



Article

DTT-CGINet: A Dual Temporal Transformer Network with Multi-Scale Contour-Guided Graph Interaction for Change Detection

Ming Chen , Wanshou Jiang * and Yuan Zhou

State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; 2448431638qwe@gmail.com (M.C.); zhoyuan@whu.edu.cn (Y.Z.)

* Correspondence: jws@whu.edu.cn

Abstract: Deep learning has dramatically enhanced remote sensing change detection. However, existing neural network models often face challenges like false positives and missed detections due to factors like lighting changes, scale differences, and noise interruptions. Additionally, change detection results often fail to capture target contours accurately. To address these issues, we propose a novel transformer-based hybrid network. In this study, we analyze the structural relationship in bi-temporal images and introduce a cross-attention-based transformer to model this relationship. First, we use a tokenizer to express the high-level features of the bi-temporal image into several semantic tokens. Then, we use a dual temporal transformer (DTT) encoder to capture dense spatiotemporal contextual relationships among the tokens. The features extracted at the coarse scale are refined into finer details through the DTT decoder. Concurrently, we input the backbone's low-level features into a contour-guided graph interaction module (CGIM) that utilizes joint attention to capture semantic relationships between object regions and the contour. Then, we use the feature pyramid decoder to integrate the multi-scale outputs of the CGIM. The convolutional block attention modules (CBAMs) employ channel and spatial attention to reweight feature maps. Finally, the classifier discriminates change pixels and generates the final change map of the difference feature map. Several experiments have demonstrated that our model shows significant advantages over other methods in terms of efficiency, accuracy, and visual effects.



Citation: Chen, M.; Jiang, W.; Zhou, Y. DTT-CGINet: A Dual Temporal Transformer Network with Multi-Scale Contour-Guided Graph Interaction for Change Detection. *Remote Sens.* **2024**, *16*, 844. <https://doi.org/10.3390/rs16050844>

Academic Editor: Andrea Garzelli

Received: 17 December 2023

Revised: 19 February 2024

Accepted: 21 February 2024

Published: 28 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: change detection; transformer; attention; graph convolutional network (GCN); remote sensing

1. Introduction

Change detection (CD) is crucial in remote sensing image analysis. Its primary purpose is to identify differences between images of the same area captured at different times. These differences, or “changes”, refer to the appearance or disappearance of objects. CD methods aim to isolate relevant changes, such as monitoring changes in buildings, while filtering out irrelevant changes caused by factors like lighting, the season, and other areas of change not of interest, such as roads and rivers. With the rapid progress of remote sensing technology, change detection has become increasingly important in various fields, including geological disaster monitoring [1,2], land cover analysis [3], and urban planning [4]. High-precision change detection is essential in understanding remote sensing scenes, particularly in conserving natural resources. Therefore, there is an urgent need to develop an efficient and semi-automated method for detecting changes in remote sensing images.

In earlier research, many traditional algorithms such as linear predictors [5] and Cluster Kernel [6] were widely used for hyperspectral change detection, achieving excellent results. Ref. [7] provides a detailed introduction to some representative change detection algorithms based on multispectral (MS) and hyperspectral (HS) images. In recent years, with the booming development of artificial intelligence technology, remote sensing image

change detection based on deep learning has achieved remarkable results. The Convolutional Neural Network (CNN) can extract rich multi-scale spatiotemporal features from remote sensing images through its powerful feature extraction capability. Many CNN-based CD methods [8–10] have been proposed, providing better accuracy and efficiency compared to traditional methods. Peng et al. [11] used Unet-type networks with dense skip connections to fuse multi-scale semantic features. However, due to the limited receptive field of convolution in CNNs, it is difficult for these methods to model long-range contextual semantic relationships in images, making it hard to detect the complete boundary of large-scale changed areas. Some subsequent works use pyramid structures, attention mechanisms [12–14], and dilated convolutions [15] to expand the receptive field of convolutional operations. Despite numerous proposed improvements, the receptive field of convolutional neural networks still remains limited.

The transformer was first proposed in Natural Language Processing (NLP), triggering a new round of innovation in language processing technology. Inspired by this, ViT [16] pioneered the introduction of the transformer architecture to large-scale image recognition with great success. Researchers have progressively applied it to change detection tasks [17–20]. Self-attention is the core component of the transformer architecture, which explicitly models one-dimensional sequence relations. Specifically, it first defines three learnable weight matrices W^Q, W^K, W^V , then input X is first projected onto three weight matrices to obtain $Q = XW^Q, K = XW^K$, and $V = XW^V$. Finally, it computes weighted combinations of input elements for each element. However, these methods inadequately explore the attention mechanism between bi-temporal images in the CD task. The self-attention mechanism of [17–20] only models the non-local structural relations within a single temporal phase in Figure 1a and indiscriminately weights combinations of features in both changed and unchanged regions in the same way while ignoring the non-local structural relationships between the dual-temporal images in Figure 1b,c.

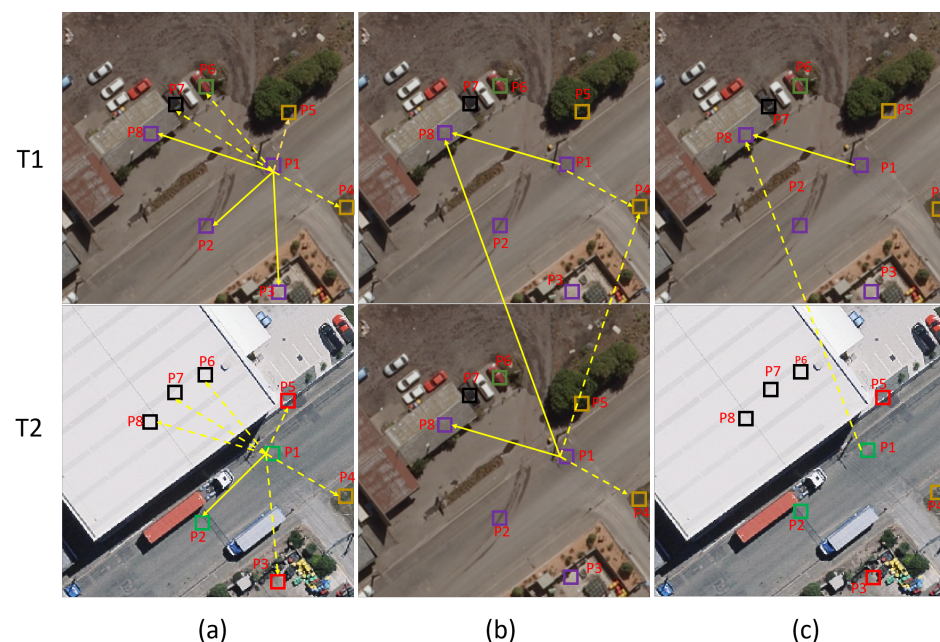


Figure 1. The non-local structural relationship of bi-temporal images. (a): Self-attention modeling the relationship in a single temporal image. (b,c): The non-local structural relationships between the dual-temporal images. The boxes of the same color represent two similar detection regions.

Figure 1 illustrates the non-local structural relationships within the images, with the first row representing the “pre-change” image and the second row representing the “post-change” image. In this context, each rectangular box represents a region within the image, which can also be understood as a token within the transformer framework. The

arrow lines indicate the relationships between two regions (tokens), where solid lines indicate similarity between the regions and dashed lines indicate dissimilarity. Figure 1a demonstrates the relationships between regions after modeling with the self-attention mechanism in a single temporal image. In the “pre-change image”, regions P1 and P2 show similarity, while P5 is dissimilar. Figure 1b shows that region P1 of the “pre-change image” remains unchanged in the “post-change image”. As a result, the similarity relationship between P1 and P8 in the “pre-change image” may be retained in the “post-change image”. The dissimilarity between region P1 and region P4 is preserved as well. Figure 1c shows that due to the change in region P1 in the “post-change image”, the similarity relationship with region P8 is lost. Effectively extracting non-local structural relationships between bi-temporal images will contribute to better modeling of spatiotemporal relationships and more efficient change detection.

We propose a dual temporal transformer (DTT) based on dual temporal attention to model the non-local structural relationships between dual-temporal images. Specifically, we first use Siamese ResNet to extract the multi-scale features of the dual temporal remote sensing images. Then, we use the tokenizer in BIT [17] to aggregate the high-level feature maps of each temporal phase into some tokens. Next, the DTT encoder models the non-local structural relationships between the tokens of the dual-temporal images. Immediately after, we use the DTT decoder to reweight the original features based on the generated tokens to obtain the refined features considering the dual-temporal contextual relationships. While some works, such as [21–23], employ transformers based on cross-attention for change detection, their proposed cross-attention merely involves the straightforward calculation of attention matrices using the query (Q) from another temporal phase and the key (K) from the current temporal phase. This approach fails to adequately model the non-local structural relationships depicted in Figure 1.

However, transformer requires expensive computational resources and is time-consuming; most existing transformer-based methods are similar to [17], only modeling the contextual relationship of high-level features and ineffectively fusing low-level features. While [20] introduced a transformer that incorporates multi-scale features, both the parameter count and computational cost significantly increased (several tens of times that of BIT [17]). Meanwhile, modeling only contextual information of the high-level features leads to blurred and incomplete boundaries of the change detection results, as shown in Figure 2 row (1). To address this, we use a contour extraction module (CEM) to extract the bi-temporal image’s contour, then use three improved CGRMs [24], named contour-guided graph interaction modules (CGIMs), to extract multi-scale semantic features further. Then, a simple feature pyramid decoder performs feature fusion on the multi-scale features output from the CGIMs. We also utilize CBAMs [25] to reweight the feature map based on the feature map’s channel attention and spatial attention. Finally, the difference feature maps obtained from the feature pyramid decoder and the DTT decoder are fused to predict the final change map. Meanwhile, the acquisition process of bi-temporal remote sensing images frequently introduces extraneous variations like illumination and seasonal changes, which can often result in false detections (Figure 2 row (2)). Therefore, we also explore the transfer learning of unsupervised remote sensing pre-training weights based on the seasonal comparison method [26] to the change detection task.

In summary, most previous methods still have shortcomings: (1) an inability to distinguish pseudo-changes like shadows, vegetation, and illumination in sensitive areas; (2) a lack of boundary information in complex remote sensing regions, resulting in extracted change areas being hollow and incomplete; and (3) insufficient exploration of spatiotemporal information in bi-temporal images. Our main contributions to this study can be summarized as follows.

- (1) We propose a novel dual temporal attention mechanism, considering non-local structural relationships between bi-temporal images and effectively modeling contextual semantic relationships.

- (2) We propose a contour extraction module (CEM) based on the Sobel convolutional block to effectively extract contours from remote sensing images. Additionally, we introduce the contour-guided graph reasoning module (CGRM), which utilizes contour maps to guide the generation of graph representations within contour-enclosed regions. To enhance CGRM's graph reasoning, we employ a joint attention mechanism to improve information propagation between graph vertices, ensuring the preservation of boundary integrity in change detection results.
- (3) Extensive experiments on three CD datasets demonstrate that our proposed method outperforms previous state-of-the-art methods in terms of accuracy and robustness.

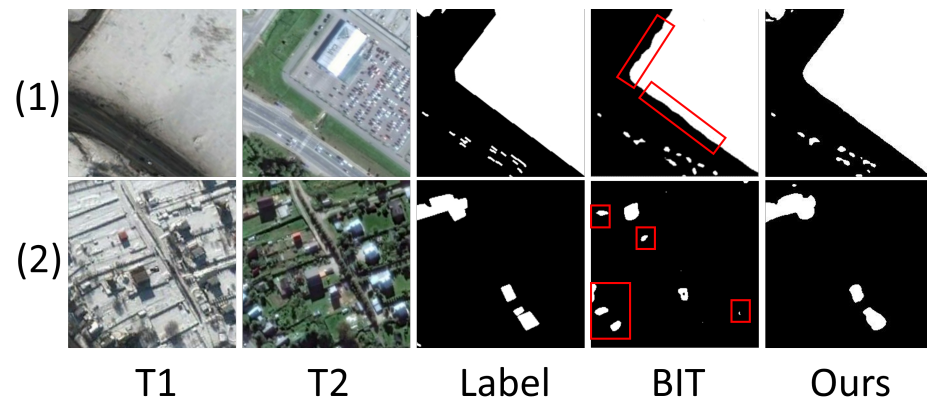


Figure 2. The first row illustrates the incomplete boundaries in the change detection results of existing transformers, such as BIT [17]. The second row demonstrates results with pseudo-changes caused by seasonal variations.

2. Related Work

Over the past few decades, change detection techniques have been increasingly developed and achieved great success. There are four main types of remote sensing change detection techniques: traditional methods, convolutional neural networks, transformers, and graph convolutional networks (GCN), respectively.

2.1. Traditional Methods

Early researchers used spectral information from remote sensing images to detect changes and proposed many traditional CD methods [27–29]. Change Vector Analysis (CVA) [28] calculates change vectors to determine the type and extent of surface changes by analyzing the spectral changes of image elements across different bands. Bourdais et al. [27] used Constrained Optical Flow Techniques for Aerial Image Change Detection, and Nielsen et al. [29,30] proposed a novel change detection method based on visual saliency and random forest, named Multivariate Change Detection (MAD). Principal component analysis (PCA) [31] was used to process remote sensing data from multiple time periods and different satellite sensors, reducing data dimensionality and extracting critical information. Slow feature analysis (SFA) [32] extracts temporal features from multi-temporal images while suppressing differences that reduce unchanged pixels. However, achieving suitable thresholds for varying scenes in the decision stage is time-consuming for most such methods. Therefore, many machine learning methods have been proposed to obtain automatic decision models. For instance, Lv et al. [33] proposed an unsupervised change detection method using a mixed-conditional random field model, analyzing spectral features in high spatial resolution remote sensing images. This approach eliminates the need for supervision during training. Support vector machines (SVM) [34] classify changing and pseudo-changing pixels in remote sensing images. Im et al. [35] proposed a Decision tree model for detecting changes, while [36] combined visual saliency and random forest techniques to improve the effectiveness of change detection. Lastly, Moser et al. [37]

proposed a multi-scale, unsupervised change detection method that combined Markov random fields and wavelet transforms to identify surface changes in optical images. However, these methods depend on hand-crafted features, which confine their performance and make it challenging to adjust information in real, intricate scenes.

2.2. CNN-Based Model

CNNs have been widely used in remote sensing due to their powerful local feature extraction capability [38,39]. Daudt et al. [40] presented three fully convolutional neural network architectures for change detection. However, their feature fusion strategy was too simple to meet the needs of multi-scale change objects. Therefore, [11,41,42] introduced denser skip connections for multi-scale feature fusion. Shi et al. [9] used deep supervision to supervise the feature maps in the lower layers of the backbone to train the model stably. Zhang et al. [43] proposed a dual-stream change detection network with a hierarchical fusion strategy. In addition, due to the limited receiving field of convolutional networks, many methods have been proposed to increase the size of the receiving field for convolutional operations, such as stacking deeper structures [41], dilated convolutional [15], and attention mechanisms [13,21,44]. Attention takes various forms: [21] uses self-attention to explain temporal and spatial correlations on semantic relations. Peng et al. [13] extract information from features' spatial and temporal dimensions, and Huang et al. [44] consider global and local channel information simultaneously to fuse features. However, CNN weakly models global dependencies and long-range contextual semantic relations. Therefore, we use CNN only as a backbone for feature extraction while using transformers and graph neural networks to model long-range contextual semantic relationships in images better.

2.3. Transformer

Vaswani et al. [45] first proposed the transformer, which was a massive success in Natural Language Processing (NLP) due to its ability to easily model remote dependencies of one-dimensional sequences, even wholly replacing convolution-based Recurrent Neural Networks (RNN) [46] and LSTM [47]. Chen et al. [16] (BIT) expressed a diachronic image as multiple tokens and used a transformer encoder to model the compact marker-based spatiotemporal space context. Liu et al. [19] and Song et al. [48] proposed a deep supervised network based on the swin transformer (MST) for change detection. Bandara et al. [49] proposed a hierarchically structured transformer to improve change detection performance further. However, almost all of the above transformers ignore the multiscale features extracted by the backbone. Liu et al. [19] proposed a method called MSCANet for detecting agricultural changes in high-resolution images. MSCANet adopts a CNN–transformer hybrid structure, combining the advantages of both CNN and transformer. It processes features extracted from each layer of the backbone using transformer, which impacts the network's parameter and speed. TransUNetCD [50] is also a CNN–transformer hybrid structure. However, it directly upsamples the low-level features extracted from the backbone, uses concatenation to interact between the low-level features from the two temporal phases, and then fuses them with the features extracted by the transformer. It does not fully exploit the information contained in the low-level features of the backbone. Inspired by this, we use graph interaction modules to fuse multi-scale semantic features further.

2.4. Graph Convolutional Network

Unlike transformer, Graph Convolutional Networks (GCNs) can model long-term dependencies among image regions with minimal computational cost. A GCN propagates feature information through graph structures to capture relationships between nodes, having applications in knowledge graphs (Li et al. [51]), recommendation systems (Wang et al. [52]), and data mining (Gao et al. [53]). Recently, graph convolution has been applied for semantic segmentation. Li et al. [54] transformed 2D images into graphs in which each vertex represented a region. They used graph convolution operations to

propagate information among all vertices, capturing long-term dependencies. For medical image segmentation, Wang et al. [24] introduced a contour-guided graph reasoning module (CGRM) that effectively captures semantic relationships between contours and object regions, enhancing segmentation map quality.

In remote sensing, Liu et al. [55] utilized both CNN and GCN for feature learning in small-scale regular regions and large-scale irregular regions, generating complementary spectral-spatial features at pixel and superpixel levels. Zhang et al. [56] represented objects as graph structures and used graph convolution to analyze inter-object relationships, achieving accurate classification results. Liu et al. [57] proposed a graph-convolution-based dual-flow change detection network. Inspired by [24,54], we introduce a novel contour-guided graph interaction module (CGIM) leveraging graph convolution and joint attention for capturing relationships between change regions and contours in bi-temporal images.

3. Materials and Methods

In this section, we first introduce the overall architecture of DTT-CGINet, followed by the individual submodules that compose the network. Finally, we provide an overview of the training approach and loss function used.

3.1. Overall Architecture

As shown in Figure 3, the proposed network architecture utilizes a modified Siamese ResNet18 as a feature extraction backbone for bi-temporal images. A contour extraction module (CEM) subsequently obtains contour maps of images. We then employ three differently sized contour-guided graph interaction modules (CGIMs) to accept the backbone and contour map's multi-scale feature maps. The CGIM captures semantic relationships between changing regions and contours. The multi-scale outputs of each CGIM are fused into a single feature map using a feature pyramid decoder, followed by two convolutional block attention modules (CBAMs) to obtain refined feature maps in both channel and spatial dimensions. Furthermore, the deepest feature map from the backbone is fed into a Siamese semantic tokenizer [17], which converts each temporal feature map into a set of compact semantic tokens. These tokens are then processed through a dual temporal transformer (DTT) encoder to model the context between the two token sets. The DTT decoder projects the semantic tokens back into pixel space, yielding refined feature maps. Next, feature difference maps are separately extracted from the outputs of the CGIM and DTT decoder using absolute subtraction. Finally, these differencing features are concatenated and processed through a pixel-wise classifier to obtain the final change map.

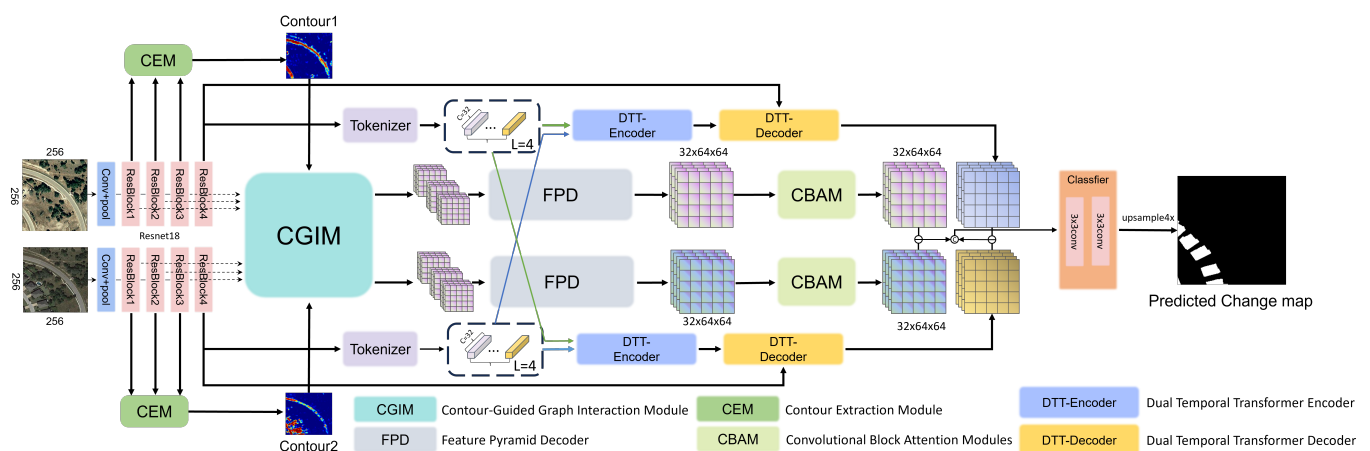


Figure 3. Overall network structure.

3.2. Feature Extraction Backbone

We adopted a Siamese network based on ResNet [58] with weight sharing. Compared to the VGG architecture [59], ResNet addresses the issue of gradient vanishing and model

degradation by introducing residual connections. Considering the model size, we used a modified ResNet18 and loaded pre-trained weights from ImageNet [60]. Furthermore, recognizing the domain gap between ImageNet and remote sensing images that can affect the generalization of change detection models, we employed a self-supervised pre-training method [26] on remote sensing images to mitigate this issue. In subsequent ablation experiments, we explored the impact of fine-tuning our change detection model using pre-trained weights from [26] via transfer learning.

Table 1 shows the detailed feature extraction backbone. The original ResNet18 consists of 1 conv layer, 4 ResBlocks layers, and a fully connected layer. Each layer has a $2\times$ downsampling. A convolutional layer with a 7×7 kernel size extracts local features from the input image, followed by batch normalization (BN) and ReLU activation. Then, to further reduce feature dimensions and increase receptive fields, max pooling with stride 2 is applied, halving feature map size. Following this, four layers of ResBlocks from ResNet18 are applied. For layer 4, the 2×2 stride is replaced with dilated convolution, expanding receptive fields. Additionally, to preserve fine-grained details, we use an upsampling operation to quadruple the feature map size. Given the computational complexity of the transformer, a conv2 layer reduces the original 512-channel dimension to 32. Thus, the feature map dimensions input to the DTT encoder are $32\times 64\times 64$ ($C\times W\times H$).

Table 1. Detailed feature extractor backbone.

Layer Name	Output Size ($C\times W\times H$)	Details
Conv1	$64\times 128\times 128$	$7\times 7, 64$, stride 2
Max Pooling	$64\times 64\times 64$	3×3 max pool, stride 2
layer1	$64\times 64\times 64$	$\begin{bmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{bmatrix} \times 2$
layer2	$128\times 32\times 32$	$\begin{bmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{bmatrix} \times 2$
layer3	$256\times 16\times 16$	$\begin{bmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{bmatrix} \times 2$
layer4	$512\times 16\times 16$	$\begin{bmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{bmatrix} \times 2$
Upsample $4\times$	$512\times 64\times 64$	
Conv2	$32\times 64\times 64$	$3\times 3, 32$ stride 1

The input image size is 256×256 .

3.3. Contour-Guided Graph Interaction Module

This module comprises three main parts, as shown in Figure 4, including (1) contour-guided graph projection, (2) the graph interaction module, and (3) graph reprojection. First, we input features extracted by ResNet18 from its first three layers into the contour extraction module to obtain temporal image contours. Then, we introduced an improved CGRM [24], named CGIM, that explicitly models relationships between contour and region features through graph interaction, enhancing boundary completeness in change detection results. We trained three distinct CGIMs with varying vertex numbers (16, 36, 64) to account for multi-scale contextual relationships.

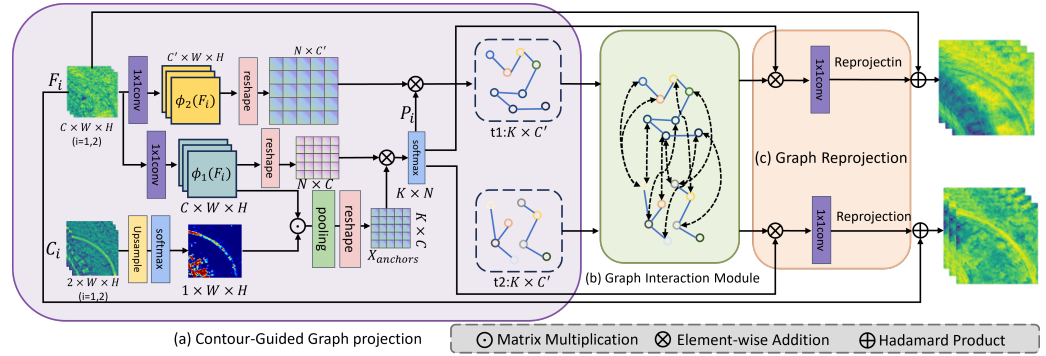


Figure 4. Illustration of the contour-guided graph interaction module depicted in Figure 3.

3.3.1. Contour Extraction Module

We designed a contour extraction module based on the Sobel convolutional block (SCB), effectively enhancing image details and emphasizing edges (Figure 5). Specifically, we extract output feature maps from layers 1, 2, and 3 of the backbone. These maps are individually processed through an SCB, consisting of a convolutional layer, batch normalization, and Sobel enhancement operations. The Sobel convolutional kernel includes horizontal and vertical kernels, detecting horizontal and vertical edges. The output feature maps are upsampled to the same size and summed to produce the contour.

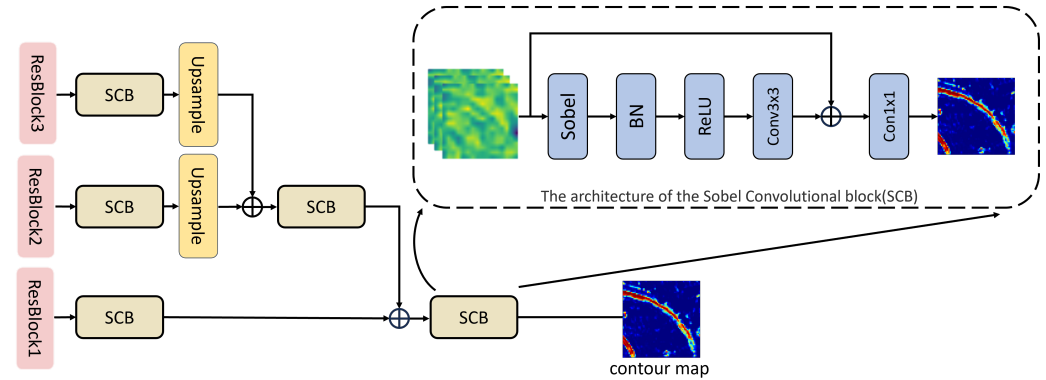


Figure 5. Illustration of the contour extraction module depicted in Figure 3.

3.3.2. Contour-Guided Graph Projection

As shown in Figure 4a, the input consists of the current temporal feature map $F_i^j \in \mathbb{R}^{H \times W \times C}$ ($i = 1, 2$; $j = 1, 2, 3$) and the contour map $C_i^j \in \mathbb{R}^{H \times W \times 2}$ ($i = 1, 2$; $j = 1, 2, 3$) extracted by the contour extraction module (CEM) (where i represents two temporal images, and j represents the output feature of layer 1, layer 2, and layer 3 of the backbone). It is noted that this module is Siamese and shares parameters. To obtain the graph representation, we project F_i^j to the vertices guided by C_i^j to obtain the projection matrix $P_i^j \in \mathbb{R}^{K \times N}$, where K is the number of vertices in the projected graph and $N = H \times W$.

First, we upsample the contour to the same resolution as the input feature map F_i^j . Then, we apply a 1×1 convolution to F_i^j to reduce its dimension, yielding $\phi_1(F_i^j)$. Next, we compute the Hadamard product of $\phi_1(F_i^j)$ and C_i^j , projecting contour information into the feature dimension and obtaining X_{mask} . The Hadamard product assigns higher weights to the building's contours. Simultaneously, we apply an average pooling on X_{mask} , obtaining $X_{anchors}$, representing regions in the image. Finally, we compute the similarity between each pixel and the anchors using the matrix multiplication between $\phi_1(F_i^j)$ and $X_{anchors}$,

normalize using softmax, and derive the projection matrix P_i^j . The formal equation is given by:

$$P_i^j = \text{Softmax}\left(\text{Avgpooling}(\phi_1(F_i^j) \odot C_i^j) \cdot \phi_1(F_i^j)^T\right) \quad (1)$$

After obtaining the projection matrix, we use Equation (2) to obtain graph representations.

$$G_i^j = P_i^j \phi_2(F_i^j) \quad (2)$$

where $\phi_2(\cdot)$ is a 1×1 convolutional layer. This process assigns pixels with similar features to the same vertices, with each vertex representing a region of the image. Ultimately, this projects the feature map into the graph domain, resulting in $G_i^j \in \mathbb{R}^{C^j \times K^j}$. Specifically, $C^1 = 64, C^2 = 64, C^3 = 128$, where C^1 represents the dimension of the features after graph projection for the layer1 output of the backbone. $K^1 = 64, K^2 = 36, K^3 = 16$ corresponds to the number of vertices in the projected graph.

3.3.3. Graph Interaction Module

After projecting feature maps from pixel space to a graph representation, we utilize the graph interaction module (GIM) to propagate information between dual-temporal images, as illustrated in Figure 6. Specifically, the GIM initially employs joint attention [61] to propagate global information and then further disseminates local information of a single temporal image via graph convolution [62]. The operation is defined as:

$$(G_1^i)', (G_2^i)' = \text{GCN}(\text{JointAtt}(G_1^i, G_2^i)) \quad (3)$$

Here, G_i^i (where $i = 1, 2, 3$) represents the graph representation from layer i of temporal 1.

- **Joint Attention:** To facilitate information interaction between the graph representations of bi-temporal images, we introduced joint attention from [61] to focus on graph nodes that undergo genuine changes. Specifically, as the graph itself is a one-dimensional sequence, we utilize a 1×1 convolutional operation to generate the query, key, and value for the graph representations of temporal images 1 and 2, denoted as $Q_1, K_1, V_1; Q_2, K_2, V_2$. Note that the channel dimension of Q needs to be halved. Subsequently, we concatenate Q_1 and Q_2 to obtain Q_{cat} , and through sequential matrix multiplication and Softmax, we obtain the similarity matrices Attn_i between Q_{cat} and K_i (where $i = 1, 2$). As Q_{cat} is a joint query from both temporal phases, it enables dual temporal interaction among graph nodes. Mathematically, JointAtt can be expressed as:

$$\begin{aligned} \hat{G}_1 &= \text{softmax}[\text{concat}(Q_1, Q_2) \cdot K_1] \cdot V_1 \\ \hat{G}_2 &= \text{softmax}[\text{concat}(Q_1, Q_2) \cdot K_2] \cdot V_2 \end{aligned} \quad (4)$$

- **Graph Convolution:** The architecture of the graph convolution unit is illustrated in Figure 6b, consisting of two 1D convolution layers that operate independently on the channel and node dimensions. The final output can be expressed as:

$$\hat{G} = ((I - A)G)W \quad (5)$$

Here, $I \in \mathbb{R}^{N \times N}$ denotes the identity matrix, $A \in \mathbb{R}^{N \times N}$ represents the adjacency matrix, and W denotes the updated parameters of the convolutional layer. A and W are randomly initialized during training for gradient descent.

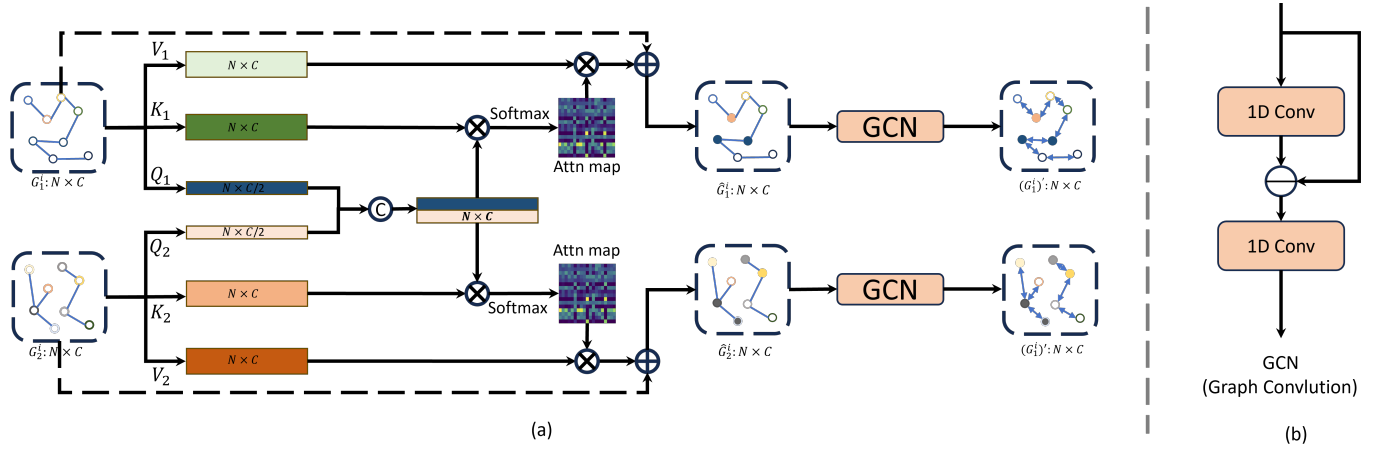


Figure 6. Illustration of graph interaction module depicted in Figure 4. (a) The whole architecture of the GIM. (b) The specific structure of graph convolution.

3.3.4. Graph Reprojection

To project the graph representation back to the original feature map space after graph interaction, for the graph representation $G \in \mathbb{R}^{C \times K}$, the projection matrix from the graph to the feature map is $Q \in \mathbb{R}^{HW \times K}$. Intuitively, one could think of the reprojection matrix Q as the inverse of the projection matrix P , denoted as P^{-1} . However, since matrix P is not square, it is non-invertible. According to [54], Q can be viewed as the transpose of the projection matrix P^T , where P_{ij}^T represents the similarity between pixel i and vertex j . $\hat{F}_i^j = (P_{ij}^T)^T G_i^j$ is used to project the graph back into pixel space, followed by a 1×1 convolution operation to restore the channel dimension to match the input feature map. Simultaneously, the original feature map is introduced through residual connections to obtain the final feature map X_i^j . The above process can be defined as

$$X_i^j = F_i^j + \varphi((P_{ij}^T)^T G_i^j) \quad (6)$$

We perform graph projection on the feature maps from the first three layers of the backbone to establish multi-scale contextual relationships. In summary, after CGIM processing, we obtain three multi-scale output features, denoted as:

$$\begin{aligned} X_1^1, X_2^1 &= \text{CGIM}(F_1^1, F_2^1) = G_{\text{reproj}}(G_{\text{interact}}(G_{\text{proj}}(F_1^1, F_2^1))) \\ X_1^2, X_2^2 &= \text{CGIM}(F_1^2, F_2^2) = G_{\text{reproj}}(G_{\text{interact}}(G_{\text{proj}}(F_1^2, F_2^2))) \\ X_1^3, X_2^3 &= \text{CGIM}(F_1^3, F_2^3) = G_{\text{reproj}}(G_{\text{interact}}(G_{\text{proj}}(F_1^3, F_2^3))) \end{aligned} \quad (7)$$

3.4. Feature Pyramid Decoder

We utilize two simple feature pyramid decoders (FPDs) to aggregate the multi-scale features obtained from the CGIM for the two temporal images. For temporal 1, the FPD takes inputs X_1^1, X_1^2, X_1^3 and produces fusion feature f_1 . The FPD achieves this by restoring original resolution through upsampling and parallel convolutions, as shown in Figure 7. Specifically, we first employ three separate convolutions to adjust the channel dimensions of feature maps X_1^1, X_1^2, X_1^3 to 64 for the current temporal image. Then, we individually upsample X_1^2 and X_1^3 by a factor of 2 and 4 to match the resolution of X_1^1 . Finally, we utilize two convolution blocks, each consisting of convolution, Batch Normalization, and ReLU activation, to obtain merged feature F_1 .

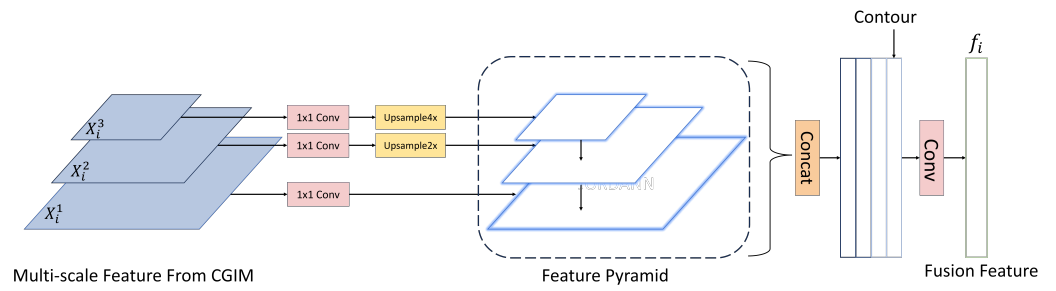


Figure 7. Illustration of feature pyramid decoder depicted in Figure 3.

3.5. Convolutional Block Attention Module (CBAM)

After using the feature pyramid decoder for multi-scale feature fusion, we obtained the output features F_i ($i = 1, 2$). To further enhance feature representation and model perception, we introduced two different Convolutional Block Attention Modules (CBAMs) [25], which incur negligible memory and computational overhead. As shown in Figure 8a, the CBAM comprises two modules: the Channel Attention Module (b) and the Spatial Attention Module (c). The CBAM employs a sequential approach, where input features pass through the Channel Attention Module and then to the Spatial Attention Module. Formally, this process can be represented by the following equation:

$$\begin{aligned}
 M_c(F_{in}) &= \sigma(\text{MLP}_{\text{share}}(\text{AvgPool}(F_{in})) + \text{MLP}_{\text{share}}(\text{MaxPool}(F_{in}))) \\
 F_c &= M_c(F_{in}) \otimes F_{in} \\
 M_s(F_c) &= \sigma(f^{(7 \times 7)}(\text{AvgPool}(F_c); \text{MaxPool}(F_c))) \\
 F_{out} &= M_s(F_c) \otimes F_c
 \end{aligned} \tag{8}$$

where F_{in} denotes the input feature, M_c and M_s are the channel attention and spatial attention, respectively, and F_{out} is the final output feature. The CBAM helps to refine and augment the feature map, improving model performance.

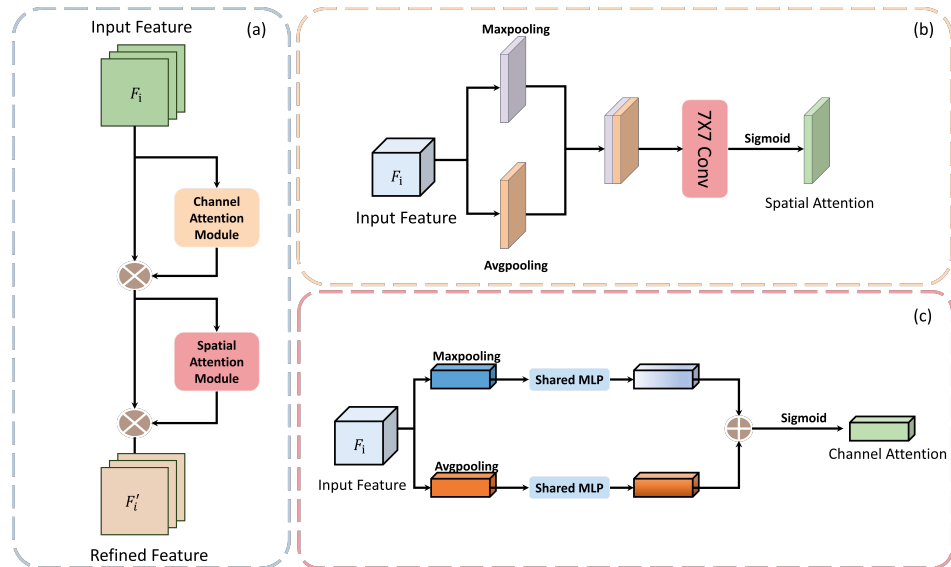


Figure 8. Architecture of CBAM block depicted in Figure 3. (a) CBAM block; (b) Channel Attention Module; (c) Spatial Attention Module.

3.6. Dual Temporal Transformer

For the first three layers of the backbone, we used the contour-guided graph interaction module (CGIM) to model semantic relationships within contours and regions. However, the output features from layer 4 of the backbone, denoted as X_i^4 (where $i = 1, 2$), still need devel-

opment. We introduced a transformer to model long-range semantic contextual relationships to address this. Inspired by BIT [17], we propose a dual-temporal transformer (DTT) that considers relations between bi-temporal images, as shown in Figure 9. The DTT consists of three main components: (1) a tokenizer, (2) DTT encoder, and (3) DTT decoder.

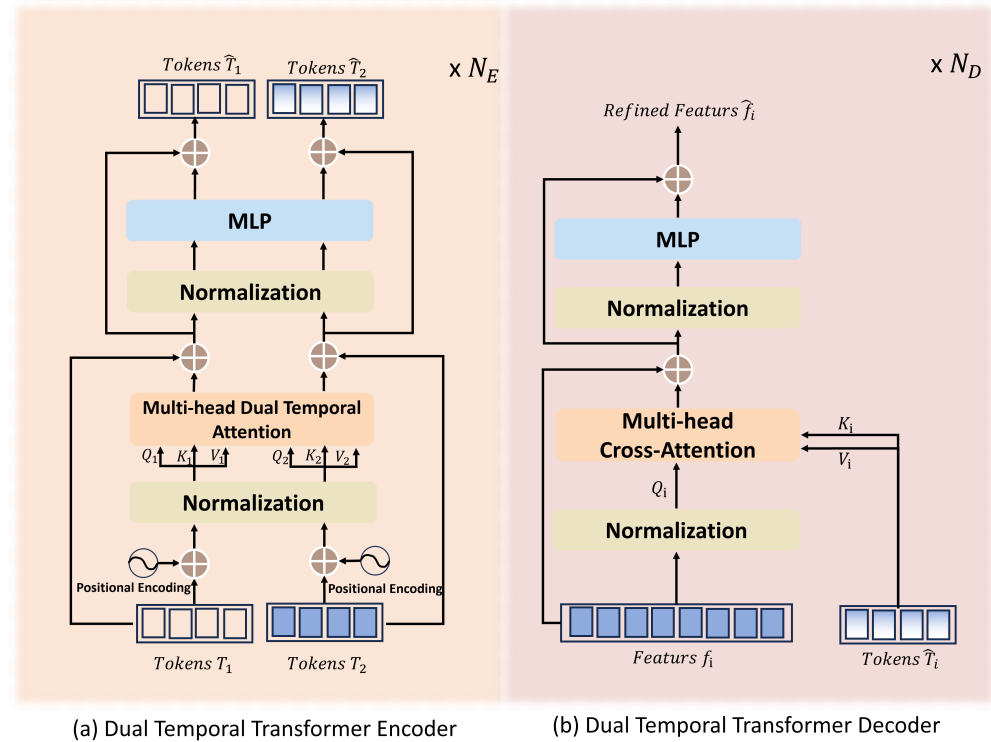


Figure 9. Illustration of dual temporal transformer depicted in Figure 3.

3.6.1. Tokenizer

To capture long-range contextual relationships within images, transformer-based methods typically start by partitioning the image into multiple image patches and then modeling the relationships between these patches using attention mechanisms. To achieve this, we introduce the Semantic Tokenizer from BIT [17], which uses a Siamese semantic tokenizer to extract compact semantic tokens for each temporal feature map.

Specifically, the tokenizer divides the 2D image into several patches and represents each patch with a single token. Figure 10 illustrates this process. In Figure 10, we apply convolution to bi-temporal feature maps $f_1, f_2 \in \mathbb{R}^{C \times HW}$. A softmax operation then computes spatial attention maps $M_1, M_2 \in \mathbb{R}^{L \times HW}$. The convolution output channel size is the number of tokens, denoted by L . Finally, matrix multiplication takes place between X_i and attention map M_i to compute a weighted sum, yielding tokens that aggregate semantic relations, namely semantic tokens T_i .

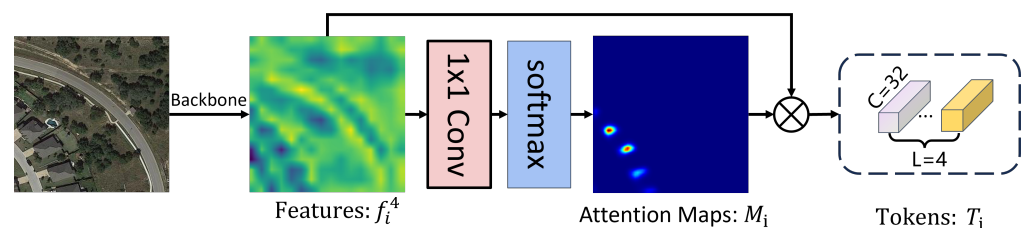


Figure 10. Illustration of tokenizer depicted in Figure 3.

3.6.2. DTT Encoder

1. Self-Attention in Transformer

In most transformer networks such as BIT [17] and ChangeFormer [20], self-attention is applied to model the relationship between tokens from single temporal images. The attention output $\hat{T}_1^i$ of the i -th token from temporal one is first multiplied with weight matrix W_q to obtain the query q . Similarly, by multiplying with weight matrices W_k and W_v , we obtain the key k and value v :

$$\begin{aligned} q_1^i &= t_1^i \cdot W_q \\ k_1^i &= t_1^i \cdot W_k \\ v_1^i &= t_1^i \cdot W_v \end{aligned} \quad (9)$$

Finally, we obtain $\hat{T}_1^i$ by summing the self-attention weighted contributions from other tokens' values v_1^j :

$$\hat{T}_1^i = \sum_{j=1}^n \text{softmax}\left(\frac{q_1^i \cdot k_1^j}{\sqrt{d_k}}\right) \cdot v_1^j \quad (10)$$

For all tokens in temporal one and temporal two, the final self-attention output can be calculated as follows:

$$\begin{aligned} \hat{T}_1 &= \text{softmax}\left(\frac{Q_1 K_1^T}{\sqrt{d_k}}\right) \cdot V_1 \\ \hat{T}_2 &= \text{softmax}\left(\frac{Q_2 K_2^T}{\sqrt{d_k}}\right) \cdot V_2 \end{aligned} \quad (11)$$

In this formula, the query, key, and value are defined as $Q_i = X_i W_Q$ ($i = 1, 2$), $K_i = X_i W_K$, and $V_i = X_i W_V$ where d_k represents the dimension of the key, corresponding to the scaling factor. The matrix $W_{1,1} = \text{softmax}\left(\frac{Q_1 K_1^T}{\sqrt{d_k}}\right)$ represents the similarity matrix between the tokens in temporal one.

2. The Dual Temporal Attention in DTT Encoder

However, applying self-attention to model semantic relationships between tokens in a single temporal image does not consider information integration across bi-temporal images, as shown in Figure 1b,c. Therefore, we propose dual-temporal attention to model the relation changes between bi-temporal images, as shown in Figure 11. We first compute the similarity between t_1^i and all tokens in T_2 using cross-attention by the following formula:

$$W_{1,2}^i = \text{softmax}\left(\frac{q_1^i K_2^T}{\sqrt{d_k}}\right) \quad (12)$$

Intuitively, in Figure 1b, region P8 of temporal one is similar to region P1 of temporal two, hence $W_{1,2}^8[8, 1] \approx 1$. Similarly, in Figure 1c, region P2 of temporal one is dissimilar to region P1 of temporal two, so $W_{1,2}^2[2, 1] \approx 0$. Similarly, we compute the similarity relationships between all tokens in temporal one and all tokens in temporal two using the following formula:

$$\begin{aligned} M_{1,2} &= \text{softmax}\left(\frac{Q_1 K_2^T}{\sqrt{d_k}}\right) \\ M_{2,1} &= \text{softmax}\left(\frac{Q_2 K_1^T}{\sqrt{d_k}}\right) \end{aligned} \quad (13)$$

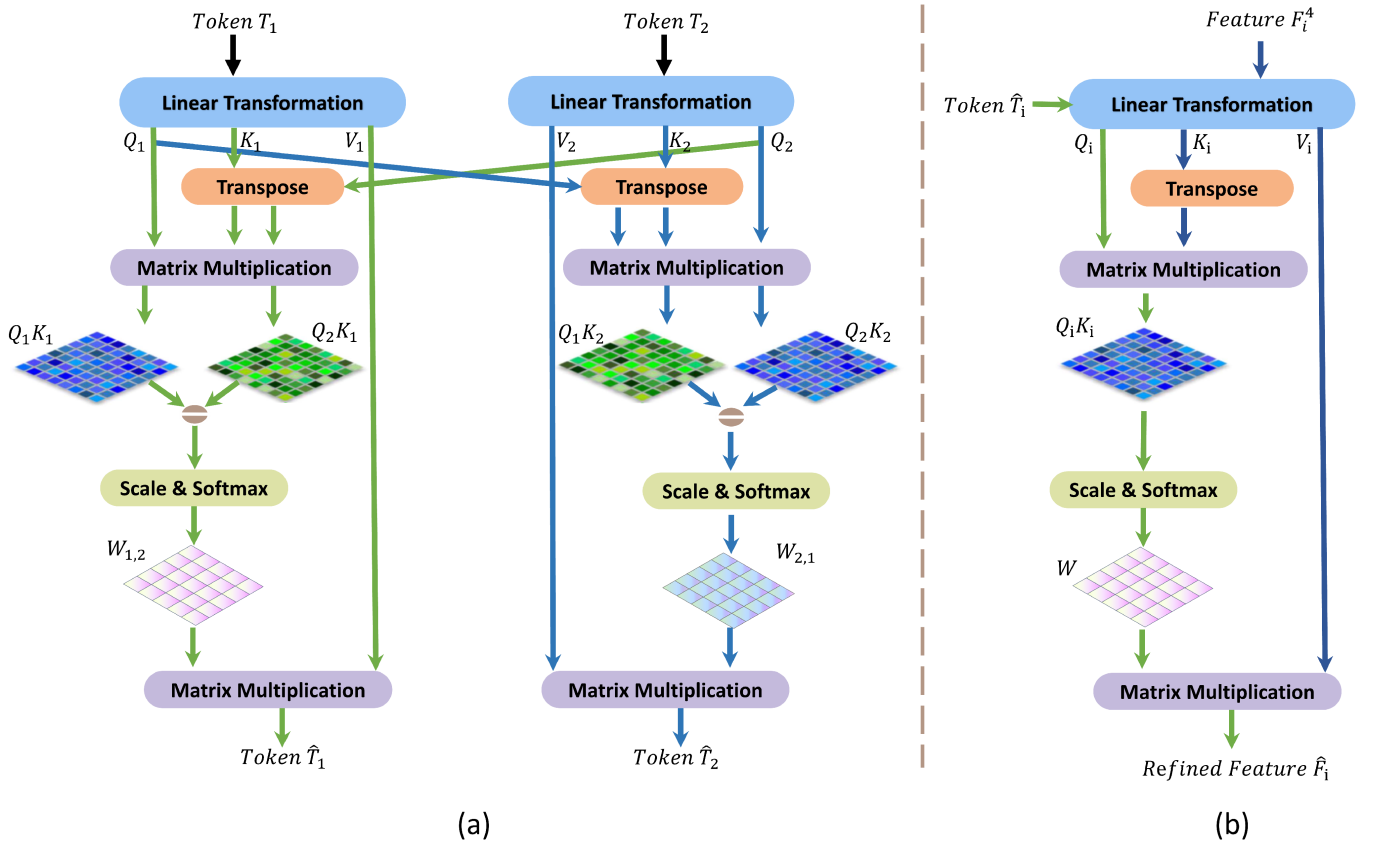


Figure 11. (a) Structure diagram of dual temporal attention in DTT encoder (Figure 9a); (b) Structure diagram of cross-attention in DTT decoder (Figure 9b).

Finally, to comprehensively consider $W_{1,1}$ and $W_{1,2}$, we calculate the cross-attention map using the following formula, denoted as W_1, W_2 , and compute the output $\hat{T}_1$ and $\hat{T}_2$ based on the obtained attention maps:

$$\begin{aligned} W_1 &= \text{softmax} \left[\frac{(Q_1 K_1 - Q_2 K_1)}{\sqrt{d_k}} \right], \hat{T}_1 = W_1 \cdot V_1 \\ W_2 &= \text{softmax} \left[\frac{(Q_2 K_2 - Q_1 K_2)}{\sqrt{d_k}} \right], \hat{T}_2 = W_2 \cdot V_2 \end{aligned} \quad (14)$$

- **Situation of unchanged:** As shown in Figure 12a, if $(Q_1^i K_1^j - Q_2^i K_1^j) > 0$, and assuming token i at temporal 2 is similar to token j at temporal 1. The $Q \cdot K$ in transformer can be intuitively understood as calculating the similarity between two tokens; thus, we have $|Q_2^i K_1^j| = a > 0$, where a is a relatively large positive value. It can also be obtained that $|Q_1^i K_1^j| = a > 0$, indicating that token i in temporal 1 is similar to token j . At the same time, we have $W_1[i, j] = b < \text{softmax} \left[\frac{Q_1^i K_1^j}{\sqrt{d_k}} \right]$. Through transitive similarity, it can be inferred that token i is similar in both temporal images. In the context of bi-temporal images, this also implies that the region represented by token i has remained unchanged. Consequently, when the attention output of token j is computed by weighted summation, the feature component from token i will be suppressed.
- **Situation of changed:** As shown in Figure 12b, assuming that tokens i and j of temporal image 1 are dissimilar, we have $|Q_1^i K_1^j| \rightarrow 0$. If $W_1[i, j] = (Q_1^i K_1^j - Q_2^i K_1^j) = a > 0$, this indicates the region represented by token i has remained changed. At the same time, we have $W_1[i, j] > \text{softmax} \left[\frac{Q_1^i K_1^j}{\sqrt{d_k}} \right]$. It means that when computing attention output $\hat{T}_i$,

the features of token i will be strengthened to highlight changed regions. In contrast, for self-attention in Equation (11), the tokens representing changed and unchanged regions are treated equally when calculating attention output. We think this is not conducive to highlighting features of changed regions while suppressing features of unchanged regions.

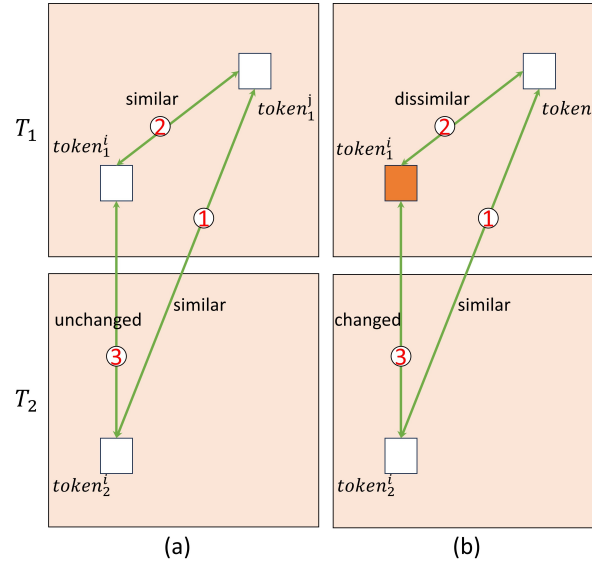


Figure 12. (a) One unchanged situation in cross-attention. (b) One changed situation in cross-attention.

Based on the cross-attention in Equation (14), we designed the DTT encoder to extract features of the changed regions in the bi-temporal image pairs. Specifically, this encoder consists of a multi-head DTT attention block and a multilayer perceptron (MLP) block, repeated for N_E layers, as illustrated in Figure 9a. The process can be represented as follows:

$$\begin{aligned} a_i &= \mathbf{MDTTAtt}(\mathbf{LN}(T_1), \mathbf{LN}(T_2)) + T_i (i = 1, 2) \\ \hat{T}_1 &= \mathbf{MLP}(\mathbf{LN}(a_1)) + a_1 \\ \hat{T}_2 &= \mathbf{MLP}(\mathbf{LN}(a_2)) + a_2 \end{aligned} \quad (15)$$

3.6.3. DTT decoder

Through the encoder shown in Figure 10a, dual temporal relationships are aggregated in two sets of new tokens $\hat{T}_i$ ($i = 1, 2$). To project this high-level semantic information represented by these tokens back into two-dimensional pixel space, we constructed the DTT decoder, obtaining refined feature maps $\hat{f}_i$, as shown in Figure 10b. Unlike MSA, we do not use f_i to compute key K and value V , as computing attention maps over the original input sequence f_i would require numerous computations. Instead, we first compute the query Q using the feature maps f_i , then compute the key K and value V using the tokens $\hat{T}_i$, resulting in a similarity matrix M . Then, a softmax operation is performed to obtain an attention matrix W , which is used to weight and sum the values V to obtain the refined features. The DTT decoder comprises a multi-head DTT attention (MDTTA) and MLP blocks, stacked for N_D layers. Formally, MDTTA for the decoder is defined as:

$$\begin{aligned} a_i &= \mathbf{MDTTAtt}(\mathbf{LN}(f_i), \mathbf{LN}(\hat{T}_i)) + f_i (i = 1, 2) \\ \hat{F}_1 &= \mathbf{MLP}(\mathbf{LN}(a_1)) + a_1 \\ \hat{F}_2 &= \mathbf{MLP}(\mathbf{LN}(a_2)) + a_2 \end{aligned} \quad (16)$$

3.7. Loss Function

Given a set of training image pairs $X_n = \{(x_n^{t1}, x_n^{t2}), n = 1, \dots, N\}$ and ground truth $Y_n = \{y_n, n = 1, \dots, N\}$, where N represents the number of training bi-temporal images, we use a hybrid loss consisting of three components: (1) focal loss, (2) dice loss, and (3) contrastive loss:

$$\mathcal{L} = \mathcal{L}_{focal} + \mathcal{L}_{dice} + \lambda \mathcal{L}_{con} \quad (17)$$

3.7.1. Focal Loss

We analyzed the ratio of changed and unchanged pixels in all three datasets, as shown in Table 2. The number of pixels in the changed region was significantly smaller than in the unchanged region. Change detection is a binary classification problem, with a severe class imbalance between positive and negative samples. To address this, we introduced focal loss [63], which effectively mitigates the class imbalance problem. Formally, it can be defined as

$$\mathcal{L}_{focal}(X_n, Y_n) = -\alpha(1 - \hat{y}_{i,j})^\gamma \log(\hat{y}_{i,j}) \quad (18)$$

where α and γ are fixed constants, and we set $\alpha = 2$ and $\gamma = 0.2$. $\hat{y}_{i,j}$ is the predicted value at position (i, j) .

Table 2. Statistical characteristics of WHU-CD, LEVIR-CD, and CDD datasets.

Dataset	Pairs	Size	Change Pixels	Change Ratio
WHU-CD [64]	1	$32,507 \times 15,354$	21,352,815	4.27%
LEVIR-CD [65]	637	1024×1024	30,913,975	4.62%
CDD [66]	7 and 4	4725×2700 and 1900×1000	9,198,562 and 400,279	9.91%

3.7.2. Dice Loss

The dice loss is a similarity-based loss function that measures the overlap between predicted and ground truth masks. It excels in tasks where precise definition of object boundaries is critical. Therefore, we introduced it to further ensure the boundary integrity of the change detection result. The dice loss encourages the model to generate segmentation masks with distinct, well-defined boundaries by penalizing discrepancies in the overlap between predicted and ground truth masks. Mathematically, dice loss is defined as follows:

$$\mathcal{L}_{Dice}(X_n, Y_n) = 1 - \frac{2\hat{y}y}{\hat{y} + y} \quad (19)$$

Here, $\hat{y}$ is a positive example of model prediction and y is a positive example of ground truth. It is obvious that the calculation of the F1 score is the same as the dice. Thus, we can optimize the F1 metrics by calculating dice loss.

3.7.3. Contrastive Loss

The contrastive loss [67] is often used to measure the similarity between two sets and can effectively handle paired data relationships in a Siamese network for change detection. Its core idea is distinguishing different categories or entities by learning feature representations, bringing similar samples closer, and pushing dissimilar samples further apart in feature space. However, the original formulation requires computing Euclidean distances in feature space for paired segmentation maps to obtain a distance map. We find this introduces extra computational overhead and hinders network optimization. Therefore, we directly compute contrastive loss between the predicted map and the ground truth. The contrastive loss can be expressed formally as follows:

$$\mathcal{L}_{Con}(X_n, Y_n) = \sum_{i,j=0}^M \frac{1}{2} [(1 - y_{i,j}) \cdot \argmax(\hat{y}_{i,j}) + y_{i,j} \cdot \max(\argmax(\hat{y}_{i,j}) - m)]^2 \quad (20)$$

where M is the size of the predicted map, 0 denotes unchanged, while 1 denotes changed, and m is the margin to filter out pixel pairs with a greater distance.

4. Results

In this section, we provide a comprehensive comparison on three CD datasets with other state-of-the-art methods to demonstrate the effectiveness of the proposed DTT-CGINet.

4.1. Description of Datasets

- **WHU-CD** [64] is a public building CD dataset from Christchurch, New Zealand, with a spatial size of $32,507 \times 15,354$ pixels at a resolution of 0.2 m. It comprises images in the red (R), green (G), and blue (B) bands. To facilitate efficient handling, we divided the large image into non-overlapping slices of 256×256 pixels. Then, the training/validation/test sample numbers were 6096/762/762, respectively.
- **LEVIR-CD** [65] consists of 637 very high-resolution (VHR, 0.5 m/pixel) Google Earth image patch pairs with a size of 1024×1024 pixels. These bi-temporal images with a time span of 5 to 14 years have significant land-use changes, especially construction growth. LEVIR-CD covers various types of buildings, such as villa residences, tall apartments, small garages, and large warehouses. We followed the default configuration to facilitate model training and partitioned the input images into smaller patches of 256×256 pixels. The dataset was split into 7120 image pairs for training, 1024 for validation, and 2048 for testing.
- **CDD** [66] is a dataset of 11 pairs of multispectral images for remote sensing change detection. The dataset contains seven pairs of seasonal images with a dimension of 4725×2200 pixels and four pairs of images with a dimension of 1900×1000 pixels. The spatial resolution of these images varies from 3 to 100 cm per pixel. The authors divided the image pairs into non-overlapping image patches of 256×256 pixels to make them suitable for processing. They obtained 15,998 pairs of bi-temporal remote sensing images and split them into training, validation, and test sets with 10,000, 2998, and 3000 pairs, respectively.

4.2. Metrics

Change detection primarily distinguishes changed and unchanged pixels, comprising a fundamental binary classification task. We evaluated using multiple metrics, precision (P), recall (R), F1 score (F1), Overall Accuracy (OA), and Intersection over Union (IoU), providing a comprehensive assessment of the proposed method. We also report the parameter count and FLOPs (Floating Point Operations Per Second) for each comparative model. We utilized MACC (Multiply-ACCumulate) operations to approximate a model's computational speed (FLOPs). They are expressed as follows.

$$\begin{aligned}
 P &= TP / (TP + FP) \\
 R &= TP / (TP + FN) \\
 F1 &= 2PR / (P + R) \\
 OA &= (TP + TN) / (TP + TN + FP + FN) \\
 IoU &= TP / (TP + FP + FN)
 \end{aligned} \tag{21}$$

where True positive (TP) refers to the number of pixels correctly identified as changed. True negative (TN) represents the number of pixels correctly identified as unchanged. False negative (FN) signifies the number of pixels that genuinely underwent a change but were erroneously categorized as unchanged. False positive (FP) indicates the number of pixels that remained unchanged but were incorrectly identified as having changed. F1 is a comprehensive measurement metric that performs weighted average reconciliation on precision and recall, better reflecting the comprehensive performance of the model. The Intersection over Union (IoU) measures the extent of overlap between the predicted change pixel area and the actual change pixel area.

4.3. Experimental Settings

All experiments were conducted using the PyTorch 1.7.1 library on one NVIDIA GeForce RTX 3090 GPU and Xeon E5-2680 CPU. We employed the AdamW optimizer to train our model. The optimizer was configured with a learning rate (lr) of 0.0004 and a weight decay of 5×10^{-4} . We utilized a custom learning rate scheduler, which adjusted learning rates based on step count and number of epochs. The learning rate scheduler implemented a warm-up strategy, gradually increasing the learning rate during initial epochs. Specifically, the warm-up spanned five epochs, during which the learning rate factor gradually increased from 1×10^{-3} to 1. After the warm-up phase, the learning rate was reduced with a power factor of 0.9, ensuring a smooth decrease throughout the training process. This configuration was instrumental in achieving a balance between rapid convergence during the initial training stages and fine-tuning the model parameters in later epochs.

The batch size was set to 32, and the model was trained for 100 epochs. To enhance the diversity of our training samples, we employed the following data augmentation techniques: (1) random cropping, (2) random flipping, and (3) random exchange.

4.4. Compared Methods

- **FC-EF [40]**: FC-EF is a change detection network that employs early fusion, based on the U-Net architecture of a fully convolutional network.
- **FC-Siam-Conc [40]**: FC-Siam-Conc is a variant of FC-EF that employs a late fusion strategy during the decoding stage, using concatenation as the fusion method.
- **FC-Siam-Diff [40]**: FC-Siam-Diff is another FC-EF variant, utilizing the difference's absolute value as the feature fusion method.
- **STANet [65]**: STANet improves remote sensing image change detection by capturing spatial-temporal dependencies at different scales.
- **SNUNet [41]**: SNUNet is a Unet-type network that introduces dense skip connections.
- **DMINet [61]**: DMINet is a novel dual temporal change detection network based on joint attention.
- **BIT [17]**: BIT enhances high-resolution remote sensing change detection by efficiently capturing spatial-temporal contexts through a transformer-based approach.
- **ICIF-Net [68]**: ICIF-Net is a hybrid network that combines CNN and transformer in parallel, aiming to leverage the respective strengths of CNN and transformer.
- **ChangeFormer [20]**: ChangeFormer introduced a novel multi-scale transformer.

4.5. Evaluation Results

4.5.1. Quantitative Comparisons

- **On the LEVIR-CD dataset**, Table 3 shows our method demonstrates significant improvement, achieving precision, recall, and F1 values of 92.45%, 89.83%, and 91.12%, respectively. FC-Diff achieves the highest F1 score among the first three comparison methods, reaching 85.72%. Compared to the second-best method, our approach increases the F1 value by 0.8%.
- **On the WHU-CD dataset**, our method demonstrates significant improvement, as shown in Table 4. It shows enhancements in all evaluation metrics, with precision, recall, and F1 values being 95.51%, 92.80%, and 94.14%, respectively. Simultaneously, our model has a relatively small number of parameters, 4.65M. Consequently, our approach achieves a commendable balance between model complexity, time cost, and accuracy. In contrast, earlier methods such as FC-EF, FC-Conc, and FC-Diff show less satisfactory performance. The results also indicate that concatenation preserves more useful information for change detection on the WHU-CD dataset than the difference operation. Moreover, our approach exhibits notable advantages over transformer-based methods, including BIT [17] and ChangeFormer [20]. It also outperforms the hybrid network ICIF-Net [68], which employs a parallel structure of CNN and transformer.

Table 3. Quantitative results on LEVIR-CD dataset.

Model	Precision	Recall	F1	IoU	OA	Params (M)	FLOPs (G)
FC-EF [40]	86.69	77.95	82.09	77.95	98.28	1.35	3.58
FC-Siam-Conc [40]	84.37	81.51	82.91	70.80	98.49	1.55	5.33
FC-Siam-Diff [40]	89.33	82.39	85.72	75.00	98.61	1.35	4.73
STANet [65]	70.88	96.01	81.56	68.86	97.79	16.89	16.9
DMINet [61]	91.09	85.26	88.08	78.70	98.82	6.24	14.55
SNUNet [41]	91.21	86.69	88.89	78.83	98.82	12.04	54.83
BIT [17]	89.10	89.16	89.13	80.68	98.92	3.50	10.63
ICIF-Net [68]	91.63	88.10	89.83	81.54	98.98	23.83	25.37
ChangeFormer [20]	91.94	88.81	90.34	82.37	99.04	41.03	202.79
Ours	92.45	89.83	91.12	83.50	99.12	4.71	18.42

All metrics are based on the category “change” and computed on the test set. Color convention: **best**, **2nd-best**, and **3rd-best**.

Table 4. Quantitative results on WHU-CD dataset.

Model	Precision	Recall	F1	IoU	OA	Params (M)	FLOPs (G)
FC-EF [40]	77.67	77.16	77.42	63.16	98.08	1.35	3.58
FC-Siam-Conc [40]	36.49	82.75	50.65	40.03	93.40	1.55	5.33
FC-Siam-Diff [40]	45.18	82.28	58.33	41.17	94.98	1.35	4.73
STANet [65]	79.37	85.50	82.32	69.95	98.66	16.89	16.9
DMINet [61]	83.98	91.09	87.39	77.61	98.88	6.24	14.55
SNUNet [41]	91.72	86.75	89.16	80.43	99.10	12.04	54.83
BIT [17]	90.46	77.55	83.51	71.69	98.69	3.50	10.63
ICIF-Net [68]	92.25	89.28	90.74	83.04	99.22	23.83	25.37
ChangeFormer [20]	94.18	89.14	91.86	89.24	99.37	41.03	202.79
Ours	95.51	92.80	94.14	88.93	99.51	4.71	18.42

All metrics are based on the category “change” and computed on the test set. Color convention: **best**, **2nd-best**, and **3rd-best**.

- **On the CDD dataset**, our method also achieved the best performance, with an F1 score of 95.78%, as shown in Table 5. Due to the larger training set of the CDD dataset, which consists of 10,000 images compared to 6096 pairs for WHU-CD and 7120 pairs for LEVIR-CD, neural network models can better learn feature representations, leading to superior generalization capabilities. Consequently, almost all methods significantly improve their F1 scores on the CDD dataset, with transformer-based approaches showing particularly notable enhancements.

Table 5. Quantitative results on CDD dataset.

Model	Precision	Recall	F1	IoU	OA	Params (M)	FLOPs (G)
FC-EF [40]	88.46	49.73	63.67	46.70	92.99	1.35	3.58
FC-Siam-Conc [40]	89.71	58.73	70.98	55.02	94.21	1.55	5.33
FC-Siam-Diff [40]	90.16	51.38	65.46	48.65	93.31	1.35	4.73
STANet [65]	76.97	94.55	84.86	73.71	95.84	16.89	16.9
DMINet [61]	96.02	95.23	95.61	91.88	98.89	6.24	14.55
SNUNet [41]	94.46	89.72	92.03	85.23	98.08	12.035	54.83
BIT [17]	95.46	90.68	93.01	86.94	98.32	3.50	10.63
ICIF-Net [68]	95.04	93.79	94.41	89.41	98.03	23.83	25.37
ChangeFormer [20]	95.47	94.31	94.88	90.27	98.74	41.03	202.79
Ours	96.79	94.78	95.78	91.90	98.92	4.71	18.42

All metrics are based on the category “change” and computed on the test set. Color convention: **best**, **2nd-best**, and **3rd-best**.

4.5.2. Qualitative Comparisons

- **LEVIR-CD dataset:** To further demonstrate the effectiveness of our approach, Figure 13 shows the visual results of different methods in three types of regions. These regions

include isolated regions, dense regions, and large-span regions, demonstrating our method's superiority over others. In particular, Figure 13(1,2) suggests that our method can effectively capture isolated regions, while many previous methods fail to provide complete segmentation results (d–f). Furthermore, our approach outperforms other methods in detecting dense regions, as shown in Figure 13(3,4). The change results effectively ensure boundary integrity, with more obvious gaps between boundaries. Our result is more consistent than other transformer-based methods regarding large-span area boundaries in complex scenes, as Figure 13(j6,k6,l6) shows. At the same time, some previous methods (FC-EF, FC-Siam-Conc, FC-diff) exhibit not only unclear boundaries but also significant voids in changing regions, as shown in Figure 13d–f. These experiments show that DTT-CGINet performs excellently on the LEVIR-CD dataset, with clear, complete boundaries, sensitivity to small targets, and no holes in large-area detection.

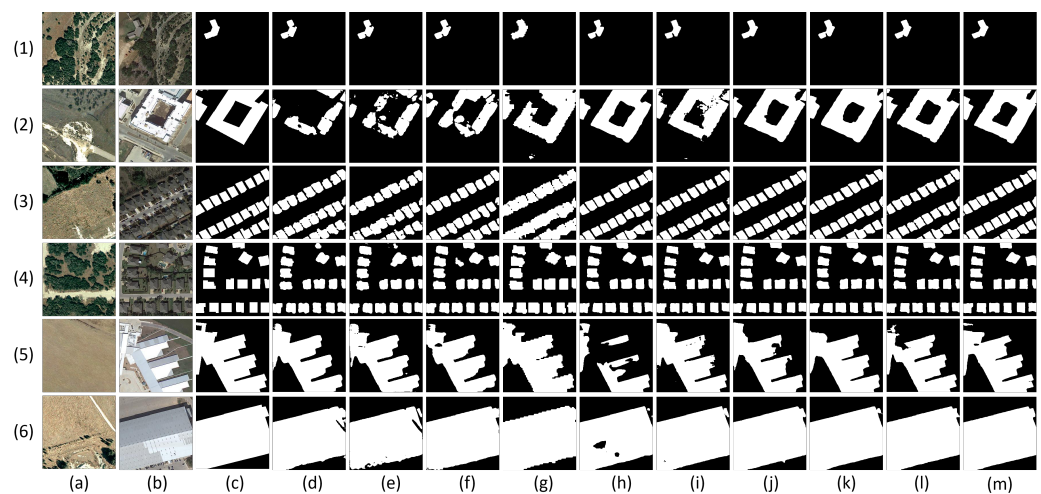


Figure 13. Visualization comparisons on LEVIR-CD dataset. (1–6) Different image pairs; (a) image T1; (b) image T2; (c) ground truth; (d) FC-EF; (e) FC-Siam-Conc; (f) FC-Siam-Diff; (g) STANet; (h) DMINet; (i) SNUNet; (j) BIT; (k) ICIF-Net; (l) ChangeFormer; (m) ours.

- **WHU-CD dataset:** Consistent with LEVIR-CD, we selected three region types representing independent, dense, and large-span areas to evaluate our method visually from diverse perspectives. For the independent area in Figure 14(1,2), many previous methods tended to produce more false positives and false negatives. Additionally, previous methods commonly exhibited boundary sticking issues for the dense area in Figure 14(3,4), whereas our results were more accurate. As for the large-span area in Figure 14(5,6), although most methods detected it fairly well, our approach had fewer holes, higher boundary integrity, and clearer boundaries.
- **CDD Dataset:** We also selected three different types of regions: (1) isolated regions, (2) dense regions, and (3) large-span regions—to visually validate our method's performance. Some early methods, like FC-EF, FC-Conc, and FC-Diff, could not detect changes in isolated or dense regions and performed poorly in large-span regions. Subsequent methods, like STANet and SNUNet, could more completely detect isolated region changes but had sticking phenomena in dense regions and unclear, incomplete boundaries for large spans. The latest transformer-based methods could provide relatively satisfactory results, but our method outperformed them in maintaining boundary integrity, as demonstrated in Figure 15(m6,m3).

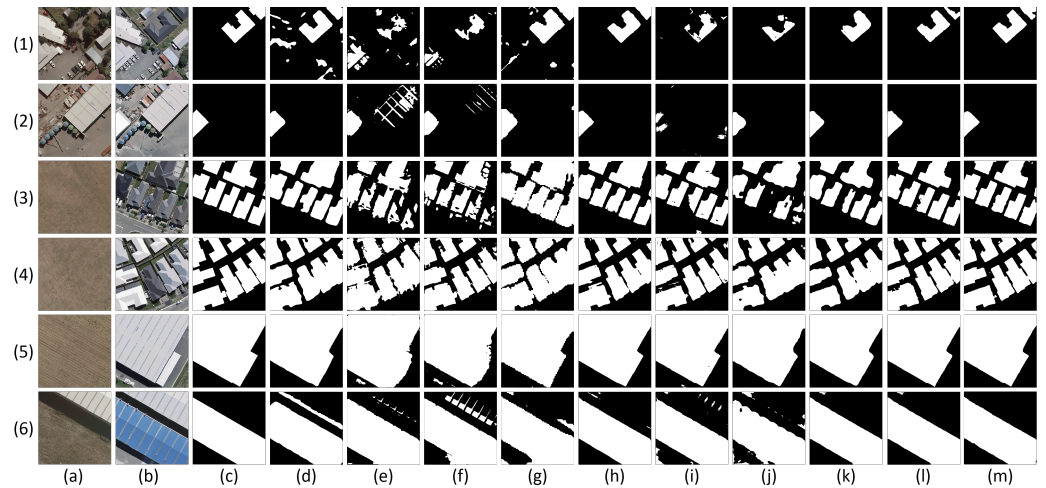


Figure 14. Visualization comparisons on WHU-CD dataset. (1–6) Different image pairs; (a) image T1; (b) image T2; (c) ground truth; (d) FC-EF; (e) FC-Siam-Conc; (f) FC-Siam-Diff; (g) STANet; (h) DMINet; (i) SNUNet; (j) BIT; (k) ICIF-Net; (l) ChangeFormer; (m) ours.

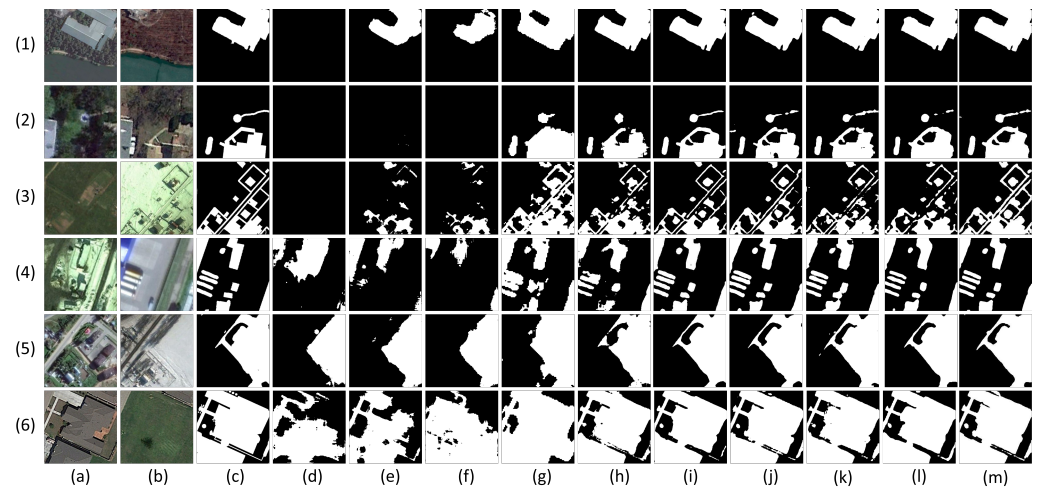


Figure 15. Visualization comparisons on CDD dataset. (1–6) Different image pairs; (a) image T1; (b) image T2; (c) ground truth; (d) FC-EF; (e) FC-Siam-Conc; (f) FC-Siam-Diff; (g) STANet; (h) SUNNet; (i) BIT; (j) ICIF-Net; (k) ChangeFormer; (l) DMINet; (m) ours.

5. Discussion

5.1. Ablation Study on the Network Components

To evaluate each component's contribution to overall performance and prove our method's effectiveness, we conducted four ablation experiments on the LEVIR-CD dataset. Table 6 displays each ablation experiment's results.

Table 6. Quantitative results of the different ablation experiments on the LEVIR-CD dataset.

Model	Precision	Recall	F1
No CGIM	90.19	89.43	89.80
No FPD	91.35	89.43	90.38
NO CBAM	91.28	90.24	90.76
NO DTT	90.47	86.51	88.45
Ours	92.45	89.83	91.12

- **In Ablation 1**, we removed the CGIM module to validate its effectiveness. The CGIM module is crucial in constraining graph projection to generate graph nodes within edge regions, yielding well-defined boundary segmentation results. As indicated in

Table 6, removing CGIM led to a decrease in all evaluation metrics for the network. In particular, the primary evaluation metric, F1, declines 1.32%. Figure 16(1,2) illustrates boundary blurring resulting from CGIM absence.

- **In Ablation 2**, to demonstrate the effectiveness of the Feature Pyramid Decoder (FPD) in DTT-CGINet, we conducted an ablation study by removing it. Since the FPD can aggregate features from multiple scales, its absence would result in the network solely using the CGIM on the output features from the backbone's layer 3, underutilizing the shallow output features from layers 1 and 2. Therefore, as indicated in the second row of Table 6, removing the FPD led to an overall performance decline in the network, with the F1 score experiencing a decrease of 0.74%. Visually, this impacted the model's ability to detect changes across regions of varying scales. Figure 16(d3) shows that many small regions are fused, and the network fails to capture finer local details. Figure 16(d4) illustrates missing detections for small objects.
- **In Ablation 3**, we conducted an ablation study by removing the CBAM submodule. CBAM reweights feature maps based on channel and spatial attention, guiding the network to focus on relevant changes and ignore irrelevant ones. Removing CBAM resulted in an overall performance decline, as shown in Table 6, with a 0.36% decrease in the F1 score.
- **In Ablation 4**, we conducted an ablation study to demonstrate the efficacy of the dual temporal transformer (DTT). As described in Section 3.6, DTT can model non-local structural relationships between bi-temporal images. The fourth row of Table 6 indicates overall performance declines when DTT is absent, with the F1 score dropping 2.67%. Visually, as shown in Figure 16(5,6), the lack of DTT hinders the network's long-range context modeling capability, impacting change detection in large-span regions.

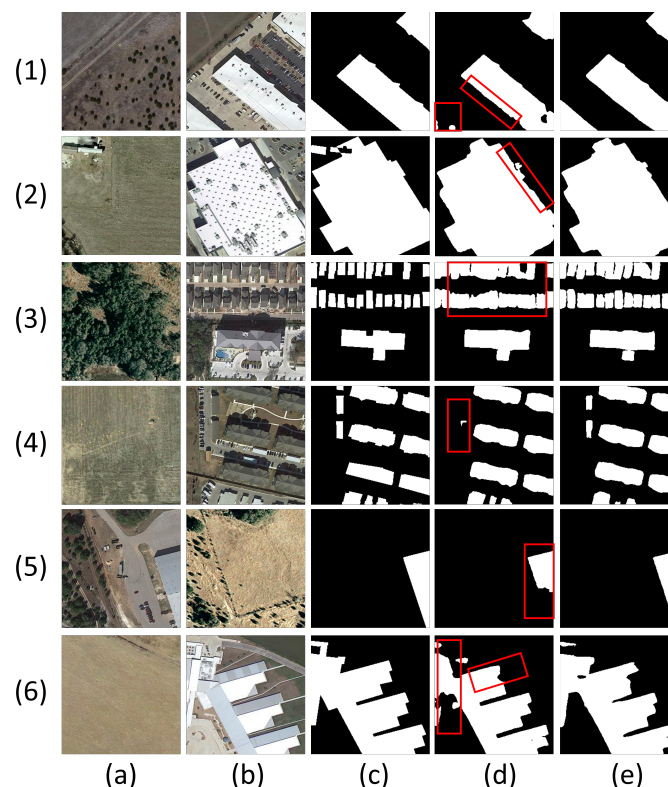


Figure 16. Visualization comparisons of ablation experiments. (1,2) Different image pairs of Ablation 1; (3,4) Different image pairs of Ablation 2; (5,6) Different image pairs of Ablation 4; (a) image T1; (b) image T2; (c) ground truth; (d) prediction results of ablation operation; (e) prediction results for networks with all components.

5.2. Parameter Analysis of Loss

In Section 3.7, we utilized a hybrid loss for network training, combining focal loss, dice loss, and contrastive loss. To reduce the tuning burden, we introduced a single parameter, λ , to balance the impact of the contrastive loss. We conducted ablation experiments to explore the effects of different λ values on training. Specifically, we varied λ from 0 to 1 to observe performance changes on the WHU-CD dataset. The results of these experiments have been collected and are presented in Table 7.

Table 7. Ablation study of the the loss function on the WHU-CD dataset.

λ	Precision	Recall	F1
0	95.18	91.33	93.31
0.1	95.52	91.07	93.54
0.3	95.94	91.63	93.74
0.5	95.67	92.80	94.14
0.7	95.51	91.25	93.63
1	95.48	91.66	93.79

λ is the coefficient of contrastive loss in the hybrid loss function and the bolded data represents the best results.

5.3. Ablation on the CGIM

We conducted an ablation study on the quantity of the contour-guided graph interaction module (CGIM) and the number of vertices from the graph projection in CGIM. As shown in Table 8, we varied the number of CGIM modules, with the corresponding quantities of graph vertices after projection set to 64, 36, and 16. It is observed that employing multiple CGIM modules enhances model performance. Furthermore, introducing three CGIM modules brings only 0.30 M parameters and 0.23 GFLOPs. In Table 9, we explore the impact of three sets of different vertices on model performance. Our intuition suggests that the number of vertices in the graph projection should align with the dimensions of the feature map (H/W). Additionally, adding more nodes increases run-time and memory costs, yet does not seem to improve the performance.

Table 8. Performance comparison with different numbers of CGIM on LEVIR-CD.

Number	Precision	Recall	F1	Params (M)	FLOPs (G)
0	90.25	89.28	89.76	4.41	18.21
1	91.19	89.45	90.40	4.62	18.26
2	91.83	89.94	90.72	4.67	18.31
3	92.45	89.83	91.12	4.71	18.42

Table 9. Performance comparison with different vertices of graph projection on LEVIR-CD.

Number	Precision	Recall	F1	Params(M)	FLOPs(G)
(16,16,16)	90.25	89.28	89.76	4.56	18.39
(64,36,16)	92.45	89.83	91.12	4.71	18.42
(64,64,64)	91.83	89.94	90.88	4.93	18.66

5.4. Ablation on the Tokenizer

As shown in Tables 10 and 11, we conducted an ablation study by varying the number of tokens output by the tokenizer. Specifically, we tested different token quantities $L \in 0, 2, 4, 8$ on the LEVIR-CD and WHU-CD datasets to assess their impact on the model. When the token number is 0, it implies the direct removal of the tokenizer. Using a larger number of tokens allows for better extraction of refined details. However, it may weaken the model's ability to capture large-scale changes in regions and introduce higher computational complexity. Conversely, a smaller number of tokens can yield more concise semantic representations, reducing computational complexity, but may cause the model to overlook

changes in some small areas. Observing the results, it is evident that our model achieves the best performance when L is set to 4.

Table 10. Impact of different token numbers on the LEVIR-CD dataset.

Number	Precision	Recall	F1
0	92.62	88.79	90.66
2	92.53	89.16	90.81
4	92.45	89.83	91.12
8	91.71	89.85	90.77

Table 11. Impact of different token numbers on the WHU-CD dataset.

Number	Precision	Recall	F1
0	92.13	91.80	92.95
2	95.63	92.07	93.82
4	95.51	92.80	94.14
8	95.08	91.65	93.33

5.5. Ablation Study on Pre-training

Many recent studies have focused on remote sensing pre-training. The authors of [26] proposed Seasonal Contrast (SeCo), an effective pipeline for pre-training in the remote sensing domain using unlabeled data. The results in Table 12 compare the performance of the backbones using pre-trained weights from ImageNet and pre-trained weights from Seco [26]. Notably, the transformer branch is trained entirely from scratch. When using only 10% of the samples from the LEVIR-CD training set to supervise our model's training, fine-tuning our network with Seco's pre-trained weights resulted in a 1.46% improvement in the F1 score compared to ImageNet. However, when supervising the model with all samples from the training set, using Seco's pre-trained weights led to a slight decrease in the F1 score compared to ImageNet's pre-trained weights. This may be because there is still a gap between the Sentinel-2 multispectral images and RGB aerial images. Figure 17 presents two visual change detection results.

Table 12. Performance comparison on the effect of pre-training on LEVIR-CD.

Pre-Training	10% Samples of Training Set			100% Samples of Training Set		
	Precision	Recall	F1	Precision	Recall	F1
ImageNet [60]	87.46	80.70	83.95	92.45	89.83	91.12
Seco [26]	87.99	82.98	85.41	91.73	90.11	90.91

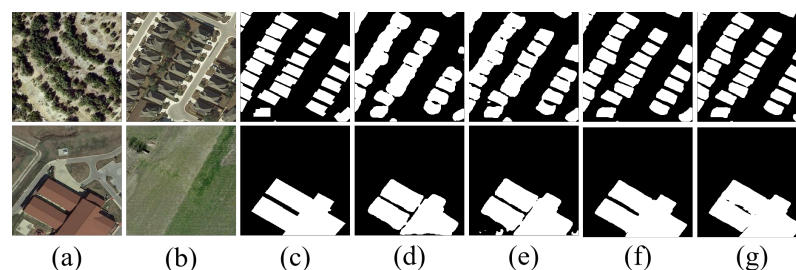


Figure 17. Visual change detection results on LEVIR-CD. (a) Image T1; (b) image T2; (c) ground truth; (d) training by ImageNet pre-trained weights and 10% of the training set; (e) training by Seco pre-trained weights and 10% of the training set; (f) training by ImageNet pre-trained weights and 100% of the training set; (g) training by Seco pre-trained weights and 100% of the training set.

5.6. Model Efficiency Analysis

In Figure 18, we present the detection accuracy in terms of the F1 score, model parameters (Params), and computation costs (FLOPs) for all the compared methods. As observed from the results, foundational models like FC-EF achieve minimal computational resource utilization but exhibit shortcomings in detection accuracy. Subsequently, proposed methods such as ICIF-Net [68] and DMINet [61] enhance detection accuracy further. ChangeFormer [20] closely approximates our approach regarding the F1 metric. However, the model parameters and computation costs are several tens of times higher than our method. Consequently, our proposed approach strikes a favorable balance between accuracy and computational expenditure, highlighting its effectiveness and efficiency in change detection.

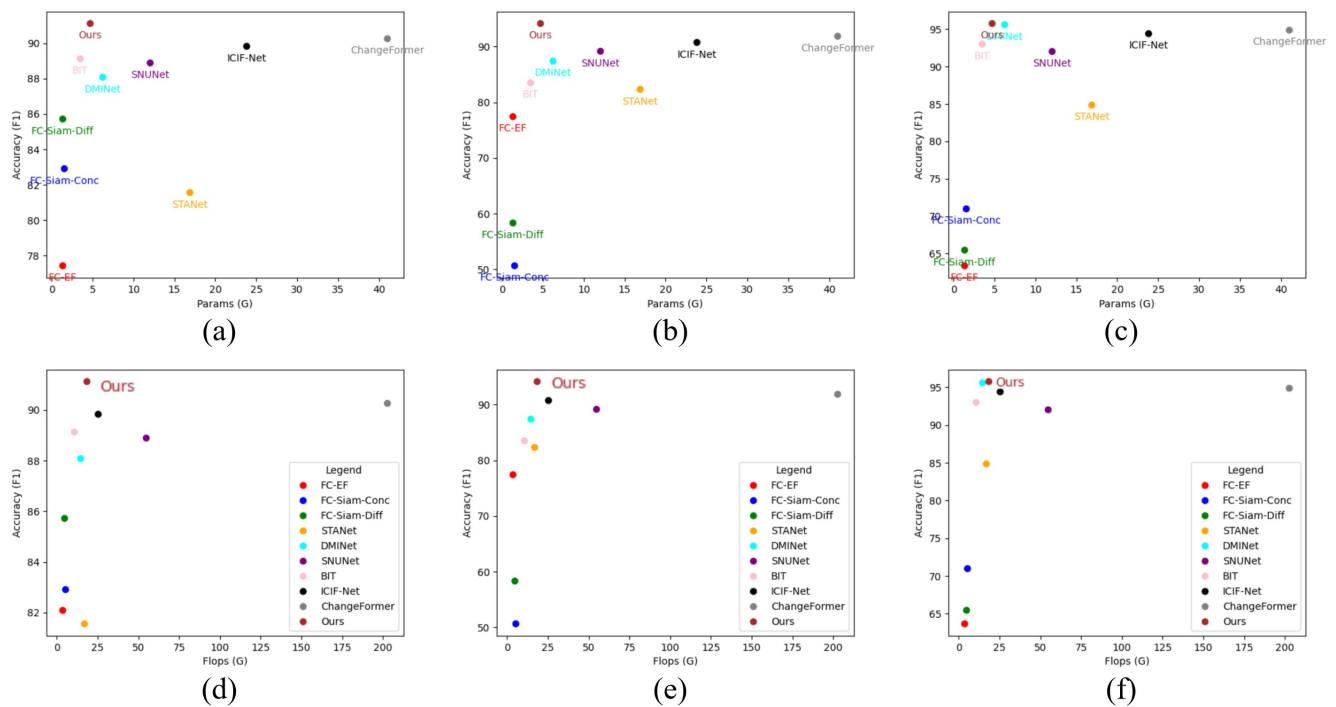


Figure 18. Comparison of compared methods in terms of Params (memory cost), FLOPs (computational cost), and F1 score on the LEVIR-CD, WHU-CD, and CDD dataset, respectively. (a,d) LEVIR-CD dataset; (b,e) WHU-CD dataset; (c,f) CDD dataset.

5.7. Visualization of Network

To further illustrate the effectiveness of our network, we utilized Grad-CAM [69] to visualize the predicted change map by our network. Grad-CAM is a visualization method that utilizes gradients to calculate the importance of spatial positions in the convolutional layer. Simultaneously, we employed heatmaps to visualize the output features of the main modules. Given a pair of temporal images (Figure 19a), the Siamese ResNet extracts multi-scale features, and we visualized the deep features of layer3 (Figure 19b). Subsequently, we used the CGIM to aggregate boundary features while employing the FPD to fuse multi-scale features, resulting in X_1 and X_2 (Figure 19d). Furthermore, we enhanced the features using the CBAM (Figure 19e).

Simultaneously, we visualized the token attention map obtained via the tokenizer (Figure 19f), demonstrating effective highlighting of building pixels by the tokenizer. Furthermore, we presented the refined features obtained through the DTT decoder (Figure 19g). Figure 19h,i, respectively, show the difference feature maps after processing through the CGIM module and dual temporal transformer. Finally, we used Grad-CAM to visualize the change probability map obtained via the classifier (Figure 19j). It can be observed that the

CAM assigns high density only in the center of the changed region, and DTT-CGINet can label almost the entire shape of changed landscapes with the highest weight.

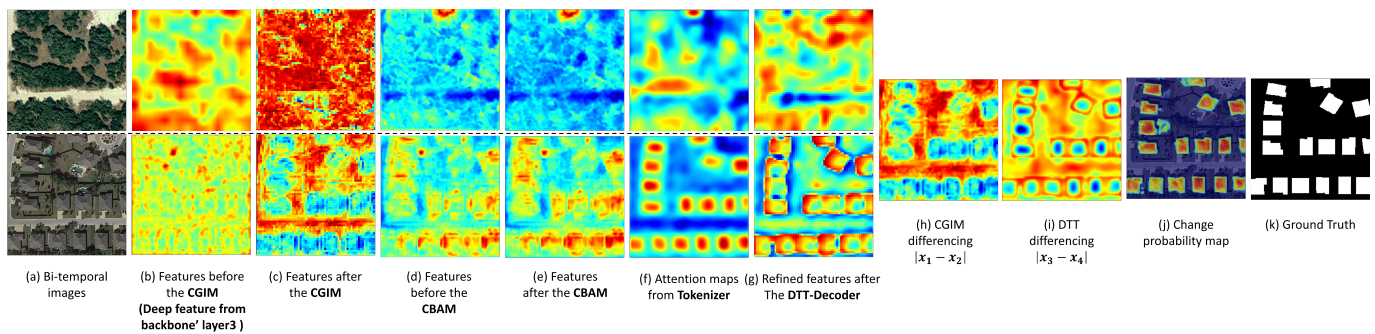


Figure 19. An example of network visualization. (a) Input images; (b) deep features extracted by the backbone; (c) features after the CGIM; (d) features before the CBAM; (e) features after the CBAM; (f) token visualization; (g) refined features map by DTT decoder; (h) bi-temporal feature differencing image from CGIM; (i) bi-temporal feature differencing image from dual temporal transformer; (j) Class Activation Map (CAM) by the Grad-cam [69]; (k) ground truth.

6. Conclusions

In this paper, we propose a novel hybrid network called DTT-CGINet. Our network utilizes dual temporal attention to model non-local structural relationships between bi-temporal images. To alleviate the computational complexity of attention calculations in transformers, we introduce a tokenizer to transform high-level features from the backbone into tokens with rich contextual semantic information. Subsequently, we map the tokens to a two-dimensional pixel space to obtain refined features considering contextual information. Additionally, to fully leverage the local information extracted from the low-level feature maps by the backbone. We use the CGIM to capture fine-grained details and preserve the integrity of change boundaries. Simultaneously, FPD is used to fuse multi-scale feature maps, and the CBAM is introduced to emphasize important pixels. Finally, experiments on three publicly available change detection datasets demonstrate that our approach outperforms several other methods.

Limitation

In Section 5.5, we explored the impact of using remotely sensed pre-trained weights on the performance of DTT-CGINet, though no significant improvement was observed. We analyze that this is due to the current lack of effective large-scale pre-training weights for aerial remote sensing images, which inevitably limits the fine-tuning performance of downstream aerial scene tasks. However, in remote sensing, the exploration and development of large-scale pre-trained models remains a hot topic, and we will continue following research progress to improve our work. In the future, we will focus on proposing a more lightweight change detection model or using techniques such as knowledge distillation to compress the original model to make it more suitable for practical applications. Additionally, we plan to extend the application of our method to other object types, such as roads, vegetation, and rivers. We also intend to explore the use of DTT-CGINet for multi-class change detection.

Author Contributions: Conceptualization, M.C. and W.J.; data curation, M.C.; funding acquisition, W.J.; investigation, M.C. and Y.Z.; methodology, M.C.; supervision, W.J.; visualization, M.C. and Y.Z.; writing—original draft, M.C.; writing—review and editing, M.C. and W.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the High-Resolution Remote Sensing Application Demonstration System for Urban Fine Management grant 06-Y30F04-9001-20/22 and in part by the National Natural Science Foundation of China grant number 42371452.

Data Availability Statement: (1) LEVIR-CD: <https://chenhao.in/LEVIR/> (accessed on 20 February 2024); (2) WHU-CD: <http://gpcv.whu.edu.cn/data/> (accessed on 20 February 2024); (3) CDD: https://drive.google.com/file/d/1GX656JqqOyBi_Ef0w65kDGVto-nHrNs9/edit (accessed on 20 February 2024); (4) Code: https://github.com/WesternTrail/DTT_CGINet (accessed on 20 February 2024).

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

References

- Gong, M.; Zhao, J.; Liu, J.; Miao, Q.; Jiao, L. Change detection in synthetic aperture radar images based on deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 125–138. [\[CrossRef\]](#)
- Radke, R.J.; Andra, S.; Al-Kofahi, O.; Roysam, B. Image change detection algorithms: A systematic survey. *IEEE Trans. Image Process.* **2005**, *14*, 294–307. [\[CrossRef\]](#)
- Zerrouki, N.; Harrou, F.; Sun, Y.; Hocini, L. A machine learning-based approach for land cover change detection using remote sensing and radiometric measurements. *IEEE Sens. J.* **2019**, *19*, 5843–5850. [\[CrossRef\]](#)
- Marin, C.; Bovolo, F.; Bruzzone, L. Building change detection in multitemporal very high resolution SAR images. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 2664–2682. [\[CrossRef\]](#)
- Eismann, M.T.; Meola, J.; Hardie, R.C. Hyperspectral change detection in the presence of diurnal and seasonal variations. *IEEE Trans. Geosci. Remote Sens.* **2007**, *46*, 237–249. [\[CrossRef\]](#)
- Zhou, J.; Kwan, C.; Ayhan, B.; Eismann, M.T. A novel cluster kernel RX algorithm for anomaly and change detection using hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6497–6504. [\[CrossRef\]](#)
- Kwan, C. Methods and challenges using multispectral and hyperspectral images for practical change detection applications. *Information* **2019**, *10*, 353. [\[CrossRef\]](#)
- Marmanis, D.; Datcu, M.; Esch, T.; Stilla, U. Deep learning earth observation classification using ImageNet pretrained networks. *IEEE Geosci. Remote Sens. Lett.* **2015**, *13*, 105–109. [\[CrossRef\]](#)
- Shi, Q.; Liu, M.; Li, S.; Liu, X.; Wang, F.; Zhang, L. A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5604816. [\[CrossRef\]](#)
- Chen, H.; Li, W.; Shi, Z. Adversarial instance augmentation for building change detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5603216. [\[CrossRef\]](#)
- Peng, D.; Zhang, Y.; Guan, H. End-to-end change detection for high resolution satellite images using improved UNet++. *Remote Sens.* **2019**, *11*, 1382. [\[CrossRef\]](#)
- Liu, Y.; Pang, C.; Zhan, Z.; Zhang, X.; Yang, X. Building change detection for remote sensing images using a dual-task constrained deep siamese convolutional network model. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 811–815. [\[CrossRef\]](#)
- Peng, X.; Zhong, R.; Li, Z.; Li, Q. Optical remote sensing image change detection based on attention mechanism and image difference. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 7296–7307. [\[CrossRef\]](#)
- Jiang, H.; Hu, X.; Li, K.; Zhang, J.; Gong, J.; Zhang, M. PGA-SiamNet: Pyramid feature-based attention-guided Siamese network for remote sensing orthoimagery building change detection. *Remote Sens.* **2020**, *12*, 484. [\[CrossRef\]](#)
- Zhang, M.; Xu, G.; Chen, K.; Yan, M.; Sun, X. Triplet-based semantic relation learning for aerial remote sensing image change detection. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 266–270. [\[CrossRef\]](#)
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
- Chen, H.; Qi, Z.; Shi, Z. Remote sensing image change detection with transformers. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5607514. [\[CrossRef\]](#)
- Song, F.; Zhang, S.; Lei, T.; Song, Y.; Peng, Z. MSTDSNet-CD: Multiscale swin transformer and deeply supervised network for change detection of the fast-growing urban regions. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 6508505. [\[CrossRef\]](#)
- Liu, M.; Chai, Z.; Deng, H.; Liu, R. A CNN-transformer network with multiscale context aggregation for fine-grained cropland change detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 4297–4306. [\[CrossRef\]](#)
- Bandara, W.G.C.; Patel, V.M. A transformer-based siamese network for change detection. In Proceedings of the IGARSS 2022—2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, 17–22 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 207–210.
- Ding, L.; Guo, H.; Liu, S.; Mou, L.; Zhang, J.; Bruzzone, L. Bi-temporal semantic reasoning for the semantic change detection in HR remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5620014. [\[CrossRef\]](#)
- Zhou, Y.; Huo, C.; Zhu, J.; Huo, L.; Pan, C. DCAT: Dual Cross-Attention-Based Transformer for Change Detection. *Remote Sens.* **2023**, *15*, 2395. [\[CrossRef\]](#)
- Xu, C.; Ye, Z.; Mei, L.; Shen, S.; Zhang, Q.; Sui, H.; Yang, W.; Sun, S. SCAD: A Siamese Cross-Attention Discrimination Network for Bitemporal Building Change Detection. *Remote Sens.* **2022**, *14*, 6213. [\[CrossRef\]](#)
- Wang, K.; Zhang, X.; Lu, Y.; Zhang, X.; Zhang, W. CGRNet: Contour-guided graph reasoning network for ambiguous biomedical image segmentation. *Biomed. Signal Process. Control* **2022**, *75*, 103621. [\[CrossRef\]](#)
- Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.

26. Manas, O.; Lacoste, A.; Giró-i Nieto, X.; Vazquez, D.; Rodriguez, P. Seasonal contrast: Unsupervised pre-training from uncured remote sensing data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 9414–9423.
27. Bourdis, N.; Marraud, D.; Sahbi, H. Constrained optical flow for aerial image change detection. In Proceedings of the 2011 IEEE International Geoscience and Remote Sensing Symposium, Vancouver, BC, Canada, 24–29 July 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 4176–4179.
28. Johnson, R.D.; Kasischke, E. Change vector analysis: A technique for the multispectral monitoring of land cover and condition. *Int. J. Remote Sens.* **1998**, *19*, 411–426. [\[CrossRef\]](#)
29. Nielsen, A.A.; Conradsen, K.; Simpson, J.J. Multivariate alteration detection (MAD) and MAF postprocessing in multispectral, bitemporal image data: New approaches to change detection studies. *Remote Sens. Environ.* **1998**, *64*, 1–19. [\[CrossRef\]](#)
30. Nielsen, A.A. The regularized iteratively reweighted MAD method for change detection in multi-and hyperspectral data. *IEEE Trans. Image Process.* **2007**, *16*, 463–478. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Deng, J.; Wang, K.; Deng, Y.; Qi, G. PCA-based land-use change detection and analysis using multitemporal and multisensor satellite data. *Int. J. Remote Sens.* **2008**, *29*, 4823–4838. [\[CrossRef\]](#)
32. Wu, C.; Du, B.; Zhang, L. Slow feature analysis for change detection in multispectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 2858–2874. [\[CrossRef\]](#)
33. Lv, P.; Zhong, Y.; Zhao, J.; Zhang, L. Unsupervised change detection based on hybrid conditional random field model for high spatial resolution remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4002–4015. [\[CrossRef\]](#)
34. Nemmour, H.; Chibani, Y. Multiple support vector machines for land cover change detection: An application for mapping urban extensions. *ISPRS J. Photogramm. Remote Sens.* **2006**, *61*, 125–133. [\[CrossRef\]](#)
35. Im, J.; Jensen, J.R. A change detection model based on neighborhood correlation image analysis and decision tree classification. *Remote Sens. Environ.* **2005**, *99*, 326–340. [\[CrossRef\]](#)
36. Wessels, K.J.; Van den Bergh, F.; Roy, D.P.; Salmon, B.P.; Steenkamp, K.C.; MacAlister, B.; Swanepoel, D.; Jewitt, D. Rapid land cover map updates using change detection and robust random forest classifiers. *Remote Sens.* **2016**, *8*, 888. [\[CrossRef\]](#)
37. Moser, G.; Angiati, E.; Serpico, S.B. Multiscale unsupervised change detection on optical images by Markov random fields and wavelets. *IEEE Geosci. Remote Sens. Lett.* **2011**, *8*, 725–729. [\[CrossRef\]](#)
38. Ma, B.; Chang, C.Y. Semantic segmentation of high-resolution remote sensing images using multiscale skip connection network. *IEEE Sens. J.* **2021**, *22*, 3745–3755. [\[CrossRef\]](#)
39. Sun, L.; Cheng, S.; Zheng, Y.; Wu, Z.; Zhang, J. SPANet: Successive pooling attention network for semantic segmentation of remote sensing images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 4045–4057. [\[CrossRef\]](#)
40. Daudt, R.C.; Le Saux, B.; Boulch, A. Fully convolutional siamese networks for change detection. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 4063–4067.
41. Fang, S.; Li, K.; Shao, J.; Li, Z. SNUNet-CD: A densely connected Siamese network for change detection of VHR images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 8007805. [\[CrossRef\]](#)
42. Zhang, C.; Yue, P.; Tapete, D.; Jiang, L.; Shangguan, B.; Huang, L.; Liu, G. A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2020**, *166*, 183–200. [\[CrossRef\]](#)
43. Zhang, Y.; Fu, L.; Li, Y.; Zhang, Y. HDFNet: Hierarchical dynamic fusion network for change detection in optical aerial images. *Remote Sens.* **2021**, *13*, 1440. [\[CrossRef\]](#)
44. Huang, J.; Shen, Q.; Wang, M.; Yang, M. Multiple attention Siamese network for high-resolution image change detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5406216. [\[CrossRef\]](#)
45. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
46. Yin, W.; Kann, K.; Yu, M.; Schütze, H. Comparative study of CNN and RNN for natural language processing. *arXiv* **2017**, arXiv:1702.01923.
47. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [\[CrossRef\]](#)
48. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
49. Ding, M.; Yang, Z.; Hong, W.; Zheng, W.; Zhou, C.; Yin, D.; Lin, J.; Zou, X.; Shao, Z.; Yang, H. Cogview: Mastering text-to-image generation via transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 19822–19835.
50. Li, Q.; Zhong, R.; Du, X.; Du, Y. TransUNetCD: A hybrid transformer network for change detection in optical remote-sensing images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5622519. [\[CrossRef\]](#)
51. Pujara, J.; Miao, H.; Getoor, L.; Cohen, W. Knowledge graph identification. In *The Semantic Web—ISWC 2013: Proceedings of the 12th International Semantic Web Conference, Sydney, NSW, Australia, 21–25 October 2013*; Proceedings, Part I 12; Springer: Berlin/Heidelberg, Germany, 2013; pp. 542–557.
52. Isinkaye, F.O.; Folajimi, Y.O.; Ojokoh, B.A. Recommendation systems: Principles, methods and evaluation. *Egypt. Inform. J.* **2015**, *16*, 261–273. [\[CrossRef\]](#)
53. Romero, C.; Ventura, S. Data mining in education. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2013**, *3*, 12–27. [\[CrossRef\]](#)

54. Li, Y.; Gupta, A. Beyond grids: Learning graph representations for visual recognition. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 9245–9255.
55. Liu, Q.; Xiao, L.; Yang, J.; Wei, Z. CNN-enhanced graph convolutional network with pixel-and superpixel-level feature fusion for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 8657–8671. [\[CrossRef\]](#)
56. Zhang, X.; Tan, X.; Chen, G.; Zhu, K.; Liao, P.; Wang, T. Object-based classification framework of remote sensing images with graph convolutional networks. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 8010905. [\[CrossRef\]](#)
57. Liu, C. Remote Sensing Image Change Detection with Graph Interaction. *arXiv* **2023**, arXiv:2307.02007.
58. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
59. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
60. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1–9. [\[CrossRef\]](#)
61. Feng, Y.; Jiang, J.; Xu, H.; Zheng, J. Change detection on remote sensing images using dual-branch multilevel intertemporal network. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 4401015. [\[CrossRef\]](#)
62. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
63. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
64. Ji, S.; Wei, S.; Lu, M. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 574–586. [\[CrossRef\]](#)
65. Chen, H.; Shi, Z. A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. *Remote Sens.* **2020**, *12*, 1662. [\[CrossRef\]](#)
66. Lebedev, M.; Vizilter, Y.V.; Vygodov, O.; Knyaz, V.A.; Rubis, A.Y. Change detection in remote sensing images using conditional adversarial networks. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *42*, 565–571. [\[CrossRef\]](#)
67. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; IEEE: Piscataway, NJ, USA, 2006; Volume 2, pp. 1735–1742.
68. Feng, Y.; Xu, H.; Jiang, J.; Liu, H.; Zheng, J. ICIF-Net: Intra-scale cross-interaction and inter-scale feature fusion network for bitemporal remote sensing images change detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4410213. [\[CrossRef\]](#)
69. Selvaraju, R.R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; Batra, D. Grad-CAM: Why did you say that? *arXiv* **2016**, arXiv:1611.07450.

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