CLDRNet: A Difference Refinement Network Based on Category Context Learning for Remote Sensing Image Change Detection

Ling Wan, Ye Tian, Wenchao Kang, and Lei Ma

***Abstract*—In recent years, change detection (CD) of optical remote sensing images has made remarkable progress through using deep learning. However, current CD deep learning methods are usually improved from the semantic segmentation models, and focus on enhancing the separability of changed and unchanged features. They ignore the essential characteristics of CD, i.e., different land cover changes exhibit different change magnitudes, resulting in limited accuracy and serious false alarms. To address this limitation, in this article, a category context learning-based difference refinement network (CLDRNet) based on our previous work is proposed. Considering the semantic content differences of heterogeneous land covers, a category context learning module is designed, which introduces a clustering learning procedure to generate an overall representation for each category, guiding the category context modeling. The clustering learning process is differentiable and can be integrated into the end-to-end trainable CD network, so it considers the semantic content differences from the CD perspective, thereby improving the CD performance. In addition, to address the magnitude differences of different land cover changes, a two-stage CD strategy is introduced. The two stages correspond to difference map learning and difference map refinement, aiming at ensuring high detection rates and revising false alarms, respectively. Finally, experimental results on three CD datasets verify the effectiveness of our CLDRNet in both visual and quantitative analysis.**

***Index Terms*—Category context learning (CCL), clustering learning (CL), difference map refinement (DMR), optical remote sensing image, change detection (CD).**

I. Introduction

CHANGE detection (CD) is a technique to qualitatively or quantitatively discriminate change information from multitemporal images acquired over the same geographical area at different times [1]. With the rapid development of satellite sensor technology, remote sensing image CD has been providing important support for many applications, such as environmental monitoring, agricultural survey, urban research, and disaster emergency management [2].

Generally, CD is a pixel-to-pixel task that takes multitemporal images as input and predicts "where" the change occurs [3]. As early as the 1960s, scholars have carried out research related to CD in remote sensing images. Traditional CD methods usually analyze the spectral information of the image first, and then select the threshold to accomplish the CD task. Representative methods include the arithmetic operations-based methods [4], [5], and the image transformation-based methods [6], [7], [8], [9]. Later, with the development of machine learning, model-based methods, such as support vector machine (SVM) [10], extreme learning machine [11], and decision tree [12] are applied to CD tasks. However, the abovementioned methods rely on expert domain knowledge. With the continuous improvement of the spatial resolution of remote sensing images, the features extracted by manual design are not enough to represent the key information of the images, and cannot meet the accuracy requirements.

Compared with the traditional models, the deep learning models can extract informative deep-level features. With the remarkable achievements of deep learning in remote sensing image interpretation, recently, many deep learning models have been applied to CD task, including stacked autoencoder, deep belief network, recurrent neural network (RNN), long short-term memory (LSTM) [13], generative adversarial network (GAN) [14], and transformer [15], [16], [17].

Although existing deep learning-based CD models have achieved remarkable detection results in some scenarios, their network structures are usually modified from the semantic segmentation models, ignoring the essential characteristics of CD [18]. Specifically, it is manifested in two aspects as follows. 1) CD task needs to consider the heterogeneity of multitype, multiscale, and multishape land covers by using heterogeneous feature extractors. However, existing "relation-attention" is usually achieved by global max or average pooling operations, which treat all pixels equally, resulting in insufficient discrimination of semantic content differences. 2) CD task requires to consider the different change magnitudes of different land cover changes. Because pre- and postchange images are usually acquired at different imaging angles, the geometrical properties of the targets in images are changed, which means the multitemporal images cannot be perfectly registered for all pixels. However, different land covers have different sensitivities to this

Manuscript received 14 August 2023; revised 25 September 2023; accepted 12 October 2023. Date of publication 25 October 2023; date of current version 2 January 2024. This work was supported by the Research Funding of Satellite Information Intelligent Processing and Application Research Laboratory under Grant 2022-ZZKY-JJ-17-02, Grant 2022-ZZKY-ZD-05-03, and Grant 2022-ZZKY-ZD-01-01. *(Corresponding author: Lei Ma.)*

Ling Wan, Ye Tian, Wenchao Kang, and Lei Ma are with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100039, China (e-mail: wanling15@mails.ucas.ac.cn; ye.tian@ia.ac.cn; xshzhdm@163.com; lei.ma@ia.ac.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3327340

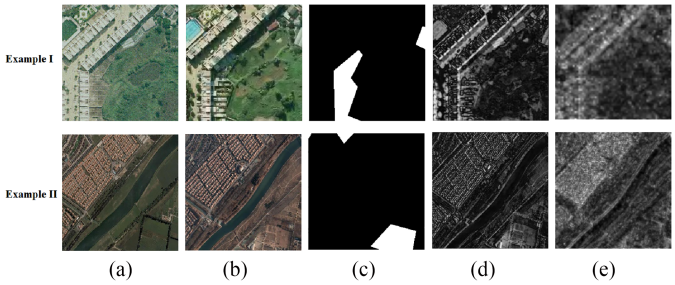

Fig. 1. Illustration of different land cover changes representing different change magnitudes. (a) Prechange image. (b) Postchange image. (c) Ground truth. (d) DM obtained by mean-differencing operation on spectral domain. (e) DM obtained by differencing operation on deep features (ResNet50 trained with ImageNet). Example I are aerial images with 0.5 m resolution. Example II are satellite images with 2 m resolution.

misalignment. As shown in Fig. 1, buildings usually exhibit high spatial heterogeneities, resulting in high difference indexes even for unchanged regions. And these difference indexes are even higher than some changed regions, meaning a fixed threshold cannot produce satisfactory CD performance. However, existing methods usually focus on enhancing the separability of changed and unchanged features, which ignore the effect of thresholds and lead to serious false alarms in CD results.

To tackle the abovementioned problems, in our previous work [18], a category-awareness-based difference-threshold alternative learning network (D-TNet) is proposed for remote sensing image CD. First, D-TNet introduces a category-awareness attention (CAA) mechanism to enhance the feature discrimination of heterogeneous land covers by learning a pixel-to-category relation. Second, D-TNet consists of a difference map (DM) learning path and a threshold map learning path, realizing self-adapting thresholds selection by assigning each pixel a unique threshold. However, D-TNet remains some problems should be reformulated: 1) The CAA mechanism is guided by the category-awareness descriptors (CA-DESCs), which are learned with conducting K-means on the feature maps pretrained on ImageNet. The K-means is computed only once, and the centroid-features are kept fixed during the network training. Thus, the learned CA-DESCs may lead to suboptimal adaptation performance. 2) The essence of two learning paths is to solve the problem of false detections and misdetections by assigning each pixel a unique threshold. However, the selection of change threshold is a complicated process, which is not only related to the land cover changes, but also related to the imaging conditions, such as illumination conditions and imaging angles. Therefore, the negative influence of the unknown imaging conditions is difficult to avoid, making it difficult to alleviate both false detections and misdetections problems simultaneously.

To address these limitations, in this article, a category context learning-based difference refinement network (CLDRNet) is proposed for remote sensing image CD. CLDRNet inherits the advantages of D-TNet and improves D-TNet in two aspects. First, to enhance the feature discriminations of heterogeneous land covers, a category context learning (CCL) module is proposed, which learns an overall representation of each category by clustering learning (CL) instead of K-means. The CL process is differentiable, and thus the mining of discriminative CA-DESCs and the CD task can be unified into one single framework. Therefore, the leaned CA-DESCs are adaptive to the CD task. Second, to address the magnitude differences of different land cover changes, a two-stage CD strategy is introduced. The two stages correspond to DM learning and DM refinement, aiming at ensuring high detection rates and revising false alarms, respectively.

The contributions of this article are summarized as follows.

1) A CL procedure is introduced to generate the CA-DESCs, which guides CCL. This process is differentiable, and can be integrated into the end-to-end trainable CD network.
2) A two-stage CD strategy is designed, corresponding to DM learning and DM refinement, aiming at ensuring high detection rates and revising false alarms, respectively.
3) Extensive experiments on three CD datasets verify the effectiveness of the CLDRNet in both visual and quantitative analysis.

II. Related Works

According to whether the feature is extracted manually, CD methods can be divided into traditional methods and deep learning-based methods.

A. Traditional Change Detection Methods

Generally, traditional CD methods can be roughly divided into postclassification comparison (PCC) methods and direct comparison methods.

PCC methods first classify the multitemporal image data, and then compare the pixel attributes of the classified images to obtain the CD results [19]. Many machine learning-based classifiers, including SVM [10] and decision tree [12] are used to classify the multitemporal image data. These methods can minimize the effect of different imaging conditions and obtain the from–to change types. However, they are highly dependent on the performance of the classifiers.

The direct comparison methods first generate difference image, and then analyze the difference image to determine whether the region has changed. The key procedures of these methods are difference image generation and difference image analysis. For difference image generation, commonly used methods include image algebra (e.g., differencing, ratioing, image regression, and change vector analysis [4]), image transformation [e.g., principal component analysis (PCA) [6], multivariate alteration detection [7], and slow feature analysis]. For difference image analysis, specific methods include thresholding (e.g., OTSU) and clustering (e.g., K-means and Markov random field [20]).

However, traditional CD methods rely on traditional manually designed features, which are insufficient to represent the informative features of images in various scenarios.

B. Deep Learning-Based Change Detection

With the remarkable achievements of deep learning, it shows the potential to deal with various changes in remote sensing.

In the beginning, due to the lack of large-scale CD datasets, unsupervised and self-supervised methods were first employed to determine whether a region has changed or unchanged. The unsupervised methods first use the available pretrained CNN model to obtain multiscale deep features, and then employ feature selection strategy to select the most discriminative features, and finally identify the changed pixels by comparing pixelwise features [21], [22]. However, the performance of these methods depends on the discriminability of the selected features. The self-supervised methods first employ traditional CD methods to generate pseudolabels, and then use the pseudolabels to train a CNN [23], [24], [25], realizing detecting changes. However, these methods are sensitive to the pseudolabels, and the accuracy of the pseudolabels directly effects the performance of CD.

With the increasing availability of CD datasets [26], [27], [28], [29], supervised methods have received much attention. These methods take advantage of deep learning models to extract discriminative features to improve the CD performance. Fully convolution network (FCN) is the mainstream architecture for CD tasks. According to the input form of image data, FCN-based methods can be divided into one-stream methods and two-stream methods. The one-stream methods first concatenate multitemporal images on channel dimension, and then send the stacked images into networks to accomplish CD. The two-stream methods take the Siamese network architecture as encoder, and then analyze the differences between the bitemporal features to obtain change maps. Typically, Daudt et al. [30] designed three FCN-based end-to-end CD architectures, they are fully convolutional early fusion (FC-EF), fully convolutional Siamese-concatenation (FC-Siam-conc), and fully convolutional Siamese-difference (FC-Siam-Di).

To improve the CD performance, researchers usually employ dense connection, multilevel aggregation and receptive field expansion to consider the multiscale characteristics of changed targets, and add attention mechanism to enhance the feature extraction capability. For example, Zheng et al. [31] proposed a cross-layer convolutional neural network, which adopts Unet structure as the backbone and embeds cross layer blocks to enhance the multiscale and multilevel feature extraction capabilities. Jiang et al. [32] proposed an efficient self-weighted spatial–temporal attention network, which introduces a multicore channel-aligning attention module to aggregate multiscale context information. Chen et al. [29] proposed a Siamese-based spatial–temporal attention neural network, which employs the spatial–temporal attention mechanism to exploit the global spatial–temporal relationship. Ke et al. [33] proposed a cross-Siamese CD network based on hierarchical-split attention, which captures cross-dimensional long-range relationship between channel with height, channel with width, and channel with channel. Zhang et al. [34] proposed an attentive differential high-resolution CD network, which designs the backbone with a differential pyramid module to extract multilevel and multiscale substantive changed features.

In addition, other state-of-the-art deep learning models, such as LSTM [13], GAN [14], and transformer [15], [16], [17] are also applied to detect changes. Chen et al. [13] proposed a deep Siamese convolutional multiple-layers recurrent neural network, which introduces a multiple-layers RNN module stacked by LSTM units to learn the spectral, spatial, and temporal feature representation. Liu et al. [14] proposed a superresolution-based CD network, which overcomes the resolution difference between bitemporal dates through adversarial learning. Recently, many scholars have applied transformer to the field of CD, and have achieved comparable performance with the CNN-based methods. Chen et al. [15] first employed transformer to realize CD, and verified the potential semantic representation capability of the transformer in differences extraction. Later, Bandara et al. [17] presented a transformer-based Siamese network architecture (ChangeFormer), which adopts hierarchically structured transformer encoder and multilayer perception decoder to capture multiscale long-range details. Liu et al. [16] proposed a CNN-transformer network with multiscale context aggregation (MSCANet), which combines the CNN and transformer architecture to capture and aggregate hierarchical context information.

However, existing deep learning-based methods simply treat the CD task as the semantic segmentation task, and ignore the essential characteristics of CD, thus limiting their applications.

III. Methodology

The overall architecture of CLDRNet is illustrated in Fig. 2. CLDRNet mainly contains four parts: backbone feature extraction (BFE) module, CCL module, DMG module, and difference map refinement (DMR) module.

CLDRNet starts with BFE module, which follows the Siamese structure, processing the multitemporal images in parallel and capturing the multitemporal feature maps. Then, the feature maps are fed into the CCL module, which uses the pixel-to-category relation to calibrate the feature maps response and obtains the category context features. After that, DMG module decodes the multitemporal category context features to calculate the DM. Finally, DMR module focuses on the most interesting category information to refine the change map obtained by DMG module.

A. Backbone Feature Extraction

The backbone network follows our previous work D-TNet, and adopt the ResNet50 with feature pyramid network (FPN) structure as the basic feature extractor branch.

The structure of the BFE module is shown in Fig. 3. First, for each temporal image (t represents the prechange or postchange), the multiscale features $\{\mathbf{E}_i^t | i = 2, 3, 4, 5\}$ are extracted from the last four stages of the modified ResNet50, whose sizes are 1/4, 1/8, 1/16, and 1/32 of the input size, and the channels is 256, 512, 1024, and 2048, respectively. Then, the FPN structure is implemented through top–down pathway and lateral connections, which comprehensively utilizes shallow high-level spatial information and deep strong semantic information to generate the enhanced multiscale features, denoted as $\{\mathbf{F}_i^t | i = 2, 3, 4, 5\}$. The size of $\mathbf{F}_i^t$ is $1/2^i$ of the input size, and the channel is 32.

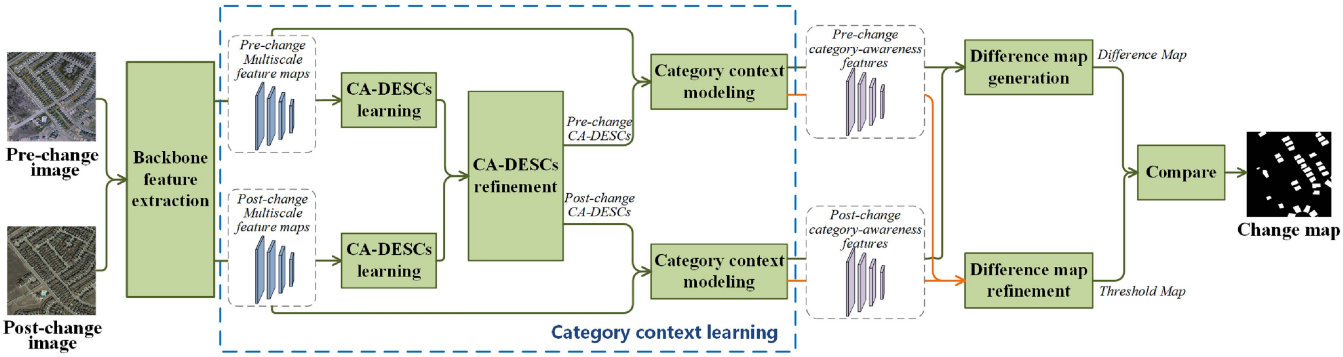

Fig. 2. Overview of the proposed CLDRNet architecture.

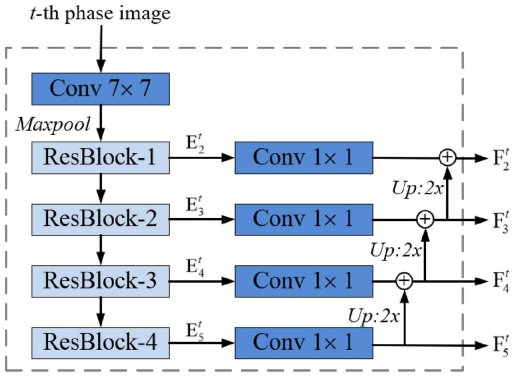

Fig. 3. Illustration of BFE module.

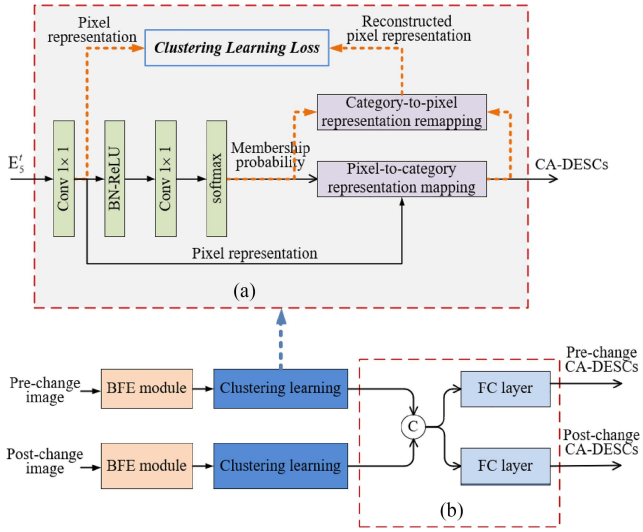

Fig. 4. Illustration of CA-DESC learning process. (a) CL. (b) CA-DESCs refinement.

B. Category Context Learning

To enhance the feature discrimination of heterogeneous land covers, in our previous work [18], a CAA mechanism is introduced, which first learns CA-DESCs to provide an overall representation of each category, and then uses CA-DESCs to guide the pixel-to-category relation learning. However, CA-DESCs are learned with conducting K-means on the feature maps pretrained on ImageNet. The K-means is computed only once, and the centroid-features are kept fixed during the network training, making the learned CA-DESCs may lead to suboptimal adaptation performance.

In this article, a CL is introduced, which uses convolution operation to simulate the K-means process. On the one hand, each image obtains its own CA-DESCs through convolution operation, which has self-adaptability. On the other hand, the CL process is differentiable, and thus the mining of discriminative CA-DESCs and the CD task can be unified into one single framework. In addition, a category context modeling (CCM) module is designed, which employs the transformer decoder to model the category context. It exploits the pixel-to-category relation by assigning the category relevant information to each pixel, thereby enhancing the representation of the feature maps.

1) CA-DESC Learning: We use convolution operations to predict the category of each pixel, and then calculate the centroid and the descriptor of each category by representation mapping, thereby obtaining the CA-DESCs of each image.

The CA-DESCs learning process is shown in Fig. 4. For each temporal image, the deep features extracted by BFE module is first encoded to reduce the feature dimension, generating the pixel representation. The specific formulas are as follows:

$$\mathbf{V}^t = \phi_{\mathbf{W}_E}(\mathbf{E}_5^t) \tag{1}$$

where $\phi_{\mathbf{W}_E}(\cdot)$ is a 1×1 convolutional layer, and $\mathbf{W}_E$ is learnable parameter. The pixel representation for the ith pixel is represented as $v_i^t \in \mathbb{R}^D$, where D is the number of feature dimensions.

Then, category prediction is performed on $\mathbf{V}^t$ through convolution layers and a softmax function

$$\mathbf{P}^t = \text{Softmax}\left(\phi_{\mathbf{W}_V}\left(\text{BR}(\mathbf{V}^t)\right)\right) \tag{2}$$

where $\text{BR}(\cdot)$ is a batch normalization operation followed by ReLU function; $\phi_{\mathbf{W}_V}(\cdot)$ is a 1×1 convolutional layer, and $\mathbf{W}_V$ is learnable parameter. The membership probability for the kth category is denoted as p_{ik}^t, and $\sum_{k=1}^{K} p_{ik}^t = 1$ for any pixel i and K is the number of groups. All membership probabilities form the matrix $\mathbf{P}^t$.

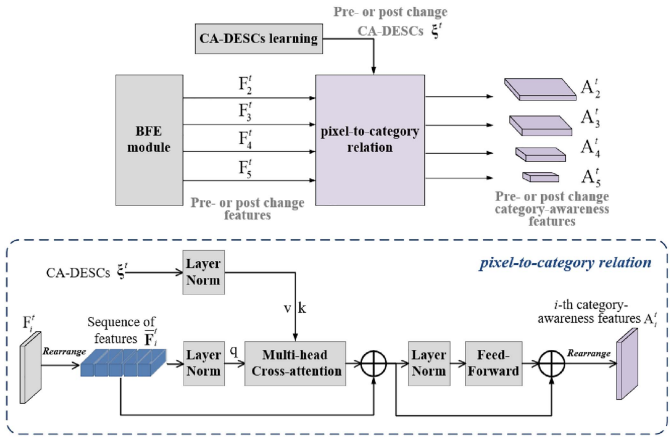

Fig. 5. Illustration of CCM process.

Next, the pixel representations $\mathbf{V}^t$ are mapped to category representation by averaging inside each category

$$c_k^t = \frac{\sum_{i=1}^{N} v_i^t \cdot p_{ik}^t}{\sum_{i=1}^{N} p_{ik}^t} \tag{3}$$

where N is the number of pixels. All category representations form the CA-DESCs, denoted as $\mathbf{C}^t \in \mathbb{R}^{K\times D}$.

The inverse mapping from category to pixel representations is achieved by assigning the category features to pixels using the soft pixel-category associations

$$\widehat{\mathbf{V}}^t = (\mathbf{P}^t)^T \times \mathbf{C}^t. \tag{4}$$

The soft pixel-category association is differentiable, thus the desirable CA-DESCs $\mathbf{C}^t$ can be obtained by minimizing the difference between the pixel representations $\mathbf{V}^t$ and the reconstructed pixel representations $\widehat{\mathbf{V}}^t$.

The advantages of the CA-DESCs learning are two folds. On the one hand, the CA-DESCs learning process is differentiable, and thus can be easily integrated into the end-to-end trainable CD network. On the other hand, the learnability of the CA-DESCs makes it adaptive for each image.

However, the CA-DESCs generated by the CL are independent of different temporals. To exploit the space–time relationships, the information in multitemporal descriptors is interacted and aligned semantically by a concatenate-split operation, as shown in Fig. 4(b). So far, the enhanced CA-DESCs can be obtained, which contain compact high-level category information, denoted as $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$.

2) Category Context Modeling: CCM module is proposed to model the category context, enhancing the discrimination of the pixel-space features extracted by the BFE module.

As shown in Fig. 5, the pixel-to-category relation is modeled by transformer decoder, which assigns the category relevant information to each pixel, making the refining pixel-space features more discriminative and reducing misjudgments caused by different imaging conditions.

The Transformer decoder is consisted of a multihead cross-attention (MCA) block and a feed-forward network (FFN). First, $\mathbf{F}_i^t$ is rearranged into the feature sequence $\overline{\mathbf{F}}_i^t$ with shape $[b, (\frac{H}{2^i} \times \frac{W}{2^i}), 32]$, where b is the minibatch size and $[H, W]$ is the input size. Then, inspired by [15], the MCA block treats $\overline{\mathbf{F}}_i^t$ as queries, and $\boldsymbol{\xi}^t$ as keys and values. For the hth head, the query, key, and value can be calculated as

$$\begin{cases} \mathbf{Q}_h = \mathrm{LN}(\overline{\mathbf{F}}_i^t)\mathbf{W}_h^q \\ \mathbf{K}_h = \mathrm{LN}(\boldsymbol{\xi}^t)\mathbf{W}_h^k \\ \mathbf{V}_h = \mathrm{LN}(\boldsymbol{\xi}^t)\mathbf{W}_h^v \end{cases} \tag{5}$$

where LN denotes LayerNorm [35]; $\mathbf{W}_h^q, \mathbf{W}_h^j, \mathbf{W}_h^v \in \mathbb{R}^{C\times d}$ are the learnable parameter matrices for linear projection; d is the dimension of each attention head and set to 64.

MCA block can be formulated as

$$\mathrm{MCA}(\overline{\mathbf{F}}_i^t, \boldsymbol{\xi}^t) = \mathrm{Concat}(\mathrm{head}_1, \ldots \mathrm{head}_{H_n})\mathbf{W}^O \tag{6}$$

and

$$\mathrm{head}_h = \mathrm{Att}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h) \tag{7}$$

$$\mathrm{Att}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h) = \sigma\left(\frac{\mathbf{Q}_h {\mathbf{K}_h}^T}{\sqrt{d}}\right)\mathbf{V}_h \tag{8}$$

where $\mathbf{W}^O \in \mathbb{R}^{hd\times C}$ are the learnable parameter matrices; H_n is the number of attention heads and set to 8; $\sigma(\cdot)$ is the softmax function conducted on the channel dimension. The feature sequence processed after MCA block is denoted as $\widehat{\mathbf{F}}_i^t$.

Then, FFN is operated on $\mathrm{LN}(\widehat{\mathbf{F}}_i^t)$. FFN consists of two linear projection layers and a GELU activation layer

$$\mathrm{FFN}\left(\mathrm{LN}(\widehat{\mathbf{F}}_i^t)\right) = \mathbf{W}_2 \cdot \mathrm{GELU}\left(\mathbf{W}_1 \cdot \mathrm{LN}(\widehat{\mathbf{F}}_i^t)\right) \tag{9}$$

where $\mathbf{W}_1, \mathbf{W}_2$ are the learnable weights. In addition, the residual connections are applied to enhance the stability of MCA and FFN.

Apply the pixel-to-category relation process to all levels features $\{\mathbf{F}_i^t | i = 2, 3, 4, 5\}$, and derive the category-awareness features, denoted as $\{\mathbf{A}_i^t | i = 2, 3, 4, 5\}$.

C. Change Detection Strategy

To tackle the different change magnitudes of different land cover changes, in our previous work [18], a threshold map learning path is introduced to alleviate error detections by assigning each pixel a unique threshold. However, the change threshold selection is a complicated process, which is not only related to the land cover changes, but also related to the imaging conditions. Nevertheless, the negative influence of the unknown imaging conditions is difficult to avoid, making it difficult to alleviate both false detections and misdetections problems simultaneously.

In this article, a two-stage CD strategy is introduced, as illustrated in Fig. 6. First, CLDRNet is trained with DMR module frozen, and generate the change map by comparing DM with 0.5. At this stage, CLDTNet is optimized under the premise of high detection rates, i.e., few misdetections. Next, DMR module is added to the training. Specifically, all parameters in CLDRNet are fixed except DMG and DMR modules. At this stage, the change map is obtained by comparing DM and the output of DMR module (denoted as threshold map, TM). That is, the determination of the changed region requires that the DM value

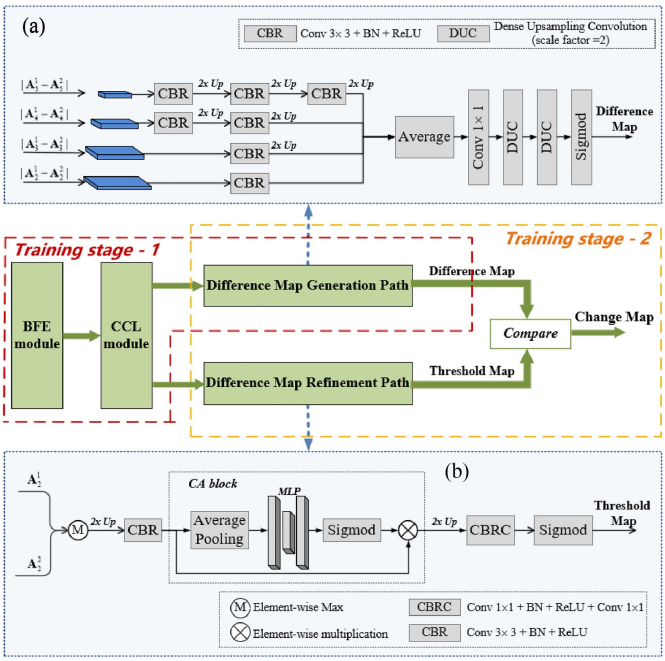

Fig. 6. Illustration of two-stage change detection strategy. (a) Architecture of DMG module. (b) Architecture of DMR module.

Algorithm 1: Training Process of CLDTNet.

Input: a set of image pairs and corresponding ground truth; parameters include learning rates [Stage-1: lr_1; Stage-2: lr_2], batch sizes [Stage-1: n_1; Stage-2: n_2], training epochs [Stage-1: Ep_1; Stage-2: Ep_2], number of groups K and number of feature dimensions D in Section III-B1.

Output: change maps

1) Stage— 1

for $Ep \in [1, Ep_1]$ **do**

Freeze the parameters in DMR module, and minimize the objective function in (14) using the back-propagation algorithm to train the BFE, CCL, and DMG modules.

end for

2) Stage— 2

for $Ep \in [1, Ep_2]$ **do**

Fix all parameters in CLDTNet except DMG and DMR modules, and minimize the objective function in (20) using the back-propagation algorithm.

end for

3) Determine the changed region where the DM value is greater than 0.5 and also greater than the TM value.

is greater than 0.5 and also greater than the TM value. Thus, the CD results can be improved by reducing the false detections. Training details are shown in Algorithm 1.

1) Difference Map Generation: The architecture of DMG module is shown in Fig. 6(a). A light decoder is operated on $\mathbf{A}_i^1$ and $\mathbf{A}_i^2$ to generate the DM. First, a difference feature map $\mathbf{T}$ is calculated by averaging all scale difference features that computed by the elementwise absolute of the subtraction of the two temporal category-awareness features

$$\mathbf{T} = \mathrm{Avg}\left\{F(\mathbf{X}_2) + \mathrm{UF}(\mathbf{X}_3) + \mathrm{UF}\left(\mathrm{UF}(\mathbf{X}_4)\right) + \mathrm{UF}\left[\mathrm{UF}\left(\mathrm{UF}(\mathbf{X}_5)\right)\right]\right\} \tag{10}$$

where $\mathbf{X}_i = |\mathbf{A}_i^1 - \mathbf{A}_i^2|$; $F(\cdot)$ denotes a 3×3 convolution—batch normalization—ReLU layer with 16 output channels; $\mathrm{UF}(\cdot)$ is $F(\cdot)$ function followed by a bilinear upsampling with a scale factor of 2.

Then, the DM is generated with a 1×1 convolutional layer, two dense upsampling convolutional layers [36] and a sigmoid function.

2) Difference Map Refinement: The architecture of DMR module is shown in Fig. 6(b). First, elementwise maximization operation is applied to $\mathbf{A}_2^1$ and $\mathbf{A}_2^2$, defined as

$$\overline{\mathbf{A}} = \mathrm{Max}(\mathbf{A}_2^1, \mathbf{A}_2^2). \tag{11}$$

Elementwise maximization operation makes the feature maximum tensor contain the most interesting category information within the temporal domain, which is useful for analyzing the sensitivity of different land covers to pseudochanges caused by factors, such as imaging conditions. In addition, only the category-awareness features with the highest spatial resolution are used, because high spatial resolution features preserve more location information, which is helpful for refining the boundaries of CD results.

Then, an upsampling layer, a CBR block, a channel attention (CA) block and a sigmoid function are sequentially operated on $\overline{\mathbf{A}}$, obtaining the threshold features

$$\mathbf{Th} = \delta\left(\mathrm{CA}\left(F\left(U(\overline{\mathbf{A}})\right)\right)\right) \tag{12}$$

where $U(\cdot)$ denotes the bilinear upsampling with a scale factor of 2; $F(\cdot)$ is a 3×3 convolution—batch normalization—ReLU layer with 16 output channels; $\delta(\cdot)$ is the sigmoid function; $\mathrm{CA}(\cdot)$ is the channel attention operation, defined as

$$\mathrm{CA}(F_{\mathrm{in}}) = \delta[\mathrm{CRC}(\mathrm{AvgPool}(F_{\mathrm{in}}))] \otimes F_{\mathrm{in}} \tag{13}$$

where $\mathrm{AvgPool}(\cdot)$ aggregates spatial information by average pooling operator, and $\mathrm{AvgPool}(F_{\mathrm{in}}) \in \mathbb{R}^{1\times1\times C}$; $\mathrm{CRC}(\cdot)$ is a FC–ReLU–FC layer with four hidden channels; $\otimes$ is the elementwise multiplication operation; $\delta(\cdot)$ is the sigmoid function.

Finally, an upsampling layer, a CBRC block (1×1 convolution—batch normalization—ReLU layer –1×1 convolution with 16 hidden channels) and a sigmoid function are sequentially operated on $\mathbf{Th}$, producing the TM with one channel.

3) Loss Functions. Stage one: At stage one, DM and CA-DESCs are optimized together by minimizing a hybrid loss

$$L_1 = L_{\mathrm{DM}} + L_{\mathrm{DA}}. \tag{14}$$

As described in Section III-B1, L_{DA} measures the difference between the pixel representations and the reconstructed pixel representations

$$L_{\mathrm{DA}} = \sum_{t=1}^{2}\left[\frac{1}{N}\sum_{i=1}^{N}||v_i^t - \widehat{v}_i^t||^2\right] \tag{15}$$

TABLE I
ANALYSIS OF THE FOUR CASES OF TP, TN, FN, AND FP

True positive (TP)	G_n=1 and $P_n > 0.5$	The correctly detected changed regions should not be destroyed, and thus DM value should be kept greater than 0.5 and DM value should be greater than TM value.
True negative (TN)	G_n=0 and $P_n \leq 0.5$	The correctly detected unchanged regions should not be destroyed, and thus DM value should be kept smaller than 0.5 or DM value smaller than TM value.
False negative (FN)	G_n=1 and $P_n \leq 0.5$	Misdetections should be further optimized, making the DM value be greater than 0.5 and be greater than TM value.
False positive (FP)	G_n=0 and $P_n > 0.5$	False detections should be corrected, making the TM value be greater than DM value.

where t represents pre- or postchange image; N is the total number of pixels; $||\cdot||^2$ is the Euclidean distance.

L_{DM} composes of cross-entropy loss, contrastive loss, and Tversky loss. The formula is as follows:

$$L_{\text{DM}} = L_{\text{ce}} + L_{\text{tver}} + L_{\text{contra}} \tag{16}$$

and

$$L_{\text{ce}} = -\frac{1}{N}\sum_{n=1}^{N}\left[G_n \log P_n + (1-G_n)\log(1-P_n)\right] \tag{17}$$

$$L_{\text{contra}} = \sum_{n=1}^{N}\left[\frac{1}{2}(1-G_n)(P_n)^2 + \frac{1}{2}G_n\{\max(0, m-P_n)^2\}\right] \tag{18}$$

$$L_{\text{tver}} = 1 - \frac{\sum_{n=1}^{N} G_n P_n}{\sum_{n=1}^{N} G_n P_n + \alpha\sum_{n=1}^{N} G_n(1-P_n) + (1-\alpha)\sum_{n=1}^{N}(1-G_n)P_n} \tag{19}$$

where L_{ce} is the cross-entropy loss. G_n and P_n are the label and DM value for the nth pixel, respectively.

L_{contra} is the contrastive loss. The changed pixels affect the loss function only when their DM values are within the margin m. We set $m = 0.5$ to train the DM value of changed and unchanged pixel separated from 0.5.

L_{tver} is the Tversky loss [37]. α is a hyperparameter that controls the tradeoff between recall and precision. To ensure a high detection rate of the changed regions at stage one, α is set to 0.9.

Stage two: At stage two, DMG and DMR modules are jointly trained to reduce false alarms without breaking the correctly detected changed regions

$$L_2 = L_{\text{DM}} + L_{\text{TM}}. \tag{20}$$

L_{DM} is calculated as (16). For L_{TM}, according to the four cases of true positive (TP), true negative (TN), false negative (FN), and false positive (FP) of the CD results of the stage one, the specific analysis is shown in Table I.

The following can be concluded from Table I.

1) For the changed pixel, DM value should be greater than 0.5 and greater than TM value.
2) For the unchanged pixel, DM value should be smaller than 0.5 or smaller than TM value.
3) To revise the false alarms, let the TM value greater than DM value.
4) We further optimize the DMG module to make the DM value greater than 0.5.

Based on the abovementioned analysis, L_{TM} is calculated by referring to the form of cross-entropy loss and contrast loss. The formula is defined as

$$L_{\text{TM}} = L_C + L_{\text{UC}} + L_{\text{FP}} + L_{\text{FN}} \tag{21}$$

and

$$\begin{cases} L_C = \frac{1}{N}\sum_{n=1}^{N}\left[G_n \cdot \left(\{\max(0, PT_n - P_n)^2\} + \{\max(0, m-P_n)^2\}\right)\right] \\ L_{\text{UC}} = \frac{1}{N}\sum_{n=1}^{N}\left[(1-G_n)\{\max(0, P_n - PT_n)^2\}\right] \\ L_{\text{FP}} = \frac{1}{N}\sum_{n=1}^{N}\left[-(1-G_n)\cdot I_{[P_n>0.5]}\cdot\log(\frac{PT_n - P_n + 1}{2})\right] \\ L_{\text{FN}} = \frac{1}{N}\sum_{n=1}^{N}\left[G_n \cdot I_{[P_n\leq 0.5]}\cdot(-\log(P_n))\right] \end{cases} \tag{22}$$

where PT_n is the TM value for the nth pixel; $I_{[\cdot]}$ is a characteristic function that returns 1 if the condition is met, and 0 otherwise.

The final change map is calculated as CM $=$ (DM > 0.5) and (DM $>$ TM).

IV. EXPERIMENTS

A. Datasets Description

Learning, vision, and remote sensing change detection (LEVIR-CD) [29] is a building CD dataset covering building types in various urban scenarios, including residential, commercial, and industrial areas. This dataset contains 637 pairs of optical images collected from Google Earth platform with 0.5 m resolution and 1024 $\times$ 1024 size. We follow its default dataset split and cut the images into 512 $\times$ 512 size nonoverlapping patches, deriving 1780/256/512 pairs of patches for training/validation/test.

Building change detection dataset (BCDD) [27] is a building CD dataset, covering an area where a 6.3-magnitude earthquake occurred in February 2011 and rebuilt in the following years. This dataset contains two aerial images with 0.3 m resolution

and 32 507 × 15 354 size. We process the images into 512 × 512 size nonoverlapping patches, and randomly divided them into three parts: 924/318/585 for training/validation/test.

Change detection dataset (CDD) [26] is a public CD dataset covering various change information, including cars, tanks, forests, roads, buildings, etc. This dataset contains 16 000 pairs of optical images collected from Google Earth platform with 3–100 cm resolution and 256 × 256 size. We follow its default dataset split and derive 10 000/3000/3000 pairs of patches for training/validation/test.

B. Experimental Setup

1) Evaluation Metrics: Five evaluation metrics are adopted to make quantitative analysis: Recall, precision, F1-score, intersection over union (IoU) of the change category, and overall accuracy (OA). The definitions are as follows:

$$\begin{aligned} \text{Precision} &= \text{TP}/(\text{TP}+\text{FP}) \\ \text{Recall} &= \text{TP}/(\text{TP}+\text{FN}) \\ \text{F1} &= 2\cdot\text{Precision}\cdot\text{Recall}/(\text{Precision}+\text{Recall}) \\ \text{IoU} &= \text{TP}/(\text{TP}+\text{FN}+\text{FP}) \\ \text{OA} &= (\text{TP}+\text{TN})/(\text{TP}+\text{TN}+\text{FN}+\text{FP}) \end{aligned} \tag{23}$$

where TP, FP, FN, and TN are the numbers of true positives, false positives, false negatives, and true negatives, respectively.

2) Implementation Details: All experiments are implemented on a computing server equipped with an Intel Core i9-10900X and an NVIDIA GTX 3090 GPU. Our model is conducted under the PyTorch deep-learning framework, and normal data augmentation methods are employed to the training data, including flipped horizontally and vertically. The stochastic gradient descent with momentum is applied for optimization, and the momentum is set to 0.99 and the weight decay is set to 0.0005. The learning rates are initially set to 0.01 and 0.001 for the first and the second training stages, respectively. We adopt the poly policy to gradually reduce the learning rate, that is, the learning rate is multiplied by $(1-\frac{\text{epoch}}{Ep_n})^{\text{power}}$ with power $= 0.8$, where Ep_n is the total training epochs, and we set $Ep_1 = 200$, $Ep_2 = 100$ for the first and the second training stages, respectively. The batch size is set to 8 for LEVIR-CD and BCDD, and 32 for CDD. In addition, in the CCL module, the number of groups K is set to 8 according to D-TNet [18], and the feature dimensions D is set to 3. Validation is conducted after each training epoch, and the best model on the validation set is used for evaluation on the test set.

C. Comparison to State of the Art

Eight methods are implemented for comparison purposes: the purely convolutional-based methods, including FC-EF [38], FC-Siam-Di [38], and FC-Siam-Conc [38], and related state-of-the-art methods in recent years, including the deep metric learning change detection network (CDNet) [14], the bitemporal image transformer-based model (BIT S4) [39], the transformer-based Siamese network (ChangeFormer) [17], a CNN-transformer MSCANet [16], and our previous work D-TNet [18].

FC-EF [38]: A U-Net structure-based method that first concatenates bitemporal images and then feeds them to the FCN.

FC-Siam-Di [38]: A Siamese FCN-based method that first extracts each temporal features, and then feeds the feature difference to the decoder.

FC-Siam-Conc [38]: A Siamese FCN-based method that first extracts each temporal features, and then feeds the connected features to the decoder.

CDNet [14]: A deep metric learning-based method that employs the stacked attention module to extract features, and then uses the metric learning-based change decision module to obtain the change map.

BIT S4 [39]: A transformer-based method that first employees the transformer encoder to express the bitemporal images as context-rich tokens, and then uses the transformer decoder to feed the tokens back to the pixel-space to refine the original features.

ChangeFormer [17]: A transformer-based Siamese network that uses the hierarchical transformer encoder to extract multiscale features, and then employs the multilayer perception decoder to fuse the multilevel feature differences and predict the CD mask

MSCANet [16]: A CNN-transformer network that first uses the CNN-based feature extractor to capture hierarchical features, and then employs the transformer-based block to encode and aggregate context information.

D-TNet [18]: A D-TNet that introduces a CAA mechanism to enhance the feature discrimination and uses difference-threshold learning to realize self-adapting thresholds selection.

For the comparison methods, we use their public codes with default hyperparameters.

1) Qualitative Analysis: Figs. 7, 8, and 9 provide the visualization comparisons of different methods for LEVIR, BCDD, and CDD datasets, respectively. For a more intuitive comparison of the results, we use white for TP, black for TN, red for false detections, and green for misdetections.

As shown in Fig. 7, for the LEVIR-CD dataset, CLDRNet achieves better performance than the competing methods. For example, as shown in Fig. 7(a), the newly constructed buildings in the postchange image have similar spectral characteristics with the surrounding environment, and we can see that the fully convolutional-based methods, including FC-EF, FC-Siam-Di, FC-Siam-Conc, suffer from serious misdetections, thus their feature extraction abilities are insufficient to distinguish the change information. In addition, the multitemporal images contain dense building changes, and there are some adhesions in CDNet. We can also see that there are some false alarms in BIT S4, ChangeFormer, MSCANet, and D-TNet. In contrast, the proposed CLDRNet can alleviate the abovementioned phenomena to a certain extent. Observing Fig. 7(b), the newly constructed buildings in the postchange image exhibit different spectral and size characteristics, especially the buildings in the red circle have similar spectral features to the surrounding land, leading to misdetections in most methods. However, our proposed CLDRNet effectively avoids the misdetection phenomenon, benefiting from the CCL that can distinguish multiscale and multishape buildings. Observing Fig. 7(c), because the feature differences

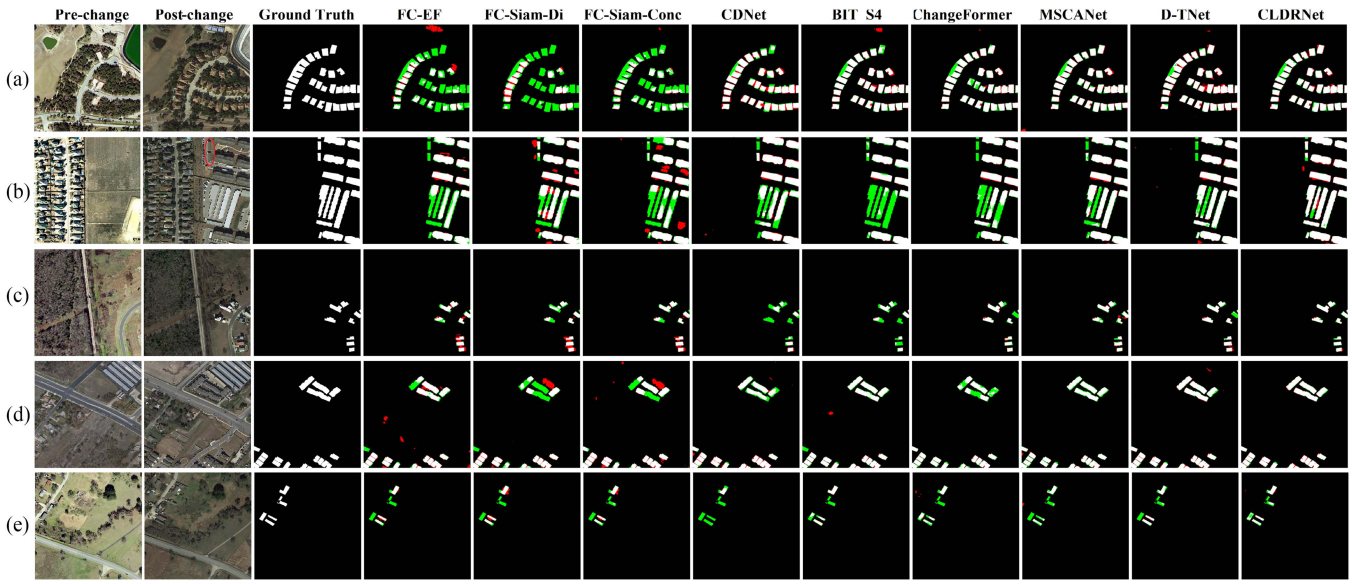

Fig. 7. Change detection results on LEVIR-CD. Legend: white for TP, black for TN, red for false detections, and green for misdetections.

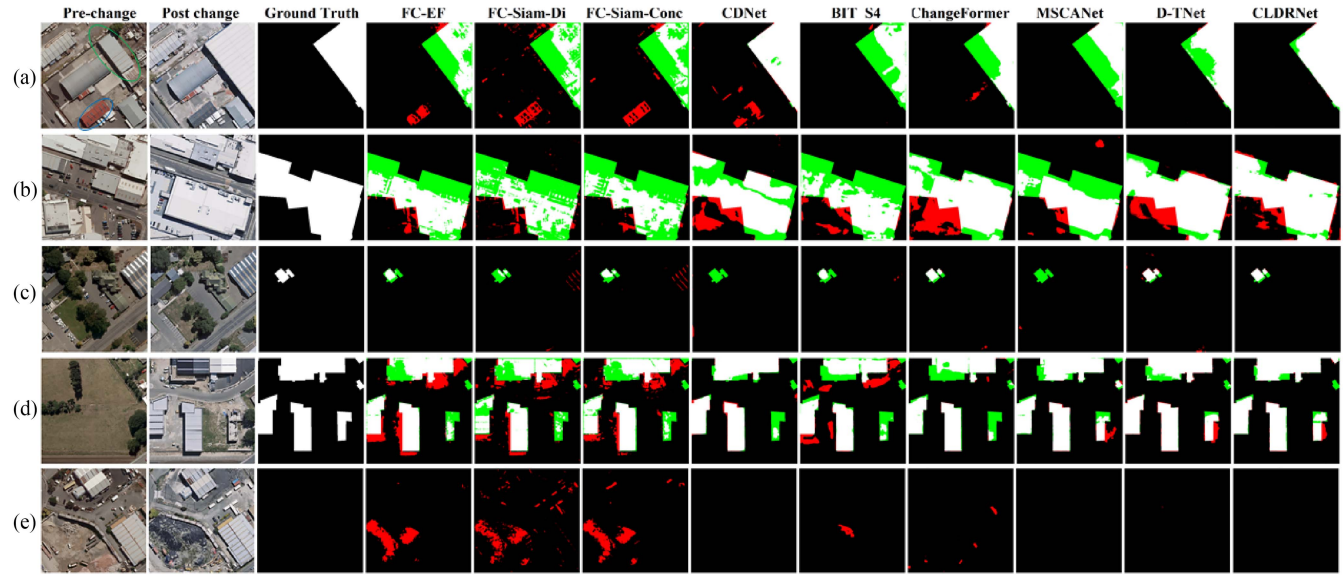

Fig. 8. Change detection results on BCDD. Legend: white for TP, black for TN, red for false detections, and green for misdetections.

of the changed regions between multitemporal images are obvious, all methods can successfully detect the building changes. However, our CLDRNet is closest to the ground truth, reflecting that CLDRNet is robust to the pseudochanges caused by building shadows. As shown in Fig. 7(d), there are multiscale, dense and irregular shape building changes between multitemporal images. In comparison, CLDRNet shows the best visual performance with fewer error detections, complete building structures and clear building boundaries. As shown in Fig. 7(e), the multitemporal images contain small size building, as well as surrounding car changes, making it difficult to accurately detect building changes. We can see that our CLDRNet obtains the best result with the fewest misdetections in comparison.

Fig. 8 shows the CD results of different methods on BCDD dataset. Compared with LEVIR-CD, the images in BCDD have a higher resolution, and at the same time, the size of the city buildings is larger, making the texture of the roof to be relatively clear. Thus, the CD results are prone to contain salt-and-pepper noise for unchanged buildings, and incomplete building structures for changed buildings. However, in comparison, we observe that our CLDRNet achieves the most satisfactory results with fewer error detections and higher internal compactness. Specifically, observing Fig. 8(a), there contains a building to building changed areas but with similar spectral and texture features (in green circle), and an unchanged area but with the roof color changed (in blue circle), making it difficult to detect the changes of interest.

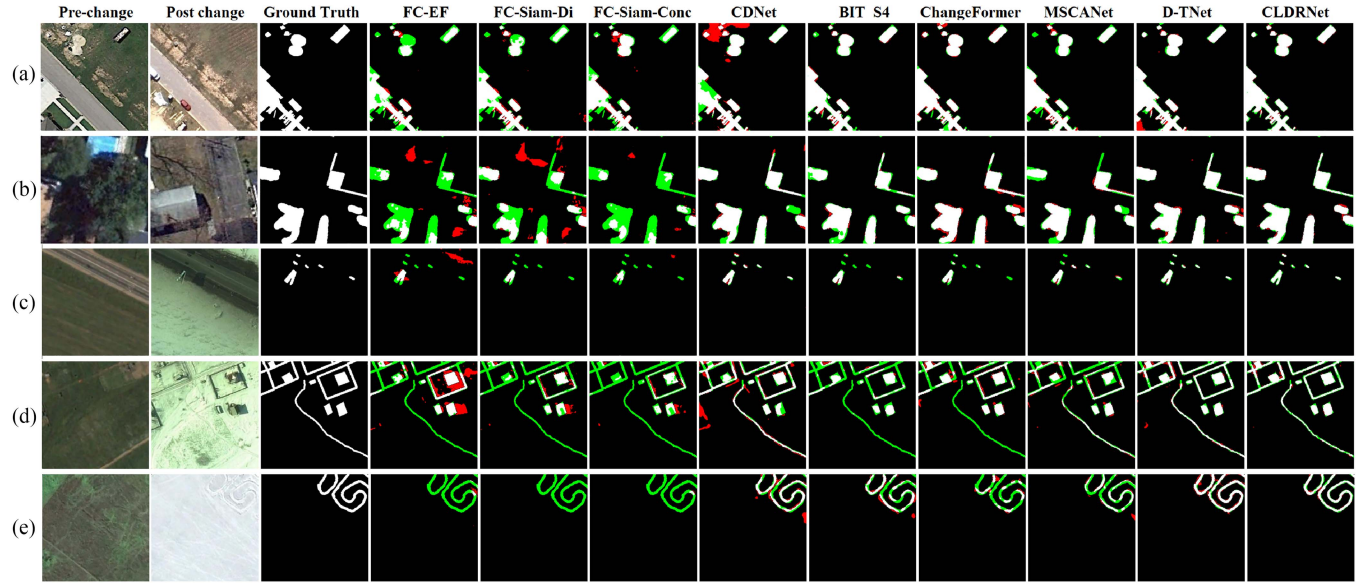

Fig. 9. Change detection results on CDD. Legend: white for TP, black for TN, red for false detections, and green for misdetections.

We can see that the fully convolutional-based methods suffer from serious false detection and misdetections, and the CDNet contains false detections. In addition, although BIT S4, ChangeFormer, and MSCANet have the semantic context modeling capabilities, they are difficult to identify the change information with the same semantic. D-TNet and CLDRNet benefit from the category-awareness information learning, which can distinguish multiscale and multishape buildings. However, CLDRNet is more satisfactory than D-TNet with more complete building structures, since the centering learning process makes the CA-DESCs more representative and adaptive. As shown in Fig. 8(b), all methods can successfully detect the main body of the changed area. But comparatively, our CLDRNet contains fewer error detections, which effectively overcomes the problem that the spectral characteristics of the newly constructed buildings are close to the surrounding land. Observing Fig. 8(c) and (d), the multitemporal images contain seasonal changes, atmospheric condition changes and nonbuilding changes (such as cars). Our CLDRNet obtains the most satisfactory result, successfully overcoming the interference of pseudochanges, and other comparison methods exist error detections in the building areas. As shown in Fig. 8(e), there is no change between the multitemporal images, and our CLDRNet has no false alarm, reflecting that the CLDRNet is robust to the appearance differences caused by different imaging conditions.

Fig. 9 shows several challenging and representative results on CDD dataset. Compared with LEVIR-CD and BCDD, CDD is also concerned with the changes caused by cars, tanks, forests, roads, etc., in addition to building changes. As shown in Fig. 9(a), because the changed areas have obvious appearance differences between multitemporal images, most methods can successfully detect the changes. But in contrast, CLDRNet has more accurate result with higher internal compactness. Observing Fig. 9(b), we can see that our CLDRNet can effectively counter against the pseudochanges caused by seasonal differences, that is, CLDRNet has fewer false alarms and fewer misdetections compared with other eight methods. As shown Fig. 9(c), there are obvious differences in imaging conditions between multitemporal images, and the sizes of the changed cars in postchange image are very small, making it difficult to accurately detect the changes. CLDRNet detects all changed targets, while the competing methods all suffer from misdetections. In addition, jointly observing Fig. 9(a) and (c), our approach is able to detect both small and large changed cars, reflecting that CLDRNet focuses on the multiscale characteristics of remote sensing images. Observing Fig. 9(d), the newly constructed buildings in postchange image have obvious shadow phenomenon, which has a greater impact on the fully convolutional-based methods method, and other methods have certain robustness to this. In addition, jointly observing Fig. 9(d) and (e), there are very narrow road changes. We can see that the fully convolutional-based methods are completely useless for this phenomenon, which indicates that their feature extraction ability is insufficient to distinguish narrow change information. For the transform-based methods, including BIT S4, ChangeFormer, and MSCANet, the CD results are incomplete and contain salt-and-pepper noises. This may be due to the loss of spatial details during tokens embedding and feature reconstruction. In comparison, D-TNet and CLDRNet are more satisfactory, because their TM maps are generated from the feature maps with the highest spatial resolution, which have a refinement effect on the boundaries. However, the TM map generation process of CLDRNet considers the misdetection and false detection problems separately, making the CD map more accurate.

2) Qualitative Analysis: Table II provides the quantitative analysis on three test sets. We can see that our CLDRNet always outperforms the competing methods in F1 and IoU scores, which is consistent with the visual inspection. Specifically, the fully convolutional-based methods, including FC-EF, FC-Siam-Di, FC-Siam-Conc, have lower recall and precision scores compared with other methods. ChangeFormer and MSCANet have relatively high F1 and IoU scores, but their precision

TABLE II
QUANTITATIVE ANALYSIS ON THREE DATASETS

	LEVIR-CD						
Method	Params.(M)	FLOPs(G)	Recall	Precision	F1	IoU	OA
FC-EF	1.29	14.29	79.85	72.20	75.83	61.07	97.41
FC-Siam-Di	1.29	18.87	80.99	77.83	79.38	65.81	97.86
FC-Siam-Conc	1.47	21.29	80.54	80.42	80.48	67.33	98.01
CDNet	16.19	286.32	91.33	81.52	86.15	75.67	98.50
BIT S4	3.39	34.85	89.37	89.24	89.31	80.68	98.92
ChangeFormer	39.13	811.15	88.83	92.48	90.61	82.84	99.06
MSCANet	15.66	59.20	87.91	91.22	89.54	81.06	98.95
D-TNet	35.40	92.49	90.62	88.75	89.67	81.28	98.94
CLDRNet	22.92	46.30	91.40	90.17	90.78	83.12	99.05
	BCDD						
Method	Params.(M)	FLOPs(G)	Recall	Precision	F1	IoU	OA
FC-EF	1.29	14.29	71.28	68.09	69.65	53.43	97.76
FC-Siam-Di	1.29	18.87	65.61	57.54	61.31	44.21	97.01
FC-Siam-Conc	1.47	21.29	62.71	67.72	65.11	48.28	97.57
CDNet	16.19	286.32	82.53	87.10	84.75	73.54	98.87
BIT S4	3.39	34.85	81.91	86.15	83.97	72.38	98.62
ChangeFormer	39.13	811.15	80.80	83.25	82.00	69.50	98.72
MSCANet	15.66	59.20	82.17	90.55	86.15	75.68	99.05
D-TNet	35.40	92.49	87.48	88.92	88.19	78.88	99.15
CLDRNet	22.92	46.30	89.91	90.05	89.98	81.78	99.28
	CDD						
Method	Params.(M)	FLOPs(G)	Recall	Precision	F1	IoU	OA
FC-EF	1.29	3.57	75.57	85.10	80.06	66.74	95.56
FC-Siam-Di	1.29	4.72	78.99	86.85	82.73	70.55	96.11
FC-Siam-Conc	1.47	5.32	75.97	90.83	82.74	70.56	96.26
CDNet	16.19	71.58	91.22	90.07	90.64	82.89	97.78
BIT S4	3.39	8.71	91.73	95.69	93.66	88.07	98.53
ChangeFormer	39.13	202.79	93.94	94.53	94.23	89.10	98.64
MSCANet	15.66	14.80	92.46	94.56	93.50	87.79	98.48
D-TNet	35.40	23.13	96.75	94.06	95.39	91.18	98.90
CLDRNet	22.92	11.57	96.16	95.64	95.90	92.12	99.03

All the scores are described in percentage (%).

values are much higher than their recall values. However, in the CD task, recall is usually more important. BIT S4 has similar problems in BCDD and CDD dataset. CDNet is also unable to strike a balance between recall and precision. The proposed CLDRNet is an improvement of our previous work D-TNet, and they achieve a relative balance between false detections and misdetections, compared with other methods. However, CLDRNet is more satisfactory in terms of recall/prescion/F1 values, benefiting from the CL and DM refinement processes.

3) Model Efficiency: As shown in Table II, the number of parameters (Params.) and the floating-point operations per second (FLOPs) for each image are used to compare the model efficiency of different methods. We can see that the fully convolutional-based methods have fewer parameters and lower FLOPs, but their CD performance is limited. Among the other six methods, CLDRNet has a moderate number of parameters and relatively low FLOPs. Therefore, our CLDRNet achieves the best F1/OA scores at an acceptable computational cost.

D. Discussion

1) Effect of CA-DESC Learning: We perform ablation on the effect of CL process by replacing it with the centroid-features learning process described in D-TNet. That is, the comparison method obtains CA-DESCs through feature maps extraction with ResNet50 pretrained on ImageNet, PCA, and K-means process.

As shown in Table III, we can see that the CA-DESCs generated by CL process are more effective for CD. The CL process unifies the CA-DESCs prediction into the CD framework, which considers the semantic content differences from the CD perspective, making the learned CA-DESCs more adaptable to CD task. Therefore, it can effectively improve the CD performance.

In addition, we display the high-level category-awareness features in Fig. 10, where yellow and blue color denote higher and lower attention values, respectively. We can see that the category-awareness maps obtained with the CL are more compact and have higher information density than the ablation method, indicating that the CA-DESCs generated by the CL can promote the effective extraction of change information, which is consistent with theoretical analysis.

2) Effect of DMR: We perform ablation on the effect of DMR by removing DMR module from the CLDRNet, i.e., the CD result of Stage 1. In addition, because CLDRNet is an improved method based on D-TNet, we also perform ablation by replacing the DMG module with the TMG module in D-TNet, and the CD map is obtained by comparing DM and TM.

TABLE III
ABLATION STUDY OF THE CL PROCESS

	LEVIR-CD			BCDD			CDD		
CL	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
✗	90.78	90.19	90.49	88.12	89.52	88.81	96.46	95.17	95.81
✓	91.40	90.17	90.78	89.91	90.05	89.98	96.16	95.65	95.90

All the scores are described in percentage (%).

TABLE IV
ABLATION STUDY OF THE DMR

	LEVIR-CD			BCDD			CDD		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
remove DMR	93.92	86.32	89.96	91.33	85.89	88.53	97.58	93.69	95.60
with TMG	89.48	91.81	90.63	88.05	90.16	89.09	95.83	95.92	95.87
with DMR	91.40	90.17	90.78	89.91	90.05	89.98	96.16	95.65	95.90

All the scores are described in percentage (%).

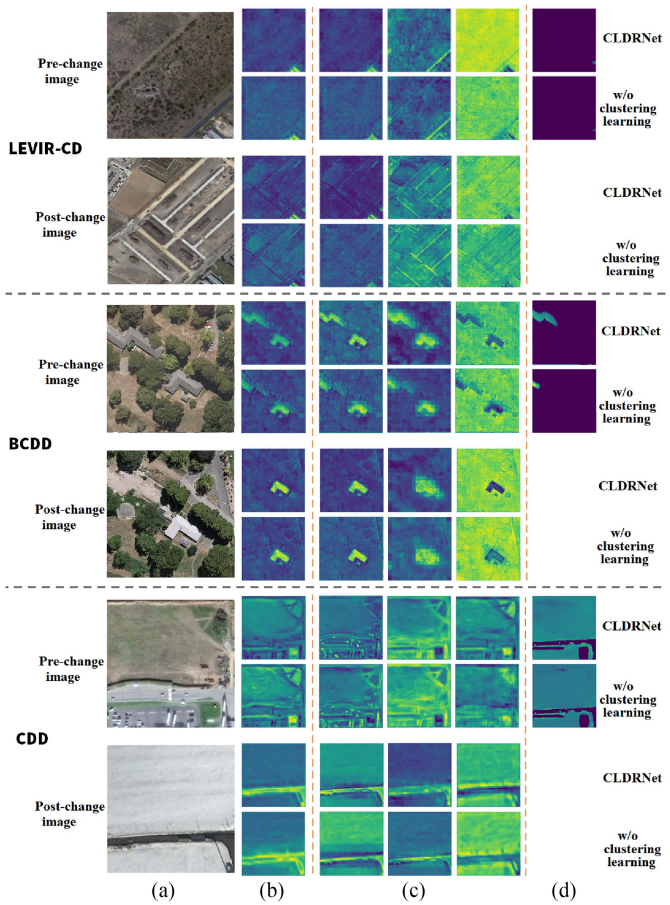

Fig. 10. Category-awareness feature visualization on the LEVIR-CD, BCDD, and CDD test sets. (a) Pre- or postchange image. (b) High-level category-awareness feature map with PCA. (c) Three selected high-level category-awareness feature maps. (d) DM.

As shown in Table IV, we can see that both TMG and DMG can improve the CD performance in terms of the balance between false alarms and misdetections. For TMG, although it aims at correcting misdetections and false alarms, it obtains a lower recall value than "remove DMR." The selection of change threshold is a complicated process, making it difficult to alleviate both false detections and misdetections problems simultaneously. In particular, the recall of "remove DMR" is high, so it is difficult to further improve. Therefore, TMG improves the main problem of "remove DMR," i.e., false alarms. However, DMG is more effective than TMG. First, DMG achieves a better balance between recall and precision than TMG. Second, the changed regions account for a small proportion than the unchanged regions, so a high precision usually leads to a high F1-score within a certain range. However, in CD tasks, recall is usually more important. We can observe that DMG obtains higher Recall and F1-score than TMG. Thus, although DMG does not specifically correct misdetections, it corrects false alarms while constraining the correctly detected changed areas, which is more effective than TMG that corrects false alarms and misdetections at the same time.

In addition, Fig. 12 provides the visual comparisons to make a deeper understanding of the DMG module. On the one hand, we can see that TMG corrects the false alarms by reducing the DM value to less than 0.5, while DMG corrects the false alarms by making the TM value greater than DM value. In other words, the DMG assigns high TM values to the pseudochanged regions, which is more effective in refining false alarms than the TMG. This is because the changed and the pseudochanged regions usually exhibit lower TM values for TMG and higher TM values for DMG. The loss function of TMG optimizes $P_n - PT_n$ for changed regions and optimize $PT_n - P_n$ for unchanged regions, so the optimization procedure will expand the distance between P_n and PT_n, i.e., the distance between DM and TM values. The loss function of DMG uses the contrastive loss to prevent the correctly detected changed and unchanged regions from being destroyed, and optimize $PT_n - P_n$ for false alarms, so the optimization procedure may pay more attention to the indistinguishable samples. On the other hand, TMG corrects false alarms and misdetections at the same time, which may lead to confusion, for example, in Fig. 12(b), the correctly

TABLE V
PARAMETER ANALYSIS OF FEATURE DIMENSIONS D ON THREE DATASETS

	LEVIR-CD			BCDD			CDD		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
$D = 2$	90.44	90.91	90.67	88.63	85.95	87.27	96.50	95.29	95.89
$D = 3$	91.40	90.17	90.78	89.91	90.05	89.98	96.16	95.65	95.90
$D = 4$	90.71	90.22	90.46	88.23	89.28	88.75	96.63	94.86	95.74
$D = 5$	90.91	90.57	90.74	88.50	88.89	88.70	96.39	95.43	95.91
$D = 6$	91.34	89.42	90.37	88.24	89.48	88.86	96.60	94.85	95.72
$D = 7$	91.72	88.80	90.24	88.04	88.87	88.45	95.81	95.31	95.56
$D = 8$	91.30	88.86	90.06	87.88	88.47	88.17	95.56	95.66	95.61

All the scores are described in percentage (%).

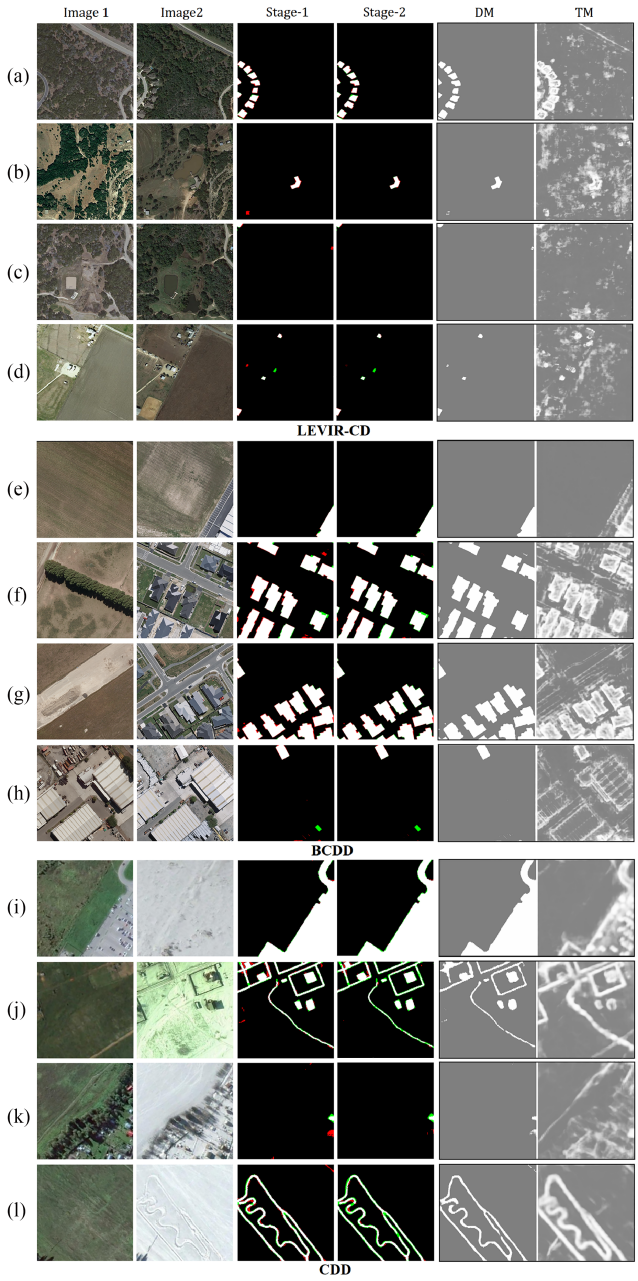

Fig. 11. Examples of DM and threshold map visualization.

detected changed regions are instead determined as unchanged regions. Whereas, DMG focus on correcting the false alarms while constraining the correctly detected changed areas, which is more effective than TMG.

Furthermore, Fig. 11 gives more examples of DM and TM. We can see that the DMG module is effective for correcting false alarms. However, although we continue to train the DM learning path in the Stage 2, it is difficult to further optimizing the DM [see Fig. 11(d) and (h)]. This is because we first train the DM learning path, and when the training stabilizes, join the TM learning path to correct the false alarms. Thus, DM learning has been stabilized in Stage 1, and it is difficult to further distinguish between the changed and unchanged areas through the DM training in Stage 2. In addition, we can observe that TMs usually exhibits higher values for the pseudochanged regions, including the misaligned regions [e.g., the roads in Fig. 11(a) and (k)], the targets that sensitive to imaging conditions [e.g., the cars in Fig. 11(i) and (k), and the edges of buildings], and the seasonal variations [e.g., the seasonal changes of vegetation in Fig. 11(a)–(c)]. Apart from the design of the loss function, this also benefits from the elementwise maximization operation of the DMG module, which makes the TM focus on the most interesting category information that are sensitive to pseudochanges within the temporal domain. Therefore, TM can effectively correct the false alarms by assigning high values to the indistinguishable unchanged regions.

3) Parameter Analysis: In addition to the parameters related to network training, including learning rates, batch sizes, and training epochs, there is another parameter in CLDRNet, i.e., feature dimensions D.

As described in Section III-B1, D is the hidden feature dimension before the category prediction layer, which is directly expressed as the feature dimension of CA-DESCs. Table V provides the parameter analysis of D on the LEVIR-CD, BCDD, and CDD datasets. We can observe that the F1-score starts to decrease when D is larger than 6. The reason may be that the CL is achieved by measuring the difference between the pixel representations and the reconstructed pixel representations, as shown in (15), and thus a higher dimension D would increase ambiguity rather than discriminability to disturb the training process. Besides, the selection of D affects the number of parameters for CADL and CCM modules. Comprehensively evaluate the results of the three datasets, we set $D = 3$.

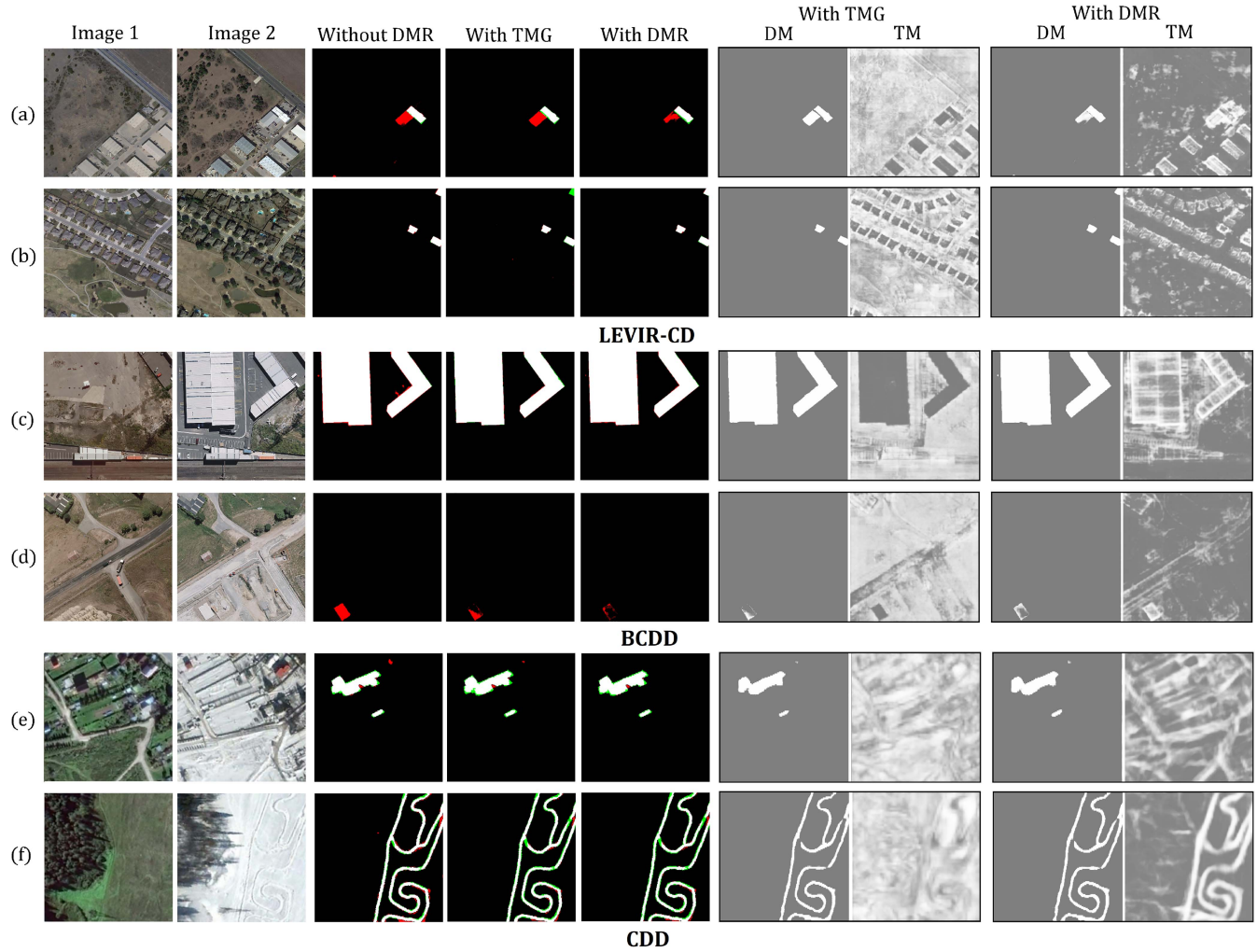

Fig. 12. Effect of DMR on the three datasets.

TABLE VI
EXPERIMENTAL COMPARISON OF DIFFERENT INPUT SIZES

	LEVIR-CD					BCDD				
	Recall	Precision	F1	Params.(M)	FLOPs(G)	Recall	Precision	F1	Params.(M)	FLOPs(G)
128×128	89.35	89.53	89.44	22.92	2.89	88.02	88.63	88.32	22.92	2.89
256×256	90.88	89.89	90.38	22.92	11.57	89.43	88.69	89.06	22.92	11.57
512×512	91.40	90.17	90.78	22.92	46.30	89.91	90.05	89.98	22.92	46.30

All the scores are described in percentage (%).

4) Analysis of the Input Sizes: To analyze the effect of input size on network performance and behavior, we conduct a comparative analysis by varying the input sizes. Table VI provides the quantitative analysis on LEVIR-CD and BCDD with input sizes of 512 × 512, 256 × 256, and 128 × 128. For network performance, we can observe that all input sizes achieve satisfactory results. However, the results of 512 × 512 outperform 256 × 256 and 128 × 128 in comparison, which is consistent with theoretical analysis, i.e., 512 × 512 patches maintain target integrity better than 256 × 256 and 128 × 128 patches. In addition, we can see that the smaller the input size, the lower the computational complexity, but the number of parameters is not affected by the input size.

V. CONCLUSION

In this work, a CLDRNet is proposed for remote sensing image CD. CLDRNet improves on our previous work D-TNet, and focuses on the essential characteristics of CD, i.e., different land cover changes exhibit different change magnitudes. On the one hand, to characterize the semantic content differences of heterogeneous land covers, a CL procedure is introduced into the

CCL module, which generates an overall representation for each category to guide the CCM. The CL process is differentiable and can be unified into the CD network, so it considers the semantic content differences from the CD perspective, thereby improving the CD performance. On the other hand, to adaptively address the magnitude differences of different land cover changes, a two-stage CD strategy is introduced for DM learning and DMR. The optimizations of DM and TM learning paths are aimed at ensuring high detection rates and revising false alarms, respectively. Extensive experiments on three CD datasets verify the effectiveness of the CLDRNet in both visual and quantitative analysis.

However, CLDRNet enhances the representations of different land covers through CL procedure, which is supervised by the reconstruction loss, i.e., to minimize the difference between pixel representations and reconstructed pixel representations. Therefore, CLDRNet deals with the binary change detection task. In the further, we will extend CLDRNet to the semantic change detection, which can determine both change extents and change types. The semantic labels will produce supervision for CL, improving the feature discriminations of heterogeneous land covers.

REFERENCES

[1] A. Singh, "Review article digital change detection techniques using remotely-sensed data," *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, 1989.

[2] L. Wan, M. Liu, F. Wang, T. Zhang, and H. J. You, "Automatic extraction of flood inundation areas from SAR images: A case study of Jilin, China during the 2017 flood disaster," *Int. J. Remote Sens.*, vol. 40, no. 13/14, pp. 5050–5077, 2019.

[3] S. Fang, K. Li, and Z. Li, "Changer: Feature interaction is what you need for change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5610111, doi: 10.1109/TGRS.2023.3277496.

[4] T. Celik, "Unsupervised change detection in satellite images using principal component analysis and k-means clustering," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 4, pp. 772–776, Oct. 2009.

[5] F. Bovolo and L. Bruzzone, "A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 1, pp. 218–236, Jan. 2007.

[6] E. Zhao, Y. Qian, C. Gao, H. Huo, X. Jiang, and X. Kong, "Characterization of land transitions patterns from multivariate time series using seasonal trend analysis and principal component analysis," *Remote Sens.*, vol. 6, no. 12, pp. 12639–12665, 2014.

[7] A. A. Nielsen, "The regularized iteratively reweighted MAD method for change detection in multi-and hyperspectral data," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 463–478, Feb. 2007.

[8] C. Wu, B. Du, and L. Zhang, "A subspace-based change detection method for hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 815–830, Apr. 2013.

[9] A. Ertürk, M.-D. Iordache, and A. Plaza, "Sparse unmixing-based change detection for multitemporal hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 708–719, Feb. 2016.

[10] T. Habib, J. Inglada, G. Mercier, and J. Chanussot, "Support vector reduction in SVM algorithm for abrupt change detection in remote sensing," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 3, pp. 606–610, Jul. 2009.

[11] F. Gao, J. Dong, B. Li, Q. Xu, and C. Xie, "Change detection from synthetic aperture radar images based on neighborhood-based ratio and extreme learning machine," *J. Appl. Remote Sens.*, vol. 10, no. 4, 2016, Art. no. 046019.

[12] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote Sens. Environ.*, vol. 61, no. 3, pp. 399–409, 1997.

[13] H. Chen, C. Wu, B. Du, L. Zhang, and L. Wang, "Change detection in multisource VHR images via deep siamese convolutional multiple-layers recurrent neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2848–2864, Apr. 2020.

[14] M. Liu, Q. Shi, A. Marinoni, D. He, X. Liu, and L. Zhang, "Super-resolution-based change detection network with stacked attention module for images with different resolutions," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4403718.

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

[16] M. Liu, Z. Chai, H. Deng, and R. Liu, "A CNN-transformer network with multiscale context aggregation for fine-grained cropland change detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4297–4306, 2022.

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

[18] L. Wan, Y. Tian, W. Kang, and L. Ma, "D-TNet: Category-awareness based difference-threshold alternative learning network for remote sensing image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5633316.

[19] L. Wan, Y. Xiang, and H. You, "An object-based hierarchical compound classification method for change detection in heterogeneous optical and SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9941–9959, Dec. 2019.

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

[21] S. Saha, F. Bovolo, and L. Bruzzone, "Unsupervised deep change vector analysis for multiple-change detection in VHR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3677–3693, Jun. 2019.

[22] L. Bergamasco, S. Saha, F. Bovolo, and L. Bruzzone, "Unsupervised change detection using convolutional-autoencoder multiresolution features," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4408119.

[23] S. Saha, F. Bovolo, and L. Bruzzone, "Unsupervised deep change vector analysis for multiple-change detection in VHR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3677–3693, Jun. 2019.

[24] L. T. Luppino et al., "Deep image translation with an affinity-based change prior for unsupervised multimodal change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4700422.

[25] Q. Li et al., "Unsupervised hyperspectral image change detection via deep learning self-generated credible labels," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 9012–9024, 2021.

[26] M. A. Lebedev, Y. V. Vizilter, O. V. Vygolov, V. A. Knyaz, and A. Y. Rubis, "Change detection in remote sensing images using conditional adversarial networks," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. XLII-2, pp. 565–571, 2018, doi: 10.5194/isprs-archives-XLII-2-565-2018.

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

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

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

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

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

[32] K. Jiang, W. Zhang, J. Liu, F. Liu, and L. Xiao, "Joint variation learning of fusion and difference features for change detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4709918.

[33] Q. Ke and P. Zhang, "CS-HSNet: A cross-siamese change detection network based on hierarchical-split attention," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 9987–10002, 2021.

[34] X. Zhang et al., "ADHR-CDNet: Attentive differential high-resolution change detection network for remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5634013.

[35] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[36] P. Wang et al., "Understanding convolution for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 1451–1460.

[37] A. Tversky, "Features of similarity," *Psychol. Rev.*, vol. 84, no. 4, 1977, Art. no. 327.
[38] T. Lei, Y. Zhang, Z. Lv, S. Li, S. Liu, and A. K. Nandi, "Landslide inventory mapping from bitemporal images using deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 6, pp. 982–986, Jun. 2019.
[39] H. Wei, R. Chen, C. Yu, H. Yang, and S. An, "BASNet: A boundary-aware siamese network for accurate remote-sensing change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2021, Art. no. 8022705.

Ling Wan received the Ph.D. degree in signal and information processing from the Aerospace Information Research Institute, Chinese Academy of Sciences (CAS), Beijing, China, in 2020.

She is currently an Assistant Researcher with the Institute of Automation, CAS. Her research interests include multisource remote sensing image change detection and feature detection.

Ye Tian received the B.S. degree in computer science and technology from the China University of Mining and Technology, Beijing, China, in 2012, and the M.E. degree in computer application technology from the China University of Mining and Technology, Beijing, China, in 2015.

He is currently an Engineer with the Institute of Automation, Chinese Academy of Sciences, Beijing. His research interests include image processing and knowledge map mining analysis.

Wenchao Kang received the Ph.D. degree in signal and information processing from the Aerospace Information Research Institute, Chinese Academy of Sciences (CAS), Beijing, China, in 2021.

He is currently an Assistant Researcher with the Institute of Automation, CAS. His research focuses on optical and SAR remote sensing image land cover classification.

Lei Ma received the Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, in 2011.

He is currently an Associate Professor with the Institute of Automation, CAS. His research interests include image processing and remote sensing intelligent interpretation.