TCIANet: Transformer-Based Context Information Aggregation Network for Remote Sensing Image Change Detection

Xintao Xu, Jinjiang Li, and Zheng Chen

***Abstract*—Change detection based on remote sensing data is an important method to detect the earth surface changes. With the development of deep learning, convolutional neural networks have excelled in the field of change detection. However, the existing neural network models are susceptible to external factors in the change detection process, leading to pseudo change and missed detection in the detection results. In order to better achieve the change detection effect and improve the ability to discriminate pseudo change, this article proposes a new method, namely, transformer-based context information aggregation network for remote sensing image change detection. First, we use a filter-based visual tokenizer to segment each temporal feature map into multiple visual semantic tokens. Second, the addition of the progressive sampling vision transformer not only effectively excludes the interference of irrelevant changes, but also uses the transformer encoder to obtain compact spatiotemporal context information in the token set. Then, the tokens containing rich semantic information are fed into the pixel space, and the transformer decoder is used to acquire pixel-level features. In addition, we use the feature fusion module to fuse low-level semantic feature information to complete the extraction of coarse contour information of the changed region. Then, the semantic relationships between object regions and contours are captured by the contour-graph reasoning module to obtain feature maps with complete edge information. Finally, the prediction model is used to discriminate the change of feature information and generate the final change map. Numerous experimental results show that our method has more obvious advantages in visual effect and quantitative evaluation than other methods.**

***Index Terms*—Attention mechanism, bitemporal remote sensing images, change detection (CD), graph convolutional network (GCN), transformers.**

I. Introduction

As AN important application of remote sensing images, change detection aims to analyze the images of the same geographical area captured at different moments and detect the change information of surface features. The purpose of the research is to find the change information of interest and filter out the irrelevant change information that appears as interference factors. With the rapid development of remote sensing technology, change detection technology has been applied in various fields such as disaster monitoring and assessment [1], [2], land surveying [3], [4], and urban planning [5]. In most of the change detection applications, the commonly used methods are still visual interpretation and human–computer interaction interpretation, which consumes a lot of manpower, time, and other resources and has low processing efficiency. Therefore, an efficient and automatic method for remote sensing image change detection is particularly important.

In change detection, changes in the surrounding environment can make it more difficult to extract important information from remote sensing images. Therefore, some traditional methods have been proposed by domestic and foreign scholars to solve various problems in change detection [6], [7], [8]. Chen and Lin [6] used a multithreshold strategy for the change detection of urban buildings by Lidar and aerial images. Bourdis et al. [7] proposed an optical-flow-based change detection method to solve the parallax problem. In addition, Benedek and Szirányi [8] used a multilayer conditional mixed Markov model for change detection in aerial images. However, these methods are vulnerable to parameters and other environmental factors (e.g., shadows and obscured objects). And these methods require expertise to construct and select features manually, with low generalization performance across regions and datasets.

In recent years, with the development of artificial intelligence technology, remote sensing image change detection methods based on deep learning are gradually applied to the field of change detection by virtue of their ultrahigh accuracy and automation. Convolutional neural networks (CNNs) can extract high-level semantic features of interest from each temporal image due to their powerful recognition ability, which provides better robustness compared to traditional methods. Most deep-learning-based change detection methods [9], [10], [11], [12] usually first use CNN models based on Siamese networks (e.g., UNet [13] and DeepLab [14]) to extract deep-level

Manuscript received 18 October 2022; revised 13 December 2022; accepted 27 January 2023. Date of publication 31 January 2023; date of current version 16 February 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62002200, Grant 62202268, and Grant 62272281, in part by the Shandong Provincial Science and Technology Support Program of Youth Innovation Team in Colleges under Grant 2021KJ069 and Grant 2019KJN042, in part by the Shandong Natural Science Foundation of China under Grant ZR2021MF107 and Grant ZR2022MA076, and in part by the Yantai Science and Technology Innovation Development Plan under Grant 2022JCYJ031. *(Corresponding author: Jinjiang Li.)*

Xintao Xu is with the School of Information and Electronic Engineering, Shandong Technology and Business University, Yantai 264005, China (e-mail: 2865247945@qq.com).

Jinjiang Li and Zheng Chen are with the School of Computer Science and Technology, Shandong Technology and Business University, Yantai 264005, China (e-mail: lijinjiang@gmail.com; chenzheng@sdtbu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3241157

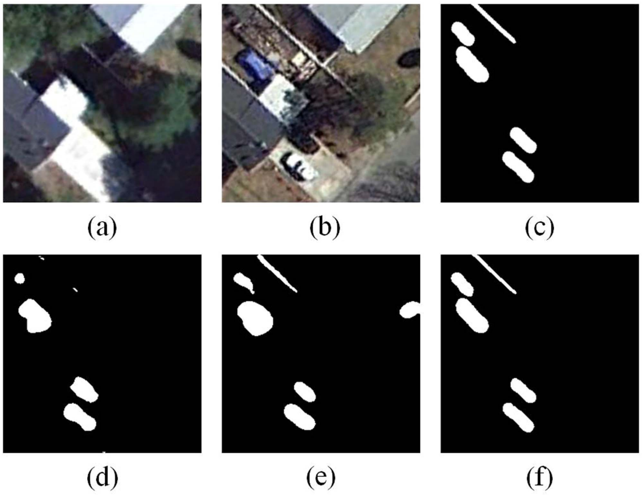

Fig. 1. Bitemporal remote sensing images with seasonal changes in the CDD dataset. (a) and (b) Remote sensing images before and after the change, respectively. (c) Ground truth. (d)–(f) Results obtained by IFN, STANet, and our method, respectively.

feature information from dual-temporal remote sensing images. The full convolutional network is then used to obtain the final change detection results. In addition, many new techniques have been introduced in recent networks in order to highlight the change information and improve the feature discrimination of the network. These include the use of deeper CNN models [15], [16], multiscale feature fusion [12], [17], and the application of dilated convolution [15]. For example, Zhang et al. [16] designed a feature difference CNN to generate feature difference maps at different scales and depths. Xu et al. [17] proposed a multiscale fusion network with multichannel information flows, which makes data transmission more flexible and can highlight important features.

Based on the prior analysis, although many deep-learning-based models have been proposed and used for remote sensing image change detection tasks, they still have some problems. First, the existing methods ignore spatiotemporal context information in the feature encoding and decoding process and do not link high-level features with low-level features. This often leads to unclear boundaries of the changing target area as well as ignoring changes in small objects, as shown in Fig. 1(d). Second, bitemporal remote sensing images are easily affected by factors such as light and seasonal changes during the acquisition process, resulting in many differences in the images with irrelevant geographical elements. These differences can cause the feature mapping of the bitemporal images to be underutilized, resulting in pseudo change in the final results, as shown in Fig. 1(e). In addition, as the transformer is widely used in the field of computer vision (CV), change detection tasks have also started to try to use the transformer to process remote sensing image features with good results. However, the extraction and processing of feature information in bitemporal images need to be enhanced.

To address the above problems, in this article, we propose a transformer-based context information aggregation network (TCIANet) for remote sensing image change detection. First, we use the Siamese backbone network to extract the features at different levels of the bitemporal remote sensing images. Then, the deep feature information is extracted by a filter-based visual tokenizer (FVT) [18] into several compact semantic tokens that contain high-level semantic information of the image. Immediately after, we employ a progressive sampling vision transformer (PS-ViT) [19] and a transformer decoder (TD) to model and aggregate the rich spatiotemporal context semantic information in the token set, so as to better reveal the changes in the feature targets of interest. Moreover, the interference of irrelevant factors is excluded, and the change features can be extracted more accurately. In addition, we use a feature fusion module (FFM) and a contour-graph reasoning module (CGRM) [20] to strengthen the intrinsic connection between higher level features and lower level features. Thus, we can better retain the edge information of the target change region, reduce the boundary error, and enhance the feature representation. Finally, we pass the feature difference map through the prediction module (PM) to obtain the final change detection result. As shown in Fig. 1(f), our method can detect the changing buildings and roads well, and the results are more satisfactory.

The main contributions of this article can be summarized as follows.

1) We propose a novel change detection method for remote sensing images based on transformer architecture. Compared with the existing methods, our method achieves better change detection performance on all three public change detection datasets. The effectiveness and superiority of our method in image processing is demonstrated.
2) In order to strengthen the spatiotemporal connection of objects of the same type, we use an FVT. The tokenizer can express the feature map as multiple compact semantic tokens and represent high-level concepts through a set of tokens.
3) We process the semantic information in tokens with the PS-ViT, which uses a progressive iterative sampling strategy to locate regions of change. And PS-ViT can model the spatiotemporal context information in the set of tokens to detect changes of interest and exclude irrelevant changes. In addition, we use an improved TD model to project the learned high-level semantic concepts back into the pixel space, thus helping the original features to be optimized.
4) We use the FFM to fuse low-level semantic feature information to obtain rough contour feature maps. Meanwhile, we introduce an effective CGRM. This module can capture the semantic relationship between regions and contour features through graph reasoning, thus reducing boundary errors and improving change detection performance.

The rest of this article is organized as follows. Section II presents the related work, which describes the current development of remote sensing image change detection techniques. Section III gives the detailed description of the proposed method. Section IV conducts some experiments and discusses the experimental results. Finally, Section V concludes this article.

II. Related Work

Over the past few decades, change detection technology has been gradually developed, and many scholars at home and abroad have conducted intensive research on it and achieved many results. According to the principle of remote sensing image change detection method and the way of processing data, we briefly introduce several common change detection methods, including change detection based on traditional methods and change detection based on deep learning. In addition, we briefly review the development and application of transformer and graph convolutional network (GCN).

A. Change Detection Based on Traditional Methods

Depending on the unit of image analysis, traditional change detection methods can be divided into pixel-based and object-based methods [3]. The pixel-based approach simply analyzes the pixel-by-pixel spectral differences in the remotely sensed image and selects a suitable threshold to classify the pixels, so as to obtain the final change detection difference map [21], [22], [23], [24]. For example, change vector analysis [21] obtains a change feature vector by calculating the difference between the corresponding bands of the image. The length of the change vector represents the change intensity, and the direction of the change vector describes the type of change. Principal component analysis (PCA) [22] is used to enhance the change information in multisensor data by first calculating the difference images and then extracting the principal components using PCA. Multivariate alteration detection [23] is based on the criterion of maximum variance of the projected feature differences, thus minimizing the radiometric variability in the differences to highlight the change information. Slow feature analysis [24] extracts time-dimensional invariant features from multitemporal remote sensing images and suppresses the differences between unchanged pixels, so as to better separate the changed pixels. Generally speaking, pixel-based methods are more suitable for low- and medium-resolution remote sensing images.

In contrast, the object-based approach treats the feature object as the minimum processing unit. Differences in temporal images are analyzed by making full use of the spectral and spatial features of the objects [25], [26], [27], [28], [29]. For example, Qin et al. [25] proposed an object-based land cover change detection method for cross-sensor remote sensing images. Feng et al. [26] can effectively improve the accuracy of change detection by combining visual saliency and random forest. Huo et al. [28] enhance the ability to discriminate between change and nonchange classes by object-level features and use progressive change feature classification to improve performance. Chen et al. [29] used the image object detection approach to identify changing regions in high-resolution satellite images.

B. Change Detection Based on Deep Learning

In recent years, deep learning methods have been widely used in the field of remote sensing [30], [31], [32] due to their powerful feature extraction capability, and change detection methods based on deep learning have emerged. Since the input of the change detection task is bitemporal or multitemporal remote sensing data, the change detection method based on deep learning can be divided into a single-stream network [13], [33], [34], [35] and a dual-stream network [36], [37], [38], [39] according to the process of feature extraction or potential feature representation of different temporal remote sensing data.

Single-stream networks are usually semantic segmentation networks. The bitemporal remote sensing images are usually fused directly, and the fused data are fed into a classification network for change detection. Peng et al. [13] first connected bitemporal remote sensing images in the channel dimension and then fed into a modified UNet++ network [40] for change detection, which directly outputs the final change detection difference map. The dual-stream network processes the bitemporal remote sensing images separately by using the Siamese network, and then, the relationship between the two is considered to obtain the final change detection difference map. Zhang et al. [36] proposed a Siamese network framework with hierarchical fusion strategy for change detection tasks. Owing to its features, such as weight sharing strategy and improved detection accuracy, dual-stream networks have become the dominant framework for most change detection tasks.

Although convolutional networks can process multispectral and hyperspectral images well, the results obtained are still flawed due to the infrequent spectral information in remote sensing maps and the complexity of objects in different scenes. In addition, consider that pure convolutional networks are intrinsically limited by the size of the received field (RF) per pixel. Many recent studies use attention mechanisms [9], [41], [42], [43] to further extend the RF of the model and increase the distinction between parts of interest, so as to better utilize the rich spatial information in remotely sensed images. Chen et al. [41] used a dual-attention module to emphasize the change information in diachronic features. Liu et al. [43] used a stacked attention module consisting of multiple attention modules to fully extract multilevel information from remote sensing images.

Different from the existing deep-learning-based approaches that directly model dense relationships between any elements in pixel-based space, we are modeling the global semantic information in the bitemporal feature maps. Specifically, we aggregate the feature information extracted from the images into several compact semantic tokens and model the context based on these tokens. Then, the tokens that have learned rich semantic information are used to enhance the original features in the pixel space. In addition, we use graph convolution to capture more spatiotemporal information features and further improve the change detection capability of the network.

C. Transformer

Vaswani et al. [44] first proposed the transformer due to its unique design endowing the transformer with the ability to handle indefinitely long inputs, capture long-distance dependencies, and sequence-to-sequence properties. Since then, the transformer has achieved excellent results in natural language processing tasks [45]. Compared with the CNN, the transformer mainly uses the self-attention mechanism to extract the intrinsic features of the target, which can effectively extract and process the global features. With its own powerful feature representation capability, researchers have gradually applied transformer to CV tasks, including image classification [46], super-resolution [47], [48], image segmentation [16], [49], and object detection [50].

In view of the high performance of transformer and the absence of human-defined perceptual bias, it has also attracted the interest of researchers to apply it in the field of remote sensing, including hyperspectral image classification [51], [52], remote sensing image captioning [53], [54], and scene classification [55]. For example, He et al. [51] proposed a HIS-BERT to capture the global dependencies between pixels in hyperspectral images and can realize flexible and dynamic input areas. Shen et al. [53] used a transformer to decode image features into multiple sentences and improve the quality of sentences through reinforcement learning. Bazi et al. [55] used the multihead attention mechanism as the main module to acquire the remote context relationship between pixels in an image. Since the transformer has achieved good results in several fields of remote sensing images, researchers have also started to try to apply it to change detection tasks. Chen et al. [18] used the transformer to better learn the context of bitemporal images, which facilitates the identification of changes of interest and excludes irrelevant changes. Feng et al. [56] extracted local and global features of images by the CNN and the transformer, respectively, and used the attention module for interactive communication. In this article, we also apply the transformer to our change detection network, so as to acquire the global environment of the input image and capture the dependence between pixels.

D. Graph Convolutional Network

Since CNNs cannot handle unstructured graph data, many researchers began to extend neural networks in the hope of processing graph data, thus giving birth to graph neural networks. The proposed GCN is a highly landmark stage in the development of graph neural networks. The GCN realizes convolution operation in the spatial domain by using approximation in the frequency domain and has made great progress in many fields.

In recent years, graph convolution methods have been gradually applied to remote sensing due to the powerful analysis capability of the GCN for graph data. Liu et al. [57] used the CNN and the GCN to learn features for areas of different sizes and generate complementary spectral spatial features, respectively. Zhang et al. [58] used graph convolution to construct a graph structure in the generated feature objects, which is used to leverage the relativity between objects to produce accurate classification. Tang et al. [59] captured short- and long-range contextual patterns in feature maps by a multiscale dynamic GCN to fully extract the changed and unchanged regions. Qu et al. [60] proposed a novel dual-branch difference amplification GCN method by extracting and amplifying the difference features of multitemporal remote sensing images for change detection. For our remote sensing image change detection task, we introduce a new CGRM. This module uses graph reasoning to capture the correlation between contour features and contextual information of different regions.

III. Methodology

In this section, we describe in detail the architecture of the proposed network. First, the overall architecture of the network proposed in this article is introduced. Then, the various parts of the model are described in detail. Finally, the loss function we use is presented.

A. Network Architecture

The overall structure of the proposed network is shown in Fig. 2. We use multitemporal image pairs as input to the network. First, we use two weight-shared ResNet18s [61] to feed bitemporal remote sensing image pairs (I_1 and I_2) into the feature extractor in order to obtain features at different levels of each input image for multiscale representation. The feature mapping of each image is then converted into compact visual semantic tokens using the FVT [18] with differential fusion. Immediately after, they are fed into the PS-ViT [19] to obtain the global semantic information in the token sets and generate a rich context representation for each temporal. Subsequently, a modified TD is used to project the corresponding semantic tokens into the pixel space to obtain the features of each temporal refinement. In addition, we use the FFM to fuse the feature information of low-level semantics to generate a rough initial contour map. Then, the contour and deep feature maps are fed into the CGRM [20], which learns the intrinsic graph representation to capture the semantic relationships between regions and edges to obtain a refined feature map. Next, the bitemporal features extracted in the TD and the CGRM are pixel-subtracted to obtain two feature difference maps, respectively. Finally, we connect the two feature maps and go through the PM to get the final change map.

The algorithm flowchart of our proposed method is shown in Algorithm 1.

B. Feature Extractor

The detailed structure of the feature extractor is shown in Fig. 2(a), where we use the Siamese-network-based ResNet. The Siamese network is to extract the remote sensing image features at moments I_1 and I_2 with the same network structure and shared parameters. For ResNet, it not only has strong feature extraction ability, but also does not show performance degradation with the increase in the number of network layers. Therefore, in this article, we use the improved ResNet18 to extract bitemporal image feature maps. Since the original fully connected layer is removed from the classical ResNet18, our feature extractor contains two convolutional layers, four residual blocks (ResBlocks), and a bilinear interpolation layer. The detailed configuration of the feature extractor is shown in Table I.

As can be seen in Table I, the first convolutional layer in the feature extractor with a step size of 2 to extract shallow features is half the size of the original image. The features of $1/4$ of the image are then obtained using a maximum pooling layer of 3×3 with a step size of 2, and the important features can be filtered. In addition, each ResBlock contains two convolution layers, a batch normalization layer, and a rectified linear unit (ReLU) function.

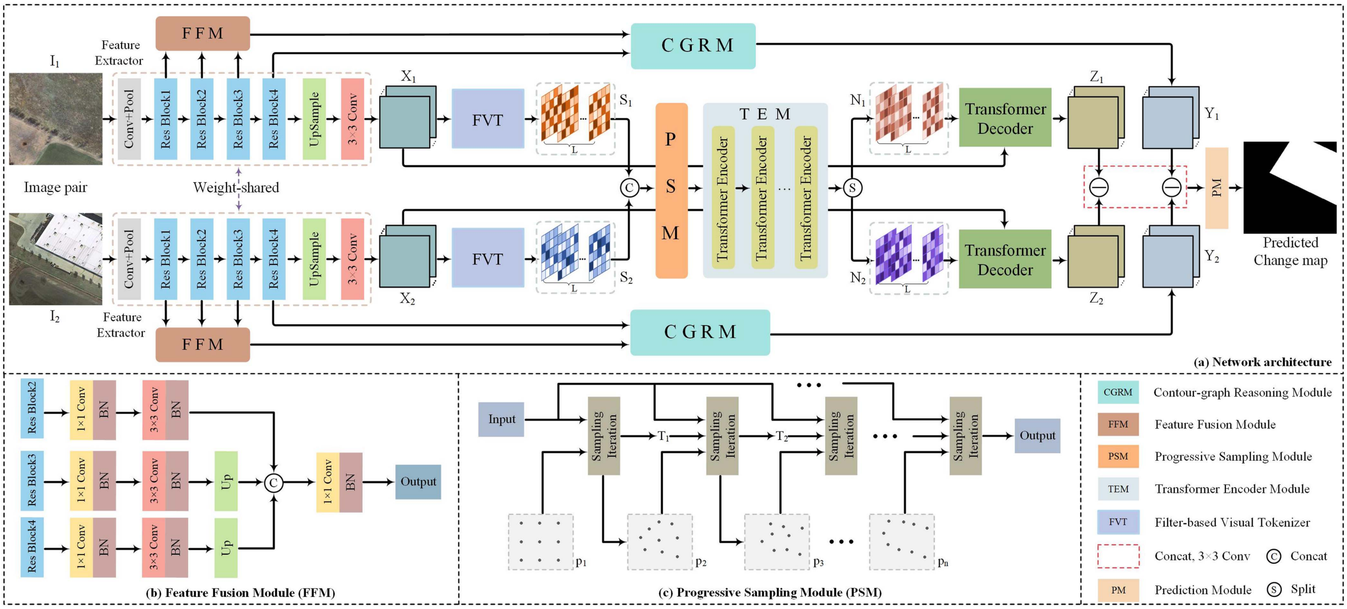

Fig. 2. Overall structure of the proposed network model. (a) Network architecture. (b) Components of the FFM. (c) Components of the PSM.

TABLE I
DETAILED CONFIGURATION OF THE FEATURE EXTRACTOR

Blocks	Module Name	Module Details	Output Size
ResNet	Conv_1	7 × 7, 64, stride 2	128 × 128
	Resb_1	3 × 3 max pool, stride 2	64 × 64
		$\begin{bmatrix} 3\times 3, & 64 \\ 3\times 3, & 64 \end{bmatrix} \times 2$	64 × 64
	Resb_2	$\begin{bmatrix} 3\times 3, & 128 \\ 3\times 3, & 128 \end{bmatrix} \times 2$	32 × 32
	Resb_3	$\begin{bmatrix} 3\times 3, & 256 \\ 3\times 3, & 256 \end{bmatrix} \times 2$	16 × 16
	Resb_4	$\begin{bmatrix} 3\times 3, & 512 \\ 3\times 3, & 512 \end{bmatrix} \times 2$	16 × 16
_	Bilinear	_	64 × 64
_	Conv_2	3 × 3, 32, stride 1	64 × 64

We use the CNN backbone as the feature extractor, which is modified from ResNet18. The size of the original input image is 256 × 256.

Since the step size of the first convolution layer in ResBlock1 and ResBlock4 is 1, the size of the output feature map is the same as the input. The step size of the first convolution layer in ResBlock2 and ResBlock3 is 2, so the output feature map is half the size of the input feature map. In addition, to compensate for the reduction of global semantic information in the deep network due to successive downsampling operations, we added a bilinear difference layer and a convolutional layer of 3 × 3 to the back of ResNet. Thus, the perceptual field can be increased and the loss of spatial information can be reduced, and richer semantic features can be extracted. Finally, the feature extractor outputs the feature map size of 64 × 64 and the number of channels is 32.

C. *Filter-Based Visual Tokenizer*

Since remote sensing images of different temporal phases are imaged in different seasons and lighting conditions, it may result in ground targets with the same semantic concept exhibiting different spectral characteristics in different times and different spatial positions. We think that the changes between two images can be described by several high-level concepts (semantic tokens). For this purpose, we introduce the FVT [18] in the network to extract compact visual semantic tokens from each temporal feature map. In simple terms, our tokenizer is to split the whole image into parts and represent each part with a token. Moreover, the semantic information between bitemporal images can be shared, so we use the Siamese tokenizer. Fig. 3(a) shows the detailed processing of the FVT.

In Fig. 3, we use $X_i \in R^{H\times W\times C}$ (height H, width W, channel dimension $C, i = 1, 2$) to represent the input bitemporal feature maps. For each pixel in the feature maps $X_i (i = 1, 2)$, we apply 1×1 convolution to divide it into L semantic groups, and these semantic groups represent the semantic information of the feature map. Then, within each semantic group, we use the

Algorithm 1: Transformer-Based Context Information Aggregation Network.

```
Input: Bitemporal remote sensing images $I = \{(I_1, I_2)\}$
Output: Predicted change map $M^*$
// Obtain different levels of features by feature extractor
for $i$ in $\{1, 2\}$ do
    for $j$ in $\{1, 2, 3, 4\}$ do
        $X_{ij} = Feature\_Extractor(I_i)$
    end for
end for
// Use FVT to convert each temporal feature into the compact semantic token
for $i$ in $\{1, 2\}$ do
    $S_i = FVT(X_{i4})$
end for
$S = Concat(S_1, S_2)$
// Use PS-ViT to get compact global context tokens
$S_{new} = PS{-}ViT(S)$
$N_1, N_2 = Split(S_{new})$
// Use transformer decoder to get refined pixel-level features
for $i$ in $\{1, 2\}$ do
    $Z_i = Transformer\_Decoder(X_{i4}, N_i)$
end for
// Use FFM and CGRM to refine the boundaries of the change region and get accurate feature maps
for $i$ in $\{1, 2\}$ do
    $C_i = FFM(X_{i1}, X_{i2}, X_{i3})$
    $Y_i = CGRM(C_i, X_{i4})$
end for
// Generate feature difference maps
$Z = |Z_1 - Z_2|$
$Y = |Y_1 - Y_2|$
$M = Concat(Z, Y)$
// The final change map is obtained through the prediction module
$M^* = PM(M)$
```

softmax function to operate on their $H \times W$ spaces to generate the spatial attention maps $A_i (i = 1, 2)$. Finally, we multiply the spatial attention feature maps A_i by X_i. And the weighted average sum operation is performed on the pixels in X_i to obtain L compact visual sets, i.e., semantic tokens S_i. Formally

$$S_i = (A_i)^T X_i = (soft\max(\varphi(X_i; W_A)))^T X_i \qquad (1)$$

where $\varphi(\cdot)$ denotes 1×1 convolution, $W_A \in R^{C\times L}$ is the semantic group formed on the basis of X_i, and $S_i \in R^{L\times C} (L \ll HW)$ denotes the generated semantic tokens. L is the number of tokens, which is set to 64 in the module. $soft\max(\cdot)$ denotes regularizing each semantic group with softmax function and converting its activation into spatial attention map $A_i \in R^{HW\times L}$.

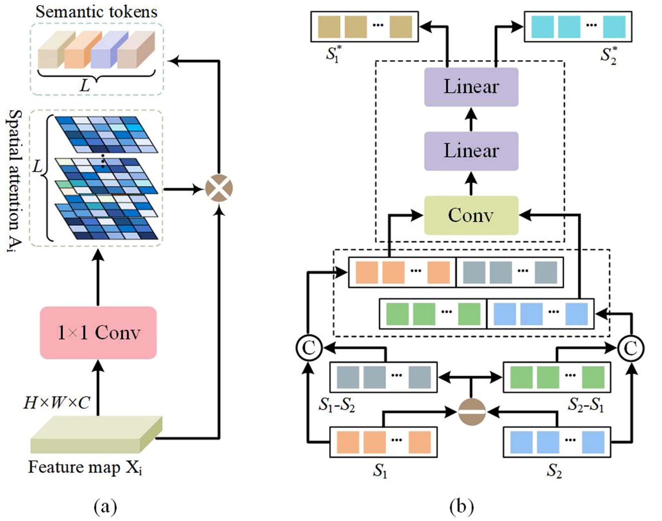

Fig. 3. Illustration of the FVT. (a) Running process of the FVT. (b) Process of differentiated fusion.

After we obtain the semantic tokens $S_i \in R^{L\times C} (i = 1, 2)$ of two remote sensing images, we perform the subtraction operation of these two semantic tokens with each other to generate two new semantic token differences $S_1 - S_2$ and $S_2 - S_1$. Then, these two semantic token differences are connected with the corresponding semantic tokens $S_i (i = 1, 2)$, respectively, to obtain two new token sequences. Immediately after, these two token sequences are fed into a shallow CNN (consisting of one convolutional layer and two linear layers) to obtain two differentially fused semantic tokens $S_i^* \in R^{L\times C} (i = 1, 2)$, respectively. And the generated differentially fused semantic tokens $S_i^* (i = 1, 2)$ and $S_i (i = 1, 2)$ have the same dimensionality. The specific process is shown in Fig. 3(b).

D. *Progressive Sampling Vision Transformer*

ViT [62] can simply divide the image into tokens of fixed length and use the transformer to learn the semantic relationships between these tokens. Therefore, after obtaining two sets of differentially fused semantic tokens $S_i^* (i = 1, 2)$ of bitemporal feature images, we make full use of the global spatial–temporal semantic relations based on the tokens by the transformer so that we can extract rich context information in each temporal. However, we consider that the traditional ViT [62] simply segments the image, which destroys the inherent object structure and makes it difficult for the network to focus on the important object regions. Therefore, we introduce the PS-ViT [19]. As shown in Fig. 2(a), the PS-ViT is composed of a progressive sampling module (PSM) and a transformer encoder module (TEM). This module follows the architecture used in ViT [62] and reduces the damage of tokenization on the image structure by adopting a progressive iterative sampling strategy to locate discriminative regions. Moreover, it tends to sample object regions that are relevant to the semantic structure, thus detecting changes in the feature target of interest and excluding irrelevant changes.

1) Progressive Sampling Module: Fig. 2(c) shows the detailed structure of the PSM. It can be seen that the PSM is an

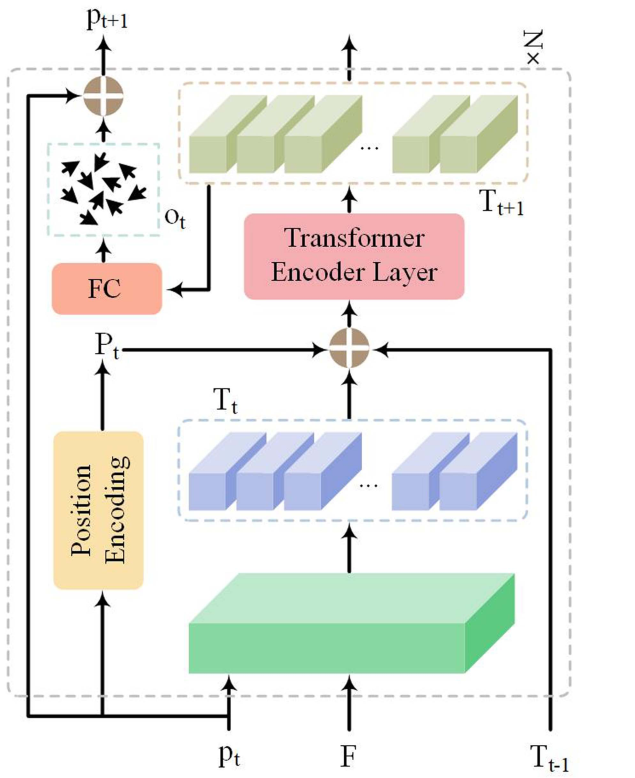

Fig. 4. Running process of PSM. During each iteration, given the sampling position p_t and the feature map F, the initial token is first sampled at p_t of F to obtain the initial sampled token T_t. Then, T_t, the position encoding P_t, and the output token T_{t-1} from the previous iteration are added element by element and transmitted to the transformer encoder layer to obtain the predicted output token T_{t+1}. In addition, T_{t+1} is passed through a fully connected layer to obtain the offset matrix o_t. Finally, p_t and o_t are added to obtain the next sampling position p_{t+1}. The running process requires N iterations.

iterative framework consisting of multiple sampling iteration blocks. In addition, Fig. 4 illustrates the operation process of the PSM in each iteration.

Before entering the PSM, we first concatenate two sets of differentially fused semantic tokens together. The connected set of tokens is then reshaped into a 2-D feature mapping $F \in R^{H\times W\times C}$ (height H, width W, and channel dimension C), and F is used as the input to the PSM. Then, after progressive sampling, it is expanded into a token set $T_N \in R^{C\times(n\times n)}$, where $(n \times n)$ denotes the number of samples in the input feature map and N is the total number of iterations of the PSM, which we set to 4 in the module. The detailed process is shown in Fig. 4; during the tth iteration, the initial token is first sampled at the sampling position p_t in the input feature map

$$T_t = F(p_t), \quad t \in \{1, \ldots, N\} \tag{2}$$

where $T_t \in R^{C\times(n\times n)}$ is the initial sampling token at the tth iteration, $F(\cdot)$ represents the sampling operation in the feature map, and $p_t \in R^{2\times(n\times n)}$ represents the sampling position matrix. Then, we send the sampling position p_t to the position encoding layer to obtain the coding matrix P_t of size $C \times (n \times n)$. Next, we add the initial sampling token T_t generated during the tth iteration, the token T_{t-1} output from the $(t-1)$th iteration, and the position encoding matrix P_t generated in the tth iteration element by element to obtain the intermediate token H_t. Finally, H_t is conveyed to the transformer encoder layer to get the output token T_{t+1} of the current iteration. Formally

$$P_t = W_t p_t \tag{3}$$

$$H_t = T_t \oplus P_t \oplus T_{t-1} \tag{4}$$

$$T_{t+1} = Transformer\,(H_t)\,, t \in \{1, \ldots, N\} \tag{5}$$

where $W_t \in R^{C\times 2}$ denotes the linear transformation operation of the position encoding layer, and $\oplus$ denotes element summation. $Transformer(\cdot)$ is the transformer encoder layer based on multiheaded attention, and the detailed structure will be described in the next subsection. In addition, except for the last iteration, the output token T_{t+1} is passed through a full connection layer to obtain the predicted offset matrix o_t. Then, the current sampling position p_t is added to the generated offset vector o_t to obtain the next sampling position p_{t+1}:

$$o_t = M_t T_{t+1}, \quad t \in \{1, \ldots, N-1\} \tag{6}$$

$$p_{t+1} = p_t + o_t, \quad t \in \{1, \ldots, N-1\} \tag{7}$$

where $M_t \in R^{2\times C}$ is the learnable linear transformation matrix, T_{t+1} is the output token of the current iteration, $p_t \in R^{2\times(n\times n)}$ denotes the sampling position matrix, and $o_t \in R^{2\times(n\times n)}$ denotes the offset matrix.

By using the progressive sampling strategy, the PSM continuously updates the sampling position in an iterative manner. And by using the transformer's ability to capture global information, it makes the network adaptively focus on the region of interest in the object by combining the local context and the current tokens' position.

2) Transformer Encoder: Since the transformer model can capture the dependency between pixels in the image by cascading multiple transformer layers to obtain the global environment of the input image. Moreover, the transformer can generate rich token representation for each temporal image by exploiting the global semantic relations in the token. Therefore, we use the TEM to model the context in T_N after obtaining the final output token T_N by the PSM. As can be seen from Fig. 2(a), the TEM consists of multiple transformer encoder layers. Here, we should note that the positional information is already preserved in the token T_N output by the last iteration in the PSM. Therefore, we do not need to add additional positional embedding. Fig. 5 shows the detailed structure of the transformer encoder layer. From the figure, it can be seen that the transformer encoder layer has a standard transformer structure [55], which consists of N_E layers of multihead self-attention (MSA) blocks and multilayer perceptron (MLP) blocks. And, a layer normalization (LN) block is used before both MSA and MLP for normalizing the activation values of each layer. In addition, the residual connections in the figure are used to prevent network degradation.

According to Fig. 5, MSA contains different heads, and each head cannot share parameters between them. Among them, each head contains two steps: linear transformation and scale dot product attention (SDPA). First, in each layer l, three learnable linear projection layers are applied to map the input token $T_N^{(l-1)}$ into different weight matrices, i.e., query (Q), key (K), and value

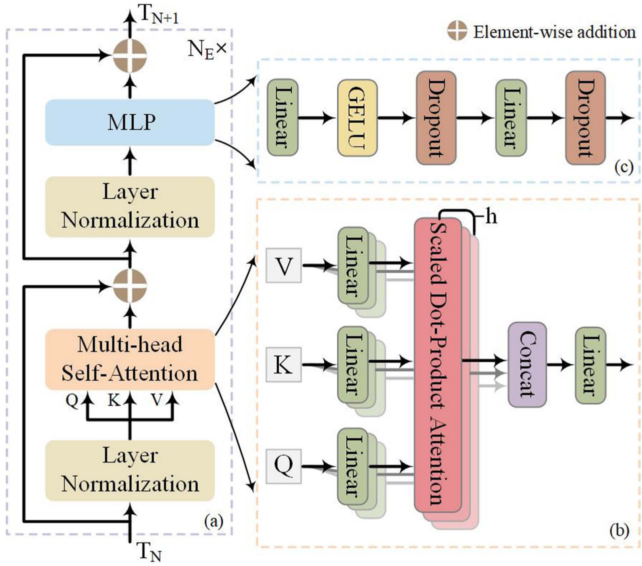

Fig. 5. Illustration of the transformer encoder layer. (a) Basic structure. (b) Details of the MSA block. (c) Details of the MLP block.

(V), and use them as inputs to the MSA. This can be expressed as

$$Q = T_N^{(l-1)} W^Q \tag{8}$$

$$K = T_N^{(l-1)} W^K \tag{9}$$

$$V = T_N^{(l-1)} W^V \tag{10}$$

where W^Q, W^K, and W^V are the weights of the linear transform layers where the maps Q, K, and V are located, respectively. Then, the correlation between Q and K is calculated in SDPA using dot product operation and softmax activation function and generates an attention map as the weight of V as follows:

$$\text{SDPA}(Q, K, V) = soft\max\left(QK^T/\sqrt{d}\right) V \tag{11}$$

where K^T denotes the transpose of K, and d is the number of columns of Q and K matrices. $soft\max(\cdot)$ denotes softmax operation on each row of the weight matrix in the channel dimension. Finally, since MSA performs multiple independent attention heads in parallel, we concatenate the output of each attention head to obtain the final output. Therefore, we connect the outputs of multiple attention heads together and pass them into a linear mapping layer to obtain the final output of the MSA. The formula is expressed as follows:

$$head_i = \text{SDPA}\left(T_N^{(l-1)} W_i^Q, T_N^{(l-1)} W_i^K, T_N^{(l-1)} W_i^V\right) \tag{12}$$

$$\text{MSA}(Q, K, V) = Concat\left(head_1, \ldots, head_h\right) W^O \tag{13}$$

where W_i^Q, W_i^K, and W_i^V denote the weights of the linear layer of the ith head for the maps Q, K, and V, respectively. h is the number of attention heads, W^O is the weight of the last linear layer in the MSA, and $Concat(\cdot)$ denotes stacking in column vectors.

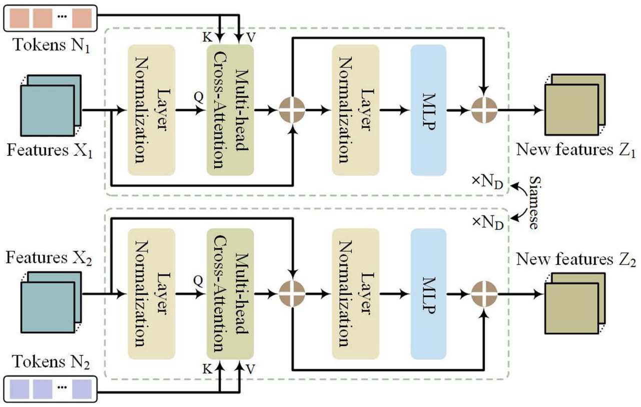

Fig. 6. Specific structure of the TD model. The ⊕ represents elementwise addition.

Also, as seen in Fig. 5, the MLP block consists of two linear transform layers and a Gaussian error linear unit (GELU) function. Formally

$$\text{MLP}\left(T^{(l-1)}\right) = \phi\left(T^{(l-1)} W_1\right) W_2 \tag{14}$$

where W_1 and W_2 are the projection matrices in the linear layer, respectively, and $\phi(\cdot)$ is the GELU function.

E. Transformer Decoder

With the PS-ViT [19] module, we have extracted two new sets of tokens $N_i (i = 1, 2)$ with context information for the bitemporal images. These tokens allow the network to better focus on regions where changes occur between bitemporal images. Then, we need to project these high-level semantic information into pixel space to obtain pixel-level features of the image. Therefore, we use the improved Siamese TD [44] to refine the image features for each temporal, allowing the network to clearly distinguish the difference between the two new feature maps generated. The detailed operation is shown in Fig. 6. First, we represent the feature maps $X_i (i = 1, 2)$ by two sequences. Then, improved TDs obtain refined feature maps $Z_i (i = 1, 2)$ based on the association of each pixel with each semantic token N_i.

As can be seen in Fig. 6, our TD has a similar structure to the encoder layer, consisting of N_D layers of multihead cross attention (MCA) block and an MLP block. Also, an LN block is used before both MCA and MLP. Here, we do not use the MSA block to avoid too much computation of dense information between pixels inside X_i. For the input of the MCA block, it is different from the input of the MSA block in the transformer encoder. We consider that the compact semantic token can be used to represent each pixel on the feature map. Therefore, we use the pixels in the original image features $X_i (i = 1, 2)$ as query (Q) and tokens $N_i (i = 1, 2)$ as key (K) and value (V). In each layer l, MCA can be expressed by the following formula:

$$head_j = Atten\left(X_{i,(l-1)} W_j^Q, M_i W_j^K, M_i W_j^V\right) \tag{15}$$

$$\text{MCA}(Q, K, V) = Concat\left(head_1, \ldots, head_h\right) W^O \tag{16}$$

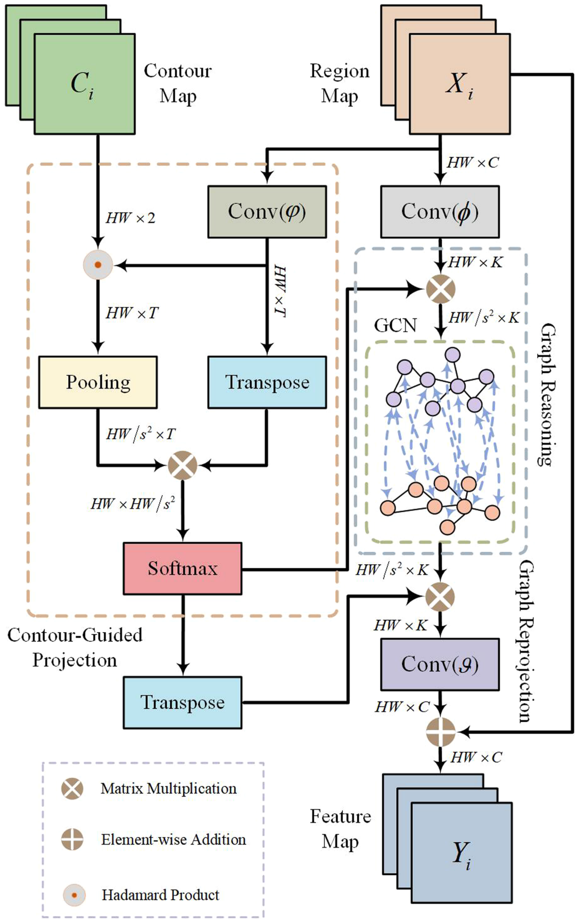

Fig. 7. Detailed structure of the CGRM. Take contour map and region map as inputs to generate high-quality feature maps.

where W_j^Q, W_j^K, and W_j^V denote the weights of the linear layer of the jth head to the maps Q, K, and V, respectively. h is the number of attention heads, W^O is the weight of the last linear layer in the MCA, and $Concat(\cdot)$ denotes stacking in column vectors.

F. Contour-Graph Reasoning Module

Since the initial features $X_i(i=1,2)$ of our obtained bitemporal images contain more semantic and global information, but lack image details, only the approximate area where the object has changed can be estimated. Therefore, we first use the Siamese FFM to fuse the low-level semantic feature information in each temporal and use the rich texture, edge, and other features in it to generate rough initial contour prediction maps $C_i(i=1,2)$. The details of the FFM are shown in Fig. 2(b). Then, we feed feature maps $X_i(i=1,2)$ with more semantic information and initial contour maps $C_i(i=1,2)$ containing detail information in each temporal to the Siamese CGRM [20]. The aim is to learn the intrinsic graph representation to capture the semantic relationships between regions and contours. The overall composition architecture of the CGRM is shown in Fig. 7. It can be seen that the CGRM block consists of three main parts: contour-guided graph projection, graph reasoning, and graph reprojection. In this way, the CGRM can map the original feature map to vertices guided by contour maps and interpret the relationships between vertices in the graph. Then, the learned graph representation is reprojected to the pixel space of the original feature map, thus reducing the boundary error, enhancing the feature representation, and obtaining a more accurate feature map.

1) Contour-Guided Graph Projection: According to Fig. 7, we use the initial contour maps $C_i \in R^{HW\times 2}(i=1,2)$ and the region feature maps $X_i \in R^{HW\times C}(i=1,2)$ as the input of the CGRM block, where H and W are the height and width of the original image, respectively, and C is the number of channels. In this section, the most important content is to construct the projection matrices $P_i(i=1,2)$ by mapping X_i to the vertices of a graph with C_i as *a priori*. Therefore, we first downscale the region feature maps X_i using 1×1 convolution layer to obtain the new features $H_i \in R^{HW\times T}(T<C, i=1,2)$. To facilitate the computation, we make the dimensionality of the contour maps C_i consistent with that of H_i. Then, H_i and C_i perform Hadamard product operation to fuse the contour information into the projection. Our purpose of using Hadamard product is to assign weights to the features of contour pixels, so that the pixel information in the contour feature map can be made to have a larger weight. Next, the anchor points of the vertices are obtained by averaging the pooling layers with span s and size 6×6. Here, the anchor point we mentioned is the center of each pixel region. Subsequently, we transpose H_i and multiply the result of the transposition with the anchor points to capture the similarity between the anchor points and each pixel. Finally, we use the softmax function to normalize the result of the phase multiplication to obtain the projection matrices $P_i \in R^{HW/s^2\times HW}(i=1,2)$. It can be expressed as

$$H_i = \varphi(X_i) \tag{17}$$

$$P_i = soft\max\left(\sigma(H_i \odot C_i)(H_i)^T\right) \tag{18}$$

where $\varphi(\cdot)$ is the 1×1 convolution layer, $\sigma(\cdot)$ denotes average pooling operation, and $\odot$ represents the Hadamard product. In (18), the contour attention and the pooling operation are the two key steps. The contour attention emphasizes the feature information of the contour by assigning greater weights to the contour pixels. The pooling operation eliminates redundant information and obtains a compact feature representation.

After we obtain the projection matrices P_i, we project the region feature maps X_i onto the image domain as follows:

$$J_i = P_i\phi(X_i) \tag{19}$$

where $\phi(\cdot)$ denotes the 1×1 convolution operation. The projection operation in (19) allows aggregating pixels that share similar features at each node, i.e., subregions in the image can be represented intrinsically by vertices. Therefore, we associate similar pixels with relevant regions by introducing contour-guided graph projection to obtain the features of projected vertices in graph $J_i \in R^{HW/s^2\times K}(i=1,2)$.

2) Graph Reasoning: After we get the vertex features, we need to further learn the connectivity between vertices in J_i, i.e., the relationship between region features and contour features.

Furthermore, these relations can be reasoned about during the propagation of information between vertices, so that higher level semantic information can be learned. Based on the above considerations, we use a single-layer GCN to implement it. In essence, the GCN is a first-order local approximation of spectral graph convolution. As shown in Fig. 7, we feed the vertex features in J_i into the GCN to obtain the graph representations $\widehat{J}_i \in R^{HW/s^2 \times K} (i = 1, 2)$. It can be expressed as

$$\widehat{J}_i = \delta \left[(I - A) P_i \phi (X_i) W \right] \tag{20}$$

where I denotes the identity matrix, A denotes the adjacency matrix of the learning graph connection, W is the weight of the GCN, and $\delta(\cdot)$ denotes the ReLU activation function. Note that, similar to [63], A is randomly initialized and learned from the vertex features.

3) Graph Reprojection: We need to reproject the learned graph representations $\widehat{J}_i$ into the original pixel space in order to obtain the final feature maps $Y_i \in R^{HW \times C} (i = 1, 2)$. According to Fig. 7, we first transpose the projection matrices P_i. However, we consider that P_i are not square matrix and the calculation steps are more complicated. Therefore, we use the reprojection matrices [64] as the transpose matrices of P_i. Then, $\widehat{J}_i$ are multiplied with $P_i{}^T$, i.e., the graph representations are reprojected to the pixel grid. Immediately after, we use the convolution operations of 1×1 to add feature channels that are consistent with the region feature maps X_i. Finally, we add the original region feature maps with the refinement features obtained after reprojection, and the resulting sums are used as the output pixel-level feature maps Y_i. It can be expressed as

$$Y_i = X_i + \vartheta \left((P_i)^T \widehat{J}_i \right) \tag{21}$$

where $\vartheta(\cdot)$ denotes the 1×1 convolutional layer.

G. Network Details

As shown in Fig. 2(a), we perform pixel subtraction operations on the feature maps $Z_i (i = 1, 2)$ output by the TD and the feature maps $Y_i (i = 1, 2)$ output by the CGRM to obtain two feature difference maps: $Z, Y \in R^{H_0 \times W_0 \times C}$ (H_0 and W_0 are the height and width of the original image, respectively). Then, Z and Y are connected together to obtain the feature map $M \in R^{H_0 \times W_0 \times C}$. In addition, a PM is added to make better use of the extracted high-level semantic features. The model uses a very shallow FCN for change discrimination to generate the predicted change result $M^* \in R^{H_0 \times W_0 \times 2}$. The formula is expressed as follows:

$$M^* = \sigma \left(g \left(|Z_1 - Z_2| + |Y_1 - Y_2| \right) \right) \tag{22}$$

where $g(\cdot) : R^{H_0 \times W_0 \times C} \Rightarrow R^{H_0 \times W_0 \times 2}$ is the change classifier, and $\sigma(\cdot)$ denotes a pixelated softmax operation on the channel dimension of the classifier output. Our change classifier mainly consists of two 3×3 convolutional layers and batch normalization.

For the selection of the loss function, we use the minimized cross-entropy loss to optimize the network parameters. The loss function is defined as follows:

$$L = \frac{1}{H_0 \times W_0} \sum_{h=1, w=1}^{H, W} l \left(P_{hw}, Y_{hw} \right) \tag{23}$$

where $l(P_{hw}, y) = -\log(P_{hwy})$ is the cross-entropy loss, and Y_{hw} denotes the label of the pixel at position (h, w).

IV. Experiments

In this section, we detail the process of verifying the proposed method on multiple change detection datasets and proving its effectiveness and rationality. First, we introduce the three experimental datasets used, namely, the CDD dataset [65], the LEVIR-CD dataset [9], and the WHU dataset [5]. Then, the implementation details of the experiment are described, which include the experimental settings, evaluation metrics, and comparative methods. Immediately after, we analyze the experimental results qualitatively and quantitatively on each of the three public datasets. In addition, we design ablation experiments to verify the rationality of the network structure and the function of each proposed module. Finally, we visualize the feature maps of the network model in several main stages.

A. Datasets

1) CDD Dataset: The CDD dataset was proposed by Lebedev et al. [65] in 2018. The dataset has 11 pairs of multispectral images, including seven pairs of images of different seasons with a size of 4725×2200 pixels and four pairs of images with a size of 1900×1000 pixels. The spatial resolution of these images ranges from 3 to 100 cm per pixel. Since the size of the image pairs is too large for direct processing, the authors crop these 11 pairs into image pairs of 256×256 pixels in size without overlapping areas. A total of 16 000 pairs of bitemporal remote sensing images are generated, of which the numbers of training dataset, validation dataset, and test dataset are 10 000, 3000, and 3000, respectively. Some examples in the CDD dataset are shown in Fig. 8(1)–(3).

2) LEVIR-CD Dataset: The LEVIR-CD dataset is a building change detection dataset proposed by Chen and Shi [9] in 2020. The dataset consists of 637 pairs of images, of which the default numbers of training, validation, and test datasets are 445, 64, and 128, respectively. The spatial resolution is 0.5 m/pixel, and the image size is 1024×1024 pixels. Considering the training memory consumption problem, we crop each image in the dataset into 16 subblocks of 256×256 pixels size without overlapping areas according to the segmentation criterion in [9]. Thus, we obtain 7120, 1024, and 2048 pairs of images for training, validation, and testing, respectively. Some examples in the LEVIR-CD dataset are shown in Fig. 8(4)–(6).

3) WHU Dataset: The WHU dataset [5] is a building change detection dataset proposed by Wuhan University. The dataset was collected from two aerial remote sensing RGB images of the Christchurch, New Zealand area in 2012 and 2016, respectively, with a size of $32\,507 \times 15\,354$ pixels and a spatial resolution of 0.075 m/pixel. Considering that the original image is too large, direct use for network training will result in insufficient video

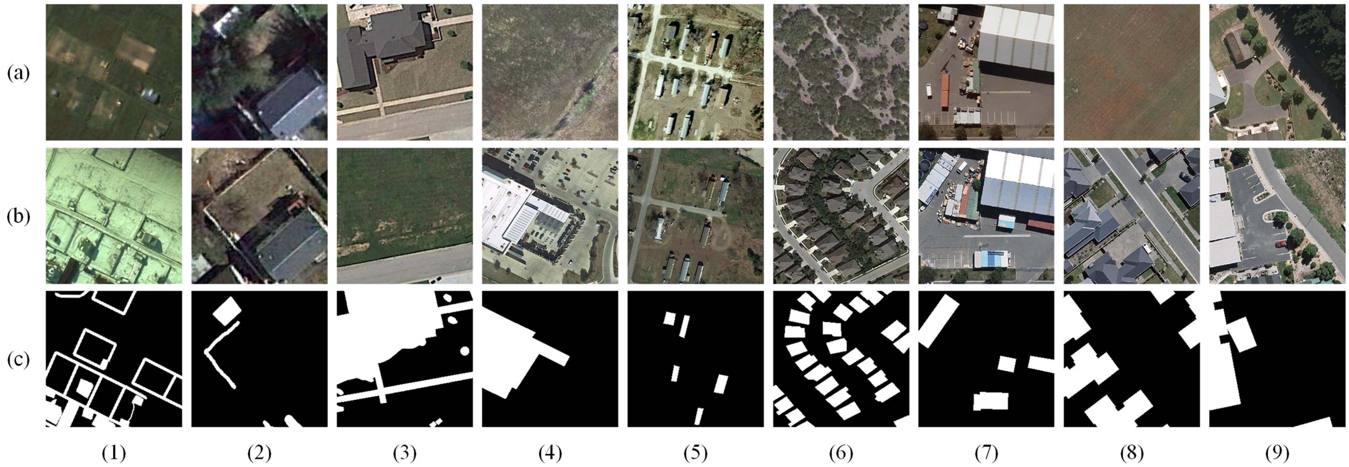

Fig. 8. Examples of CDD dataset (1)–(3), LEVIR-CD dataset (4)–(6), and WHU dataset (7)–(9). (a) Remote sensing image at the moment of T1. (b) Remote sensing image at the moment of T2. (c) Ground truth.

memory. We crop the two images into pairs of 256 × 256 pixel size without overlapping areas. Then, we randomly divide the cropped images into three parts 6096, 762, and 762 in the ratio of 8:1:1, which are used as the training dataset, test dataset, and validation dataset, respectively. Some examples in the WHU dataset are shown in Fig. 8(7)–(9).

B. Implementation Details

1) Experimental Settings: Our experiment is based on the Ubuntu 18.04 operating system, and the deep learning framework adopted is Pytorch, written in python. The training, validation, and test of the model are carried out on a server with multiple NVIDIA RTX 2080Ti graphics cards.

In the training process, the model uses stochastic gradient descent as the optimization algorithm, setting the parameter momentum = 0.9 and weight decay = 0.0005. The initial learning rate is set to 0.01 for all three training datasets, and 200 epochs are trained for each dataset. Considering the limitation of memory capacity, the batch size of all datasets is set to 8.

2) Evaluation Metrics: The purpose of change detection is to determine the changed pixels and the unchanged pixels. In essence, it can be classified as a binary classification problem. We use evaluation metrics include overall accuracy (OA), intersection over union (IoU), precision (P), recall (R), and F1-score (F1), which can reflect the performance of the proposed method in several dimensions. They are expressed as follows:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{24}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{25}$$

$$\mathrm{F1} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}} \tag{26}$$

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{27}$$

$$\mathrm{OA} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{28}$$

where TP (true positive) indicates the number of pixels where the change is correctly detected, TN (true negative) indicates that the number of pixels with no change is correctly detected, FN (false negative) indicates the number of pixels that actually changed but no change was detected, and FP (false positive) indicates the number of pixels that did not actually change but were detected to have changed.

Among the above metrics, precision (P), recall (R), and OA are the most commonly used metrics in binary partitioning problems. F1 is a comprehensive measurement metric, which performs weighted average reconciliation on precision and recall. IoU indicates the degree of overlap between the pixel area predicted to have change and the real change pixel area. Therefore, F1 and IoU can better reflect the comprehensive performance of the model.

3) Comparative Methods: To verify the effectiveness and superiority of our network model, we selected nine recent deep-learning-based change detection methods as comparative methods. These include FC-EF [66], FC-Siam-conc [66], FC-Siam-diff [66], SNUNet [67], DTCDSCN [11], IFN [12], RDP-Net [68], BIT [18], and ChangeFormer [69]. The following is a brief introduction to each method.

FC-EF [66] extracts multiscale features through the U-net structure that connects bitemporal images together as the input to the network. FC-Siam-conc [66] is an extension of FC-EF. Superposition is performed in the channel dimension as a jump connection in the U-net structure. FC-Siam-diff [66] is another extension of FC-EF. The absolute value of the difference value is used as a jump connection in the U-net structure. SNUNet [67] uses the Siamese UNet++ network [40] as a feature extraction tool and uses the integrated channel attention module to refine the features at different levels. DTCDSCN [11] contains two encoding branches and one decoding branch and adds a dual-attention module in the decoder part to extract more context features. In addition, to ensure the fairness of the experimental results, we did not use a semantic segmentation decoder. IFN [12] uses the attention module to fuse the extracted multilevel deep features with image difference features

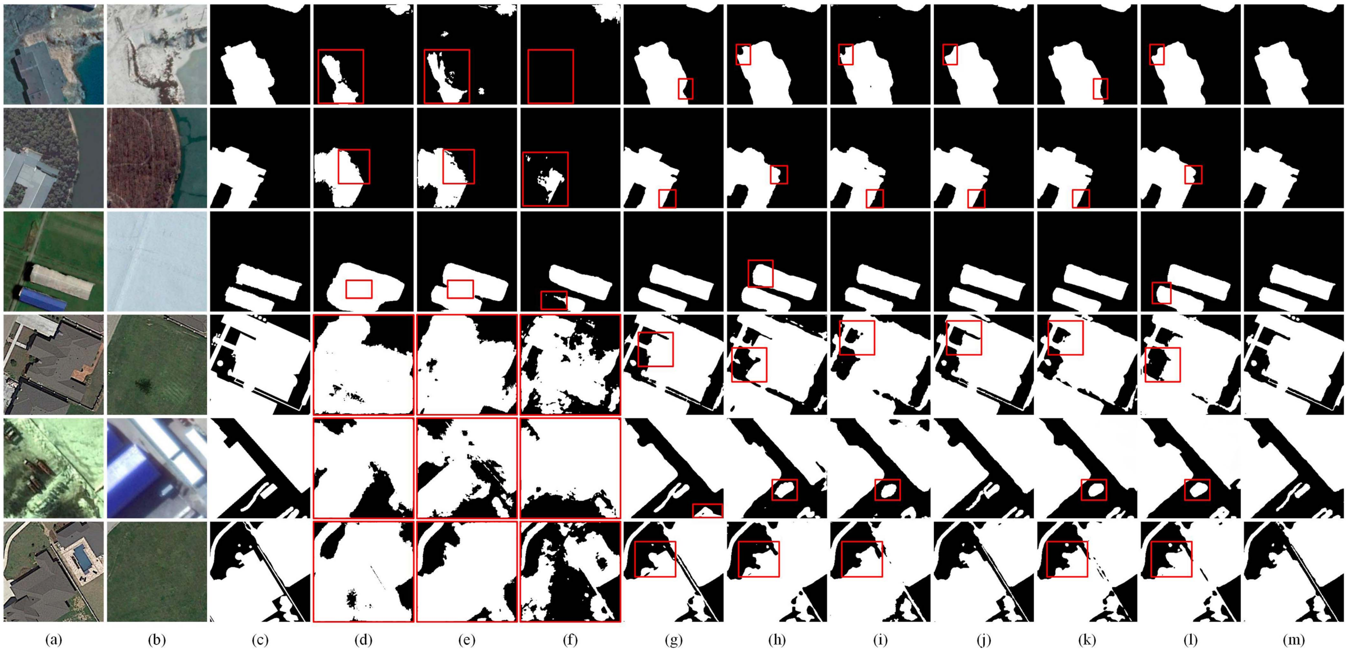

Fig. 9. Change detection of buildings in the CDD dataset by different methods and qualitative comparison of the results. (a) and (b) Remote sensing images before and after the change, respectively. (c) Ground truth. (d) FC-Siam-diff. (e) FC-Siam-conc. (f) FC-EF. (g) SNUNet. (h) DTCDSCN. (i) IFN. (j) RDPNet. (k) BIT. (l) ChangeFormer. (m) Ours. In the change map, white pixels indicate actual changes, and black pixels indicate no changes.

and uses deep supervision to train the middle layer of the network. RDPNet [68] performs nonuniform sampling based on the importance of individual samples and uses an efficient edge loss to focus on the detailed information of the boundary. BIT [18] is a transformer-based approach that integrates Siamese tokenizer and transformer encoder–decoder structures into a change detection network, thus capturing richer context information in the spatial–temporal domain. ChangeFormer [69] is a transformer-based Siamese network. It unifies a transformer encoder with a hierarchical structure and an MLP decoder so that multiscale long-range detail information can be efficiently acquired.

C. Results and Analysis

We compare the nine change detection methods mentioned above with our method on the CDD, LEVIR-CD, and WHU datasets, respectively, and analyze the results qualitatively and quantitatively. In addition, to ensure the accuracy and fairness of the results, we use the same training and validation datasets to train the network models for each method, and the same test datasets for testing. The evaluation results for each method on each dataset are shown as follows.

1) Evaluation Results on the CDD Dataset: Since the CDD dataset contains change areas at different scales and a wide variety of change types, we select three typical scenes of buildings, roads, and land for change detection, respectively. Figs. 9–11 show the change detection results of our method and other advanced methods for different scenes in the CDD dataset. Although there are obvious geographical differences between bitemporal image pairs, which are also susceptible to interference from sensors, sunlight angles, and seasonal variations, our method can effectively filter out these irrelevant factors and obtain complete and accurate areas of change.

Fig. 9 shows the change detection results for various building scenes. As can be seen from the figure, for the first three rows of small- and medium-sized buildings, most of the methods can identify the changed areas. For the changes of large buildings in the fourth to sixth rows, the change results of FC-EF and DTCDSCN are more fragmented. Although other methods can obtain the complete change area, some detailed information is lost, resulting in blurred edges of the change area. Compared with other methods, our method not only has better ability to retain detail information at the edges of large-scale change areas, but also has more complete detection of small scale change areas (rows 2 and 6 in Fig. 9).

Fig. 10 shows the change detection results of roads. As can be seen from the figure, FC-Siam-diff, FC-Siam-conc, and FC-EF show poor results and can only detect the more obvious roads (rows 5 and 6 in Fig. 10). The change results of DTCDSCN, IFN, and ChangeFormer show a large number of fragments although roads are detected (row 4 in Fig. 10). Some of the roads in the change results of SNUNet, RDPNet, and BIT are incorrectly detected or not detected (row 3 in Fig. 10). In comparison, our method can identify more detailed change information, thus generating high-quality change maps with clear and continuous boundaries. In addition, the interference of pseudo change is overcome with better robustness (rows 5 and 6 in Fig. 10).

Fig. 11 shows the change detection results of land. As shown in the figure, for the first four rows of small- and medium-sized land changes, the results obtained by the FC-Siam-diff, FC-Siam-conc, and FC-EF methods are less satisfactory in terms of visual performance, and only simple areas of change can

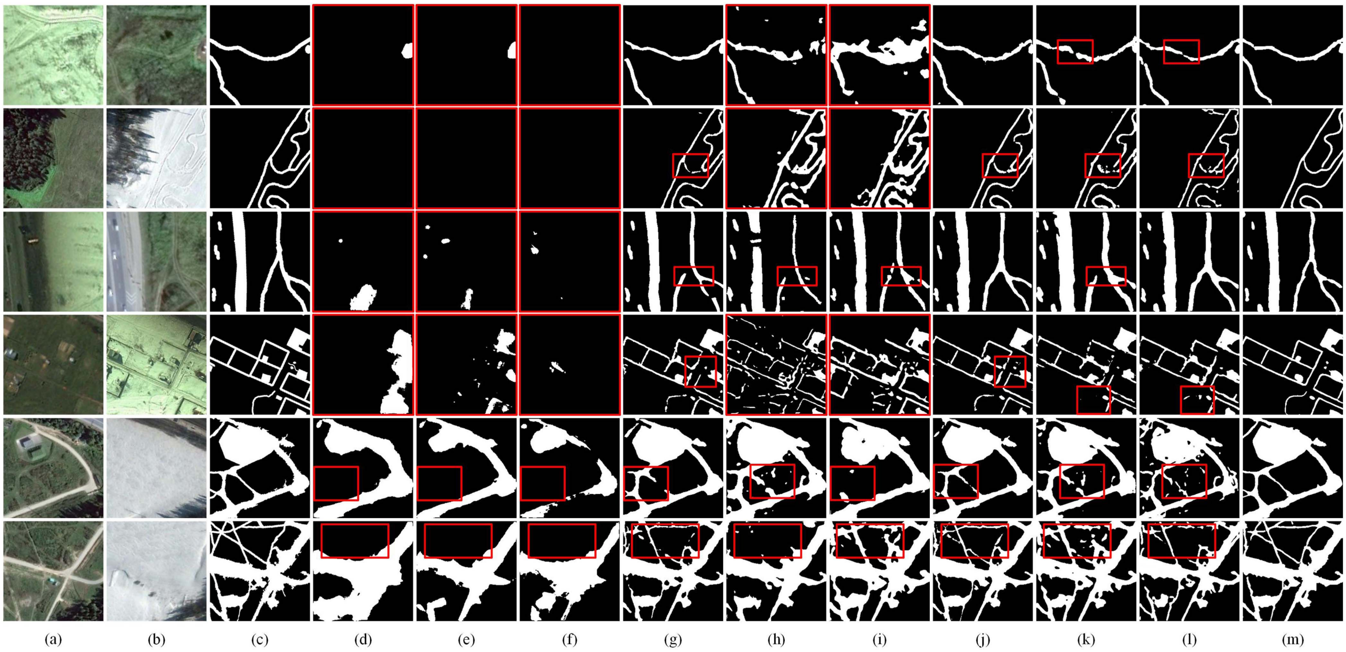

Fig. 10. Change detection of roads in the CDD dataset by different methods and qualitative comparison of the results. (a) and (b) Remote sensing images before and after the change, respectively. (c) Ground truth. (d) FC-Siam-diff. (e) FC-Siam-conc. (f) FC-EF. (g) SNUNet. (h) DTCDSCN. (i) IFN. (j) RDPNet. (k) BIT. (l) ChangeFormer. (m) Ours. In the change map, white pixels indicate actual changes, and black pixels indicate no changes.

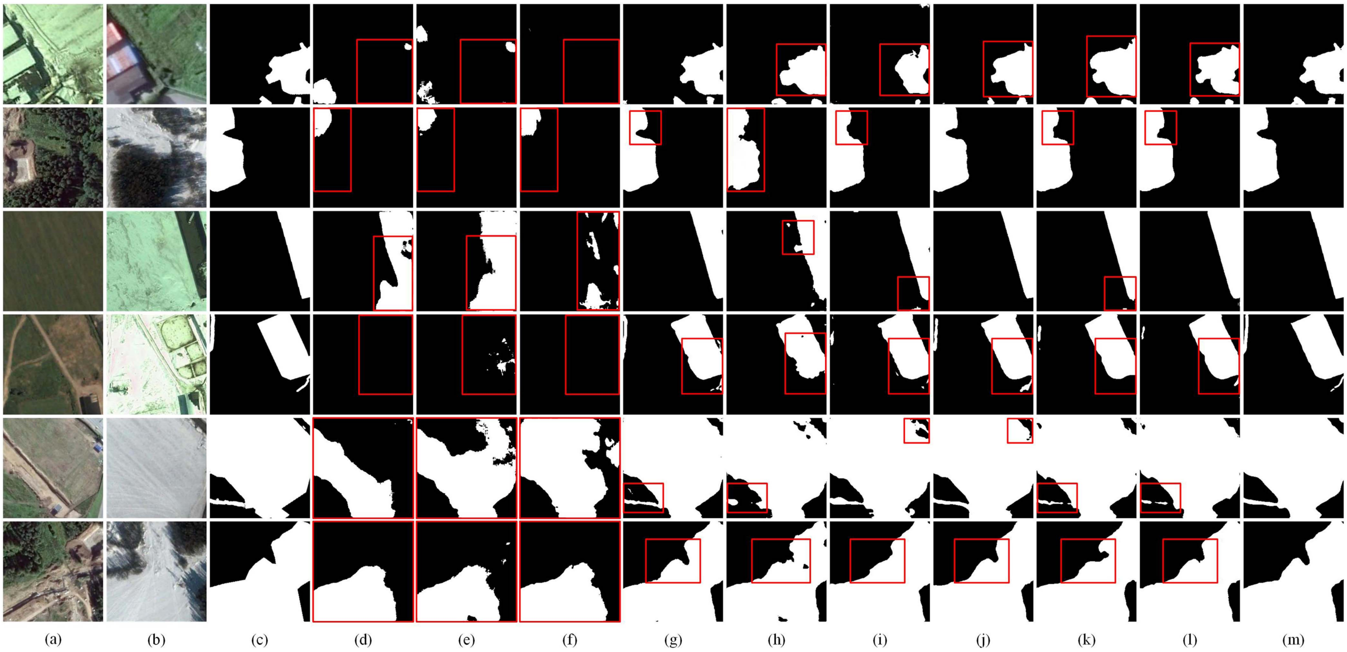

Fig. 11. Change detection of land in the CDD dataset by different methods and qualitative comparison of the results. (a) and (b) Remote sensing images before and after the change, respectively. (c) Ground truth. (d) FC-Siam-diff. (e) FC-Siam-conc. (f) FC-EF. (g) SNUNet. (h) DTCDSCN. (i) IFN. (j) RDPNet. (k) BIT. (l) ChangeFormer. (m) Ours. In the change map, white pixels indicate actual changes, and black pixels indicate no changes.

be identified. The remaining methods are better at identifying the areas where changes occur. For the large and complex land changes in the fifth and sixth rows, although these methods can extract the changed area, our method detects the changed area of the land more accurately. In addition, our approach performs better in maintaining internal integrity and recovering details (rows 4 and 5 in Fig. 11). Therefore, after the above qualitative analysis, it is clear that the change results obtained by our method on the CDD dataset outperform other advanced methods in terms of accuracy and predicted shape, achieving better visual performance.

To further demonstrate the effectiveness and superiority of our method, we also perform a quantitative evaluation on the CDD dataset. The quantitative results of various methods are

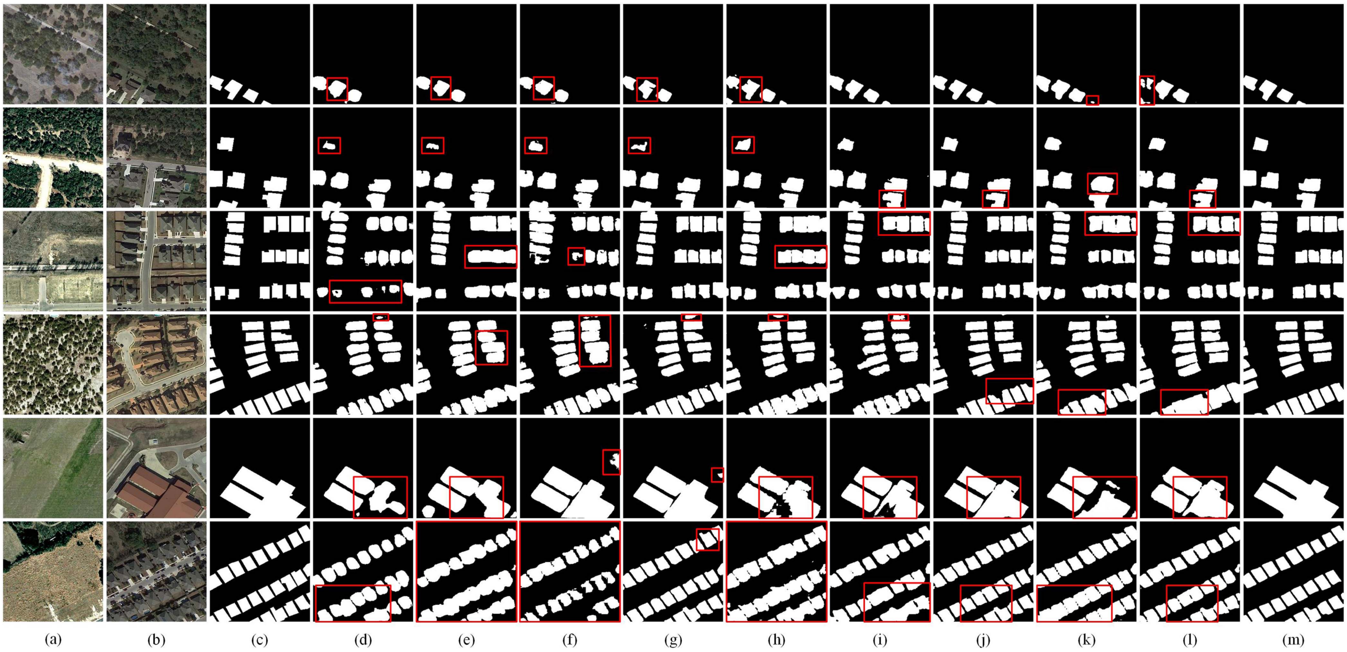

Fig. 12. Change detection by different methods on the LEVIR-CD dataset and qualitative comparison of the results. (a) and (b) Remote sensing images before and after the change, respectively. (c) Ground truth. (d) FC-Siam-diff. (e) FC-Siam-conc. (f) FC-EF. (g) SNUNet. (h) DTCDSCN. (i) IFN. (j) RDPNet. (k) BIT. (l) ChangeFormer. (m) Ours. In the change map, white pixels indicate actual changes, and black pixels indicate no changes.

TABLE II
AVERAGE QUANTITATIVE RESULTS OF DIFFERENT METHODS ON THE CDD DATASET

Method	P (%)	R (%)	F1 (%)	IoU (%)	OA (%)
FC-Siam-diff	78.25	65.76	69.04	56.57	94.44
FC-Siam-conc	74.51	73.87	71.16	60.11	94.92
FC-EF	65.60	55.01	57.65	52.20	93.58
SNUNet	**96.13**	91.52	93.88	87.11	97.36
DTCDSCN	89.74	85.24	87.09	80.24	96.56
IFN	91.30	89.08	89.13	85.34	96.70
RDPNet	94.76	92.79	94.54	90.09	97.84
BIT	93.82	89.70	90.08	86.12	97.07
ChangeFormer	94.40	92.48	93.94	89.80	97.74
Ours	95.98	**93.41**	**95.68**	**92.57**	**98.73**

Values are described using percentages, and the highest scores are highlighted in bold.

shown in Table II. As can be seen from the table, our method is superior to all comparative methods in most evaluation metrics, with recall, F1, IoU, and OA reaching 93.41%, 95.68%, 92.57%, and 98.73%, respectively. Compared with other methods, our method improves at least 1.14% and 2.48% in F1 and IoU metrics, respectively.

2) Evaluation Results on the LEVIR-CD Dataset: To further validate that the proposed method has good generalization performance in change detection tasks, we also perform effective evaluation on the LEVIR-CD dataset. It is worth noting that this dataset is a building change detection dataset. It focuses only on building changes and ignores other types of changes.

Fig. 12 shows the change detection results of our method and other advanced methods in the LEVIR-CD dataset. A qualitative analysis of Fig. 12 shows that our method predicts the correct areas of change for both scattered or dense small building change areas and large building change areas. As can be seen from the first two rows of the figure, for scattered small buildings, although all the algorithms can detect the change areas, our method yields more accurate results with less noise. As can be seen from the third, fourth, and sixth rows of the figure, most algorithms do not detect all of the change parts due to the dense distribution of buildings, irregular changes in shape, and the interference of more pseudo changes (e.g., road and tree changes). Even though SNUNet, RDPNet, and ChangeFormer methods are able to detect all the change areas, there are still cases of false detections and missed detections. Our method can be more focused on building changes and can exclude the interference of other factors, resulting in finer change detection results. In addition, as shown in the fifth row of the figure, for large buildings, most of the algorithms obtain change results that are not clear enough in terms of shape and contour, with large missed detection areas. Compared with other methods, our method can better learn the characteristics of buildings, so that the generated results can maintain accurate edge information and complete change areas.

Table III shows the quantitative results of various methods on the LEVIR-CD dataset. As can be seen from the table, our method achieves the best performance in several evaluation metrics. Specifically, our method has the best results in precision, F1, IoU, and OA, reaching 92.51%, 91.59%, 83.50%, and 99.05%, respectively. Compared with the second-ranked RDPNet, our method achieves performance improvements of 1.25% and 0.71% in the F1 and IoU metrics, respectively.

3) Evaluation Results on the WHU Dataset: Finally, we also conduct experiments on the WHU dataset. Similar to the

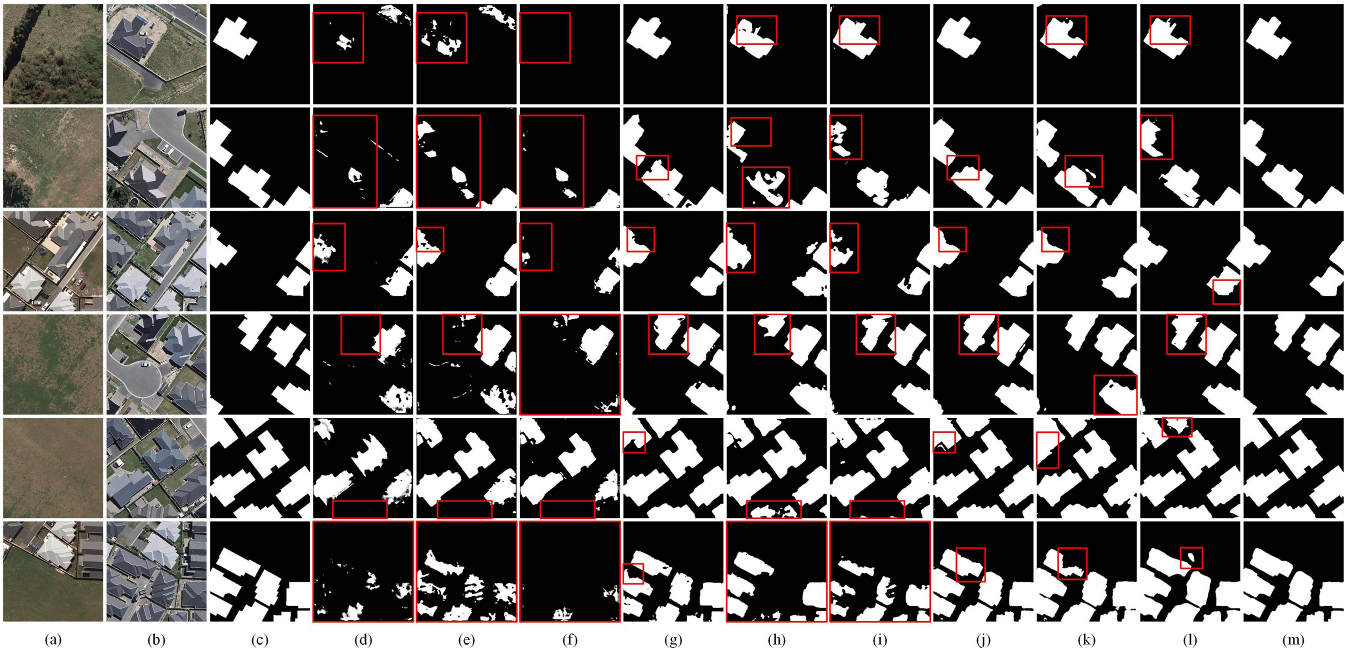

Fig. 13. Change detection by different methods on the WHU dataset and qualitative comparison of the results. (a) and (b) Remote sensing images before and after the change, respectively. (c) Ground truth. (d) FC-Siam-diff. (e) FC-Siam-conc. (f) FC-EF. (g) SNUNet. (h) DTCDSCN. (i) IFN. (j) RDPNet. (k) BIT. (l) ChangeFormer. (m) Ours. In the change map, white pixels indicate actual changes, and black pixels indicate no changes.

TABLE III
AVERAGE QUANTITATIVE RESULTS OF DIFFERENT METHODS ON THE LEVIR-CD DATASET

Method	P (%)	R (%)	F1 (%)	IoU (%)	OA (%)
FC-Siam-diff	82.35	78.71	86.58	75.63	98.12
FC-Siam-conc	89.65	84.85	83.32	72.46	97.59
FC-EF	86.37	83.54	82.35	71.80	96.97
SNUNet	91.78	88.17	89.34	81.67	98.87
DTCDSCN	86.54	88.06	87.81	78.61	98.45
IFN	89.73	86.69	88.40	79.57	98.58
RDPNet	92.05	**89.93**	90.34	82.79	98.96
BIT	89.21	89.19	89.23	80.57	98.79
ChangeFormer	91.54	89.11	90.10	82.32	98.92
Ours	**92.51**	89.57	**91.59**	**83.50**	**99.05**

Values are described using percentages, and the highest scores are highlighted in bold.

LEVIR-CD dataset, the WHU dataset is also a building change detection dataset that contains only the change areas of buildings.

Fig. 13 shows the change detection results of our method and other advanced methods in the WHU dataset. It can be seen that our method achieves the best visual effect among all the methods for building changes of different quantities and scales. In detail, in the first three rows of Fig. 13, the number of change pixels is relatively small in the whole image and unevenly distributed, and the shapes of the buildings vary in size. The change detection results obtained by FC-Siam-diff, FC-Siam-conc, and FC-EF methods are noisy and contain some unnecessary information. Other advanced comparative methods are able to detect areas of change. In contrast, our method generates the most accurate change maps. In addition, when the number of change pixels accounts for most of the whole image, that is, the fourth to sixth rows in the figure, most methods correctly mark the location of building changes. For densely distributed building change areas, DTCDSCN, IFN, and BIT methods are able to correctly mark the change locations, but there are more missed detections at the edges of the change areas. The change building area predicted by SNUNet, RDPNet, and ChangeFormer methods suffers from the problem that the contour segmentation is not accurate enough. Compared with other methods, our method not only correctly marks the location of building changes, but also accurately delineates the boundaries of buildings, enabling adequate representation of building information.

The quantitative results of the different methods on the WHU dataset are shown in Table IV. Combining the data in the table for quantitative analysis of the five evaluation metrics, it can be obtained that the SNUNet and BIT methods achieve the best performance in the precision and OA metrics, respectively. However, our method achieves optimal results in all three metrics of recall, F1, and IoU, reaching 86.32%, 85.91%, and 75.58%, respectively. Compared with other methods, our method improves F1 and IoU by at least 1.25% and 1.46%, respectively.

D. Model Efficiency Analysis

In order to further demonstrate the performance of the proposed network model, in addition to the conventional visual and data comparison of the change detection results of the network, we also used three evaluation indexes, namely, FLOPs, Params, and Time, to analyze the efficiency of the network model, in which FLOPs indicate the number of floating point operations required to run the network model once. The time complexity

TABLE IV
AVERAGE QUANTITATIVE RESULTS OF DIFFERENT METHODS ON THE WHU DATASET

Method	P (%)	R (%)	F1 (%)	IoU (%)	OA (%)
FC-Siam-diff	75.46	69.50	72.01	56.38	94.18
FC-Siam-conc	78.19	80.65	76.02	63.13	96.33
FC-EF	77.26	75.44	75.27	61.55	94.37
SNUNet	**89.67**	83.58	83.54	73.20	98.82
DTCDSCN	79.33	81.01	78.21	67.26	96.59
IFN	83.11	82.13	81.37	71.77	97.36
RDPNet	88.75	84.25	84.66	74.12	98.94
BIT	88.51	84.03	83.17	72.84	**99.01**
ChangeFormer	87.20	82.64	82.49	71.90	97.98
Ours	89.45	**86.32**	**85.91**	**75.58**	98.95

Values are described using percentages, and the highest scores are highlighted in bold.

TABLE V
COMPARISON OF THE EFFICIENCY OF DIFFERENT METHODS ON THE LEVIR-CD DATASET

Method	FLOPs (G)	Params (M)	Time (ms)
FC-Siam-diff	4.72	1.35	0.004
FC-Siam-conc	5.32	1.55	0.725
FC-EF	3.57	1.35	0.675
SNUNet	54.82	12.03	4.609
DTCDSCN	13.21	31.26	8.624
IFN	82.26	50.44	6.488
RDPNet	27.12	1.69	0.835
BIT	10.59	3.49	4.598
ChangeFormer	21.18	29.73	2.319
Ours	12.80	5.62	4.254

(length of time) of the model is measured by calculating the sum of the number of multiplication operations and the number of addition operations. Params indicates the number of parameters of the network model, i.e., the number of parameters to be learned during the training process, corresponding to the spatial complexity of the model (the size of the occupied video memory). Time indicates the time cost required by the network model to process a single image.

The FLOPs, Params, and Time of our method and other methods are recorded in Table V. Among them, the smaller the FLOPs and Params, the less the complexity of the network model. The smaller the Time, the lower the time cost of the network model. As can be seen from Table V, compared with other network models, the network we proposed is in the middle in terms of FLOPs, Params, and time cost. However, combined with the previous objective analysis, it is known that our method achieves the best change detection performance on different change detection datasets. Thus, our method can achieve a good balance in terms of model complexity, time cost, and accuracy. This also reflects the feasibility of our method in change detection tasks.

TABLE VI
CONTROL OF THE COMPONENT MODULES OF THE NETWORK STRUCTURE IN THE ABLATION EXPERIMENT

Variable	FVT	PS-ViT	TD	FFM+CGRM
Experiment 1		✓	✓	✓
Experiment 2	✓		✓	✓
Experiment 3	✓	✓		✓
Experiment 4	✓	✓	✓	
Ours	✓	✓	✓	✓

TABLE VII
AVERAGE QUANTITATIVE RESULTS OF THE DIFFERENT ABLATION EXPERIMENTS ON THE LEVIR-CD DATASET

Model	P (%)	R (%)	F1 (%)	IoU (%)	OA (%)
No FVT	90.86	89.19	89.55	81.68	98.51
No PS-ViT	87.62	89.48	87.90	79.55	98.23
No TD	90.71	87.30	88.89	81.22	98.49
No FFM+CGRM	89.35	88.17	89.31	80.63	98.37
Ours	**92.51**	**90.87**	**91.59**	**83.50**	**99.05**

Values are described using percentages, and the highest scores are highlighted in bold.

TABLE VIII
AVERAGE QUANTITATIVE RESULTS OF THE DIFFERENT ABLATION EXPERIMENTS ON THE CDD DATASET

Model	P (%)	R (%)	F1 (%)	IoU (%)	OA (%)
No FVT	93.51	92.87	93.81	90.76	98.43
No PS-ViT	91.12	91.40	91.23	87.14	98.32
No TD	92.56	89.88	92.19	89.31	98.29
No FFM+CGRM	92.97	91.73	93.52	90.27	98.52
Ours	**95.98**	**93.41**	**95.68**	**92.57**	**98.73**

Values are described using percentages, and the highest scores are highlighted in bold.

E. Ablation Study

In our investigation, we conduct ablation experiments on each component of the network model in order to demonstrate the design rationality and effectiveness of the proposed method. Under the premise that the experimental environment, parameter settings, and picture size are all uniform, we adopt the single-variable control method and conduct four groups of ablation experiments separately. The variables controlled for each group of experiments are shown in Table VI. In addition, in these four groups of experiments, we use two datasets (LEVIR-CD dataset and CDD dataset) for training and testing. Tables VII and VIII show the quantitative results of each set of ablation experiments.

In Experiment 1, we remove the FVT from the original network to ablate it. Since the function of our FVT is to represent feature maps with compact semantic tokens, the tokens contain rich semantic information. As can be seen from the first row of data in Tables VII and VIII, the network model without FVT has lower values than the full network model for all evaluation metrics. For our main evaluation metrics F1 and IoU, the two

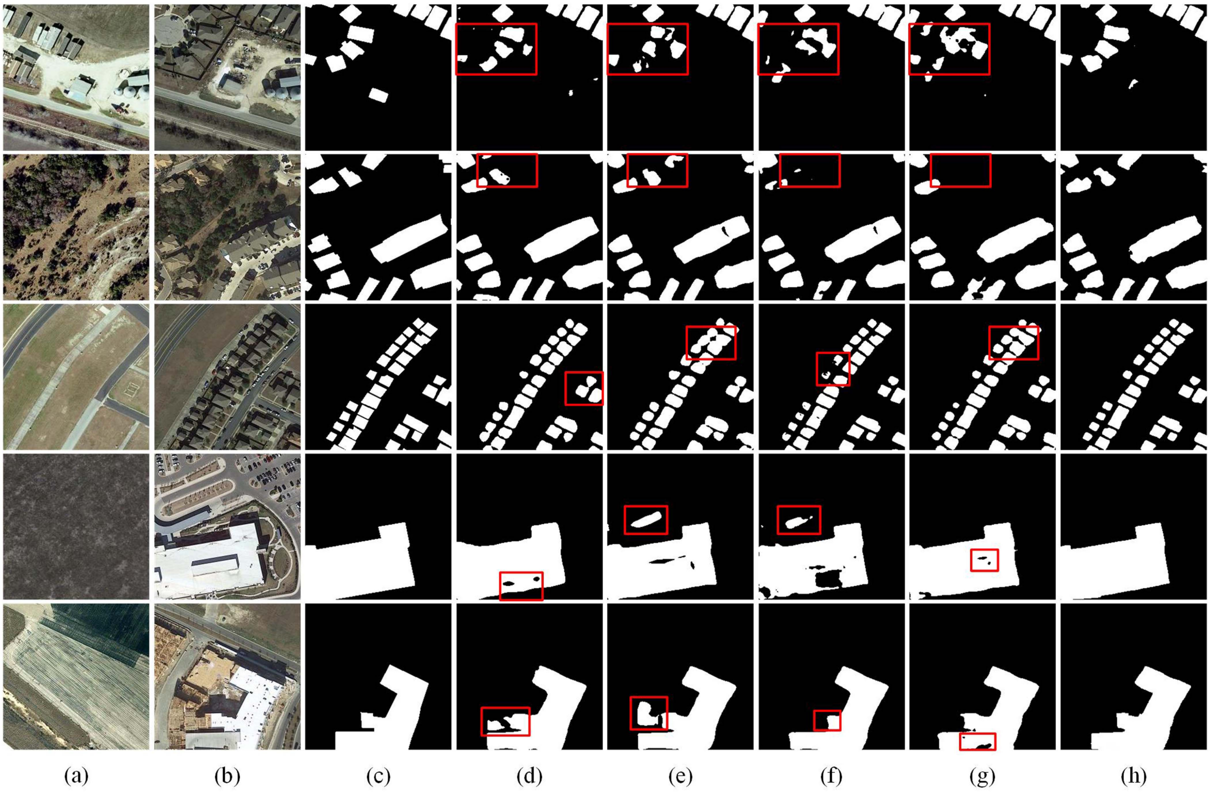

Fig. 14. Change detection results of different ablation experiments on the LEVIR-CD dataset. (a) and (b) Remote sensing images before and after the change, respectively. (c) Ground truth. (d) No FVT. (e) No PS-ViT. (f) No TD. (g) No FFM+CGRM. (h) Ours. In the change map, white pixels indicate actual changes, and black pixels indicate no changes.

datasets decreased by 2.04%/1.82% and 1.87%/1.81%, respectively. This proves the rationality of the design of FVT.

In Experiment 2, to demonstrate the effectiveness of the PS-ViT in the network model, we ablate it. Since PS-ViT uses a progressive iterative approach to locate the area of change, the effect of irrelevant changes is effectively avoided. The lack of PS-ViT module results in the network not being able to fully extract the spatiotemporal information in the token, which affects the modeling of the context. Thus, as shown in the second row of Tables VII and VIII, the network model without PS-ViT is lower than our method in all the metrics. For our main evaluation metrics F1 and IoU, the two datasets decreased by 3.69%/4.45% and 3.95%/5.43%, respectively. This indicates the essential role of PS-ViT for the overall performance improvement of the model.

In Experiment 3, we ablate the TD to verify its performance in the network. Specifically, we use a simple module in the original network instead of the TD to be able to fuse the tokens from the PS-ViT with the original features from the feature extractor. As can be observed in the third row of Tables VII and VIII, the network model that does not include TD causes some degree of degradation in all the metrics. This is because the TD uses cross-attention blocks to project high-level semantic information into pixel space and generate finer feature maps. For our main evaluation metrics F1 and IoU, the two datasets decreased by 2.70%/2.28% and 3.49%/3.26%, respectively. This shows that TD is an essential part of our network model.

In Experiment 4, we regard the FFM and the CGRM as a whole model and perform an ablation experiment to verify their effects on the experimental results. The CGRM can combine the semantic information and the detailed information extracted by the FFM to fully capture the semantic relationships between regions and contours. It also improves the accuracy of the network in processing edge information. Therefore, according to the data provided in the fourth row of Tables VII and VIII, it can be seen that the network without FFM and CGRM is degraded in all the metrics compared to the complete network model. For our main evaluation metrics F1 and IoU, the two datasets decreased by 2.28%/2.87% and 2.16%/2.30%, respectively. This shows that the FFM and the CGRM together can effectively improve the change detection performance of the network.

In addition, we qualitatively compare the change detection results of different ablation experiments. Figs. 14 and 15 show the predicted results of each group of ablation experiments in the LEVIR-CD dataset and the CDD dataset, respectively. As can be seen from the two images, the change detection results of the "No FVT" model contain some unneeded information and misjudge some changes in the building (e.g., row 5 of Fig. 14 and row 1 of Fig. 15). The change detection results of the "No PS-ViT" model have more pseudo changes and poor segmentation of the edges of the change area (e.g., row 4 of Fig. 14 and row 2 of Fig. 15). The change detection results of the "No TD" model show a small amount of fragmentation, and not all of the changes are detected, and there are some cases of missing detection (e.g., row

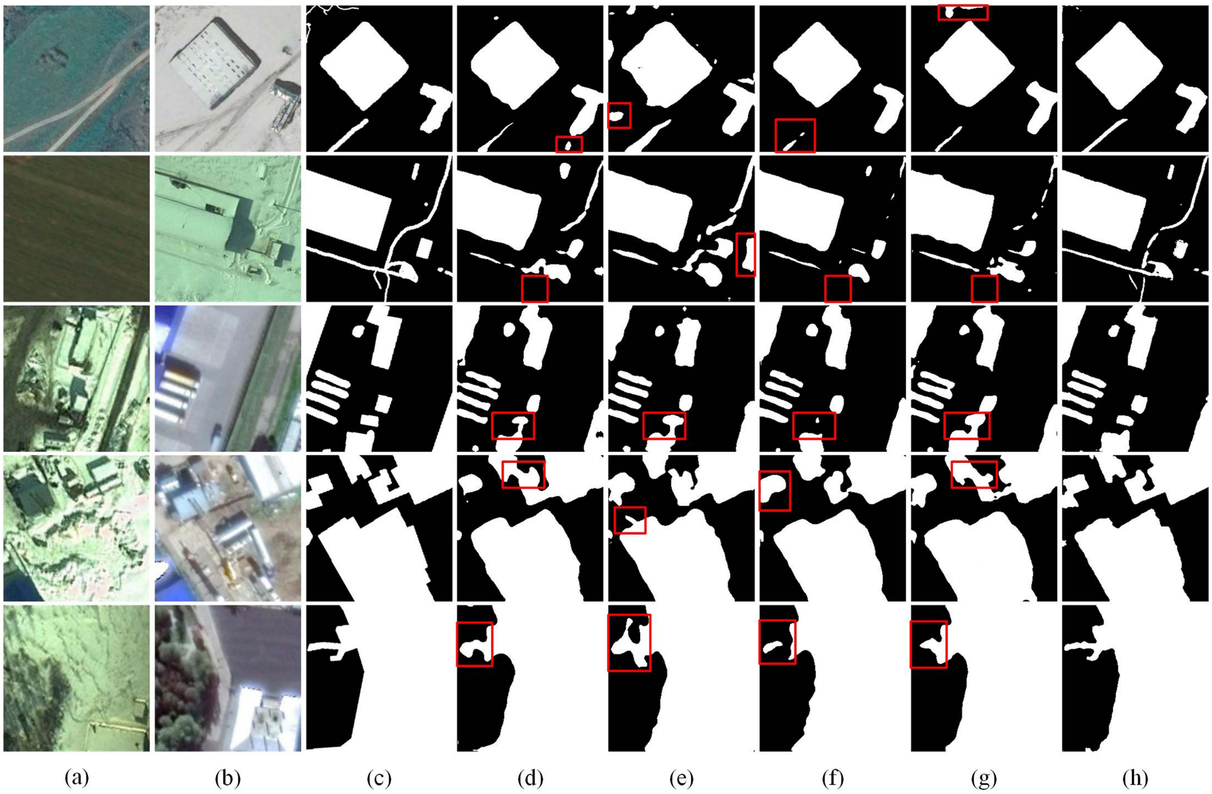

Fig. 15. Change detection results of different ablation experiments on the CDD dataset. (a) and (b) Remote sensing images before and after the change, respectively. (c) Ground truth. (d) No FVT. (e) No PS-ViT. (f) No TD. (g) No FFM+CGRM. (h) Ours. In the change map, white pixels indicate actual changes, and black pixels indicate no changes.

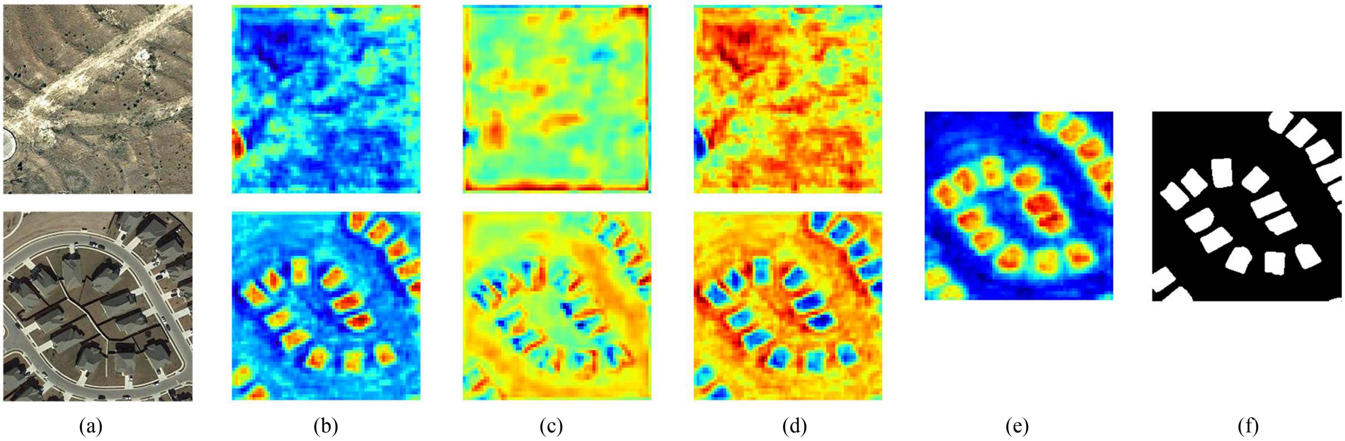

Fig. 16. Example of visualizing the main modules in a network. (a) Input bitemporal remote sensing image. (b) Deep features extracted by the Siamese backbone network (feature extractor). (c) Feature map obtained after CGRM processing. (d) Refined feature map obtained after TD processing. (e) Bitemporal feature difference map. (f) Predicted change map. The sample is from the LEVIR-CD dataset.

2 of Fig. 14 and row 2 of Fig. 15). The change detection results of the "No FFM+CGRM" model have more false detections. Meanwhile, the refinement ability of object edges in the real change area is weak, and a clear object boundary cannot be segmented (e.g., row 1 of Fig. 14 and row 3 of Fig. 15). In contrast, our approach avoids the various problems that arise in the ablation model mentioned above. The overall effect of the generated change map is the best, with advantages in both boundary extraction and false detection.

In conclusion, the qualitative and quantitative analysis of the results of the ablation experiments illustrates the usefulness of our method for change detection tasks. Meanwhile, it reflects the dependence between each module and the rationality and effectiveness of the complete model design.

F. Network Visualization

To further understand our network structure, we visualize the feature maps of several major network modules processed in the TCIANet model, as shown in Fig. 16. Given the original bitemporal remote sensing image [see Fig. 16(a)], the Siamese backbone network (feature extractor) first extracts the high-level feature mapping [see Fig. 16(b)]. Then, the low-level feature information extracted from the first three residual layers in the feature extractor and the semantic information extracted from the last residual layer are passed through the CGRM to generate a new feature mapping [see Fig. 16(c)]. Immediately afterward, we feed the context-rich tokens generated by the PS-ViT to the TD for processing. The TD projects these semantic tokens into the pixel space to obtain a refined feature map [see Fig. 16(d)]. Finally, we calculate the feature difference map [see Fig. 16(e)] and generate the predicted change map [see Fig. 16(f)] by using the PM.

V. Conclusion

In this article, we proposed a TCIANet for remote sensing image change detection. In the specific network architecture, we used the FVT to convert the pixel information in the feature map into compact visual semantic tokens. And the high-level concept, i.e., the change region existing in the bitemporal image, was represented by token sets. Next, the PS-ViT was introduced to reduce the damage of tokenization on the image structure by adopting a progressive iterative sampling strategy to locate the discriminative regions. Moreover, the transformer encoder was used to encode the global context relationships of these token sets in order to obtain abstract semantic information of feature images and improve local perception. Then, we used the TD to project these high-level semantic concepts into pixel space and refined them, thus making the acquired pixel-level features more robust and can better reveal changes in the objects of interest. In addition, the combination of the FFM and the CGRM enhanced the extraction of texture and detail information and improved the network's ability to process edge information. At the same time, the semantic relationship between regions and contours can be further obtained to highlight the contribution of important pixels, thus reducing the boundary error. Finally, through experimental analysis on three different datasets, it was demonstrated that our method outperforms other advanced methods both in terms of visual performance and quantitative evaluation.

In addition, in the future research work, we will further improve and optimize the performance of the network and apply it to multiple types of change detection tasks.

References

[1] T. Lei, Y. Zhang, Z. Lv, S. Li, S. Liu, and A. K. Nandi, "Landslide inventory mapping from bitemporal images using deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 6, pp. 982–986, Jun. 2019.

[2] P. Lu, Y. Qin, Z. Li, A. C. Mondini, and N. Casagli, "Landslide mapping from multi-sensor data through improved change detection-based Markov random field," *Remote Sens. Environ.*, vol. 231, 2019, Art. no. 111235.

[3] M. Hussain, D. Chen, A. Cheng, H. Wei, and D. Stanley, "Change detection from remotely sensed images: From pixel-based to object-based approaches," *ISPRS J. Photogrammetry Remote Sens.*, vol. 80, pp. 91–106, 2013.

[4] N. Zerrouki, F. Harrou, Y. Sun, and L. Hocini, "A machine learning-based approach for land cover change detection using remote sensing and radiometric measurements," *IEEE Sens. J.*, vol. 19, no. 14, pp. 5843–5850, Jul. 2019.

[5] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 574–586, Jan. 2019.

[6] L.-C. Chen and L.-J. Lin, "Detection of building changes from aerial images and light detection and ranging (LIDAR) data," *J. Appl. Remote Sens.*, vol. 4, no. 1, 2010, Art. no. 041870.

[7] N. Bourdis, D. Marraud, and H. Sahbi, "Constrained optical flow for aerial image change detection," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2011, pp. 4176–4179.

[8] C. Benedek and T. Szirányi, "Change detection in optical aerial images by a multilayer conditional mixed Markov model," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 10, pp. 3416–3430, Oct. 2009.

[9] H. Chen and Z. Shi, "A spatial-temporal attention-based method and a new dataset for remote sensing image change detection," *Remote Sens.*, vol. 12, no. 10, 2020, Art. no. 1662.

[10] P. P. De Bem, O. A. de Carvalho Junior, R. F. Guimarães, and R. A. T. Gomes, "Change detection of deforestation in the brazilian amazon using landsat data and convolutional neural networks," *Remote Sens.*, vol. 12, no. 6, 2020, Art. no. 901.

[11] Y. Liu, C. Pang, Z. Zhan, X. Zhang, and X. Yang, "Building change detection for remote sensing images using a dual-task constrained deep Siamese convolutional network model," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 5, pp. 811–815, May 2021.

[12] C. Zhang et al., "A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images," *ISPRS J. Photogrammetry Remote Sens.*, vol. 166, pp. 183–200, 2020.

[13] D. Peng, Y. Zhang, and H. Guan, "End-to-end change detection for high resolution satellite images using improved UNet," *Remote Sens.*, vol. 11, no. 11, 2019, Art. no. 1382.

[14] Y. Wang et al., "Mask DeepLab: End-to-end image segmentation for change detection in high-resolution remote sensing images," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 104, 2021, Art. no. 102582.

[15] M. Zhang, G. Xu, K. Chen, M. Yan, and X. Sun, "Triplet-based semantic relation learning for aerial remote sensing image change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 266–270, Feb. 2019.

[16] D. Zhang, H. Zhang, J. Tang, M. Wang, X. Hua, and Q. Sun, "Feature pyramid transformer," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 323–339.

[17] J. Xu, C. Luo, X. Chen, S. Wei, and Y. Luo, "Remote sensing change detection based on multidirectional adaptive feature fusion and perceptual similarity," *Remote Sens.*, vol. 13, no. 15, 2021, Art. no. 3053.

[18] H. Chen, Z. Qi, and Z. Shi, "Remote sensing image change detection with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5607514.

[19] X. Yue et al., "Vision transformer with progressive sampling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 387–396.

[20] K. Wang, X. Zhang, Y. Lu, X. Zhang, and W. Zhang, "CGRNet: Contour-guided graph reasoning network for ambiguous biomedical image segmentation," *Biomed. Signal Process. Control*, vol. 75, 2022, Art. no. 103621.

[21] L. Bruzzone and D. F. Prieto, "Automatic analysis of the difference image for unsupervised change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1171–1182, May 2000.

[22] J. Deng, K. Wang, Y. Deng, and G. Qi, "PCA-based land-use change detection and analysis using multitemporal and multisensor satellite data," *Int. J. Remote Sens.*, vol. 29, no. 16, pp. 4823–4838, 2008.

[23] A. A. Nielsen, K. Conradsen, and J. J. Simpson, "Multivariate alteration detection (MAD) and MAF postprocessing in multispectral, bitemporal image data: New approaches to change detection studies," *Remote Sens. Environ.*, vol. 64, no. 1, pp. 1–19, 1998.

[24] C. Wu, B. Du, and L. Zhang, "Slow feature analysis for change detection in multispectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2858–2874, May 2014.

[25] Y. Qin, Z. Niu, F. Chen, B. Li, and Y. Ban, "Object-based land cover change detection for cross-sensor images," *Int. J. Remote Sens.*, vol. 34, no. 19, pp. 6723–6737, 2013.

[26] W. Feng, H. Sui, J. Tu, W. Huang, and K. Sun, "A novel change detection approach based on visual saliency and random forest from multi-temporal high-resolution remote-sensing images," *Int. J. remote Sens.*, vol. 39, no. 22, pp. 7998–8021, 2018.

[27] C. Zhang, G. Li, and W. Cui, "High-resolution remote sensing image change detection by statistical-object-based method," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 7, pp. 2440–2447, Jul. 2018.

[28] C. Huo, Z. Zhou, H. Lu, C. Pan, and K. Chen, "Fast object-level change detection for VHR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 1, pp. 118–122, Jan. 2010.

[29] J. Chen, Z. Mao, B. Philpot, J. Li, and D. Pan, "Detecting changes in high-resolution satellite coastal imagery using an image object detection approach," *Int. J. Remote Sens.*, vol. 34, no. 7, pp. 2454–2469, 2013.

[30] B. Ma and C.-Y. Chang, "Semantic segmentation of high-resolution remote sensing images using multiscale skip connection network," *IEEE Sens. J.*, vol. 22, no. 4, pp. 3745–3755, Feb. 2022.

[31] L. Zhao, Y. Zhang, and Y. Cui, "An attention encoder-decoder network based on generative adversarial network for remote sensing image dehazing," *IEEE Sens. J.*, vol. 22, no. 11, pp. 10890–10900, Jun. 2022.

[32] L. Sun, S. Cheng, Y. Zheng, Z. Wu, and J. Zhang, "SPANet: Successive pooling attention network for semantic segmentation of remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4045–4057, 2022.

[33] X. Peng, R. Zhong, Z. Li, and Q. Li, "Optical remote sensing image change detection based on attention mechanism and image difference," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7296–7307, Sep. 2021.

[34] R. Liu, D. Jiang, L. Zhang, and Z. Zhang, "Deep depthwise separable convolutional network for change detection in optical aerial images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1109–1118, 2020.

[35] Z. Zheng, Y. Wan, Y. Zhang, S. Xiang, D. Peng, and B. Zhang, "CLNet: Cross-layer convolutional neural network for change detection in optical remote sensing imagery," *ISPRS J. Photogrammetry Remote Sens.*, vol. 175, pp. 247–267, 2021.

[36] Y. Zhang, L. Fu, Y. Li, and Y. Zhang, "HDFNet: Hierarchical dynamic fusion network for change detection in optical aerial images," *Remote Sens.*, vol. 13, no. 8, 2021, Art. no. 1440.

[37] B. Hou, Q. Liu, H. Wang, and Y. Wang, "From W-Net to CDGAN: Bitemporal change detection via deep learning techniques," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 1790–1802, Mar. 2020.

[38] L. Zhang, X. Hu, M. Zhang, Z. Shu, and H. Zhou, "Object-level change detection with a dual correlation attention-guided detector," *ISPRS J. Photogrammetry Remote Sens.*, vol. 177, pp. 147–160, 2021.

[39] F. I. Diakogiannis, F. Waldner, and P. Caccetta, "Looking for change? Roll the dice and demand attention," *Remote Sens.*, vol. 13, no. 18, 2021, Art. no. 3707.

[40] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. New York, NY, USA: Springer, 2018, pp. 3–11.

[41] J. Chen et al., "DASNet: Dual attentive fully convolutional Siamese networks for change detection in high-resolution satellite images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1194–1206, 2020.

[42] H. Jiang, X. Hu, K. Li, J. Zhang, J. Gong, and M. Zhang, "PGA-SiamNet: Pyramid feature-based attention-guided Siamese network for remote sensing orthoimagery building change detection," *Remote Sens.*, vol. 12, no. 3, 2020, Art. no. 484.

[43] M. Liu, Q. Shi, A. Marinoni, D. He, X. Liu, and L. Zhang, "Super-resolution-based change detection network with stacked attention module for images with different resolutions," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 440371.

[44] A. Vaswani et al., "Attention is all you need," *in Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[45] T. Brown et al., "Language models are few-shot learners," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 1877–1901.

[46] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[47] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning texture transformer network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5791–5800.

[48] H. Chen et al., "Pre-trained image processing transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12299–12310.

[49] S. Zheng et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6881–6890.

[50] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[51] J. He, L. Zhao, H. Yang, M. Zhang, and W. Li, "HSI-BERT: Hyperspectral image classification using the bidirectional encoder representation from transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 165–178, Jan. 2020.

[52] L. Sun, G. Zhao, Y. Zheng, and Z. Wu, "Spectral–spatial feature tokenization transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5522214.

[53] X. Shen, B. Liu, Y. Zhou, and J. Zhao, "Remote sensing image caption generation via transformer and reinforcement learning," *Multimedia Tools Appl.*, vol. 79, no. 35, pp. 26661–26682, 2020.

[54] Q. Wang, W. Huang, X. Zhang, and X. Li, "Word–sentence framework for remote sensing image captioning," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 12, pp. 10532–10543, Dec. 2021.

[55] Y. Bazi, L. Bashmal, M. M. A. Rahhal, R. A. Dayil, and N. A. Ajlan, "Vision transformers for remote sensing image classification," *Remote Sens.*, vol. 13, no. 3, 2021, Art. no. 516.

[56] Y. Feng, H. Xu, J. Jiang, H. Liu, and J. Zheng, "ICIF-Net: Intra-scale cross-interaction and inter-scale feature fusion network for bitemporal remote sensing images change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4410213.

[57] Q. Liu, L. Xiao, J. Yang, and Z. Wei, "CNN-enhanced graph convolutional network with pixel-and superpixel-level feature fusion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8657–8671, Oct. 2021.

[58] X. Zhang, X. Tan, G. Chen, K. Zhu, P. Liao, and T. Wang, "Object-based classification framework of remote sensing images with graph convolutional networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2021, Art. no. 8010905.

[59] X. Tang et al., "An unsupervised remote sensing change detection method based on multiscale graph convolutional network and metric learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5609715.

[60] J. Qu, Y. Xu, W. Dong, Y. Li, and Q. Du, "Dual-branch difference amplification graph convolutional network for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5519912.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[62] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021.

[63] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 433–442.

[64] Y. Li and A. Gupta, "Beyond grids: Learning graph representations for visual recognition," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9245–9255.

[65] M. Lebedev, Y. V. Vizilter, O. Vygolov, V. Knyaz, and A. Y. Rubis, "Change detection in remote sensing images using conditional adversarial networks," *Int. Arch. Photogrammetry Remote Sens. Spatial Inf. Sci.*, vol. 42, no. 2, pp. 565–571, 2018.

[66] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional Siamese networks for change detection," in *Proc. 25th IEEE Int. Conf. Image Process.*, 2018, pp. 4063–4067.

[67] S. Fang, K. Li, J. Shao, and Z. Li, "SNUNET-CD: A densely connected Siamese network for change detection of VHR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2021, Art. no. 8007805.

[68] H. Chen, F. Pu, R. Yang, R. Tang, and X. Xu, "RDP-Net: Region detail preserving network for change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5635010.

[69] W. G. C. Bandara and V. M. Patel, "A transformer-based Siamese network for change detection," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2022, pp. 207–210.

Xintao Xu received the B.S. degree in computer science and technology from the School of Computer Science and Technology, Shandong Women's University, Jinan, China, in 2020. He is currently working toward the M.S. degree with the School of Information and Electronic Engineering, Shandong Technology and Business University, Yantai, China.

His research interests include computer graphics, computer vision, and image processing.

Jinjiang Li received the B.S. and M.S. degrees from the Taiyuan University of Technology, Taiyuan, China, in 2001 and 2004, respectively, and the Ph.D. degree from Shandong University, Jinan, China, in 2010, all in computer science.

From 2004 to 2006, he was an Assistant Research Fellow with the Institute of Computer Science and Technology, Peking University, Beijing, China. From 2012 to 2014, he was a Postdoctoral Fellow with Tsinghua University, Beijing. He is currently a Professor with the School of Computer Science and Technology, Shandong Technology and Business University, Yantai, China. His research interests include image processing, computer graphics, computer vision, and machine learning.

Zheng Chen received the B.S. and M.S. degrees in computer science and technology from Shandong Agricultural University, Taian, Shandong, China and Shandong Normal University, Jinan, Shandong, China, in 2012 and 2015, respectively, and the Ph.D. degree in signal and information processing from Dalian University of Technology, Dalian, China, in 2022.

He is currently a Lecturer with Shandong Technology and Business University. His research interests include computer vision, hand pose estimation and hand shape recovery.