

# Dynamic Graph-Level Neural Network for SAR Image Change Detection

Rongfang Wang<sup>✉</sup>, Member, IEEE, Liang Wang, Xiaohui Wei, Member, IEEE, Jia-Wei Chen<sup>✉</sup>, Member, IEEE, and Licheng Jiao<sup>✉</sup>, Fellow, IEEE

**Abstract**—The graph neural network (GNN) has been widely applied to image analysis and recognition. Recently, a semisupervised graph convolutional network (ssGCN) method has been proposed to change detection and obtains promising performance on very-high-resolution remote sensing images. However, a synthetic aperture radar (SAR) image is subject to speckle noise, and there is no explicit structure. In this letter, an end-to-end dynamic graph-level neural network (DGLNN) is proposed to exploit the local structure of each pixel neighborhood block at a graph level and learn a more discriminative graph for change detection. Moreover, in the training of DGLNN, a  $K$ -nearest neighborhood is employed to reconstruct edges between nodes instead of the fixed edges between two nodes so that each node exploits the features from different neighbor nodes. The proposed method is verified by cross-domain SAR image change detection on four sets of SAR images and compared with five state-of-the-art deep-learning-based SAR image change detection methods. The overall experimental results show that the proposed DGLNN obtains outstanding performance.

**Index Terms**—Change detection, cross-dataset, graph neural network (GNN), graph-lever classification, synthetic aperture radar (SAR) Image.

## I. INTRODUCTION

MOST traditional synthetic aperture radar (SAR) image change detection methods are developed with a difference image (DI)-based pipeline [1]. However, an SAR image is usually subject to the speckle, and the detection accuracy is greatly affected. It is challenging to exploit accurate and clear features from the speckle-affected DI. Recently, due to the powerful ability to exploit essential and robust features, neural networks (NNs) [2] have been successfully applied to computer vision and remote sensing image interpretation. Several NN-based change detection methods [3]–[5] have been developed and obtained better detection performance than traditional algorithms.

Manuscript received August 25, 2021; revised November 7, 2021; accepted November 30, 2021. Date of publication December 1, 2021; date of current version January 7, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62176196; in part by the Fundamental Research Funds for the Central Universities, Jilin University (JLU), under Grant 93K172021K11; and in part by the Open Fund of Fujian Key Laboratory of Data Science and Statistics, Minnan Normal University, under Grant 2020L0707. (*Corresponding author: Jia-Wei Chen*)

Rongfang Wang, Liang Wang, Jia-Wei Chen, and Licheng Jiao are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China (e-mail: rfwang@xidian.edu.cn; jawaechan@gmail.com).

Xiaohui Wei is with the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin University, Changchun 130012, China.

Digital Object Identifier 10.1109/LGRS.2021.3132167

Recently, with its promising performance and high interpretability, the graph NN (GNN) [6] becomes a popular deep learning model and achieve promising results in various applications, such as semantic segmentation [7], visual question answering [8], point cloud classification, and segmentation [9].

Among the variety of GNNs, the common character is the static graph construction, where an edge between two nodes is built by the initial features in each node. However, in the training process, the edges are invariant, and the features are inaccurate. To fix this issue, Wang *et al.* [10] proposed a dynamic graph convolutional neural network (CNN) for point clouds classification and segmentation, which achieves state-of-the-art performance. Zhang *et al.* [11] employed dynamic graph message-passing networks to the task of semantic segmentation, object detection, and instance segmentation.

On the other hand, Saha *et al.* [12] proposed a semisupervised graph convolutional network (ssGCN) for change detection, where a graph is proposed to exploit the structures of bitemporal very-high-resolution (VHR) images. The multi-scale parcels are obtained via a superpixel segmentation algorithm and treated as nodes to form a region adjacency graph. The information learned from the labeled parcels is propagated to the unlabeled ones over training epochs; ssGCN has shown good performance for VHR image change detection.

However, SAR image quality is subject to speckle, and it is quite challenging to exploit its structures. The ssGCN cannot be directly applied to bitemporal SAR images due to the inaccurate superpixel segmentation, which will affect the node-based ssGCN. Furthermore, it is rigorous that the number of nodes in the bitemporal images must be consistent for the ssGCN.

To tackle this limitation, in this letter, an end-to-end dynamic graph-level NN (DGLNN) is proposed for bitemporal SAR image change detection, where a dynamic graph is constructed for each three-channel block extracted from bitemporal SAR images and DI, and its edges are dynamically updated layer by layer to exploit the local structure of each pixel neighborhood block with the graph level. Moreover, in the training process, the feature propagation and aggregation of nodes are employed in the whole graph to learn more discriminative representations for each block. In the experiments, we compare our proposed method with three NN-based methods on four cross-dataset of the SAR image. The experimental results show that the proposed method DGLNN outperforms the other methods on SAR image change detection.

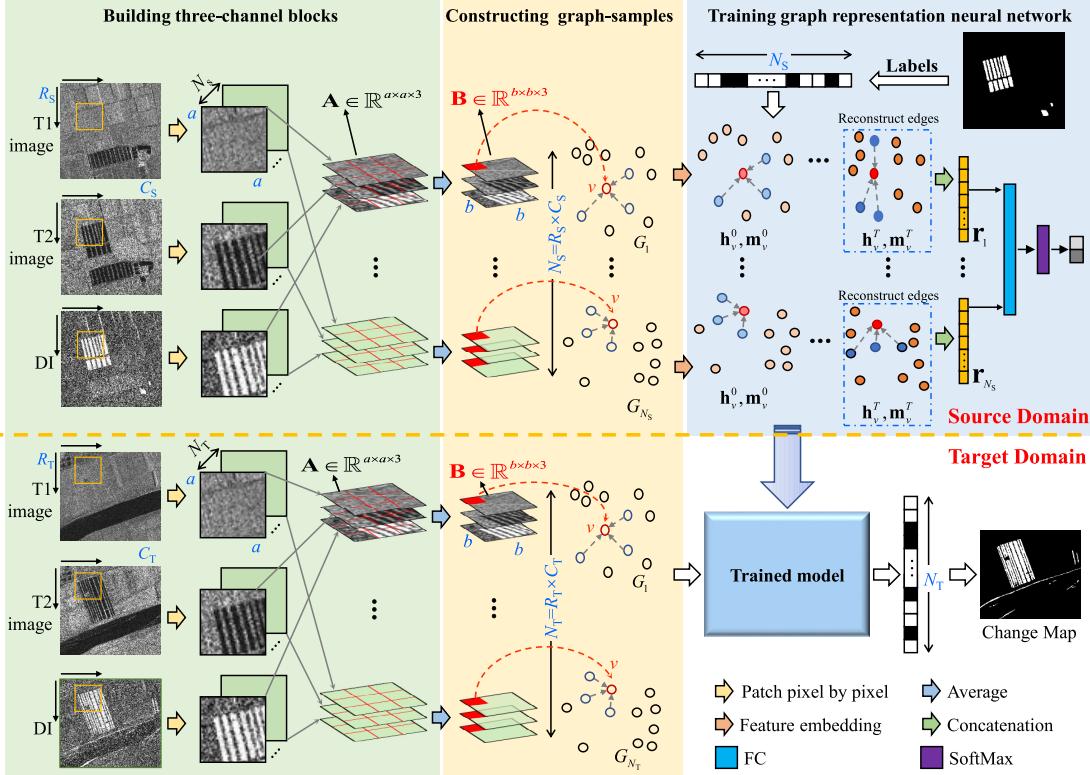


Fig. 1. Framework of the proposed DGLNN.

## II. METHODOLOGY

The framework of our method DGLNN is illustrated in Fig. 1. It consists of building a three-channel block, constructing a graph sample, and updating graph representation NN. Finally, the trained model is applied to cross-domain SAR image change detection. The details will be introduced in the following paragraphs.

### A. Building Three-Channel Blocks

Given a set of aligned bitemporal SAR images  $\mathbf{I}_1$  and  $\mathbf{I}_2$  with a size of  $R \times C$ , the corresponding DI is generated by the method of the log ratio

$$\mathbf{I}_{\text{DI}} = \log \left( \frac{\mathbf{I}_1}{\mathbf{I}_2 + 1} \right). \quad (1)$$

Both the bitemporal SAR images and generated DI are mirror-padded with the width of  $(a - 1)/2$  at the boundary, where  $a(a \geq 3)$  is the kernel size of convolution. From the padded  $\mathbf{I}_1$ ,  $\mathbf{I}_2$ , and  $\mathbf{I}_{\text{DI}}$ , the overlapping patches of size  $a \times a$  are extract at the same position. Finally, these three patches are stacked to one three-channel block  $\mathbf{A} \in \mathbb{R}^{a \times a \times 3}$ . We totally extract  $N = R \times C$  three-channel blocks.

### B. Constructing Graph Samples

A graph sample  $G = \{V, E\}$  on each three-channel block  $\mathbf{A}$  is constructed as follows. To exploit the spatial contextual information of a neighborhood to reduce the influence of speckle noise, each patch of  $\mathbf{A}$  is partitioned into  $b$  nonoverlapping regular-shaped superpixels, where the size of each one is

$s \times s$ , and  $b = a/s$ . Then, we take the mean of each superpixel on each channel separately to get a new matrix  $\mathbf{B} \in \mathbb{R}^{b^2 \times 3}$ .

Each column  $\mathbf{x}_v = [x_v^1, x_v^2, x_v^3]^T$ ,  $v = 1, 2, \dots, b^2$ , is considered as a node, and the number of graph node is  $|V| = b^2$ . The edges of each node in the first training epoch are built by  $K$ -nearest neighbors (KNNs) in the space of  $b^2$  nodes. The label of the constructed graph sample  $G$  is the value of the reference image corresponding to the central pixel of block  $\mathbf{A}$ . Because our method is cross-domain change detection, the reference image of the source domain is known, so the label information of the reference image can be used directly.

### C. Training a Graph-Level NN

The proposed DGLNN is trained on the above graph sample set  $\mathbf{G} = \{G_i = \{V, E\}, i = 1, \dots, N\}$  following two steps:

*Step 1:* Learning graph representation. Let  $\mathbf{h}_v^0 = \mathbf{x}_v \in \mathbb{R}^{3 \times 1}$  be the initial feature of the node  $v$  and  $\mathbf{h}_v^k \in \mathbb{R}^{3 \times 1}$  ( $k \in [0, K]$ ) be the hidden representation, which denotes the state of node  $v$  at the time step  $t$ . For each graph sample, motivated by Gao *et al.* [8], we update according to the following equations:

$$\mathbf{m}_v^t = \frac{1}{k} \sum_{n \in \Omega_v} \mathcal{F}_1(\mathbf{h}_n^k | n \in \Omega_v) \quad (2)$$

$$\mathbf{h}_v^{t+1} = \mathcal{F}_2(\text{cat}[\mathbf{h}_v^k, \mathbf{m}_v^k]) \quad (3)$$

where  $\text{cat}[\cdot, \cdot]$  denotes the concatenation of two vectors. In the  $k$ th time, (2) is employed to compute the node  $v$ 's message  $\mathbf{m}_v^k$ , which is aggregated from the neighbors set  $\Omega_v$  of node  $v$ , and  $k$  is the number of neighbors.  $\mathcal{F}_1(\cdot)$  denotes one fully connected

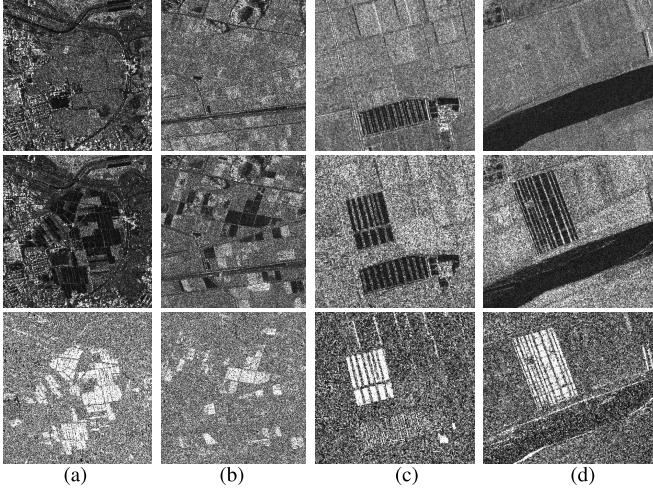


Fig. 2. Four datasets of bitemporal SAR images. The first two rows are bitemporal images, and the last row is the DI. (a) Sendai-A. (b) Sendai-B. (c) YR-A. (d) YR-B.

layer with the nonlinear activation function rectified linear unit (ReLU), which is used to compute message from neighbors. Then, we use (3) to update the hidden representation  $\mathbf{h}_v^{k+1}$  of the node  $v$  at the time step  $t+1$ .  $\mathcal{F}_2(\cdot)$  denotes another one fully connected layer as the updating function. After updating the hidden representation  $\mathbf{h}_v^{k+1}$ , the new dynamic edges between nodes in each graph samples are reconstructed by KNN based on  $\mathbf{h}_v^{t+1}$ . At time step  $k+1$ , we use the new graph sample to do the same updating the hidden representation operations for all the nodes, which is used for next time step. Finally, all the states of  $b^2$  nodes are concatenated to get the final representation vector of each graph sample  $\mathbf{r} = \text{cat}[\mathbf{h}_1^T, \mathbf{h}_2^T, \dots, \mathbf{h}_{b^2}^T]^T \in \mathbb{R}^{(3 \times b^2) \times 1}$ .

*Step 2:* Classify graph sample. We train a two layers multi-layer perceptron (MLP) based on the graph representation set  $\mathbf{R} = \{\mathbf{r}_i, i = 1, 2, \dots, N\}$  to classify a graph sample into the change class or unchange class. The loss function is a softmax cross-entropy. The model is trained with the backpropagation algorithm.

Finally, the trained GNN is applied to the cross-domain SAR image change detection.

### III. EXPERIMENT AND ANALYSIS

#### A. Datasets Description

The proposed method is verified on four datasets of bitemporal SAR images [4], as shown in Fig. 2. Two scenes (Sendai-A and Sendai-B) are acquired from the Sendai earthquake in Japan. The sizes of these two image sets are  $549 \times 560$  and  $613 \times 641$ , respectively. The other two scenes (YR-A and YR-B) are from bitemporal Yellow River SAR images acquired by the Radarsat-2 satellite in 2008 and 2009, respectively. The sizes of these two image sets are  $306 \times 291$  and  $400 \times 350$ , respectively.

#### B. Parameter Setup

For DGLNN, we set the patch size  $a = 21$  and the superpixel size  $s = 3$ . The number of nearest neighbors  $k = 3$ . The maximum training epoch is set as 7. For the

three comparison methods, all the parameters are set to the given value according to the corresponding work. Moreover, we verify the performance of all methods by leave-one-out-manner, i.e., the network trained on YR-B, Sendai-A, and Sendai-B is applied to an unknown testing dataset YR-A. For YR-B, Sendai-A, and Sendai-B, the training and testing datasets are crossed in the same way. Note that all training parameters on the four datasets remain the same.

#### C. Detection Results Analysis

We compared our proposed DGLNN with three deep-learning-based methods, CNN [3], pyramid scene parsing (PSP)-Net [4], deep neural network (DNN) [13], GraphTer [9], and deep change vector analysis (DCVA) [5], where CNN and DNN are the two-channel methods based on bitemporal SAR images, and others are the three-channel methods based on bitemporal SAR images and DI image.

The visual comparison results were shown in Fig. 3. It can be shown from the second column that DNN produces many misclassifications within the inner changed regions on Sendai-A, Sendai-B, and YR-B datasets while false alarms on Sendai-A and YR-A datasets. Next, it is shown from the third column that CNN produces many false alarms on Sendai-A, Sendai-B, and YR-A datasets while many missed alarms on the YR-B dataset. Furthermore, it is demonstrated from the fourth column that PSP produces fewer false alarms and missed alarms on both Sendai-B and YR-A datasets. However, PSP produces several false alarms on the Sendai-A dataset and missed alarms on the YR-B dataset. Finally, the proposed DGLNN produces fewer false alarms on Sendai-A, Sendai-B, and YR-A datasets and fewer missed alarms on the YR-B dataset.

Moreover, the performance was evaluated by probabilistic missed alarm (pMA), probabilistic false alarm (pFA), and kappa coefficient, where pFA (pMA) are calculated by the ratios between FA (MA) and the number of unchanged pixels (NC). The quantitative comparison results are shown in Fig. 4. It is shown from panel Fig. 4(a) and (b) that, on the Sendai-A dataset, DGLNN achieves the lowest FA, MA higher than PSP-Net, and lower than DNN and CNN. DGLNN obtains more clear visual results with less noisy spots than DNN, CNN, and PSP-Net on the Sendai-B and YR-B. On the YR-A, although PSP-Net had the least FA, its MA is highest in four methods; DGLNN had the lower FA, MA than DNN, and CNN. It is shown from Fig. 4(c) that our DGLNN gets the best kappa on all datasets. Therefore, our proposed method outperforms other compared methods.

#### D. Ablation Study

*1) Effect of Patch Size and Superpixel Size:* First, we analyze the influence of the patch size  $a$  in DGLNN. As shown in Fig. 5(a), it can be found that, when the value of  $a$  is set 21, the overall change detection performance of DGLNN is the best. If the patch size  $a$  is too small, there will be fewer features for change detection, and  $a$  is too big, which will cause the image patch contains more features, which are irrelevant to the label of the central pixel. Therefore, we set  $a = 21$ .

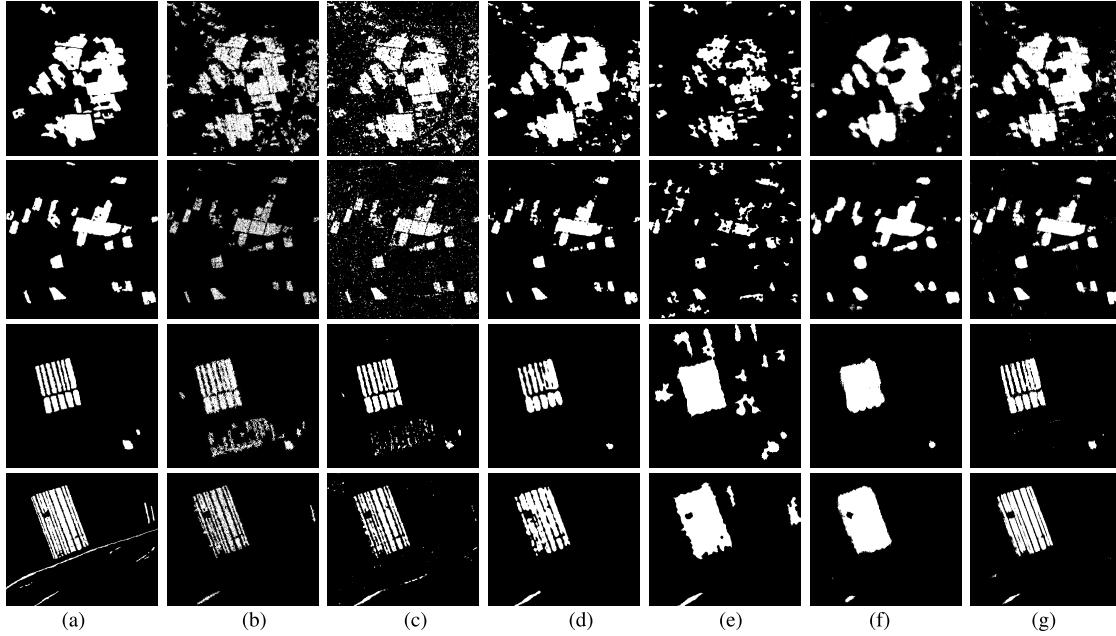


Fig. 3. Visual results of the comparison methods. The first row is the reference image. From the second row to bottom, they are the results of method DNN, CNN, PSP-Net, and DGLNN, respectively. (a) Reference. (b) DNN. (c) CNN. (d) PSP. (e) DCVA. (f) GraphTer. (g) DGLNN.

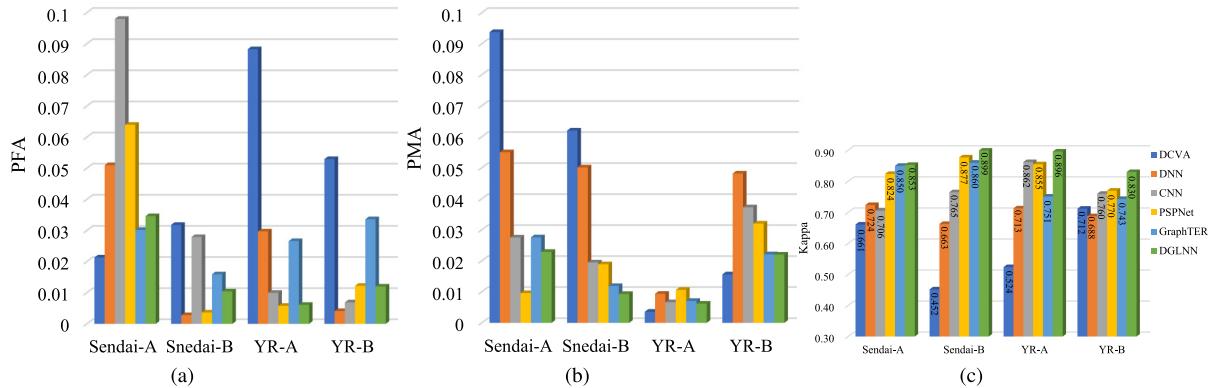


Fig. 4. Quantitative results of the comparison methods. (a) pFA. (b) pMA. (c) Kappa.

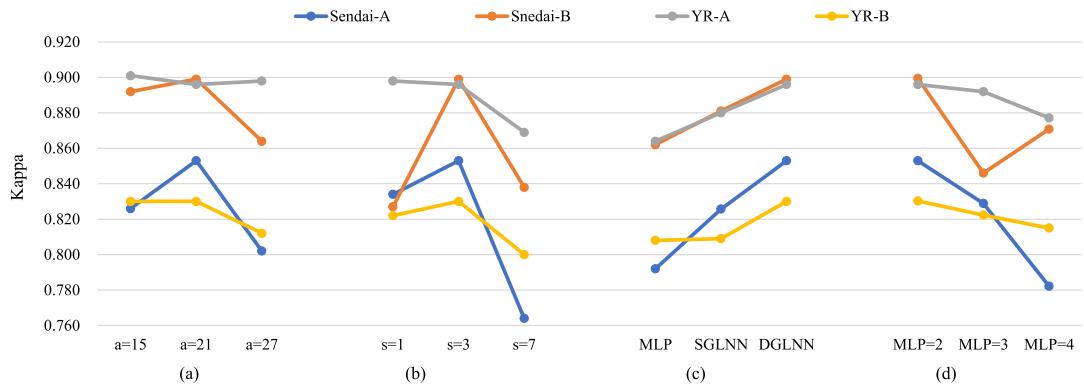


Fig. 5. Performance of ablation study. (a) Patch size  $a$ . (b) Superpixel resolution  $s$ . (c) Various components. (d) Different MLP layers.

Then, we conduct DGLNN with different values of  $s$  in the case of  $a = 21$ . The results with different values of  $s$  are shown in Fig. 5(b). It is shown in the figure that, for

Sendai-A, Sendai-B, and YR-B datasets, when  $s = 3$ , the DGLNN obtains best Kappa, and for YR-A, although the best Kappa is obtained in the condition of  $s = 1$ ,  $s = 3$  obtains

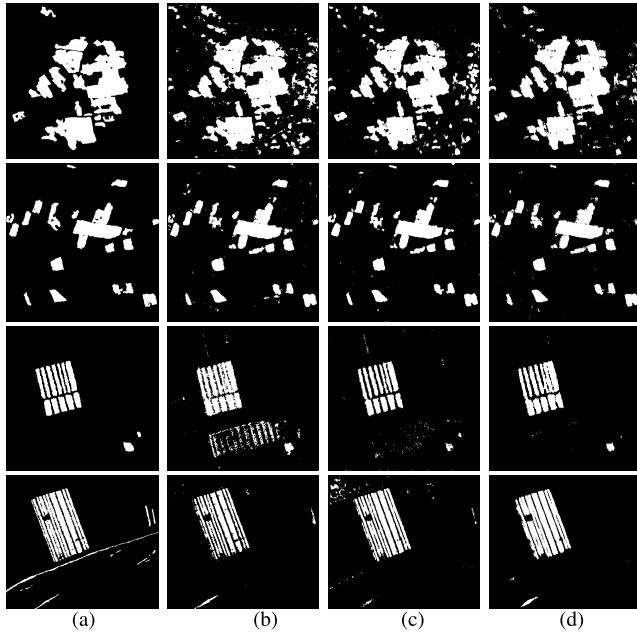


Fig. 6. Visual results of the ablation study. The first row is the reference image. From the second row to bottom, they are the results based on the method MLP, SGLNN, and DGLNN, respectively. (a) Reference. (b) MLP. (c) SGLNN. (d) DGLNN.

the comparable Kappa. The parameter  $s$  is used to extract features for each node during the initialization of the graph construction. The parameter  $s$  is set to small, resulting in that the initialization features of each node are too similar, which is not conducive to subsequent epoch, and the  $s$  is set too large, resulting in that each node cannot extract enough structured information. Thus, in this ablation study, we find that the best parameters  $s$  should be 3.

2) *Effect of Various Components*: Since we integrate three components in our proposed model DGLNN, we conducted an additional ablation study in order to analyze the impact of DGLNN's component on change detection performance. First, we have set up two ablation studies; we only removed one component for each ablation study. In ablation study 1, we use KNN to build the edge of each node in the first epoch, which means that the edges between nodes are fixed in the following epochs. This modified model was named static graph-level NN (SGLNN). In ablation study 2, we further removed the setting of the GNN. We directly fed three-channel blocks into the two layers MLP, which is used in DGLNN to generate the final change detection result. This modified model was named MLP. Except for these, training parameters are the same as those in Section III-C.

The quantitative comparison results were shown in Fig. 5(c). The visual comparison results were shown in Fig. 6. In the case of MLP and SGLNN, the visual result shows that MLP and SGLNN get more noisy spots and less clear change regions than DGLNN in general. In the quantitative comparison, DGLNN failed to achieve the best pFA and pMA in Fig. 5(c), but DGLNN achieved the best Kappa in four datasets. Comparing SGLNN with DGLNN, the result shows that the setting of dynamically update node features from

different neighbor nodes in each epoch is better than update node features from the same neighbor nodes in each epoch. Comparing MLP with DGLNN, the result demonstrates that the proposed graph structure is effective.

Furthermore, the performance on various layers of MLP can be shown in Fig. 5(d). It is shown that, when the layer of MLP is set to 2, the best performance can be obtained.

#### IV. CONCLUSION

In this letter, we have proposed a novel DGLNN model for bitemporal SAR image change detection. DGLNN is the graph-level change detection method, which constructs the graph on the three-channel pixel neighborhood block and then dynamically updates the edge connections between nodes according to the trained features. The proposed method is verified on four cross-dataset of SAR images and compared with three state-of-the-art deep-learning-based methods. The experimental results show that the proposed DGLNN obtains outstanding performance, and the performance of the three-channel method added DI is generally better than the method only with bitemporal images. The results of ablation experiments show that the dynamic construction of edges between nodes can improve the performance of change detection compared with the setting of fixed edges between nodes.

#### REFERENCES

- [1] L. Bruzzone and D. F. Prieto, "Automatic analysis of the difference image for unsupervised change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1171–1182, May 2000.
- [2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [3] Y. Li, C. Peng, Y. Chen, L. Jiao, L. Zhou, and R. Shang, "A deep learning method for change detection in synthetic aperture radar images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5751–5763, Mar. 2019.
- [4] J.-W. Chen, R. Wang, F. Ding, B. Liu, L. Jiao, and J. Zhang, "A convolutional neural network with parallel multi-scale spatial pooling to detect temporal changes in SAR images," *Remote Sens.*, vol. 12, no. 10, p. 1619, May 2020.
- [5] 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.
- [6] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020.
- [7] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3D graph neural networks for RGBD semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5199–5208.
- [8] D. Gao, K. Li, R. Wang, S. Shan, and X. Chen, "Multi-modal graph neural network for joint reasoning on vision and scene text," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12746–12756.
- [9] X. Gao, W. Hu, and G.-J. Qi, "GraphTER: Unsupervised learning of graph transformation equivariant representations via auto-encoding node-wise transformations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 7163–7172.
- [10] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.
- [11] L. Zhang, D. Xu, A. Arnab, and P. H. S. Torr, "Dynamic graph message passing networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3726–3735.
- [12] S. Saha, L. Mou, X. Zhu, F. Bovolo, and L. Bruzzone, "Semisupervised change detection using graph convolutional network," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 4, pp. 607–611, Apr. 2021.
- [13] M. Gong, J. Zhao, J. Liu, Q. Miao, and L. Jiao, "Change detection in synthetic aperture radar images based on deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 1, pp. 125–138, Jan. 2015.