	#RGB	#Sensors
SLPCCD [17]	Bi-change	4 Years	\	2	85.2M	Yes	MLS
URB3DCD [23]	Multi-change	\	7.6km^2	7	5.7M	No	Synthetic
HKCD	Bi-change	1 Year	8.1km^2	2	128.03M	Yes	UAV Photogrammetry

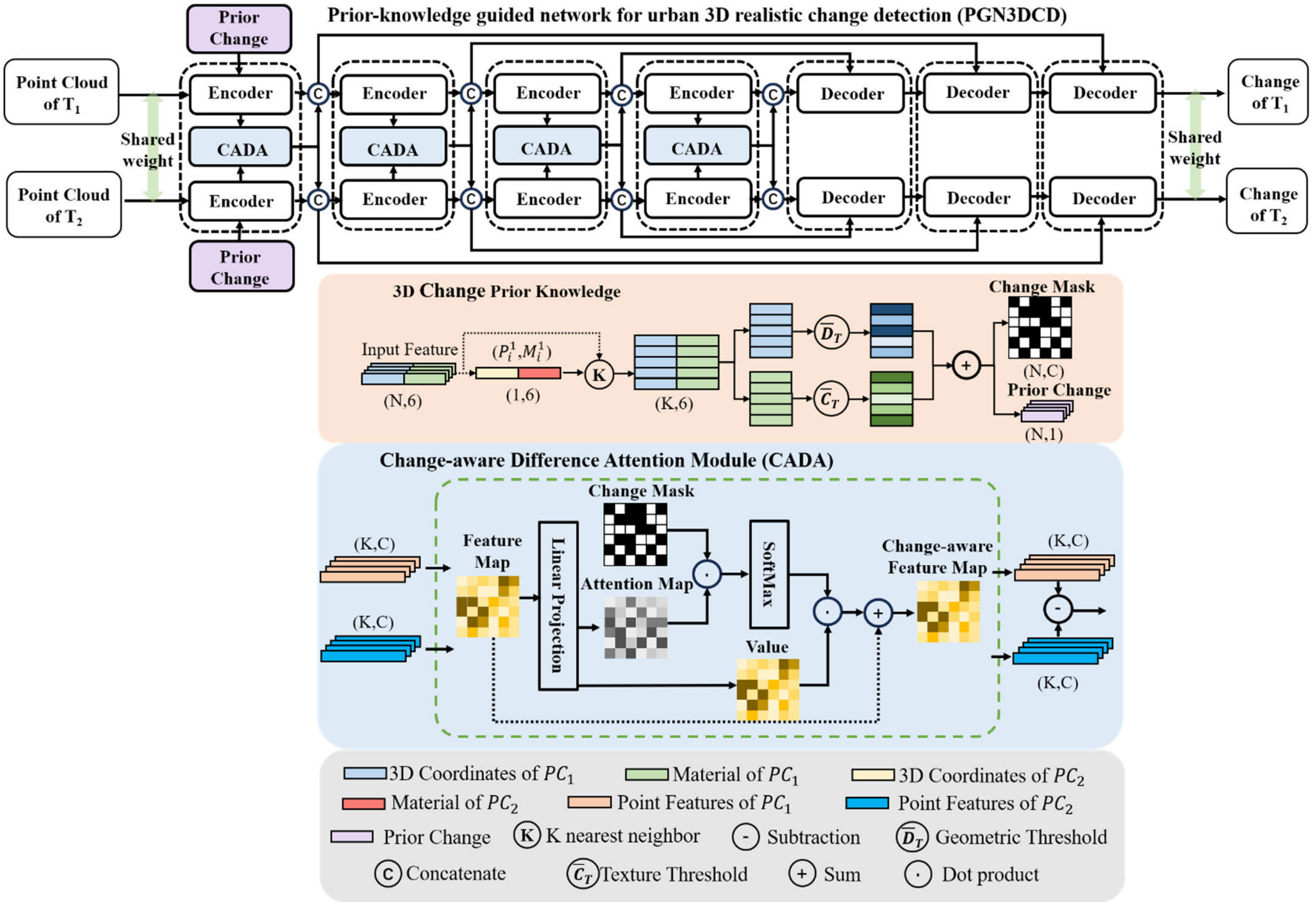


Fig. 6. Overview of the proposed PGN3DCD includes two novel components: 3-D change prior knowledge and change-aware difference attention module. Three-dimensional change prior knowledge converts the coordinates and colors to change mask and prior change. The change mask is then input to the change-aware difference attention module to guide PGN3DCD focusing on variations. Prior change is concatenated with the coordinates to be the input of the PGN3DCD. Change-aware difference attention module is used to get the difference between bi-temporal point clouds.

the judgment criterion of the corresponding positions. Given sampled points $s_1 \subset \text{PC}_1$ in point clouds PC_1 and sampled points $s_2 \subset \text{PC}_2$ in point clouds PC_2, K-nearest neighbor (KNN) is employed to get the KNNs $K_{12} \in \mathbb{R}^{N_1 \times k}(k_{12} \subset \text{PC}_2)$ of s_1 in PC_2 and $K_{21} \in \mathbb{R}^{N_2 \times k}(k_{21} \subset \text{PC}_1)$ of s_2 in PC_1. These nearest-neighbor points are the counterparts of each sampled point in the point cloud of another period.

2) Prior Computation of Change Information: After confirming the neighbors of input point clouds, PC_1 and PC_2, in another temporal point cloud, PC_{12} and PC_{21} (PC_{12} is the neighboring points of the point cloud of period 1 in the point cloud of period 2), by the above method, the potential changes are calculated in advance from the geometric space and texture space. Specifically, the variations between different time-series point clouds are reflected by the differences between them, while in this work, geometric and texture differences are used to synthesize the differences between different time-series point clouds. The geometric difference is reflected by the Euclidean distance between the point cloud and the neighboring points in another time-series point cloud, while the texture difference is reflected by the two-Norm between the point cloud and the neighboring points in another time-series point cloud in terms of color. First, the geometry difference, $\overline{D}_1$ and $\overline{D}_2$, and texture difference, $\overline{C}_1$ and $\overline{C}_2$, of each point in PC_{12} and PC_1, PC_{21} and PC_2 are calculated, respectively. Subsequently, according to the density of the input point clouds, PC_1 and PC_2, the spatial distance thresholds, $\overline{D}_T^1$ and $\overline{D}_T^2$, and texture thresholds, $\overline{C}_T^1$ and $\overline{C}_T^2$, are set. If the geometry difference or texture difference exceeds the threshold, it is assigned a value of 1. If it is less than the threshold, it is divided by the corresponding threshold value to normalize the spatial distance and texture. Finally, the geometry difference and texture difference are weighted and summed, and the results are input into the model as the prior judgments $\wp$. The closer the value is to 1, the more likely the point is a change point; the closer the value is to 0, the more likely the point is an unchanged point. This is expressed by the following formula:

$$\begin{cases} f(x) = \sqrt{\sum (\mathrm{KNN}(x) - x)^2} \\ \overline{D} = \begin{cases} \dfrac{f(P)}{\overline{D}_T}, & f(P) < \overline{D}_T \\ 1.0, & f(P) > \overline{D}_T \end{cases} \\ \overline{C} = \begin{cases} \dfrac{f(M)}{\overline{C}_T}, & f(M) < \overline{C}_T \\ 1.0, & f(M) > \overline{C}_T \end{cases} \\ \wp = \lambda_1 * \overline{D} + \lambda_2 * \overline{C} \end{cases} \tag{1}$$

where P is the spatial coordinate of the point cloud; M is the texture; $\overline{D}_T$ is the spatial distance threshold; $\overline{C}_T$ is the texture threshold; $\overline{D}$ is the geometry difference; $\overline{C}$ is the texture difference; λ_1 and λ_2 are the weighted values, which are set as 0.5; KNN is the K-nearest neighbor algorithm; $\wp$ is the prior change.

It is worth noting that due to the differences in internal and external factors of point cloud acquisition in different time series (acquisition equipment, lighting conditions, etc.), the density and texture of the acquired point clouds are different, which leads to the great difficulty in determining the geometric threshold and material threshold. Therefore, we provide a simple method for automatically determining geometric thresholds and texture thresholds, which is preprocessing without impacting the efficiency of the model. There is a variation when there is a significant difference in the geometric or texture of the point cloud at different time series. Two-Norm is used to represent the difference between geometric and texture. The distance threshold and texture thresholds are set to twice the mean difference of the geometric and texture of the nearest point in one time-series point cloud with respect to another time-series point cloud, respectively. The specific formula is as follows:

$$\begin{cases} D(x) = \dfrac{\sum_n f(x)}{n} \\ \overline{D}_T = 2 * D(P) \\ \overline{C}_T = \begin{cases} 2 * D(M), & D(M) < \dfrac{\sqrt{3}}{2} \\ D(M) + \dfrac{\sqrt{3} - D(M)}{2}, & D(M) \geq \dfrac{\sqrt{3}}{2} \end{cases} \end{cases} \tag{2}$$

where, $D(x)$ is the function to calculate the average difference. When the texture is normalized RGB, the maximum of average difference of texture is $\sqrt{3}$.

C. Change-Aware Difference Attention

In the feature encoding stage, we want the feature extraction modules to focus as much as possible on the regions where there is variation and weaken the influence of unchanged regions. Therefore, point cloud features need to be optimized before feature differences are calculated by subtraction. To this end, we design a change-aware self-attention algorithm to enhance the learning of change features by the feature encoder based on the above 3-D change detection prior knowledge, the specific structure of which is shown in Fig. 7.

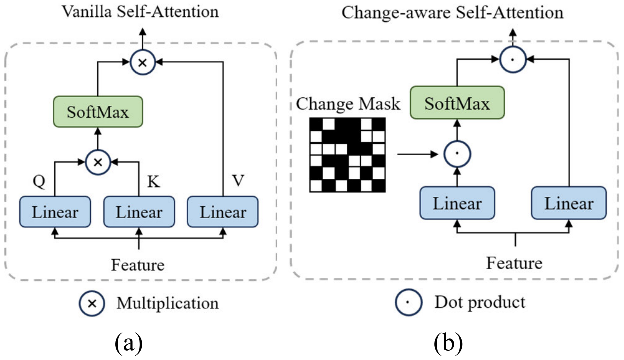


Fig. 7. Comparison of self-attention structure. (a) Vanilla self-attention. (b) Change-aware self-attention.

1) Change-Aware Self-Attention: Vanilla point cloud self-attention methods calculate the attention matrix by multiplying the matrix query $\in \mathbb{R}^{N \times C}$ with the matrix key $\in \mathbb{R}^{N \times C}$. This method has a high computational complexity of $\mathcal{O}(N^2 \times C)$. When the number of input points N is large, vanilla self-attention will cause a great memory overhead. In order to reduce the computational complexity of attention and cause the smallest possible overhead, we follow the idea of the paper [45] to use the linear function to directly transform the input features into attention weights with a computational complexity of $\mathcal{O}(N \times C)$. In the shallow feature encoding stage, the attention weights are discrete and their attention is consistent throughout the point cloud scene and will not focus on learning in the changing regions, which will not be conducive to mining features in the changing regions. In order to direct the attention to pay more attention to the variation, we design a change mask based on the 3-D change detection prior knowledge, which can increase the weight of the variations and decrease the weight of the unchanged, and after the dot product of the attention weight, the attention weight can be refined to have a clear directivity and purposefulness for 3-D change detection. The refined attention is subjected to SoftMax and the dot product of value $\in \mathbb{R}^{N \times C}$ to get the fine-grained change features. The entire change-aware self-attention can be expressed by the following equation:

$$\mathrm{CSA}(f) = \mathrm{SoftMax}(L(f) \cdot \mathrm{Mask}) \cdot L(f) \tag{3}$$

where, $\mathrm{CSA}(\cdot)$ is the change-aware self-attention; $L(\cdot)$ is the linear function; f is the point cloud features; and Mask is the change mask.

2) Change Mask: Mask in attention mechanism can guide the neural network to focus on the foreground target, suppressing the background, and strengthening the learning of the task target [46]. Inspired by this, we introduce a change mask based on the above prior knowledge of 3-D change detection to strengthen the learning of the features of the changed region and suppress the influence of the unchanged region.

Given point clouds, $\mathrm{PC}_1 \in \mathbb{R}^{N_1 \times C_1}$ and $\mathrm{PC}_2 \in \mathbb{R}^{N_2 \times C_2}$, the differences of multitemporal point clouds are calculated by the prior knowledge of 3-D change detection. The prior change $\wp$ is also used as the change mask. The closer the $\wp$ is to 1, the more likely the point is to be a potential change point, and the closer the $\wp$ is to 0, the more likely the point is to be an unchanged point. By giving greater weight to the prior changed points and smaller weight to the prior unchanged points, the learning of the changed points is strengthened and the learning of the unchanged points is inhibited, thus effectively guiding the shallow neural network to mine the features of the changed region and improve the efficiency of change detection. Other computational methods for change masks are discussed in detail in this article, see experiments for details.

V. Experiments

A. Datasets

In addition to the HKCD dataset introduced in this article, the SLPCCD dataset and the URB3DCD-V2 dataset are also used in the experiments in this article. SLPCCD is a dataset designed for street furniture in complex streetscapes, a change classification task for POIs, which was acquired in 2016 and 2020 by MLS in Schiedam, The Netherlands [17]. This bi-temporal dataset contains a total of 78 pairs of street point clouds, each pair is labeled with the coordinate and the specific type of change, namely "Unchanged" and "Change," aiming to realize the dense segmentation of changes. URB3DCD-V2 is the ALS point cloud change detection dataset synthesized by the simulator based on LoD2 models of the first and second districts of Lyon, France [18], which contains seven categories of changes, namely "Unchanged," "New Building," "Demolition," "New Vegetation," "Vegetation Growth," "Vegetation Loss," and "Mobile Objects," aiming to achieve multiclass change detection.

B. Evaluation Metrics

The essence of 3-D realistic change detection implemented in this article is the dense segmentation of 3-D point clouds. For evaluation, we select five metrics commonly used in dense segmentation tasks, namely overall accuracy (OA), mean accuracy (mAcc), recall, intersection over union (IoU), and mean IoU (mIoU).

C. Implementation Details

The method proposed in this article is a plug-and-play change detection module that can be combined with a feature extraction backbone to form a 3-D change detection model. Therefore, when dealing with different datasets, this experiment keeps the backbone of the state-of-the-art model unchanged and combines it with the proposed module to form a new change detection method. The current state-of-the-art model of the SLPCCD dataset is 3DCDNet [17], and that of the URB3DCD-V2 dataset is Encoder Fusion SKPConv [25]. We keep the backbone of the state-of-the-art models of the two datasets unchanged, 3DCDNet as well as Siamese KPConv, respectively, and add the proposed PGN3DCD to them. In the HKCD dataset, Siamese KPConv is also selected as the backbone, and the PGN3DCD is incorporated. The average distance of SLPCCD, HKCD, and URB3DCD-V2 is 0.15, 1.5, and 1.8, so the distance thresholds D_T are set to 0.3, 3.0, and 3.6. The average texture difference between SLPCCD and HKCD is 0.3 and 0.3, so the texture thresholds C_T are set to 0.6. As for URB3DCD-V2, because there are no colors in the datasets, we set the λ_2 as 0 and the λ_1 as 1 from (1), eliminating the effect of texture on threshold calculations. The training parameters of all backbones are kept consistent with the original article.

Negative log likelihood (NLL) loss is used as the loss function and is minimized by stochastic gradient descent (SGD) with a momentum of 0.98. As for HKCD, the initial learning rate is set to 10^{-2} and scheduled to decrease step by step, and the batch size is used as 10. It is worth noting that the loss of each temporal point cloud needs to be calculated separately and two losses are summed up to get the overall loss, which is calculated as follows:

$$\begin{cases} \mathrm{NLL}\left(y_t, y_p\right) = -\left(y_t \log\left(y_p\right)\right) + (1 - y_t) \log(1 - y_p) \\ \mathrm{loss} = \mathrm{NLL}\left(y_t^1, y_p^1\right) + \mathrm{NLL}\left(y_t^2, y_p^2\right) \end{cases} \tag{4}$$

where, y_t is the truth; y_p is the prediction; y_t^1 is the truth of PC_1; y_p^1 is the prediction of PC_1; y_t^2 is the truth of PC_2; y_p^2 is the prediction of PC_2.

D. Change Detection on Benchmarks

In this section, we compare the proposed PGN3DCD with multiple state-of-the-art models on benchmarks. The results of the comparison models are all taken from the original articles. The results on HKCD are experimented with by ourselves based on original papers. It is worth noting that almost all metrics of baseline in the URB3DCD-V2 and SLPCCD datasets are from the original paper [17], [25], except Recall. In this article, recall is calculated, to ensure objectivity and fairness, which are only calculated for models for which the source code and parameters are publicly available.

1) Evaluation on HKCD: We conduct comparative experiments on the test set of the HKCD dataset, and the quantitative and qualitative results are shown in Table II and Fig. 8, respectively. The PGN3DCD is compared with DGCNN [36], 3DCDNet [17], Siamese KPConv [18], and Encoder Fusion SKPConv [25]. From Table II, the performance of PGN3DCD is significantly improved, and all metrics reach the optimal results, with OA reaching 96.24 (+2.09%) and mIoU reaching 77.14% (+4.89%). From Fig. 8, there are a large number of False Positive samples in the results of Encoder Fusion SKPConv, which greatly confuses the detection of True Positive samples, while PGN3DCD is able to more comprehensively and accurately detect the real existing variations. The experimental results demonstrate the effectiveness of PGN3DCD on urban 3-D change detection.

TABLE II

QUANTITATIVE RESULTS OF DIFFERENT APPROACHES ON HKCD. THE BOLDED RESULTS ARE OPTIMAL AND THE UNDERLINED RESULTS ARE SUBOPTIMAL

Method	OA (%)	Recall (%)	mIoU (%)	IoU (%)		
				Unchanged	Removed	Added
DGCNN-based [17]	90.32	92.77	39.92	90.13	6.30	23.35
3DCDNet [17]	84.40	76.95	45.60	83.47	27.77	25.54
Siamese KPConv [18]	91.78	85.64	63.72	91.04	45.97	54.15
Encoder Fusion SKPConv [25]	94.15	89.91	72.25	93.55	55.61	67.59
PGN3DCD	**96.24**(↑2.09)	**93.93**(↑1.16)	**77.14**(↑4.89)	**95.91**(↑2.36)	**63.72**(↑8.11)	**71.79**(↑4.20)

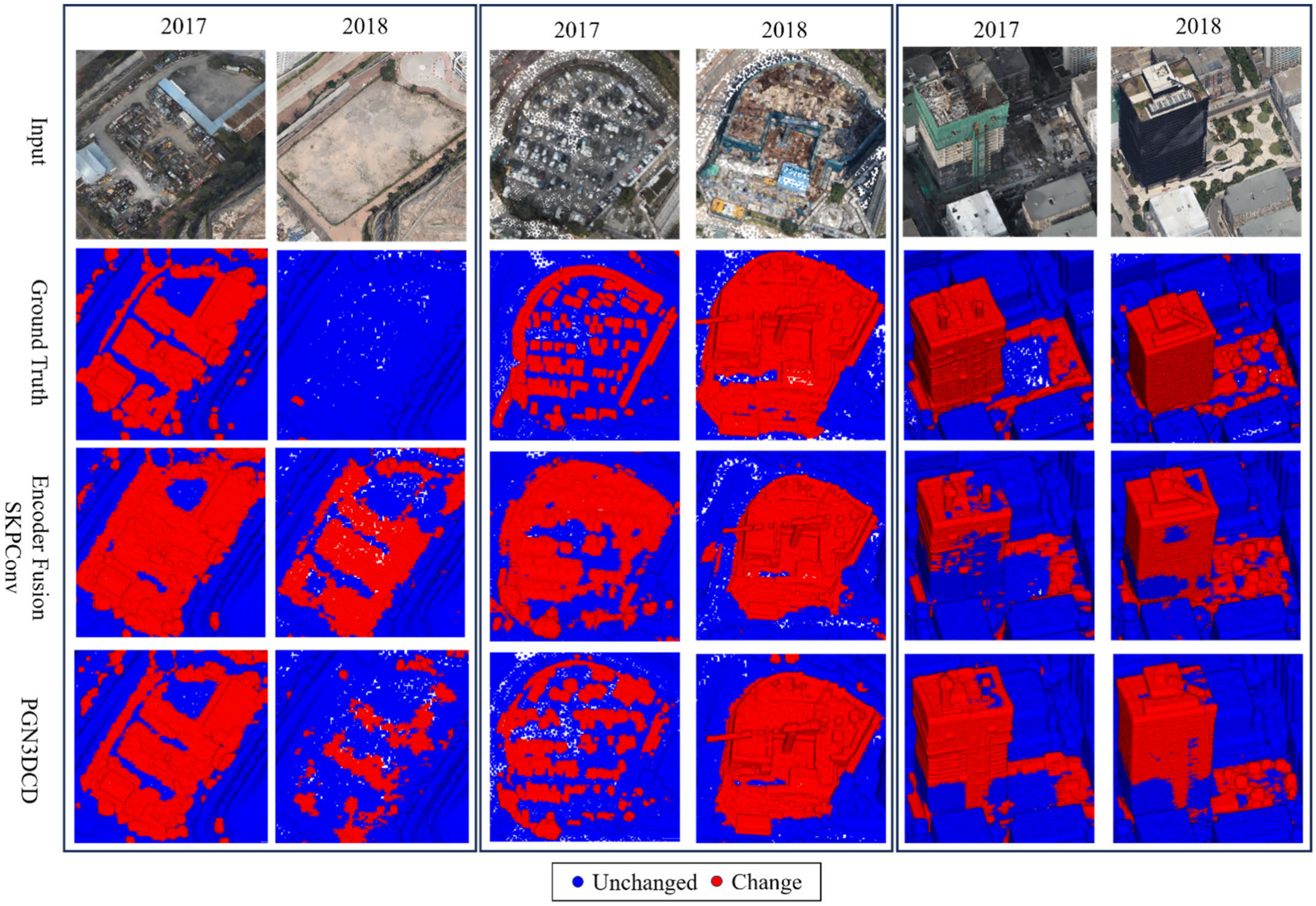


Fig. 8. Qualitative comparison on HKCD. All of the changed points in the 2017 columns represent point clouds that "removed" in 2017. All of the changed points in the 2018 columns represent point clouds that "added" in 2018. The third row is the results of Encoder Fusion SKPConv [25] on HKCD.

2) Evaluation on URB3DCD-V2: To verify the generalization, we conduct another experiment on the URB3DCD-V2 dataset. The quantitative and qualitative results on the test set of this dataset are shown in Table III with Fig. 9, respectively. Table III shows that PGN3DCD achieves 88.12% of the best performance in mIoU, which is 2.93% higher than the previous state-of-the-art method, and also achieves excellent performance in six classifications, e.g., veg. growth (+5.93%). Fig. 9 shows that the improvement of PGN3DCD over the previous model is that it reduces misclassification and improves the efficiency of detecting changes in dense objects, such as plants. This significant improvement integrates the effectiveness of the proposed method on multiclass change segmentation tasks. Most of the errors occur in the part of the object that is in contact with the ground, which is hardly recognized by the model.

3) Evaluation on SLPCCD: The above two experiments explore the change detection performance of PGN3DCD in large scenes. In this section, we further test the change detection performance of PGN3DCD against single instance objects in the city. The results of quantitative and qualitative comparisons are shown in Table IV and Fig. 10. Table IV demonstrates that the PGN3DCD performs better than the previous state-of-the-art approach in OA and mIoU metrics, with 1.12% and 7.72% improvement, respectively, and the IoU metrics are improved in all three classifications, with 1.15%

TABLE III

QUANTITATIVE RESULTS OF DIFFERENT APPROACHES ON URB3DCD-V2. THE BOLDED RESULTS ARE OPTIMAL AND THE UNDERLINED RESULTS ARE SUBOPTIMAL. "\" MEANS THAT THE RELEVANT CODES AND PARAMETERS WERE NOT PUBLIC IN THE ORIGINAL ARTICLE

				IoU (%)						
Method	**mAcc (%)**	**Recall (%)**	**mIoU(%)**	**Unchanged**	**New building**	**Demolition**	**New Veg.**	**Veg. growth**	**Missing veg.**	**Mobile Object**
Random Forest [31]	65.82	\	52.37	92.72	73.16	64.60	75.17	19.78	7.78	73.71
DSM-Siamese [47]	80.91	\	57.41	93.21	86.14	69.85	70.69	8.92	60.71	8.14
DSM-FC-EF [47]	81.47	\	56.98	94.39	91.23	71.15	68.56	1.89	62.34	46.70
Siamese KPConv [18]	91.21	85.31	80.12	95.82	86.67	78.66	93.16	65.18	65.46	91.55
OneConvFusion [25]	92.62	86.42	81.74	96.95	96.06	79.63	95.53	61.12	65.79	92.89
Triplet KPConv [25]	92.94	86.41	84.08	97.41	95.73	81.71	96.24	64.85	73.02	92.90
Encoder Fusion SKPconv [25]	<u>94.23</u>	<u>90.01</u>	<u>85.19</u>	<u>97.47</u>	<u>96.68</u>	<u>82.29</u>	<u>96.52</u>	<u>67.76</u>	<u>73.50</u>	**94.37**
PGN3DCD	**94.27** (↑0.04)	**90.53** (↑0.42)	**88.12** (↑2.93)	**97.77** (↑0.3)	**96.93** (↑0.25)	**83.08** (↑0.79)	**96.82** (↑0.3)	**73.70** (↑5.94)	**76.01** (↑2.51)	<u>92.56</u> (↓1.81)

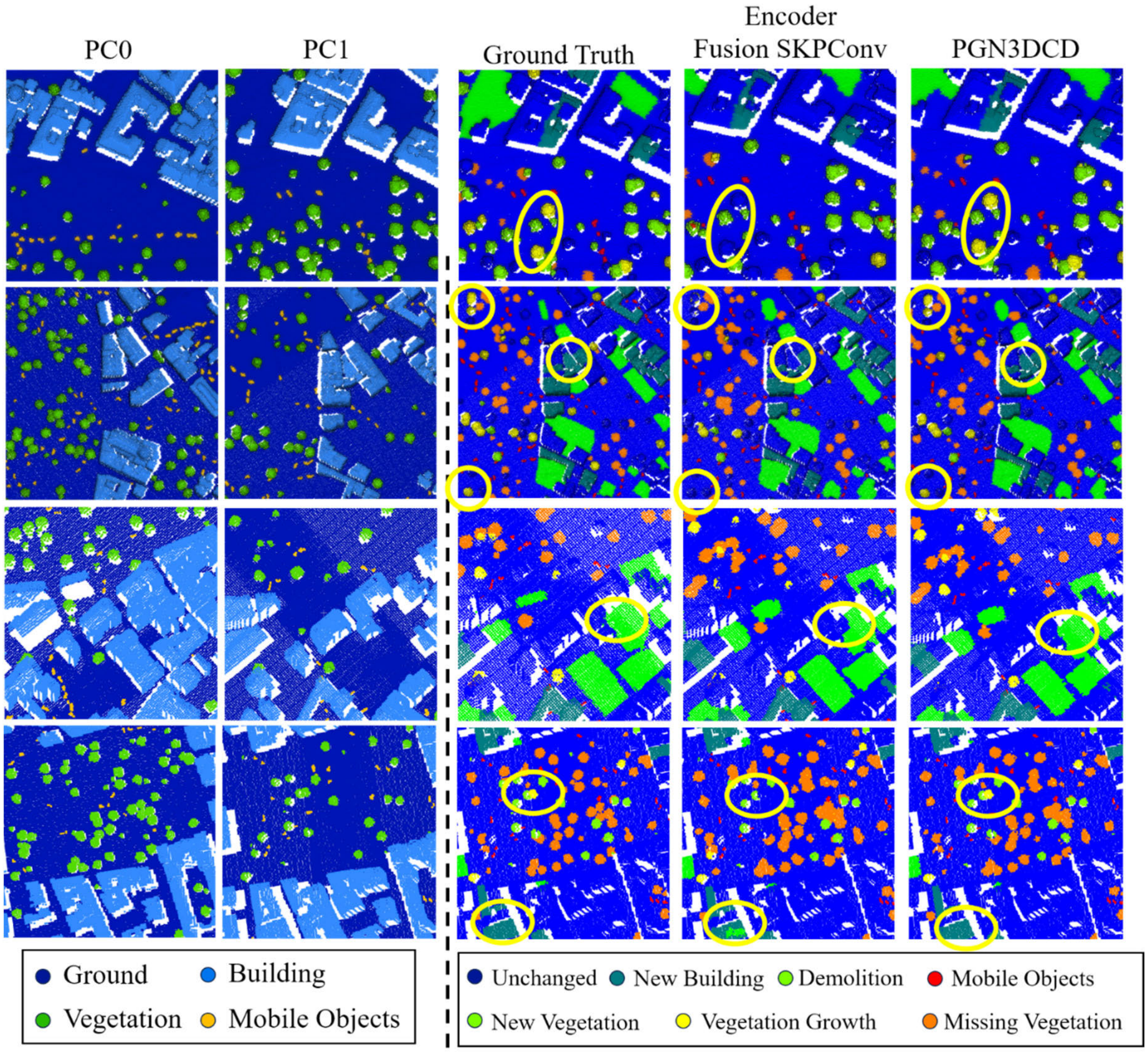


Fig. 9. Qualitative comparison of URB3DCD-V2 [25]. Some enhancements are highlighted with yellow ellipses. The fourth column is the results of Encoder Fusion SKPConv [25] on URB3DCD-V2.

improvement in "Background," 11.12% in "Removed," and 10.88% in "Added." From Fig. 10, although the 3DCDNet is able to localize variations, the variations are incomplete. After incorporating PGN3DCD, the variations are more complete. This significant enhancement proves the effectiveness of the proposed method in detecting changes in instance objects.

TABLE IV

QUANTITATIVE RESULTS OF DIFFERENT APPROACHES ON SLPCCD. THE BOLDED RESULTS ARE OPTIMAL AND THE UNDERLINED RESULTS ARE SUBOPTIMAL. "\" MEANS THAT THE RELEVANT CODES AND PARAMETERS WERE NOT PUBLIC IN THE ORIGINAL ARTICLE

Method	OA (%)	Recall (%)	mIoU (%)	IoU (%)		
				Background	Removed	Added
PointNet-based [17]	89.67	\	48.01	89.26	25.13	29.64
PointNet++-based [17]	95.09	\	67.55	94.80	47.86	60.00
PointMLP-based [17]	92.32	\	59.41	91.85	46.74	39.64
PointConv-based [17]	91.01	\	42.16	90.83	16.13	19.51
DGCNN-based [17]	94.07	\	62.30	93.76	43.73	49.40
3DCDNet [17]	95.85	71.85	74.45	95.52	63.46	64.37
PGN3DCD	**96.97**(↑1.12)	**88.63**(↑16.78)	**82.17**(↑7.72)	**96.67**(↑1.15)	**74.58**(↑11.12)	**75.25**(↑10.88)

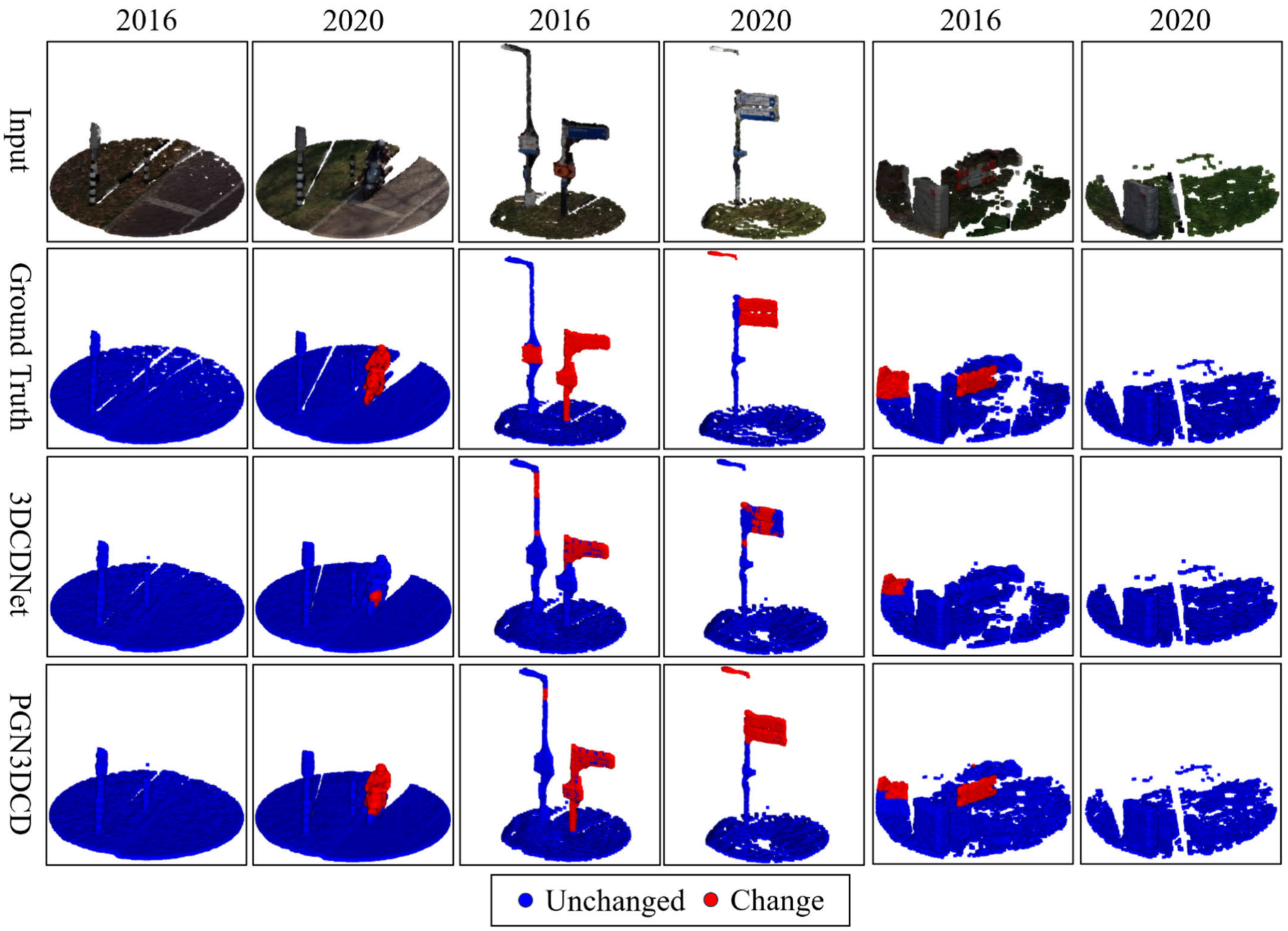


Fig. 10. Qualitative comparison of SLPCCD [17]. All of the changed points in the 2016 columns represent point clouds that were "removed" in 2016. All of the changed points in the 2020 columns represent point clouds that were "added" in 2020. The third row is the results of 3DCDNet [17] on SLPCCD.

E. Ablation and Analysis

In this section, we perform ablation experiments on the proposed method and further analyze the relevant parameters, efficiency, and interpretability.

1) Ablation of the Network: In order to validate the effectiveness of the proposed modules, the input prior change, and change-aware difference self-attention, we conduct four sets of ablation experiments on three datasets following the single variable principle. Table V exhibits the performance of three different optimization strategies on three different datasets. A denotes the input prior change, which brings improvement on all three datasets compared to the baseline, with mIoU improvements of 5.83%, 7.42%, and 4.19% on the HKCD, URB3DCD-V2, and SLPCCD datasets, respectively. B denotes the change-aware difference attention module. In all three scenarios, it brings performance improvement compared to the baseline, with mIoU improvements of 6.71%, 5.59%, and 4.84% on HKCD, URB3DCD-V2, and SLPCCD datasets, respectively. The results demonstrate that both modules realize

TABLE V

Ablation Experiments on the Optimization Strategies. A Denotes the Input Prior Change. B Denotes the Change-Aware Difference Attention Module

Model name	HKCD		URB3DCD-V2		SLPCCD	
	OA(%)	mIoU(%)	mAcc(%)	mIoU(%)	OA(%)	mIoU(%)
Opt-A: Backbone	91.78	63.72	91.21	80.12	95.85	74.45
Opt-B: Backbone + A	94.67(↑2.89)	69.55(↑5.83)	91.69(↑0.48)	87.54(↑7.42)	96.40(↑0.55)	78.64(↑4.19)
Opt-C: Backbone + B	93.89(↑2.11)	70.43(↑6.71)	94.21(↑3.00)	85.71(↑5.59)	96.67(↑0.82)	79.29(↑4.84)
Opt-D: Backbone + A + B	**96.24**(↑4.46)	**77.14**(↑13.42)	**94.27**(↑3.06)	**88.12**(↑8.00)	**96.97**(↑1.12)	**82.17**(↑7.72)

the improvement of the baseline, which proves that the proposed method can express the change characteristics more comprehensively, enabling the method to perform change detection more accurately.

2) Ablation of the Modules: The different hyperparameters have an impact on the performance of the model, and we explore the impact of the hyperparameters of the proposed model on the HKCD dataset as follows:

1) As for the change-aware attention module, different mask values have an impact on the performance, and we expect that the model enhances focus on the variations and weakens focus on the unchanged. Two patterns are designed to accomplish this. First, zero for the unchanged and one for the variations, which reduces the weight of the unchanged and retains the weight of the variations. Second, one for the unchanged and two for the variations, which retains the unchanged and expands the variations. Moreover, the discrete assignment of 0, 1, and 2 is a hard operation that may affect the performance if there is an error in the prior variation. Therefore, we design a soft operation to represent the prior state of points in the form of probability by assigning values to the mask through continuous intervals. The results are shown in Table VI. Overall, continuous mask outperforms discrete mask. Adopting the continuous mask of 0–1 outperforms the other setting, while the discrete mask of 0 and 1 is the worst outcome. The performance gap between the discrete mask of 0 and 1 and the other three is large. The reason for this is that misclassification is inevitable in the prior changes, and if changed points are misclassified as unchanged, assigning zero will completely obscure the features of these points, which is not conducive to learning on these points.
2) As for prior change input, it is an empirical prejudgment of potential variations in the multitemporal point clouds. It is necessary to compare the prior change input with the changes in the final output of the model to confirm how much the model improves on the prior change input. The mIoU is selected as the measure metric. From Table VII, it is explicit that there are improvements between prior change and predicted output on all three datasets, which effectively demonstrates the validity of the proposed PGN3DCD.

TABLE VI

Effectiveness of Different Mask of Proposed Module on HKCD

Model	Change-aware Attention				mIoU (%)
	[0,1]	[1,2]	(0,1)	(1,2)	
A	√				75.01(↓2.13)
B		√			76.94(↓0.20)
C			√		77.14
D				√	76.91(↓0.23)

TABLE VII

Evaluation of Prior Change and Output on Different Datasets

	HKCD	**URB3DCD-V2**[25]	**SLPCCD**[17]
Input	48.35%	72.83%	71.12%
Output	77.14%	88.12%	82.17%

TABLE VIII

Efficiency Analysis of Different Approaches on HKCD

Model	Time (second per 10^4 points)	Parameters (millions)
Siamese KPConv [25]	0.74	39.7
Encoder Fusion SKPConv [25]	0.83(↑0.09)	48.1(↑8.40)
PGN3DCD	0.84(↑0.10)	50.9(↑11.20)

3) Efficiency of PGN3DCD: The addition of modules inevitably affects the space and time complexity. We validate the computational efficiency of PGN3DCD by inference time and the size of the parameters. The settings of PGN3DCD are kept the same as above. Table VIII shows that the total number of parameters in the model increases by 11 million, and the computational efficiency of 10^4 points per inference is reduced by 0.1 s with PGN3DCD. The addition of the modules improves the accuracy of the model change detection, but it increases the size and reduces the computational efficiency.

4) Interpretability Verification: The change-aware self-attention module is designed to guide the neural network to pay full attention to the variations and suppress learning in unchanged regions. This section verifies that the module is functioning as designed by modeling the weights and effective field-of-view, as shown in the heat map Fig. 11. A redder color means that the model pays more attention and a lighter

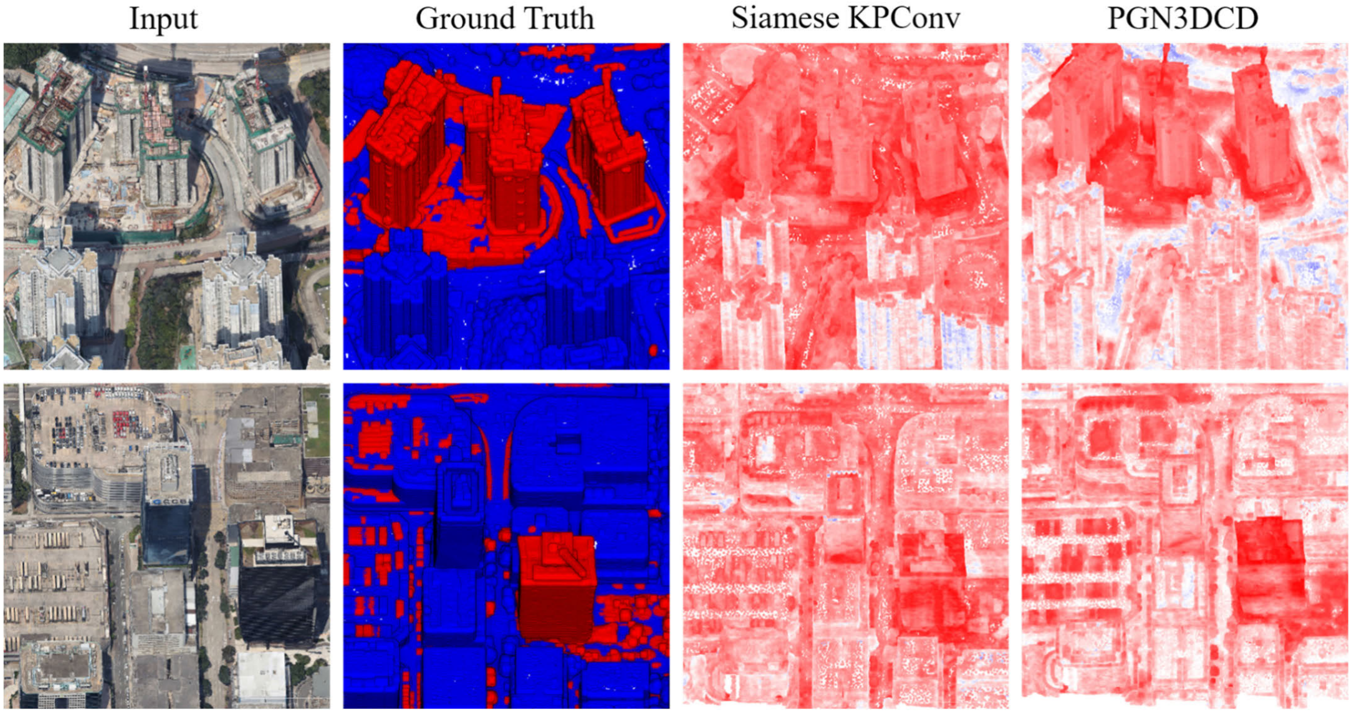


Fig. 11. Qualitative comparison of methods' attention to variations. In the "ground truth" column, red for change and blue for unchanged. The two columns on the right show the attention of the model. Redder colors mean the model is more concerned and lighter colors mean the model is more neglected. The third column is the attention of Siamese KPConv [18].

color means that the model pays less attention. From the figure, we can see that the baseline, Siamese KPConv, treats each point in the point cloud as relatively equal, which is not conducive to the learning of variations, whereas the PGN3DCD can give more weight to the regions where there is a change to strengthen the learning of variations and at the same time weaken the learning of the nonchanging regions, which proves the interpretability of the proposed method and its ability to efficiently guide the model to the learning of variations.

F. Discussion

1) Impact of Prior Change Accuracy: The above experiments effectively demonstrate that prior knowledge can bring effective performance improvement for change detection. However, if the prior change information is not always accurate, will the PGN3DCD still be able to correctly recognize the variations. Therefore, in this section, the impact of prior change accuracy on the accuracy of the model on change detection is explored on SLPCCD. We experiment with the change detection performance of the model with an accuracy of prior change at 10%, 30%, 50%, 70%, and 90%, respectively, and the results are shown in Table IX. It is apparent that when the accuracy of the prior change is higher than 50%, the performance of the model has a significant improvement, on the contrary, when the accuracy of the prior change is lower than 50%, the prior change not only cannot bring performance optimization for the model but also may even destroy the performance of the model. Therefore, ensuring the accuracy of the prior change is of great importance for the final performance of the model.

TABLE IX
IMPACT OF DIFFERENT ACCURACY OF PRIOR CHANGE ON PERFORMANCE

Model	Prior Change Accuracy	mIoU
PGN3DCD (w prior change)	10%	72.39%
	30%	73.44%
	50%	80.21%
	70%	83.62%
	90%	86.75%
PGN3DCD (w/o prior change)	\	74.45%

2) Stability of PGN3DCD: The above experiments show the best results of PGN3DCD on three datasets. However, it is not credibility and logical rigor that show the results of the experiments only once. To show the stability of the proposed method, we conduct five other experiments under equivalent conditions and show the mean and standard deviation of the five results in Table X. The means of the methods on all three datasets are very close to the best performance (HKCD-0.44, URB3DCD-V2-0.3, and SLPCCD-0.3) and all have small standard deviations, which can effectively prove the stability of the proposed method.

3) Future Work: This work focuses on 3-D change detection. From the above experiments, it can be concluded that the incorporation of prior knowledge can effectively improve the performance of the 3-D change detection deep learning model. Meanwhile, the accuracy of the prior knowledge is positively correlated with the model performance. Although this article proposes an automatic prior change detection method, the

TABLE X
STABILITY OF MODEL ON DIFFERENT DATASETS

Dataset	Means	Standard deviations	Best mIoU
HKCD	76.70 (-0.44)	0.37	77.14
URB3DCD-V2[25]	87.82 (-0.3)	0.31	88.12
SLPCCD[17]	81.87 (-0.3)	0.37	82.17

accuracy of the prior change is low in the face of complex urban environments. In the future, we consider incorporating methods for computing prior knowledge in unsupervised learning [48] into our model to obtain higher accuracy potential variations to further improve model performance. In addition, the proposed HKCD is a binary change dataset, which mainly explores whether a change has occurred or not, and the dataset will be further extended to incorporate semantic information in order to achieve multiclassification change detection. Moreover, the HKCD are all from Hong Kong, whose architectural styles differ greatly from those of Europe and America, and we will incorporate their urban variations into the dataset to accommodate different styles of urban change.

VI. Conclusion

In this article, to achieve 3-D urban realistic change detection, we establish a new publicly available dataset for realistic 3-D change detection in cities, HKCD. The dataset consists of bi-temporal photogrammetric point clouds in Hong Kong and is intended to establish a new paradigm for urban-scale 3-D change detection. In addition, we propose a novel prior-knowledge-guided 3-D change detection method, PGN3DCD, which utilizes the proposed prior knowledge to guide a model focusing on the learning of variations and acquire a more comprehensive and accurate characterization of the variation. The experiments show that PGN3DCD achieves competitive results on several benchmarks, e.g., HKCD (mIoU: 77.14, +4.89%), URB3DCD-V2 (mIoU: 88.12%, +2.93%), and SLPCCD (mIoU: 82.17%, +7.72%), which proves that the PGN3DCD is effective in 3-D change detection in various scenes. However, the addition of new modules increases the number of parameters and reduces the computational efficiency. In the future, we plan to optimize and expand the HKCD dataset, hoping it can be a stepping stone toward advancing 3-D change detection, and design a backbone dedicated to 3-D change detection to achieve more efficient change feature extraction.

Acknowledgment

The authors would like to thank the Common Spatial Data Infrastructure for sharing the 3-D models of the Hong Kong.

References

[1] R. Qin, J. Tian, and P. Reinartz, "3D change detection—Approaches and applications," *ISPRS J. Photogramm. Remote Sens.*, vol. 122, pp. 41–56, Dec. 2016, doi: 10.1016/j.isprsjprs.2016.09.013.

[2] L. Feng, Y. You, W. Liao, J. Pang, R. Hu, and L. Feng, "Multi-scale change monitoring of water environment using cloud computing in optimal resolution remote sensing images," *Energy Rep.*, vol. 8, pp. 13610–13620, Nov. 2022, doi: 10.1016/j.egyr.2022.09.134.

[3] C. He, Y. Zhao, J. Dong, and Y. Xiang, "Use of GAN to help networks to detect urban change accurately," *Remote Sens.*, vol. 14, no. 21, p. 5448, Oct. 2022, doi: 10.3390/rs14215448.

[4] D. Lü, G. Gao, Y. Lü, F. Xiao, and B. Fu, "Detailed land use transition quantification matters for smart land management in drylands: An in-depth analysis in China," *Land Use Policy*, vol. 90, Jan. 2020, Art. no. 104356, doi: 10.1016/j.landusepol.2019.104356.

[5] W. Xiao, H. Cao, M. Tang, Z. Zhang, and N. Chen, "3D urban object change detection from aerial and terrestrial point clouds: A review," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 118, Apr. 2023, Art. no. 103258, doi: 10.1016/j.jag.2023.103258.

[6] U. Stilla and Y. Xu, "Change detection of urban objects using 3D point clouds: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 197, pp. 228–255, Mar. 2023, doi: 10.1016/j.isprsjprs.2023.01.010.

[7] S. Du et al., "Building change detection using old aerial images and new LiDAR data," *Remote Sens.*, vol. 8, no. 12, p. 1030, Dec. 2016, doi: 10.3390/rs8121030.

[8] R. Richter, J. E. Kyprianidis, and J. Döllner, "Out-of-core GPU-based change detection in massive 3D point clouds," *Trans. GIS*, vol. 17, no. 5, pp. 724–741, Oct. 2013, doi: 10.1111/j.1467-9671.2012.01362.x.

[9] C. E. Akumu and S. Dennis, "Urban land cover/use mapping and change detection analysis using multi-temporal Landsat OLI with LiDAR-DEM and derived TPI," *Photogramm. Eng. Remote Sens.*, vol. 88, no. 4, pp. 243–253, Apr. 2022, doi: 10.14358/pers.21-00042r3.

[10] S. Shirowzhan, S. M. E. Sepasgozar, H. Li, J. Trinder, and P. Tang, "Comparative analysis of machine learning and point-based algorithms for detecting 3D changes in buildings over time using bi-temporal LiDAR data," *Autom. Construct.*, vol. 105, Sep. 2019, Art. no. 102841, doi: 10.1016/j.autcon.2019.102841.

[11] B. Gálai and C. Benedek, "Change detection in urban streets by a real time LiDAR scanner and MLS reference data," in *Image Analysis and Recognition*, F. Karray, A. Campilho, and F. Cheriet, Eds., Cham, Switzerland: Springer, 2017, pp. 210–220.

[12] S. Park, S. Ju, S. Yoon, M. H. Nguyen, and J. Heo, "An efficient data structure approach for BIM-to-point-cloud change detection using modifiable nested octree," *Autom. Construct.*, vol. 132, Dec. 2021, Art. no. 103922, doi: 10.1016/j.autcon.2021.103922.

[13] R. Yadav, A. Nascetti, and Y. Ban, "Building change detection using multi-temporal airborne LiDAR data," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLIII-B3, pp. 1377–1383, May 2022, doi: 10.5194/isprs-archives-xliii-b3-2022-1377-2022.

[14] I. de Gélis, S. Saha, M. Shahzad, T. Corpetti, S. Lefèvre, and X. X. Zhu, "Deep unsupervised learning for 3D ALS point clouds change detection," *ISPRS Open J. Photogramm. Remote Sens.*, vol. 9, Aug. 2023, Art. no. 100044, doi: 10.1016/j.ophoto.2023.100044.

[15] S. Fan, W. Gao, and G. Li, "Salient object detection for point clouds," in *Computer Vision—ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham, Switzerland: Springer, 2022, pp. 1–19.

[16] Y. Cao, J. Bin, J. Hamari, E. Blasch, and Z. Liu, "Multimodal object detection by channel switching and spatial attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 403–411, doi: 10.1109/CVPRW59228.2023.00046.

[17] Z. Wang, Y. Zhang, L. Luo, K. Yang, and L. Xie, "An end-to-end point-based method and a new dataset for street-level point cloud change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5703015, doi: 10.1109/TGRS.2023.3295386.

[18] I. de Gélis, S. Lefèvre, and T. Corpetti, "Siamese KPConv: 3D multiple change detection from raw point clouds using deep learning," *ISPRS J. Photogramm. Remote Sens.*, vol. 197, pp. 274–291, Mar. 2023, doi: 10.1016/j.isprsjprs.2023.02.001.

[19] M. Han, J. Sha, Y. Wang, and X. Wang, "PBFormer: Point and bi-spatiotemporal transformer for pointwise change detection of 3D urban point clouds," *Remote Sens.*, vol. 15, no. 9, p. 2314, Apr. 2023, doi: 10.3390/rs15092314.

[20] B. Nagy, L. Kovács, and C. Benedek, "ChangeGAN: A deep network for change detection in coarsely registered point clouds," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 8277–8284, Oct. 2021, doi: 10.1109/LRA.2021.3105721.

[21] Z. J. Yew and G. H. Lee, "City-scale scene change detection using point clouds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13362–13369, doi: 10.1109/ICRA48506.2021.9561855.

[22] Ö. Zováthi, B. Nagy, and C. Benedek, “Point cloud registration and change detection in urban environment using an onboard LiDAR sensor and MLS reference data,” *Int. J. Appl. Earth Observ. Geoinf.*, vol. 110, Jun. 2022, Art. no. 102767, doi: 10.1016/j.jag.2022.102767.

[23] I. de Gélis, S. Lefèvre, and T. Corpetti, “Change detection in urban point clouds: An experimental comparison with simulated 3D datasets,” *Remote Sens.*, vol. 13, no. 13, p. 2629, Jul. 2021, doi: 10.3390/rs13132629.

[24] J. P. Underwood, D. Gillsjö, T. Bailey, and V. Vlaskine, “Explicit 3D change detection using ray-tracing in spherical coordinates,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 4735–4741, doi: 10.1109/ICRA.2013.6631251.

[25] I. de Gélis, T. Corpetti, and S. Lefèvre, “Change detection needs change information: Improving deep 3-D point cloud change detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5701810, doi: 10.1109/TGRS.2024.3359484.

[26] A. G. Melo, M. F. Pinto, L. M. Honório, F. M. Dias, and J. E. N. Masson, “3D correspondence and point projection method for structures deformation analysis,” *IEEE Access*, vol. 8, pp. 177823–177836, 2020, doi: 10.1109/ACCESS.2020.3027205.

[27] X. Dang, “Application of 3D laser scanning technology in monitoring deformation of port trail,” *IOP Conf. Ser., Earth Environ. Sci.*, vol. 783, no. 1, May 2021, Art. no. 012151, doi: 10.1088/1755-1315/783/1/012151.

[28] R. P. de Souza, C. A. Sierra-Franco, P. I. N. Santos, M. P. Rios, D. L. de Mattos Nascimento, and A. B. Raposo, “Automatic deformation detection and analysis visualization of 3D steel structures in as-built point clouds,” in *Human-Computer Interaction. Design and User Experience*, M. Kurosu, Ed., Cham, Switzerland: Springer, 2020, pp. 635–654.

[29] D. Liu, D. Li, M. Wang, and Z. Wang, “3D change detection using adaptive thresholds based on local point cloud density,” *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 3, p. 127, Mar. 2021, doi: 10.3390/ijgi10030127.

[30] R. Qin, X. Huang, A. Gruen, and G. Schmitt, “Object-based 3-D building change detection on multitemporal stereo images,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 5, pp. 2125–2137, May 2015, doi: 10.1109/JSTARS.2015.2424275.

[31] T. Tran, C. Ressl, and N. Pfeifer, “Integrated change detection and classification in urban areas based on airborne laser scanning point clouds,” *Sensors*, vol. 18, no. 2, p. 448, Feb. 2018, doi: 10.3390/s18020448.

[32] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[33] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, vol. 2017. Red Hook, USA: Curran Associates, 2017, pp. 5105–5114.

[34] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, “Rethinking network design and local geometry in point cloud: A simple residual MLP framework,” in *Proc. ICLR*, 2022. [Online]. Available: https://openreview.net/pdf?id=3Pbra-_u76D

[35] W. Wu, Z. Qi, and L. Fuxin, “PointConv: Deep convolutional networks on 3D point clouds,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9613–9622, doi: 10.1109/CVPR.2019.00985.

[36] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph CNN for learning on point clouds,” *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019, doi: 10.1145/3326362.

[37] S. Xiao, H. Lin, C. Wang, S. Wang, and J. C. Rajapakse, “Graph neural networks with multiple prior knowledge for multi-omics data analysis,” *IEEE J. Biomed. Health Informat.*, vol. 27, no. 9, pp. 4591–4600, Sep. 2023, doi: 10.1109/JBHI.2023.3284794.

[38] S. L. Brunton, “Applying machine learning to study fluid mechanics,” *Acta Mechanica Sinica*, vol. 37, no. 12, pp. 1718–1726, Dec. 2021, doi: 10.1007/s10409-021-01143-6.

[39] M. Atwya and G. Panoutsos, “Structure optimization of prior-knowledge-guided neural networks,” *Neurocomputing*, vol. 491, pp. 464–488, Jun. 2022, doi: 10.1016/j.neucom.2022.03.008.

[40] S. Chen, Y. Xu, and B. Zou, “Prior-knowledge-based self-attention network for 3D human pose estimation,” *Expert Syst. Appl.*, vol. 225, Sep. 2023, Art. no. 120213, doi: 10.1016/j.eswa.2023.120213.

[41] M. Qiao et al., “KSTAGE: A knowledge-guided spatial–temporal attention graph learning network for crop yield prediction,” *Inf. Sci.*, vol. 619, pp. 19–37, Jan. 2023, doi: 10.1016/j.ins.2022.10.112.

[42] S. Chakraborty, P. P. Chattopadhyay, S. K. Ghosh, and S. Datta, “Incorporation of prior knowledge in neural network model for continuous cooling of steel using genetic algorithm,” *Appl. Soft Comput.*, vol. 58, pp. 297–306, Sep. 2017, doi: 10.1016/j.asoc.2017.05.001.

[43] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992, doi: 10.1109/34.121791.

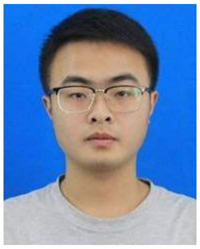
[44] H. Xu, L. Cheng, M. Li, Y. Chen, and L. Zhong, “Using octrees to detect changes to buildings and trees in the urban environment from airborne LiDAR data,” *Remote Sens.*, vol. 7, no. 8, pp. 9682–9704, Jul. 2015, doi: 10.3390/rs70809682.

[45] Q. Hu et al., “Learning semantic segmentation of large-scale point clouds with random sampling,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8338–8354, Nov. 2022, doi: 10.1109/TPAMI.2021.3083288.

[46] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, “Mask3D: Mask transformer for 3D semantic instance segmentation,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 8216–8223, doi: 10.1109/ICRA48891.2023.10160590.

[47] R. Caye Daudt, B. Le Saux, and A. Boulch, “Fully convolutional Siamese networks for change detection,” in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 4063–4067.

[48] P. Naylor, D. Di Carlo, A. Traviglia, M. Yamada, and M. Fiorucci, “Implicit neural representation for change detection,” in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 935–945.



Wenxiao Zhan received the B.E. degree in geographic information science from the School of Resource and Environmental Sciences, Wuhan University, Wuhan, China, in 2020, where he is currently pursuing the Ph.D. degree in map cartography and geographic information engineering with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing.

His research interests include deep learning, analysis of point clouds, and 3-D GIS.

Ruozhen Cheng received the Ph.D. degree in map cartography and geographic information engineering from the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China, in 2022.

Her research interests include deep learning, natural language processing, semantic networks, and 3-D GIS.

Jing Chen received the bachelor’s degree from the East China University of Technology, Nanchang, Jiangxi, China, in 1998, and the master’s and Ph.D. degrees from Wuhan University, Wuhan, China, in 2001 and 2004, respectively.

Since 2004, he has been working at the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, where he is currently a Professor Fellow. His research interests include deep learning, 3-D reconstruction, and 3-D GIS.