

ChangeMamba: Remote Sensing Change Detection with Spatio-Temporal State Space Model

Hongruixuan Chen[†], *Student Member, IEEE*, Jian Song[†], Chengxi Han, *Student Member, IEEE*, Junshi Xia, *Senior Member, IEEE*, Naoto Yokoya, *Member, IEEE*

Abstract—Convolutional neural networks (CNN) and Transformers have made impressive progress in the field of remote sensing change detection (CD). However, both architectures have inherent shortcomings: CNN are constrained by a limited receptive field that may hinder their ability to capture broader spatial contexts, while Transformers are computationally intensive, making them costly to train and deploy on large datasets. Recently, the Mamba architecture, based on state space models, has shown remarkable performance in a series of natural language processing tasks, which can effectively compensate for the shortcomings of the above two architectures. In this paper, we explore for the first time the potential of the Mamba architecture for remote sensing CD tasks. We tailor the corresponding frameworks, called MambaBCD, MambaSCD, and MambaBDA, for binary change detection (BCD), semantic change detection (SCD), and building damage assessment (BDA), respectively. All three frameworks adopt the cutting-edge Visual Mamba architecture as the encoder, which allows full learning of global spatial contextual information from the input images. For the change decoder, which is available in all three architectures, we propose three spatio-temporal relationship modeling mechanisms, which can be naturally combined with the Mamba architecture and fully utilize its attribute to achieve spatio-temporal interaction of multi-temporal features, thereby obtaining accurate change information. On five benchmark datasets, our proposed frameworks outperform current CNN- and Transformer-based approaches without using any complex training strategies or tricks, fully demonstrating the potential of the Mamba architecture in CD tasks. Specifically, we obtained 83.11%, 88.39% and 94.19% F1 scores on the three BCD datasets SYSU, LEVIR-CD+, and WHU-CD; on the SCD dataset SECOND, we obtained 24.11% SeK; and on the BDA dataset xBD, we obtained 81.41% overall F1 score. Further experiments show that our architecture is quite robust to degraded data. The source code is available in <https://github.com/ChenHongruixuan/MambaCD>.

Index Terms—State space model, Mamba, binary change detection, semantic change detection, building damage assessment, spatio-temporal relationship, optical high-resolution images

I. INTRODUCTION

Manuscript submitted April 11, 2024. This work was supported in part by the Council for Science, Technology and Innovation (CSTI), the Cross-ministerial Strategic Innovation Promotion Program (SIP), Development of a Resilient Smart Network System against Natural Disasters (Funding agency: NIED), the JSPS, KAKENHI under Grant Number 22H03609, JST, FOREST under Grant Number JPMJFR206S, Microsoft Research Asia, and Next Generation AI Research Center, The University of Tokyo. ([†]: equal contribution; Corresponding author: Naoto Yokoya)

Hongruixuan Chen, Jian Song, and Naoto Yokoya are with Graduate School of Frontier Sciences, The University of Tokyo, Chiba, Japan (e-mail: qschr@ gmail.com; song@ms.k.u-tokyo.ac.jp; yokoya@k.u-tokyo.ac.jp).

Chengxi Han is with State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan, China (e-mail: chengxihan@whu.edu.cn).

Junshi Xia is with RIKEN Center for Advanced Intelligence Project (AIP), RIKEN, Tokyo, 103-0027, Japan (e-mail: junshi.xia@riken.jp).

CHANGE detection (CD) has been a popular field within the remote sensing community since the inception of remote sensing technology. It aims to detect changes in objects on the Earth's surface from multi-temporal remote sensing images acquired at different times. Depending on the desired result from the change detector, CD tasks can be categorized into three types, namely binary CD (BCD), semantic CD (SCD), and building damage assessment (BDA). Now, CD techniques play an important role in many fields, including land cover change analysis, urban sprawl studies, disaster response, geographic information system (GIS) updating and ecological monitoring [1]–[7].

Optical high-resolution remote sensing imagery has become one of the most applied and researched data sources in the field of CD, which can provide detailed textural and geometric structural information about surface features, allowing us to achieve more refined changes [3]. Due to the increased heterogeneity within the same land-cover feature brought about by the increase in spatial resolution, traditional pixel-based CD methods [8]–[10] are difficult to achieve satisfactory detection results. To cope with this, researchers proposed object-based CD methods [11]–[13], which take the object consisting of homogeneous pixels as the basic unit for CD. However, these methods rely on hand-designed “shallow” features, which are not representative and robust enough to cope with complex ground conditions in multi-temporal high-resolution images [14], [15].

The emergence of deep learning has brought new models and paradigms for CD, greatly improving the efficiency and accuracy of CD. After the early days of developing image patch-based approaches with tiny deep learning models [16]–[18], the release of a number of large-scale benchmark datasets [19]–[21] has allowed us to start designing and introducing larger and deeper models. Convolutional neural network (CNN)-based methods have been dominant within the field of CD since the introduction of fully convolutional networks (FCNs) to the field by Daudt *et al.* [22]. By employing models within the computer vision field as well as a priori knowledge of the CD task, many representative methods have been proposed [3], [4], [23]–[27]. Despite the decent results achieved by these methods, the inherent shortcomings of the CNN model structure, *i.e.*, the inability of the limited receptive field to capture long-range dependencies between pixels, make these methods still fall short when dealing with complex and diverse multi-temporal scenes in images with different spatial-temporal resolutions [28]. The emergence of the visual Transformer [29] provides an effective way to solve the above

problems. By means of the stacked self-attention modules, the Transformer architecture can fully model the relationship between all the pixels of the whole image. Currently, more and more CD architectures adopt Transformer architectures as the encoder to extract representative and robust features [30]–[32] and utilize it in the decoder to capture the spatio-temporal relationships between multi-temporal features [28], [33], exhibiting superior performance.

Nonetheless, the self-attention operation demands quadratic complexity in terms of image sizes, resulting in expensive computational overhead, which is detrimental in dense prediction tasks, e.g., land-cover mapping, object detection, and CD, in large-scale remote sensing datasets. Some available solutions, such as limiting the size of the computational window or the step size to improve attention efficiency, come at the cost of imposing limits on the size of the reception field [34], [35]. As a viable alternative to the Transformer architecture, state space models (SSMs), especially structured state space sequence models (S4) [36], have shown cutting-edge performance in continuous long-sequence data analysis, with the promising attribute of linear scaling in sequence length. The Mamba architecture [37] further improves on the S4 model by employing a selection mechanism that allows the model to select relevant information in an input-dependent manner. By combining it with hardware-aware implementations, Mamba outperforms Transformers on a number of downstream tasks with high efficiency. Very recently, Mamba architecture has been extended to image data and shown promising results on some visual tasks [38], [39]. However, its potential for high-resolution optical images and different CD subtasks is still under-explored. How to design a suitable architecture based on a priori knowledge of CD tasks is undoubtedly valuable for subsequent research in the community.

In this paper, we present the first attempt at introducing Mamba architecture for CD tasks and propose several spatio-temporal state space models (collectively called ChangeMamba) to realize effective and robust CD. ChangeMamba can effectively model the global spatial context and spatio-temporal relationships in multi-temporal images, thereby showing very promising results on the three subtasks of CD, namely BCD, MCD, and BDA. Specifically, ChangeMamba is based on the recently proposed VMamba architecture [38], which adopts a Cross-Scan Module (CSM) to unfold image patches in different spatial directions to achieve effective modeling of global contextual information from images. Since CD tasks require the detector to adequately learn spatio-temporal features from multi-temporal images, we design three spatio-temporal relationship modeling mechanisms and naturally combine them with Mamba architecture. The proposed three mechanisms can adequately model the spatio-temporal relationships between multi-temporal features by scanning them in different ways, and thus effectively detect the different categories of changes, including binary changes, semantic changes, and building damage levels.

In summary, the main contributions of this paper are as follows:

- 1) For the first time, we explore the application of the Mamba architecture in the field of remote sensing

change detection, thereby achieving high accuracy, efficient, and robust CD.

- 2) Based on the Mamba architecture and incorporating the unique characteristics of CD tasks, we design the corresponding network frameworks for each of the three CD tasks, *i.e.*, BCD, SCD, and BDA.
- 3) Three spatio-temporal relationship modeling mechanisms are proposed, tailored for CD tasks, with the aid of the Mamba architecture to fully learn spatio-temporal features.
- 4) The proposed three frameworks show very competitive and even SOTA performance on BCD, SCD, and BDA tasks on five benchmark datasets, demonstrating their superiority. The source code for this work is publicly available for contributions to subsequent possible research.

The remainder of this paper is organized as follows. Section II reviews related work. Section III details the three ChangeMamba architectures proposed for BCD, MCD, and BDA, respectively. The experimental results and discussion are provided in Sections IV. Section V draws the conclusion.

II. RELATED WORKS

In this section, we focus on reviewing deep learning-based CD methods using optical high-resolution remote sensing images as data sources.

A. CNN-Based Method

The rapid evolution of CNN and its prowess in automatically extracting hierarchical features have led to wide applications in CD tasks. After the early stage of the patch-based framework [16], [18], Daudt *et al.* [22] presented the first attempt to introduce FCNs into the BCD field and three basic FCNs architectures were developed. After that, many representative network architectures were proposed. Zhang *et al.* [23] sought to tailor networks for semantic segmentation tasks more suitable for the BCD task by integrating feature change maps as a deep supervised signal, thus refining the final prediction. To make the framework suitable for handling multi-source images, Chen *et al.* [17] proposed a framework by combining long- and short-term memory (LSTM) networks with CNNs. Furthermore, Fang *et al.* [40] designed a Siamese architecture with denser connections to facilitate more effective shallow information exchange for BCD. To combat the challenge of uneven foreground-background distributions within BCD, Han *et al.* [41] employed a novel sampling approach and an attention mechanism to integrate global information more effectively. After that, they introduced an innovative self-attention module that not only enhances long-distance dependency, but also specifically targets the refinement of BCD predictions, significantly improving the delineation of change area boundaries and reducing the occurrence of inner holes [42]. Zhao *et al.* [43] explored the effect of different fusion strategies on BCD performance and designed a dual encoder-decoder architecture to improve detection performance. To solve the problem of insufficient samples, the transfer learning technique

was also introduced. Guo *et al.* [4] developed an attention-based network for semi-supervised building updating. Cao *et al.* [25] designed a full-level fused cross-task transfer learning architecture for building change detection. Recently, Chen *et al.* [44] has developed a sample generation method by simply exchanging image patches, which can train change detectors on single-temporal images in an unsupervised manner.

Compared to BCD, which only aims to identify “where” changes have occurred, SCD further seeks to determine “what” the changes are. Although this poses a greater challenge, it is more important for practical application analysis [24], [45]. Early explorations based on CNNs for SCD is the study [45], which adopts a multitask learning framework to simultaneously predict binary change maps and land cover maps, using the latter to assist in generating the final semantic change prediction. Mou *et al.* [46] proposed to combine recurrent neural network (RNN) and CNN for SCD, where a Siamese CNN extracts semantic features and RNN models temporal dependencies to classify multiple types of changes. Yang *et al.* [20] introduced a benchmark dataset for SCD along with a new evaluation metric. Based on the framework in [45], Ding *et al.* [47] proposed a more powerful architecture to address the insufficient exchange of information between the semantic and change decoders. A new loss function was introduced to force the network to correlate two decoders. Zheng *et al.* [24] continues to strengthen the interactions between multi-temporal features, designing a temporal-symmetric transformer (TST) module to model the spatio-temporal relationships. In [48], channel attention was used to embed change information into temporal features. In [49], temporal salient features were fused to extract and improve the representation of changes. In [50] and [14], unsupervised SCD methods were also studied, using pre-trained CNNs or unsupervised learning techniques to extract hierarchical features, then comparing these features to determine changes.

Due to the lack of suitable benchmark datasets caused by the scarcity of disaster events and difficulty in annotation, the methodology progression of BDA has been relatively slow. The establishment of the xBD benchmark dataset in the xView2 Challenge [51], in turn, provides a large-scale, common benchmark for the field. The procedural approach to BDA typically encompasses two primary components: identifying the location of the building and assessing the level of damage incurred. In the xBD dataset, there are four damage levels, i.e., “no damage”, “minor damage”, “major damage”, and “destroyed”. This bifurcation essentially positions BDA as a nuanced subset of SCD, with a focus on quantifying damage levels. Among established approaches, the xView2 challenge [51] introduced a baseline method that employs a residual UNet for building localization and damage classification, trained with pre and post-disaster imagery. The Siamese-UNet, the winning approach of the xView2 challenge [51], utilized a UNet framework for both pixel-level building localization and damage-level classification. It initialized the UNet for the damage classification task with the parameters from the UNet for the building localization task. A further advancement [52] utilized Mask-RCNN architecture [53] to address BDA, incorporating Siamese encoders, multitask heads, and instance-

level predictions. Zheng *et al.* [3] introduced a deep object-based architecture that encompasses shared and task-specific encoders, alongside an FCN-based multitask decoder. This development enhances task integration with an object-based post-processing strategy, resulting in more refined outcomes.

The development of CNN-based CD methods has matured significantly over time. However, despite their excellent ability to extract local features, CNN architectures inherently struggle to capture long-distance dependencies [29]. This limitation becomes particularly critical in CD tasks, where changed regions, compared to background areas, are often sparse and scattered. The network’s ability to model relationships beyond local areas is essential for accurately identifying these changed regions [28], [30].

B. Transformer-Based Method

Recently, with the promising results obtained by Transformers in the computer vision field [29], [35], their superior long-distance modeling capabilities, have made them effectively overcome the challenges faced by the CNNs. As a result, many studies have begun to address CD tasks using Transformer architectures. In the field of BCD, Chen *et al.* [28] emerged as a pioneering approach, transforming multi-temporal images into semantic tokens to model spatial-temporal relationships within a token-based framework, enhancing the CD process. Bandara *et al.* [30] proposed a pure Transformer-based Siamese network, which has a Siamese Transformer encoder coupled with a multilayer perceptron (MLP) decoder. This architecture eliminates the need for a CNN-based feature extractor. Similarly, Zhang *et al.* [32] proposed a pure Transformer-driven model, adopting a Siamese U-shaped structure to fully recover detailed change information. Specifically, it leveraged the Swin Transformer blocks [34] across its architecture, further demonstrating the Transformer’s capability in BCD tasks. Li *et al.* [33] introduced a hybrid model that blends the global contextual modeling capability of Transformers with the CNN block and UNet architecture. It eliminates UNet’s redundant information and enhances feature quality through a difference enhancement module, leading to more precise detection results.

For Transformer-based SCD, Niu *et al.* [54] used a PRTB backbone, combining preactivated residual and Transformer blocks, to extract semantic features from multi-temporal image pairs, which is crucial for detecting subtle changes. The model’s efficacy is further amplified by multitask prediction branches and a custom loss function, ensuring nuanced SCD with heightened accuracy. In addition, Ding *et al.* [55] introduced the Semantic Change Transformer (SCanFormer), a CSWin Transformer adaptation, to explicitly learn semantic transitions. The model has a triple encoder-decoder architecture to enhance spatial features and incorporates spatio-temporal dynamic and task-specific prior information, leading to a decrease in learning disparities. This approach significantly boosts accuracy on benchmarks, adeptly managing scenarios with sparse labels.

Transformer-based BDA methods are currently understudied. Chen *et al.* [31] proposed the first pure Transformer-based BDA architecture called DamFormer. It has a Siamese

Mix Transformer (MiT) [35] encoder and a lightweight all-MLP decoder to efficiently process multi-temporal image pairs. DamFormer employs a multi-temporal fusion module for improved information integration, and a dual-task decoder for precise building localization and damage classification. By adopting strategies like weight sharing and innovative fusion mechanisms, it efficiently handles high-resolution imagery with lower computational costs, making it effective and streamlined for BDA.

Although Transformer-based methods have significantly enhanced CD accuracy through their powerful nonlocal modeling capabilities, they come with substantial computational resource demands. Specifically, as the number of tokens increases, the consumption of computational resources escalates quadratically, posing challenges for processing large-scale input, such as optical high-resolution imagery. Recently, the emerging Mamba architecture [37] not only rivals Transformers in non-local modeling capabilities but also maintains computational overhead at a linear growth rate, which opens up the possibility of efficient and accurate CD.

C. State Space Model

The SSM concept first popped up with the S4 model [36], offering a new way to handle contextual information globally, which caught attention due to its attractive property of scaling linearly in sequence length. Based on the S4 model, Smith et al. [56] proposed a new S5 model by introducing MIMO SSM and efficient parallel scan into the S4 model. After that, the H3 model [57] further advanced these concepts, achieving a performance on par with that of Transformers in language modeling tasks. Recently, Gu *et al.* [37] proposed a data-dependent SSM layer and built a generic language model backbone called Mamba, which outperforms Transformers at different scales on large-scale real data and scales linearly in sequence length. Very recently, Visual Mamba [38] and Vision Mamba [39] have extended the Mamba architecture to 2D image data, showing superior performance on many computer vision tasks. Inspired by this progress, some pioneering work has introduced the Mamba architecture to the field of remote sensing image processing [58], [59]. Their studies show that the Mamba architecture can yield performance comparable to the advanced CNN and Transformer architectures on scene classification [59] and pansharpening [58] tasks. However, these works still focus on low-level tasks or classification tasks, and all of these tasks are single-temporal tasks [44]. The potential of the Mamba architecture for multi-temporal remote sensing image-related scenarios and dense prediction tasks remains to be explored.

III. METHODOLOGY

A. Preliminaries

The SSM-based models and Mamba are inspired by linear time-invariant systems, which map a 1-D function or sequence $x(t) \in \mathcal{R}$ to the response $y(t) \in \mathcal{R}$ through a hidden state

$h(t) \in \mathcal{R}^N$. These systems are usually formulated as linear ordinary differential equations (ODEs) as

$$\begin{cases} h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t) \\ y(t) = \mathbf{C}h(t) \end{cases} \quad (1)$$

where $\mathbf{A} \in \mathcal{R}^{N \times N}$ is the evolution parameter and $\mathbf{B} \in \mathcal{R}^{N \times 1}$, $\mathbf{C} \in \mathcal{R}^{1 \times N}$ are the projection parameters.

The S4 model [36] is a discrete counterpart of the continuous system, as continuous systems face significant challenges when integrated into deep learning algorithms. It contains a time scale parameter Δ that is used to convert the continuous parameters \mathbf{A} and \mathbf{B} into their discrete counterparts $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$. A prevalent way to achieve this transformation is to use a zero-order hold (ZOH) approach:

$$\begin{cases} \bar{\mathbf{A}} = \exp(\Delta \mathbf{A}) \\ \bar{\mathbf{B}} = (\Delta \mathbf{A})^{-1} (\exp(\Delta \mathbf{A}) - \mathbf{I}) \Delta \mathbf{B} \end{cases} \quad (2)$$

After that, the discretization of Eq. (1) can be formulated as

$$\begin{cases} h'_t = \bar{\mathbf{A}}h_{t-1} + \bar{\mathbf{B}}x_t \\ y_t = \mathbf{C}h_t \end{cases} \quad (3)$$

The output of x with sequence length L_x can be calculated directly by the following global convolution operation

$$\begin{cases} \bar{\mathbf{K}} = (\mathbf{C}\bar{\mathbf{B}}, \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{B}}, \dots, \mathbf{C}\bar{\mathbf{A}}^{L_x-1}\bar{\mathbf{B}}) \\ \mathbf{y} = \mathbf{x} * \bar{\mathbf{K}} \end{cases} \quad (4)$$

where $\bar{\mathbf{K}} \in \mathcal{R}^{L_x}$ is a structured convolutional kernel and $*$ denotes a convolutional operation.

Mamba architecture [37] builds on the S4 model by proposing a selection mechanism that allows the model to filter out irrelevant information and recall relevant information¹. In addition, a hardware-aware algorithm has been developed for recursive computation of the model but will not be implemented in the extended state, optimizing the GPU memory layout. These characteristics allow the Mamba architecture not only to obtain promising results on languages and other data with long sequences, but also to achieve fast training and inference with memory and computational overhead linearly proportional to the sequence length.

B. Problem Statement

In this paper, we focus on three sub-tasks within the CD field. They are binary change detection (BCD), semantic change detection (SCD), and building damage assessment (BDA). The definitions of the three tasks are as follows.

1) *Binary Change Detection*: BCD is the basic and the most intensively studied task in the CD field. It focuses on “where” change occurs. According to the category of interest, BCD can also be divided into category-agnostic CD focusing on general land-cover change information, and single-category CD, e.g., building CD and forest CD. Given a training set

¹Mamba architectures sometimes are abbreviated as the S6 model since they are S4 models with a selection mechanism and computed with a scan [37].

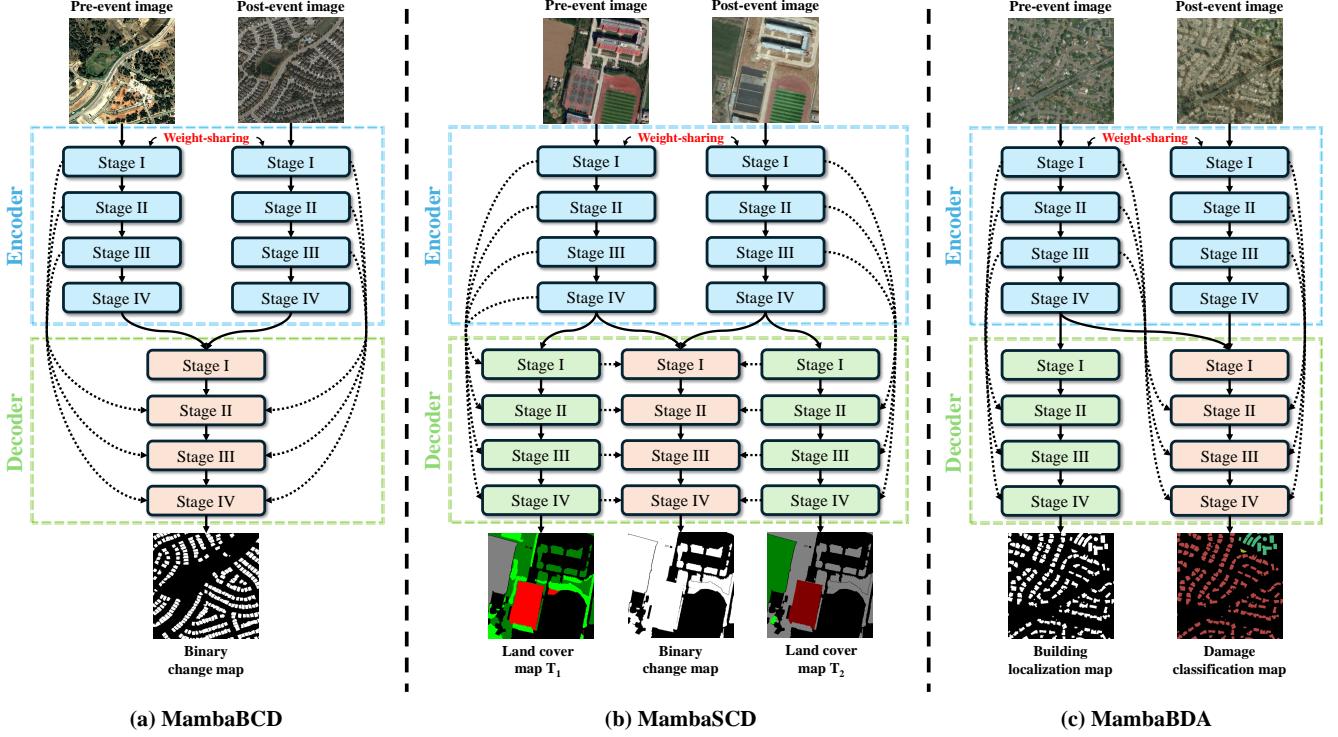


Fig. 1: The network framework of the proposed (a) MambaBCD for binary change detection, (b) MambaSCD for semantic change detection, and (c) MamabaBDA for building damage assessment.

represented as $\mathcal{D}_{train}^{bcd} = \left\{ \left(\mathbf{X}_i^{T_1}, \mathbf{X}_i^{T_2}, \mathbf{Y}_i^{bcd} \right) \right\}_{i=1}^{N_{train}^{bcd}}$, where $\mathbf{X}_i^{T_1}, \mathbf{X}_i^{T_2} \in \mathcal{R}^{H \times W \times C}$ is the i -th multi-temporal image pair acquired in T_1 and T_2 , respectively, and $\mathbf{Y}_i^{bcd} \in \{0, 1\}^{H \times W}$ is the corresponding label, the goal of BCD is to train a change detector $\mathcal{F}_{\theta}^{bcd}$ on $\mathcal{D}_{train}^{bcd}$ that can predict change maps reflecting accurate “change / non-change” binary information on new sets.

2) *Semantic Change Detection*: Compared to BCD, SCD focuses not only on “where” the change occurs, but also on the “what” of the change, i.e. “from-to” semantic change information [24], [47], [55]. The training set in the SCD task can be represented as $\mathcal{D}_{train}^{scd} = \left\{ \left(\mathbf{X}_i^{T_1}, \mathbf{X}_i^{T_2}, \mathbf{Y}_i^{T_1}, \mathbf{Y}_i^{T_2}, \mathbf{Y}_i^{bcd} \right) \right\}_{i=1}^{N_{train}^{scd}}$. Compared to BCD, the corresponding land-cover labels $\mathbf{Y}_i^{T_1}, \mathbf{Y}_i^{T_2} \in \{0, 1, \dots, C^{lcm}\}^{H \times W}$ with C^{lcm} categories of $\mathbf{X}_i^{T_1}$ and $\mathbf{X}_i^{T_2}$ are additionally required. The goal of SCD is to train a semantic change detector $\mathcal{T}_{\theta}^{scd}$ on $\mathcal{F}_{train}^{scd}$ that can predict land-cover maps of multi-temporal images and binary change maps between them on new sets as accurately as possible. By combining the information in predicted land-cover maps and binary change maps, the from-to” semantic change information can be derived.

3) *Building Damage Assessment*: BDA is a special “one-to-many” SCD task [3]. BDA requires recognizing not only the “where” of the change (damage) occurs but also post-event states of the object of interest (building) given the pre-event state of the object of interest. The training set in the BDA task can be denoted as $\mathcal{D}_{train}^{bda} =$

$\left\{ \left(\mathbf{X}_i^{T_1}, \mathbf{X}_i^{T_2}, \mathbf{Y}_i^{loc}, \mathbf{Y}_i^{clf} \right) \right\}_{i=1}^{N_{train}^{bda}}$, where $\mathbf{Y}_i^{loc} \in \{0, 1\}^{H \times W}$ is the label of the object-of-interest (building) at T_1 , and $\mathbf{Y}_i^{clf} \in \{0, 1, \dots, C^{dam}\}^{H \times W}$ is the post-event state (damage level) of the object of interest at T_2 .

C. Network Architecture

For these three CD subtasks, we design the corresponding network frameworks based on the Mamba architecture, called MambaBCD, MambaSCD, and MamabaBDA, respectively. Fig. 1 shows the network framework of these three architectures. Among them, the encoders of the three networks are weight-sharing siamese networks based on the Visual State Space Model (VMamba) architecture [38]. VMamba can adequately extract the robust and representative features of the input images that benefit from the Mamba architecture and an efficient 2D cross-scan mechanism (as shown in Fig. 3). All three architectures have a change detector to learn spatio-temporal relationships from features extracted by the encoder. In addition, MambaSCD and MambaBDA additionally possess semantic decoders for other tasks. The specific structure of the encoders and decoders in our architectures will be elaborated in Sections III-D and III-E. In this subsection, we focus on the input, output, and internal information flow of these three network architectures.

1) *MambaBCD*: MambaBCD is the architecture designed for the BCD task. First, the siamese encoder network denoted as $\mathcal{F}_{\theta}^{enc}$ extracts multi-level features from the input multi-temporal images as $\left\{ \mathbf{F}_{i,j}^{T_1} \right\}_{j=1}^4 = \mathcal{F}_{\theta}^{enc}(\mathbf{X}_i^{T_1})$ and

$\left\{ \mathbf{F}_{i,j}^{\mathrm{T}_2} \right\}_{j=1}^4 = \mathcal{F}_{\theta}^{\mathrm{enc}} \left(\mathbf{X}_i^{\mathrm{T}_2} \right)$. Next, these multi-level features are fed into a tailored change decoder $\mathcal{F}_{\theta}^{\mathrm{cdec}}$. Based on the Mamba architecture, the change decoder can fully learn the spatio-temporal relationship from multi-level features through three different mechanisms, and gradually obtains an accurate BCD result, formulated as $\mathbf{P}_i^{\mathrm{bcd}} = \mathcal{F}_{\theta}^{\mathrm{cdec}} \left(\left\{ \mathbf{F}_{i,j}^{\mathrm{T}_1} \right\}_{j=1}^4, \left\{ \mathbf{F}_{i,j}^{\mathrm{T}_2} \right\}_{j=1}^4 \right)$. The binary change map can be obtained as $\hat{\mathbf{Y}}_i^{\mathrm{bcd}} = \operatorname{argmax}_c \mathbf{P}_i^{\mathrm{bcd}}$.

2) *MambaSCD*: MambaSCD is the architecture designed for the SCD task. As shown in Fig. 1-(b), MambaSCD adds two additional semantic decoders for land cover mapping tasks based on MambaBCD, denoted as $\mathcal{F}_{\theta}^{\mathrm{T}_1}$ and $\mathcal{F}_{\theta}^{\mathrm{T}_2}$. In addition to being treated as input as $\mathcal{F}_{\theta}^{\mathrm{cdec}}$ to learn spatio-temporal relationships to predict BCD results, the multi-level features extracted by the encoder are also fed into $\mathcal{F}_{\theta}^{\mathrm{T}_1}$ and $\mathcal{F}_{\theta}^{\mathrm{T}_2}$ to predict the land-cover map of the corresponding temporal image, formulated as $\mathbf{P}_i^{\mathrm{T}_1} = \mathcal{F}_{\theta}^{\mathrm{T}_1} \left(\left\{ \mathbf{F}_{i,j}^{\mathrm{T}_1} \right\}_{j=1}^4 \right)$ and $\mathbf{P}_i^{\mathrm{T}_2} = \mathcal{F}_{\theta}^{\mathrm{T}_2} \left(\left\{ \mathbf{F}_{i,j}^{\mathrm{T}_2} \right\}_{j=1}^4 \right)$. After obtaining the land-cover maps $\hat{\mathbf{Y}}_i^{\mathrm{T}_1} = \operatorname{argmax}_c \mathbf{P}_i^{\mathrm{T}_1}$ and $\hat{\mathbf{Y}}_i^{\mathrm{T}_2} = \operatorname{argmax}_c \mathbf{P}_i^{\mathrm{T}_2}$ and the binary change map $\hat{\mathbf{Y}}_i^{\mathrm{bcd}}$, the semantic change information of $\mathrm{T}_1 \rightarrow \mathrm{T}_2$ can be obtained by performing the mask operation on $\hat{\mathbf{Y}}_i^{\mathrm{T}_1}$ and $\hat{\mathbf{Y}}_i^{\mathrm{T}_2}$ using $\hat{\mathbf{Y}}_i^{\mathrm{bcd}}$ [6], [24], [45].

3) *MambaBDA*: MambaBDA is the architecture presented for the BDA task. Unlike the typical SCD task, the BDA task only needs to predict land-cover maps (building location maps) for pre-event images. Therefore, in the MambaBDA architecture shown in Fig. 1-(c), there is only one semantic decoder $\mathcal{F}_{\theta}^{\mathrm{loc}}$ to predict the building localization map as $\mathbf{P}_i^{\mathrm{loc}} = \mathcal{F}_{\theta}^{\mathrm{loc}} \left(\left\{ \mathbf{F}_{i,j}^{\mathrm{T}_1} \right\}_{j=1}^4 \right)$. Like MambaBCD and MambaSCD, a change decoder denoted as $\mathcal{F}_{\theta}^{\mathrm{clf}}$ then learns spatio-temporal relationships from multi-temporal features to classify the damage level of buildings as $\mathbf{P}_i^{\mathrm{clf}} = \mathcal{F}_{\theta}^{\mathrm{clf}} \left(\left\{ \mathbf{F}_{i,j}^{\mathrm{T}_1} \right\}_{j=1}^4, \left\{ \mathbf{F}_{i,j}^{\mathrm{T}_2} \right\}_{j=1}^4 \right)$. The obtained building location map $\hat{\mathbf{Y}}_i^{\mathrm{loc}} = \operatorname{argmax}_c \mathbf{P}_i^{\mathrm{loc}}$ can be used for further post-processing on the obtained damage classification map $\hat{\mathbf{Y}}_i^{\mathrm{clf}} = \operatorname{argmax}_c \mathbf{P}_i^{\mathrm{clf}}$ to improve the accuracy, e.g., object-based post-processing methods [3].

D. Encoder Based on Visual State Space Model

Mamba architecture causally processes the input data and thus can only capture information within the scanned part of the data. This attribute naturally aligns with natural language data, which poses significant challenges when adapting to non-causal data such as image [38]. Directly expanding image data along a certain dimension of space and inputting it into the S6 model will result in a lack of adequately modeled spatial contextual information. Recently, VMamba solved this problem well by proposing a 2D cross-scan mechanism. As shown in Fig. 3, before inputting the tokens into the S6 model, the cross-scan mechanism rearranges the tokens in spatial dimensions, i.e., top-left to bottom-right, bottom-right to top-left, top-right to bottom-left, and bottom-left to top-right.

Finally, the resulting features are then merged. In this way, any pixel can get spatial context information from different directions. Moreover, the computational complexity of the VMamba model under the cross-scan mechanism is still $O(N)$ compared to the self-attention operation in Transformer.

Based on the VMamba architecture, the specific structure of the encoder network in our three proposed architectures is shown in Fig. 2. There are four stages, each of which first downsamples the input data, then fully models the spatial contextual information using a number of visual state space (VSS) blocks, and then outputs the features for that stage, i.e., $\mathbf{F}_{i,j}^{\mathrm{T}_1}$ and $\mathbf{F}_{i,j}^{\mathrm{T}_2}$. The structure of the VSS block is also depicted in Fig. 2. The input first passes through a linear embedding layer and the output is split into two information flows. One flow passes through a 3×3 depth-wise convolution (DWConv) layer, followed by a Silu activation function [61] before entering the core SS2D module (i.e., the integration of the S6 model with the cross-scan mechanism). The output of the SS2D module passes through a layer normalization (LN) layer and is then summed up with the outputs of the other streams that have been activated by Silu. This combination produces the final output of the VSS block. Finally, the features extracted for pre-event and post-event images from the four stages $\left\{ \mathbf{F}_{i,j}^{\mathrm{T}_1} \right\}_{j=1}^4$ and $\left\{ \mathbf{F}_{i,j}^{\mathrm{T}_2} \right\}_{j=1}^4$ are then used in the subsequent decoders responsible for specific tasks.

E. Task-Specific Decoders

1) *Spatio-Temporal Relationship Modeling Mechanism*: Although the employed encoder can extract robust and representative features, learning spatio-temporal relationships of multi-temporal images is also significant for CD tasks [24], [46]. To this end, we propose three mechanisms for modeling spatio-temporal relationships that can be aligned with the attribute of the S6 model capable of adequately modeling the global contextual information of long sequential data. Fig. 5 illustrates the three modeling mechanisms. They are spatio-temporal sequential modeling, spatio-temporal cross modeling, and spatial-temporal parallel modeling. Sequential modeling is unfolding the tokens of two temporal phases of data and then arranging them in temporal order as $\mathbf{F}_{i,j}^{\mathrm{seq}} = \left[\mathbf{F}_{i,j}^{\mathrm{T}_1}(1), \dots, \mathbf{F}_{i,j}^{\mathrm{T}_1}\left(\frac{\mathrm{HW}}{2^{1+j}}\right), \mathbf{F}_{i,j}^{\mathrm{T}_2}(1), \dots, \mathbf{F}_{i,j}^{\mathrm{T}_2}\left(\frac{\mathrm{HW}}{2^{1+j}}\right) \right]$. The cross-modeling mechanism, on the other hand, is modeling by cross-ordering the tokens of the two temporal phases as $\mathbf{F}_{i,j}^{\mathrm{crs}} = \left[\mathbf{F}_{i,j}^{\mathrm{T}_1}(1), \mathbf{F}_{i,j}^{\mathrm{T}_2}(1), \dots, \mathbf{F}_{i,j}^{\mathrm{T}_1}\left(\frac{\mathrm{HW}}{2^{1+j}}\right), \mathbf{F}_{i,j}^{\mathrm{T}_2}\left(\frac{\mathrm{HW}}{2^{1+j}}\right) \right]$. Finally, parallel modeling is done by concatenating the tokens of the two temporal phases in the channel dimension and then performing joint modeling, that is, $\mathbf{F}_{i,j}^{\mathrm{pra}} = \operatorname{concat}_c \left(\mathbf{F}_{i,j}^{\mathrm{T}_1}, \mathbf{F}_{i,j}^{\mathrm{T}_2} \right)$. Through these three mechanisms and Mamba architecture, the spatio-temporal relationships intrinsic in the multi-temporal features will be fully explored, helping the change decoder to obtain accurate change detection results.

2) *Change Decoder*: Based on the proposed three spatio-temporal learning mechanisms, the specific structure of the change decoder is then shown in Fig. 4. It fully learns the spatio-temporal relationships from the extracted multi-temporal features $\left\{ \mathbf{F}_{i,j}^{\mathrm{T}_1} \right\}_{j=1}^4$ and $\left\{ \mathbf{F}_{i,j}^{\mathrm{T}_2} \right\}_{j=1}^4$ in four stages

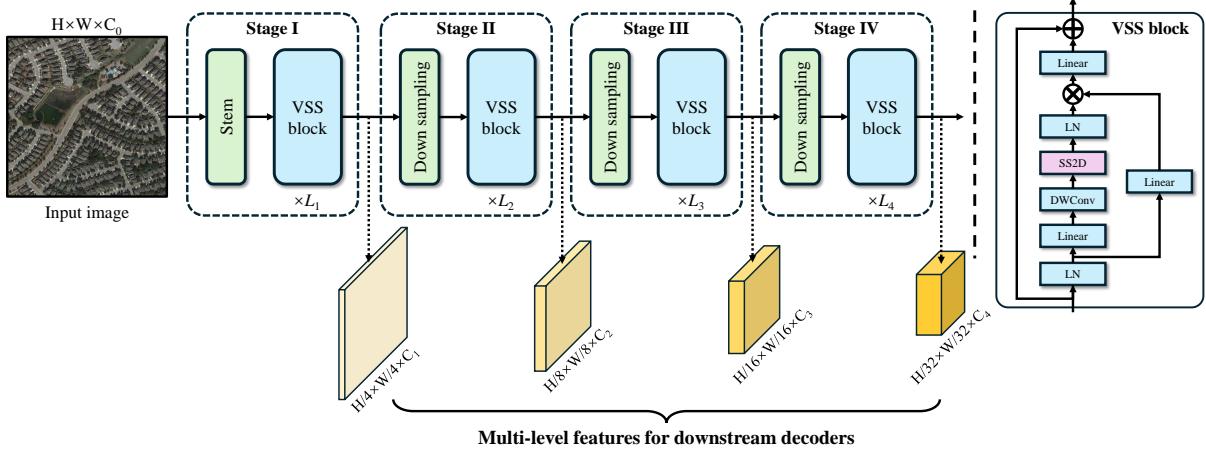


Fig. 2: The specific structure of the encoder network is based on the visual state space model used in our three architectures.

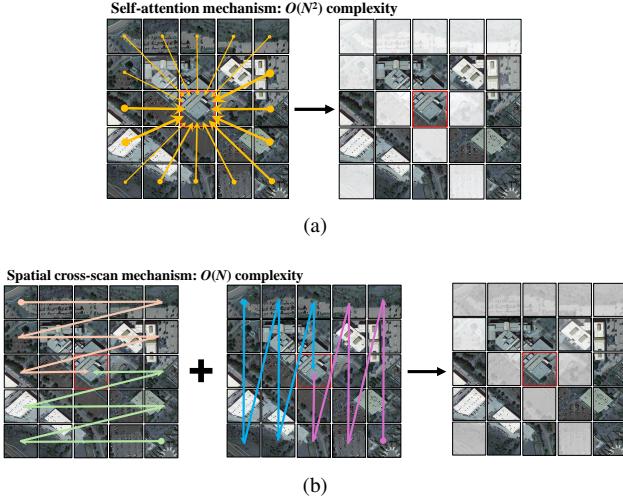


Fig. 3: Illustration of the self-attention mechanism [29], [60] and cross-scan mechanism [38] to capture global contextual information.

and obtains accurate binary change maps. At the beginning of each stage, the spatio-temporal relationship of multi-temporal features is first modeled using a spatio-temporal state space (STSS) block. In the STSS block, a spatio-temporal tokens generator module will rearrange the input multi-temporal features, which are then fed into three VSS blocks. Each block is responsible for learning one of the spatio-temporal relationships in Fig. 5. The output of the STSS block of the current stage is then integrated with the information from the feature map of the previous stage through a fusion module. In the fusion module, the number of channels of the low-level feature map is mapped to be consistent with the high-level feature map by a 1×1 convolutional layer. Then, the high-level and low-level feature maps are summed. Finally, the resulting feature map is smoothed by a residual layer. After passing through an upsampling layer, the feature map is fed to the next stage.

3) *Semantic Decoder*: The specific structure of the semantic decoder is shown in Fig. 6. It is mainly responsible for gradually recovering the land-cover maps from the corre-

sponding multi-level features extracted by the encoder. It also has four stages. At the beginning of each stage, the global spatial context information of the input data is first modeled using a VSS block. Then the feature map is up-sampled and integrated with information about the lower-level feature map with higher resolution through a fusion module. In the last stage, the features output by the VSS block are upsampled four times and used to predict the corresponding land cover map.

F. Loss Function

Since this paper focuses on exploring the potential of the Mamba architecture for CD tasks, we have optimized networks using commonly used loss functions in the field. Some tricks for improving accuracy such as focal loss [62], deep supervision [23], multi-scale training/testing [6], etc. are not adopted. Nevertheless, we can find that the proposed architecture achieves very competitive and even SOTA performance on all three subtasks in the experimental section.

1) *BCD*: Since BCD can be regarded as a special semantic segmentation task [44], we directly optimize the network using the cross-entropy loss as

$$\mathcal{L}_{ce}^{bcd} = -\frac{1}{N_{train}^{bcd}} \sum_{i=1}^{N_{train}^{bcd}} \sum_{c=0}^1 \tilde{\mathbf{Y}}_i^{bcd}(c) \log(\mathbf{P}_i^{bcd}(c)), \quad (5)$$

where $\tilde{\mathbf{Y}}_i^{bcd}$ is the one-hot form of \mathbf{Y}_i^{bcd} , $\mathbf{P}_i^{bcd} = \mathcal{F}_\theta^{bcd}(\mathbf{X}_i^{T_1}, \mathbf{X}_i^{T_2})$ is the output of the binary change detector through the softmax activation.

In addition, the Lovasz-softmax loss [63] is introduced to alleviate the problem of imbalance in the number of samples between changed and unchanged pixels. The final loss function for the BCD task is formulated as

$$\mathcal{L}_{final}^{bcd} = \mathcal{L}_{ce}^{bcd} + \mathcal{L}_{lov}^{bcd}. \quad (6)$$

2) *SCD*: For the SCD task, in addition to optimizing the BCD task, it is also necessary to optimize the land cover mapping task for both pre-event and post-event images, which can be optimized with cross-entropy loss as

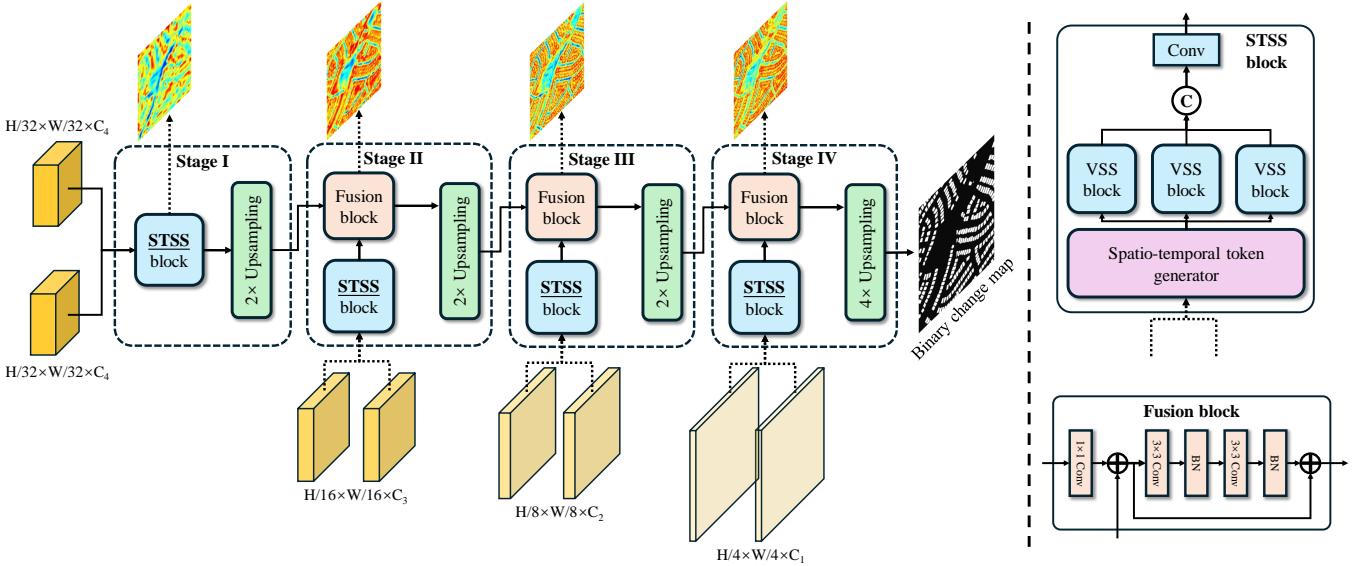


Fig. 4: The specific structure of the change decoder used in MambaBCD, MambaSCD, and MambaBDA architectures based on the proposed spatio-temporal relationship modeling mechanisms in Fig. 5.

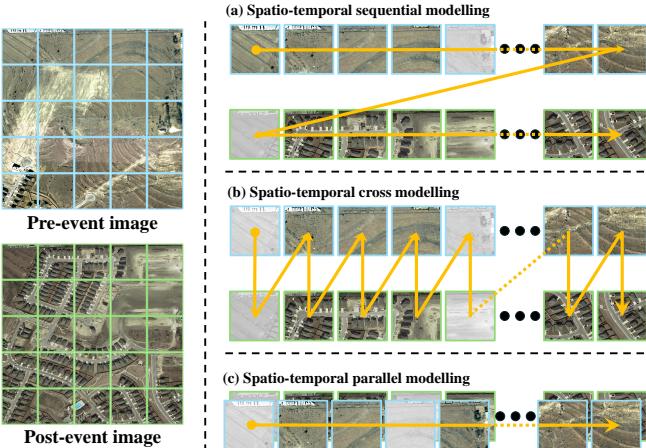


Fig. 5: Illustration of three mechanisms for learning spatio-temporal relationships in the proposed change decoder using Mamba architecture.

$$\begin{cases} \mathcal{L}_{ce}^{T_1} = -\frac{1}{N_{train}^{scd}} \sum_{i=1}^{N_{train}^{scd}} \sum_{c=0}^{C_{lcm}} \tilde{\mathbf{Y}}_i^{T_1}(c) \log(\mathbf{P}_i^{T_1}(c)) \\ \mathcal{L}_{ce}^{T_2} = -\frac{1}{N_{train}^{scd}} \sum_{i=1}^{N_{train}^{scd}} \sum_{c=0}^{C_{lcm}} \tilde{\mathbf{Y}}_i^{T_2}(c) \log(\mathbf{P}_i^{T_2}(c)) \end{cases}, \quad (7)$$

where $\mathbf{P}_i^{T_1}$ and $\mathbf{P}_i^{T_2}$ are the probabilistic map of land-cover classification for the pre-event image and post-event image output by the semantic decoder of the MambaSCD architecture; $\mathcal{L}_{ce}^{T_1}$ and $\mathcal{L}_{ce}^{T_2}$ are the corresponding cross-entropy loss.

Similarly, the Lovasz-softmax loss function is used to solve the sample imbalance problem. Thus, the final loss function

for the SCD task takes the form

$$\mathcal{L}_{final}^{scd} = \mathcal{L}_{ce}^{bcd} + \mathcal{L}_{ce}^{bcd} + \frac{1}{2} (\mathcal{L}_{ce}^{T_1} + \mathcal{L}_{ce}^{T_2} + \mathcal{L}_{lov}^{T_1} + \mathcal{L}_{lov}^{T_2}). \quad (8)$$

3) *BDA*: BDA contains a building location task and a damage classification task. The cross-entropy loss for each task can be expressed as follows

$$\begin{cases} \mathcal{L}_{ce}^{loc} = -\frac{1}{N_{train}^{bda}} \sum_{i=1}^{N_{train}^{bda}} \sum_{c=0}^1 \tilde{\mathbf{Y}}_i^{loc}(c) \log(\mathbf{P}_i^{loc}(c)) \\ \mathcal{L}_{ce}^{clf} = -\frac{1}{N_{train}^{bda}} \sum_{i=1}^{N_{train}^{bda}} \sum_{c=0}^{C_{dam}} \tilde{\mathbf{Y}}_i^{clf}(c) \log(\mathbf{P}_i^{clf}(c)) \end{cases}, \quad (9)$$

where \mathbf{P}_i^{loc} and \mathbf{P}_i^{clf} are the output of localization branch and classification branch of the MambaBDA architecture, respectively; \mathcal{L}_{ce}^{loc} and \mathcal{L}_{ce}^{clf} are the corresponding cross-entropy loss of building localization and damage classification tasks, respectively.

By utilizing the Lovasz-softmax loss function to solve the sample imbalance problem, the final loss function of BDA is formulated as

$$\mathcal{L}_{final}^{bda} = \mathcal{L}_{ce}^{loc} + \mathcal{L}_{ce}^{loc} + \mathcal{L}_{ce}^{clf} + \mathcal{L}_{ce}^{clf}. \quad (10)$$

IV. EXPERIMENTS AND DISCUSSIONS

A. Datasets

1) *SYSU-CD* [19]: This dataset is a category-agnostic CD dataset, which introduces a comprehensive collection of 20,000 pairs of 0.5 meter/pixel resolution aerial images from Hong Kong, spanning 2007 to 2014, to advance the field of CD. This dataset is distinguished by its focus on urban and coastal changes, featuring high-rise buildings and infrastructure developments, where CD poses significant challenges due to shadow and deviation effects. For effective deep learning application, the dataset is structured into training, validation,

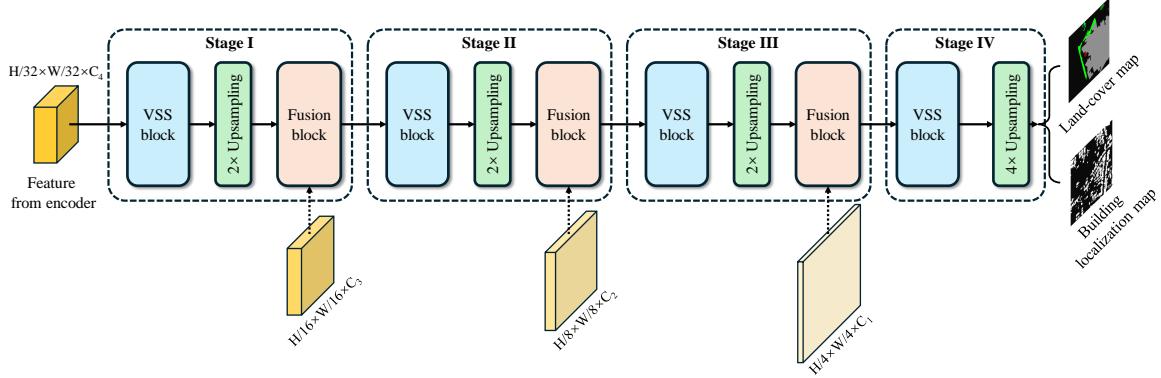


Fig. 6: The specific structure of the semantic decoder in MambaSCD and MambaBDA architectures.

TABLE I: Information of the five benchmark datasets used for experiments.

| Dataset | Study site | Number of image pairs | Size | Resolution | Evaluation task |
|-----------|---------------------------------|-----------------------|-----------------|------------|-------------------------|
| SYSU | Hong Kong | 20,000 | 256 x 256 | 0.5m | BCD (category-agnostic) |
| LEVIR-CD+ | Texas of the US | 985 | 1,024 x 1,024 | 0.5m | BCD (single-category) |
| WHU-CD | Christchurch, New Zealand | 1 | 32,508 x 15,354 | 0.3m | BCD (single-category) |
| SECOND | Hangzhou, Chengdu, and Shanghai | 4,662 | 512 x 512 | 0.5-3m | SCD |
| xBD | 15 countries | 22,068 | 1,024 x 1,024 | <0.8m | BDA |

and test sets following a 6:2:2 ratio, with 20,000 patches of 256×256 . This dataset encompasses a range of various change scenarios such as urban construction, suburban expansion, groundwork, vegetation changes, road expansion, and sea construction.

2) *WHU-CD* [64]: The WHU-CD dataset, a subset of the larger WHU Building dataset, is tailored for the building CD task. It comprises two aerial datasets from Christchurch, New Zealand, captured in April 2012 and 2016, with a spatial resolution of 0.3 meters/pixel. This dataset is particularly focused on detecting changes in large and sparse building structures. The aerial images captured in 2012 cover an area of 20.5 km^2 , featuring 12,796 buildings, while the image captured in 2016 shows an increase to 16,077 buildings within the same area, reflecting significant urban development over the four-year period. We follow the official protocol to split the dataset into training ($21,243 \times 15,354$) area and testing ($11,265 \times 15,354$) area for our experiments.

3) *LEVIR-CD+* [65]: This advanced version of the LEVIR-CD dataset is a building CD dataset. It comprises 985 pairs of very high-resolution images at 0.5 meters/pixel, each with dimensions of 1024×1024 pixels. Spanning a time interval of 5 to 14 years, these multi-temporal images document significant building construction changes. It also encompasses a wide array of building types, including villa residences, tall apartments, small garages, and large warehouses, with a focus on both the emergence of new buildings and the decline of existing structures. The final dataset features 31,333 instances of building changes, making LEVIR-CD+ a valuable benchmark for evaluating CD methodologies.

4) *SECOND* [20]: The SECOND dataset introduces a collection of 4,662 pairs of 512×512 aerial images at 0.5-3 meter/pixel resolution, annotated for SCD in cities such as

Hangzhou, Chengdu, and Shanghai. It focuses on six primary land cover classes: non-vegetated ground surfaces, trees, low vegetation, water, buildings, and playgrounds to capture a wide range of geographical changes. Annotations are done by experts, ensuring high accuracy at the pixel level. The dataset uses land-cover map pairs and non-change masks to accurately represent change, facilitating the differentiation of changes from unchanged pixels within the same class. This approach ensures its utility in addressing SCD challenges by providing a diverse and accurately labeled dataset for model training and evaluation.

5) *xBD* [51]: The xBD dataset, pivotal for advancing BDA in disaster response, integrates pre- and post-disaster satellite imagery with over 850,000 building annotations across $45,362 \text{ km}^2$. It stands as the largest dataset of its kind, enriched by environmental data and detailed damage levels for a broad spectrum of disaster events. The dataset adheres to stringent criteria, featuring high-resolution imagery (better than 0.8 meter/pixel), encapsulating multiple levels of damage (“no damage”, “minor damage”, “major damage”, and “destroyed”), and covering a diverse range of disaster types (earthquake, tsunami, storm, wildfire, volcano, and flood). By including both damaged and undamaged sites, xBD is crafted to foster the creation of adaptive models that effectively navigate the complexities of various disaster scenarios.

The information of these five datasets is summarized in Table I.

B. Experimental Setup

1) *Implementation Details*: All three of our proposed architectures are implemented in Pytorch. Depending on the size and depth of the encoder network, our proposed MambaBCD, MambaSCD, and MambaBDA architectures are available in

TABLE II: Number of layers and feature channels for encoders in ChangeMamba series.

| Stage | Tiny | Small | Base |
|-------|-------------|-------------|--------------|
| I | $L_1 = 2$ | $L_1 = 2$ | $L_1 = 2$ |
| | $C_1 = 96$ | $C_1 = 96$ | $C_1 = 128$ |
| II | $L_2 = 2$ | $L_2 = 2$ | $L_2 = 2$ |
| | $C_2 = 192$ | $C_2 = 192$ | $C_2 = 256$ |
| III | $L_3 = 9$ | $L_3 = 27$ | $L_3 = 27$ |
| | $C_3 = 384$ | $C_3 = 384$ | $C_3 = 512$ |
| IV | $L_4 = 2$ | $L_4 = 2$ | $L_4 = 2$ |
| | $C_4 = 768$ | $C_4 = 768$ | $C_4 = 1024$ |

Tiny, Small, and Base versions [38]. As shown in Table II, the difference between them is the number of VSS blocks inside each stage and the number of channels in the feature map. For datasets other than the SYSU dataset, the multi-temporal image pairs and associated labels are cropped to 256×256 pixels for input to the network, and then the data with the original size are inferred using the trained networks on the test set. During training, we optimize the network using the AdamW optimizer [66] with a learning rate of $1e^{-4}$ and a weight decay of $5e^{-3}$. The batch size is set to 16. Except for the SYSU dataset, for which we set the number of training iterations to 20000, we set the number of training iterations to 50000 on the remaining four datasets. Random rotation, left-right, and top-bottom flip are used as training data augmentation methods. Our source code is open-sourced for community reproduction and subsequent research in <https://github.com/ChenHongruixuan/MambaCD>.

2) *Evaluation Metrics*: For evaluating model performance across different CD subtasks, we employ a set of metrics tailored to the specific requirements of BCD, SCD and BDA. In evaluating the BCD performance of the model, we use the following metrics:

- **Recall (Rec)**: The proportion of all true positive pixels correctly identified as positive

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (11)$$

where TP represents true positives and FN denotes false negatives.

- **Precision (Pre)**: Reflecting the proportion of true positive pixels among all pixels identified as positive

$$\text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (12)$$

with FP standing for false positives.

- **Overall Accuracy (OA)**: The ratio of correctly predicted pixels to total pixels is expressed as

$$\text{OA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (13)$$

where TN indicates true negatives.

- **F1 Score (F1)**: Balancing both precision and recall by calculating their harmonic mean, given by

$$\text{F1} = 2 \times \frac{\text{Pre} \times \text{Rec}}{\text{Pre} + \text{Rec}}, \quad (14)$$

- **Intersection over Union (IoU)**: Also known as the Jaccard index, this metric measures the overlap between the predicted and actual positives:

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \quad (15)$$

- **Cohen's Kappa (KC)**: Assessing the agreement between two raters beyond chance, Cohen's Kappa is given by

$$\kappa_c = \frac{p_o - p_e}{1 - p_e}, \quad (16)$$

where p_o is the observed agreement, and p_e is the expected agreement by chance. Their detailed definition can be found in [14].

For SCD, we follow the existing setup in the literature [55] to use the following metrics:

- **Overall Accuracy (OA) and F1 Score (F1)**: As defined for BCD but extended into multi-class cases.
- **Mean Intersection over Union (mIoU)**: Averaging IoU across classes, mIoU is calculated as

$$\text{mIoU} = \frac{1}{\mathbf{C}^{lcm}} \sum_{i=1}^{\mathbf{C}^{lcm}} \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i + \text{FN}_i}, \quad (17)$$

where \mathbf{C}^{lcm} is the number of semantic categories. mIoU can provide a balanced measure across different semantic categories.

- **Separated Kappa (SeK) coefficient**: Specifically proposed in [20] to alleviate the influences of label imbalance in SCD model evaluation, SeK is defined in a cited work.

For BDA, we follow the protocol of the xView2 Challenge [51] to split BDA into two subtasks, i.e., building localization and damage classification tasks. The following metrics are adopted to measure the performance of each subtask and the overall BDA performance:

- **F1 Score for Building Localization (F_1^{loc})**: This metric assesses the model's accuracy in the building localization task.
- **F1 Score for Different Damage Level Classes ($F_1^{level_i}$)**: This evaluates the model's performance across various damage levels.
- **F1 Score for Damage Classification (F_1^{clf})**: This metric evaluates the overall performance of the model on the damage classification task and is defined as the harmonic mean of $F_1^{level_i}$

$$F_1^{clf} = \frac{1}{\frac{1}{\mathbf{C}^{dam}} \sum_{i=1}^{\mathbf{C}^{dam}} \frac{1}{F_1^{level_i}}}, \quad (18)$$

where the value of \mathbf{C}^{dam} in the xBD dataset is 4, from 1 to 4 being “no damage”, “minor damage”, “major damage”, and “destroyed”, respectively.

- **Overall F1 Score ($F_1^{overall}$)**: This reflects the combined performance of the model in building location and damage classification tasks. It is defined as

$$F_1^{overall} = 0.3F_1^{loc} + 0.7F_1^{clf}. \quad (19)$$

Employing this collectively comprehensive set of metrics ensures a balanced and thorough evaluation, offering a detailed

TABLE III: Accuracy assessment for different binary CD models on the SYSU dataset, where \mathcal{C} indicates CNN-based models, \mathcal{T} indicates Transformer-based models, and \mathcal{M} indicates Mamba-based models. The table highlights the highest values in red, and the second-highest results in blue

| Type | Method | Rec | Pre | OA | F1 | IoU | KC |
|---------------|---------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| \mathcal{C} | FC-EF [22] | 75.17 | 76.47 | 88.69 | 75.81 | 61.04 | 68.43 |
| | FC-Siam-Diff [22] | 75.30 | 76.28 | 88.65 | 75.79 | 61.01 | 68.38 |
| | FC-Siam-Conc [22] | 76.75 | 73.67 | 88.05 | 75.18 | 60.23 | 67.32 |
| | SiamCRNN-18 [17] | 76.83 | 84.80 | 91.29 | 80.62 | 67.54 | 75.02 |
| | SiamCRNN-34 [17] | 76.85 | 85.13 | 91.37 | 80.78 | 67.75 | 75.23 |
| | SiamCRNN-50 [17] | 78.40 | 83.41 | 91.23 | 80.83 | 67.82 | 75.15 |
| | SiamCRNN-101 [17] | 80.48 | 80.40 | 90.77 | 80.44 | 67.28 | 74.40 |
| | SNUNet [40] | 72.21 | 74.09 | 87.49 | 73.14 | 57.66 | 64.99 |
| | DSIFN [23] | 82.02 | 75.83 | 89.59 | 78.80 | 65.02 | 71.92 |
| | HANet [41] | 76.14 | 78.71 | 89.52 | 77.41 | 63.14 | 70.59 |
| \mathcal{T} | CGNet [42] | 74.37 | 86.37 | 91.19 | 79.92 | 66.55 | 74.31 |
| | ChangeFormerV1 [30] | 75.82 | 79.65 | 89.73 | 77.69 | 63.52 | 71.02 |
| | ChangeFormerV2 [30] | 75.62 | 78.14 | 89.26 | 76.86 | 62.42 | 69.87 |
| | ChangeFormerV3 [30] | 75.24 | 79.46 | 89.57 | 77.29 | 62.99 | 70.53 |
| | ChangeFormerV4 [30] | 77.90 | 79.74 | 90.12 | 78.81 | 65.03 | 72.37 |
| | ChangeFormerV5 [30] | 74.00 | 77.26 | 88.73 | 75.60 | 60.77 | 68.28 |
| | ChangeFormerV6 [30] | 72.38 | 81.70 | 89.67 | 76.76 | 62.29 | 70.15 |
| | BIT-18 [28] | 76.42 | 84.85 | 91.22 | 80.41 | 67.24 | 74.78 |
| | BIT-34 [28] | 74.63 | 82.40 | 90.26 | 78.32 | 64.37 | 72.06 |
| | BIT-50 [28] | 77.90 | 81.42 | 90.60 | 79.62 | 66.14 | 73.51 |
| \mathcal{M} | BIT-101 [28] | 75.58 | 83.64 | 90.76 | 79.41 | 65.84 | 73.47 |
| | TransUNetCD [33] | 77.73 | 82.59 | 90.88 | 80.09 | 66.79 | 74.18 |
| | SwinSUNet [32] | 79.75 | 83.50 | 91.51 | 81.58 | 68.89 | 76.06 |
| | MambaBCD-Tiny | 79.59 | 83.06 | 91.36 | 81.29 | 68.48 | 75.68 |
| \mathcal{M} | MambaBCD-Small | 78.25 | 87.99 | 92.35 | 82.83 | 70.70 | 77.94 |
| | MambaBCD-Base | 80.31 | 86.11 | 92.30 | 83.11 | 71.10 | 78.13 |

assessment of the model’s capabilities across a range of CD tasks.

C. Comparison Methods

To show the superiority of our Mamba-based architecture in CD tasks, we select a number of state-of-the-art methods based on CNN and Transformer architectures to compare across three distinct tasks: BCD, SCD, and BDA.

For BCD, our comparison methods include a suite of CNN-based methods: FC-EF [22], FC-Siam-Diff [22], FC-Siam-Conc [22], SiamCRNN [17] (utilizing ResNet architectures ranging from ResNet-18 to ResNet-101 [67]), SNUNet [40], DSIFN [23], HANet [41], and CGNet [42]. For the Transformer-based methods, we compare against ChangeFormer (versions 1 to 6) [30], BIT (with ResNet-18 to ResNet-101 as the backbone for feature extraction) [28], TransUNetCD [33], and SwinSUNet [32].

For the SCD arena, CNN-based competitors encompass HRSCD (variants S1 to S4) [45], ChangeMask [24], SSCD-1 [47], Bi-SRNet [47], and TED [55]. For Transformer-based methods, we evaluate against SMNet [54] and ScanNet [55].

Lastly, for the BDA task, we selected Siamese-UNet [51], MTF [52], and ChangeOS [3] (employing ResNet-18 to ResNet-101) as CNN-based benchmarks. DamFormer [31] stands as the solitary Transformer-based comparison method.

For the above methods, if the detection performance is reported in the original paper on the corresponding dataset, we directly adopt the accuracy in the original paper. If not, we train and test the method on the corresponding dataset based on the hyperparameters recommended in the original paper, using the loss function and data augmentation methods consistent with our method.

D. Detection Results and Benchmark Comparison in Three Tasks

1) *BCD*: Tables III to V list the performance in BCD of the proposed MambaBCD architecture and some comparison methods on the three benchmark datasets. In these tables, we differentiate between the types of these BCD architectures into CNN-based methods, Transformer-based methods, and our Mamba-based methods. It can be seen that our approaches significantly outperform either the CNN-based approaches or the Transformer-based architectures. The proposed MambaBCD-Base and MambaBCD-Small architectures achieve the highest and second-highest OA, F1, IoU, and KC for both category-agnostic CD (SYSU dataset) and single-category CD tasks (LEVIR-CD+ and WHU-CD datasets), fully demonstrating the potential of the Mamba architecture for the BCD task. Fig. 7 to Fig. 9 then show some binary change maps predicted by our three architectures on the test sets of the three datasets. It can be seen that the proposed methods accurately detect

TABLE IV: Accuracy assessment for different binary CD models on the LEVIR-CD+ dataset, where \mathcal{C} indicates CNN-based models, \mathcal{T} indicates Transformer-based models, and \mathcal{M} indicates Mamba-based models. The table highlights the highest values in **red**, and the second-highest results in **blue**.

| Type | Method | Rec | Pre | OA | F1 | IoU | KC |
|---------------|---------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| \mathcal{C} | FC-EF [22] | 71.77 | 69.12 | 97.54 | 70.42 | 54.34 | 69.14 |
| | FC-Siam-Diff [22] | 74.02 | 81.49 | 98.26 | 77.57 | 63.36 | 76.67 |
| | FC-Siam-Conc [22] | 78.49 | 78.39 | 98.24 | 78.44 | 64.53 | 77.52 |
| | SiamCRNN-18 [17] | 84.25 | 81.22 | 98.56 | 82.71 | 70.52 | 81.96 |
| | SiamCRNN-34 [17] | 83.88 | 82.28 | 98.61 | 83.08 | 71.05 | 82.35 |
| | SiamCRNN-50 [17] | 81.61 | 85.39 | 98.68 | 83.46 | 71.61 | 82.77 |
| | SiamCRNN-101 [17] | 80.96 | 85.56 | 98.67 | 83.20 | 71.23 | 82.50 |
| | DSIFN [23] | 84.36 | 83.78 | 98.70 | 84.07 | 72.52 | 83.39 |
| | SNUNet [40] | 78.73 | 71.07 | 97.83 | 74.70 | 59.62 | 73.57 |
| | HANet [41] | 75.53 | 79.70 | 98.22 | 77.56 | 63.34 | 76.63 |
| \mathcal{T} | CGNet [42] | 86.02 | 81.46 | 98.63 | 83.68 | 71.94 | 82.97 |
| | ChangeFormerV1 [30] | 77.00 | 82.18 | 98.38 | 79.51 | 65.98 | 78.66 |
| | ChangeFormerV2 [30] | 81.32 | 79.10 | 98.36 | 80.20 | 66.94 | 79.35 |
| | ChangeFormerV3 [30] | 79.97 | 81.34 | 98.44 | 80.65 | 67.58 | 79.84 |
| | ChangeFormerV4 [30] | 76.68 | 75.07 | 98.01 | 75.87 | 61.12 | 74.83 |
| | ChangeFormerV5 [30] | 77.96 | 78.50 | 98.23 | 78.23 | 64.24 | 77.31 |
| | ChangeFormerV6 [30] | 78.57 | 67.66 | 97.60 | 72.71 | 57.12 | 71.46 |
| | BIT-18 [28] | 80.86 | 83.76 | 98.58 | 82.28 | 69.90 | 81.54 |
| | BIT-34 [28] | 80.96 | 85.87 | 98.68 | 83.34 | 71.44 | 82.66 |
| | BIT-50 [28] | 81.84 | 85.02 | 98.67 | 83.40 | 71.53 | 82.71 |
| \mathcal{M} | BIT-101 [28] | 81.20 | 83.91 | 98.60 | 82.53 | 70.26 | 81.80 |
| | TransUNetCD [33] | 84.18 | 83.08 | 98.66 | 83.63 | 71.86 | 82.93 |
| | SwinSUNet [32] | 85.85 | 85.34 | 98.92 | 85.60 | 74.82 | 84.98 |
| | MambaBCD-Tiny | 81.79 | 85.51 | 98.69 | 83.61 | 71.83 | 82.93 |
| | MambaBCD-Small | 86.49 | 89.17 | 99.02 | 87.81 | 78.27 | 87.30 |
| | MambaBCD-Base | 87.57 | 89.24 | 99.06 | 88.39 | 79.20 | 87.91 |

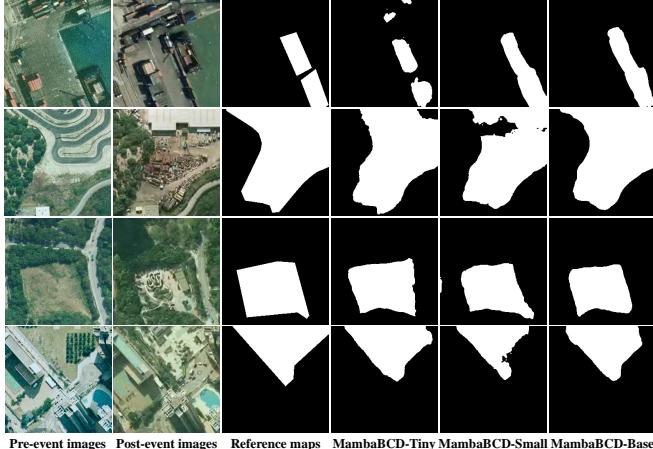


Fig. 7: Some binary change maps obtained by our methods on the test set of the SYSU dataset.

changes with varying types, scales, and numbers contained in these image pairs, very close to the change reference maps.

Table VI lists the network parameters and GFLOPs of these BCD architectures. Benefiting from the S6 model, the MambaBCD architecture can have only linear computational complexity while modeling global context information. It can be seen that even the largest model MambaBCD-Base has

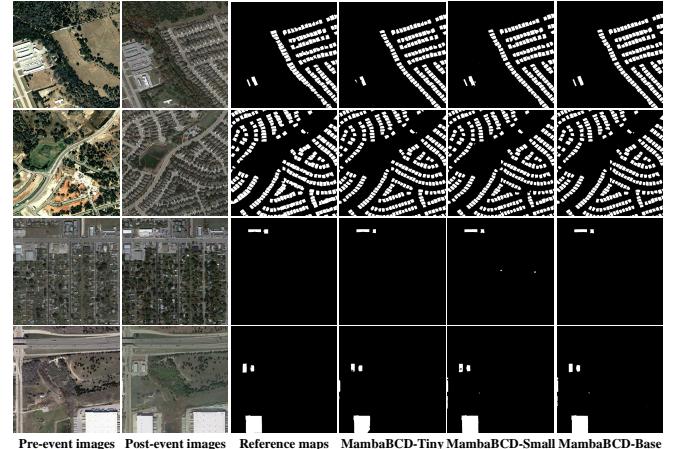


Fig. 8: Some binary change maps obtained by our methods on the test set of the LEVIR-CD+ dataset.

GFLOPs of 179.32, which is lower than many CNN and Transformer architectures, like DSIFN, CFNet, and ChangeFormerV4, comparable to some CNN SOTA methods such as SNUNet and SiamCRNN-50. However, the accuracy advantage is obvious. In particular, the MambaBCD-Tiny architecture has a clear accuracy advantage over other architectures with a similar number of parameters and computational com-

TABLE V: Accuracy assessment for different binary CD models on the WHU-CD dataset, where \mathcal{C} indicates CNN-based models, \mathcal{T} indicates Transformer-based models, and \mathcal{M} indicates Mamba-based models. The table highlights the highest values in **red**, and the second-highest results in **blue**

| Type | Method | Rec | Pre | OA | F1 | IoU | KC |
|---------------|---------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| \mathcal{C} | FC-EF [22] | 86.33 | 83.50 | 98.87 | 84.89 | 73.74 | 84.30 |
| | FC-Siam-Diff [22] | 84.69 | 90.86 | 99.13 | 87.67 | 78.04 | 87.22 |
| | FC-Siam-Conc [22] | 87.72 | 84.02 | 98.94 | 85.83 | 75.18 | 85.28 |
| | SiamCRNN-18 [17] | 90.48 | 91.56 | 99.34 | 91.02 | 83.51 | 90.68 |
| | SiamCRNN-34 [17] | 89.10 | 93.88 | 99.39 | 91.42 | 84.20 | 91.11 |
| | SiamCRNN-50 [17] | 91.45 | 86.70 | 99.30 | 90.57 | 82.76 | 90.20 |
| | SiamCRNN-101 [17] | 90.45 | 87.79 | 99.19 | 89.10 | 80.34 | 88.68 |
| | DSIFN [23] | 83.45 | 97.46 | 99.31 | 89.91 | 81.67 | 89.56 |
| | SNUNet [40] | 87.36 | 88.04 | 99.10 | 87.70 | 78.09 | 87.23 |
| | HANet [41] | 88.30 | 88.01 | 99.16 | 88.16 | 78.82 | 87.72 |
| \mathcal{T} | CGNet [42] | 90.79 | 94.47 | 99.48 | 92.59 | 86.21 | 92.33 |
| | ChangeFormerV1 [30] | 84.30 | 90.80 | 99.11 | 87.43 | 77.67 | 86.97 |
| | ChangeFormerV2 [30] | 83.41 | 88.77 | 99.00 | 86.00 | 75.45 | 85.49 |
| | ChangeFormerV3 [30] | 85.55 | 88.25 | 99.05 | 86.88 | 76.80 | 86.39 |
| | ChangeFormerV4 [30] | 84.85 | 90.09 | 99.10 | 87.39 | 77.61 | 86.93 |
| | ChangeFormerV5 [30] | 84.87 | 90.20 | 99.11 | 87.45 | 77.70 | 86.99 |
| | ChangeFormerV6 [30] | 81.90 | 85.49 | 98.83 | 83.66 | 71.91 | 83.05 |
| | BIT-18 [28] | 90.36 | 90.30 | 99.29 | 90.33 | 82.37 | 89.96 |
| | BIT-34 [28] | 90.10 | 89.14 | 99.23 | 89.62 | 81.19 | 89.22 |
| | BIT-50 [28] | 90.33 | 89.70 | 99.26 | 90.01 | 81.84 | 89.63 |
| \mathcal{M} | BIT-101 [28] | 90.24 | 89.83 | 99.27 | 90.04 | 81.88 | 89.66 |
| | TransUNetCD [33] | 90.50 | 85.48 | 99.09 | 87.79 | 78.44 | 87.44 |
| | SwinSUNet [32] | 92.03 | 94.08 | 99.50 | 93.04 | 87.00 | 92.78 |
| | MambaBCD-Tiny | 91.94 | 94.76 | 99.52 | 93.33 | 87.49 | 93.08 |
| | MambaBCD-Small | 92.29 | 95.90 | 99.57 | 94.06 | 88.79 | 93.84 |
| | MambaBCD-Base | 92.23 | 96.18 | 99.58 | 94.19 | 89.02 | 93.98 |

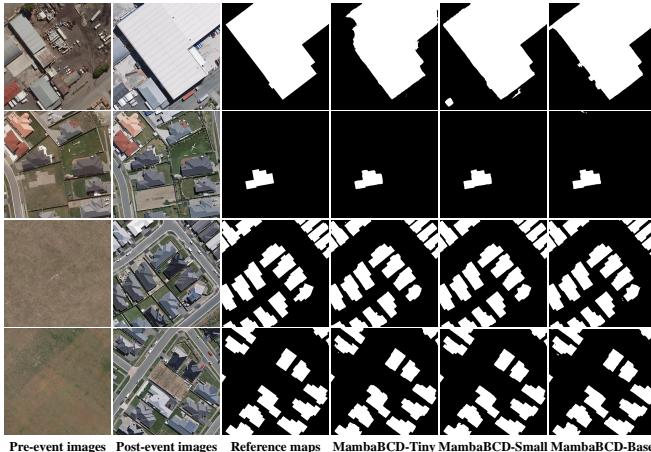


Fig. 9: Some binary change maps obtained by our methods on the test set of the WHU-CD dataset.

sumption. For example, compared to BIT-18, MambaBCD-Tiny has 56.9% lower GFLOPs but 0.88%, 1.33% and 3.00% higher F1 scores on the three datasets.

2) *SCD*: Table VII compares the SCD performance of the proposed MambaSCD architectures and representative SCD approaches. Compared to the BCD task, the SCD task requires the change detector capable of effectively modeling

not only the spatio-temporal relationships of multi-temporal images, but also learning representative and discriminative representations for accurate land-cover mapping. From Table VII, we can see that the MambaSCD architecture can satisfy both requirements well. The MambaSCD-Small architecture surpasses the Transformer-based SOTA approach, ScanNet, on all four metrics of the SCD task, with OA of 88.38%, F1 of 64.10%, mIoU of 73.61%, and SeK of 24.04 %. Note that our methods have achieved the current accuracy without employing loss functions such as those proposed for the SCD task in ScanNet [55].

Based on the SCD results, the detailed “from-to” semantic change information can be obtained, which is presented in Fig. 10. Firstly, we can find that the “from-to” transition matrices obtained by our methods are very close to the ground truth, demonstrating the effectiveness of our MambaSCD architecture again. Then, based on the contents of the matrices, we can find that the three semantic changes with the highest percentage between the two temporal phases are “non-vegetated ground surface \rightarrow building”, “low vegetation \rightarrow building”, “low vegetation \rightarrow non-vegetated ground surface”. These results indicate that the areas covered by the SECOND dataset was in the process of urbanization during this period.

Table X further lists the number of parameters and GFLOPs of these methods. Compared to the Transformer-based ScanNet method, the GFLOPs of our three architectures are lower.

TABLE VI: Comparison of different binary CD models in computational cost, where \mathcal{C} indicates CNN-based models, \mathcal{T} indicates Transformer-based models, and \mathcal{M} indicates Mamba-based models

| Type | Method | Params (M) | GFLOPs |
|---------------|---------------------|------------|--------|
| \mathcal{C} | FC-EF [22] | 1.35 | 14.13 |
| | FC-Siam-Diff [22] | 1.35 | 18.66 |
| | FC-Siam-Conc [22] | 1.54 | 21.07 |
| | SiamCRNN-18 [17] | 18.85 | 86.64 |
| | SiamCRNN-34 [17] | 28.96 | 113.28 |
| | SiamCRNN-50 [17] | 44.45 | 185.30 |
| | SiamCRNN-101 [17] | 63.44 | 224.30 |
| | DSIFN [23] | 35.73 | 329.03 |
| | SNUNet [40] | 10.21 | 176.36 |
| | HANet [41] | 2.61 | 70.68 |
| | CGNet [42] | 33.68 | 329.58 |
| \mathcal{T} | ChangeFormerV1 [30] | 29.84 | 46.62 |
| | ChangeFormerV2 [30] | 24.30 | 44.54 |
| | ChangeFormerV3 [30] | 24.30 | 33.68 |
| | ChangeFormerV4 [30] | 33.61 | 852.53 |
| | ChangeFormerV5 [30] | 55.27 | 841.08 |
| | ChangeFormerV6 [30] | 41.03 | 811.15 |
| | TransUNetCD [33] | 28.37 | 244.54 |
| | SwinSUNet [32] | 39.28 | 43.50 |
| | BIT-18 [28] | 11.50 | 106.14 |
| | BIT-34 [28] | 21.61 | 190.83 |
| \mathcal{M} | BIT-50 [28] | 24.28 | 224.61 |
| | BIT-101 [28] | 43.27 | 380.62 |
| | MambaBCD-Tiny | 17.13 | 45.74 |
| | MambaBCD-Small | 49.94 | 114.82 |
| | MambaBCD-Base | 84.70 | 179.32 |

TABLE VII: Accuracy assessment for different SCD models on the SECOND dataset, where \mathcal{C} indicates CNN-based models, \mathcal{T} indicates Transformer-based models, and \mathcal{M} indicates Mamba-based models. The table highlights the highest values in red, and the second-highest results in blue. Note that some methods like ScanNet [55] adopted pseudo-label learning to improve accuracy, while ours do not.

| Type | Method | OA | F1 | mIoU | SeK |
|---------------|-----------------|--------------|-------|-------|-------|
| \mathcal{C} | HRSCD-S1 [45] | 45.77 | 38.44 | 62.72 | 5.90 |
| | HRSCD-S2 [45] | 85.49 | 49.22 | 64.43 | 10.69 |
| | HRSCD-S3 [45] | 84.62 | 51.62 | 66.33 | 11.97 |
| | HRSCD-S4 [45] | 86.62 | 58.21 | 71.15 | 18.80 |
| | ChangeMask [24] | 86.93 | 59.74 | 71.46 | 19.50 |
| | SSCD-1 [47] | 87.19 | 61.22 | 72.60 | 21.86 |
| | Bi-SRNet [47] | 87.84 | 62.61 | 73.41 | 23.22 |
| | TED [55] | 87.39 | 60.34 | 72.79 | 22.17 |
| | \mathcal{T} | SMNet [54] | 86.68 | 60.34 | 71.95 |
| | | ScanNet [55] | 87.86 | 63.66 | 73.42 |
| \mathcal{M} | MamabaSCD-Tiny | 88.07 | 63.44 | 73.33 | 23.34 |
| | MamabaSCD-Small | 88.38 | 64.10 | 73.61 | 24.04 |
| | MamabaSCD-Base | 88.12 | 64.03 | 73.68 | 24.11 |

Compared to SMNet, another Transformer-based method, the proposed MamabaSCD-Tiny can achieve higher SeK values with a lower number of parameters and GFLOPs. These results

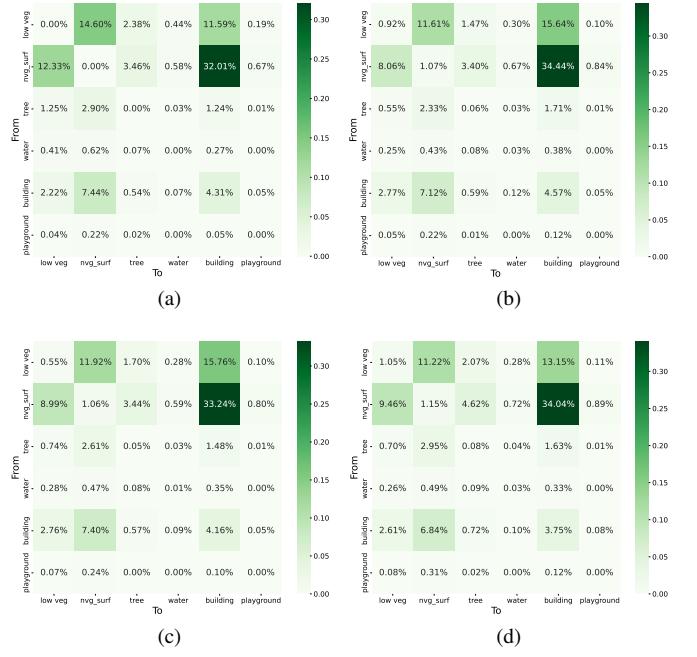


Fig. 10: Semantic change matrices providing “from-to” transition information derived from the SCD results. (a) Reference. (b) MambaSCD-Tiny. (c) MambaSCD-Small. (d) MambaSCD-Base.

TABLE VIII: Comparison of different SCD models in computational cost, where \mathcal{C} indicates CNN-based models, \mathcal{T} indicates Transformer-based models, and \mathcal{M} indicates Mamba-based models.

| Type | Method | Params | GFLOPs |
|---------------|-----------------|--------------|--------|
| \mathcal{C} | HRSCD-S1 [45] | 3.36 | 8.02 |
| | HRSCD-S2 [45] | 6.39 | 14.29 |
| | HRSCD-S3 [45] | 12.77 | 42.67 |
| | HRSCD-S4 [45] | 13.71 | 43.69 |
| | ChangeMask [24] | 2.97 | 37.16 |
| | SSCD-1 [47] | 23.31 | 189.57 |
| | Bi-SRNet [47] | 23.39 | 189.91 |
| | TED [55] | 24.19 | 204.29 |
| | \mathcal{T} | SMNet [54] | 42.16 |
| | | ScanNet [55] | 27.90 |
| \mathcal{M} | MamabaSCD-Tiny | 21.51 | 73.42 |
| | MamabaSCD-Small | 54.28 | 146.70 |
| | MamabaSCD-Base | 89.99 | 211.55 |

demonstrate the effectiveness of the proposed methods and the potential of the Mamba architecture on the SCD task.

Fig. 11 shows some semantic change maps on the test set of the SECOND dataset. Due to land-cover category combinations, SCD often involves a considerable number of categories. For ease of visualization, we follow the visualization way in [20], [24], [68], that is, only plotting the land-cover categories of the changed regions in both pre- and post-event images. In this way, the “from-to” semantic change information can be effectively reflected. From Fig. 11, we can observe that the MambaSCD architecture is not only capable of distinguishing changed features at different scales but also

TABLE IX: Accuracy assessment for different building damage assessment models on the xBD dataset, where \mathcal{C} indicates CNN-based models, \mathcal{T} indicates Transformer-based models, and \mathcal{M} indicates Mamba-based models. The table highlights the highest values in **red**, and the second-highest results in **blue**. The suffix PPS means post-processing.

| Type | Method | F_1^{loc} | F_1^{clf} | $F_1^{overall}$ | Damage F_1 per class | | | |
|---------------|----------------------|--------------|--------------|-----------------|------------------------|--------------|--------------|--------------|
| | | | | | No damage | Minor damage | Major damage | Destroyed |
| \mathcal{C} | Siamese-UNet [51] | 85.92 | 65.58 | 71.68 | 86.74 | 50.02 | 64.43 | 71.68 |
| | MTF [52] | 83.60 | 70.02 | 74.10 | 90.60 | 49.30 | 72.20 | 83.70 |
| | ChangeOS-18 [3] | 84.62 | 69.87 | 74.30 | 88.61 | 52.10 | 70.36 | 79.65 |
| | ChangeOS-34 [3] | 85.16 | 70.28 | 74.74 | 88.63 | 52.38 | 71.16 | 80.08 |
| | ChangeOS-50 [3] | 85.41 | 70.88 | 75.24 | 88.98 | 53.33 | 71.24 | 80.60 |
| | ChangeOS-101 [3] | 85.69 | 71.14 | 75.50 | 89.11 | 53.11 | 72.44 | 80.79 |
| | ChangeOS-18-PPS [3] | 84.62 | 73.89 | 77.11 | 92.38 | 57.41 | 72.54 | 82.62 |
| | ChangeOS-34-PPS [3] | 85.16 | 74.25 | 77.52 | 92.19 | 58.07 | 72.84 | 82.79 |
| | ChangeOS-50-PPS [3] | 85.41 | 75.64 | 78.57 | 92.66 | 60.14 | 74.18 | 83.45 |
| \mathcal{T} | ChangeOS-101-PPS [3] | 85.69 | 75.44 | 78.52 | 92.81 | 59.38 | 74.65 | 83.29 |
| | DamFormer [31] | 86.86 | 72.81 | 77.02 | 89.86 | 56.78 | 72.56 | 80.51 |
| | MambaBDA-Tiny | 84.76 | 77.50 | 79.68 | 95.33 | 60.15 | 75.94 | 88.27 |
| \mathcal{M} | MambaBDA-Small | 86.61 | 78.80 | 81.14 | 95.99 | 62.82 | 76.26 | 88.37 |
| | MambaBDA-Base | 87.38 | 78.84 | 81.41 | 95.94 | 62.74 | 76.46 | 88.58 |

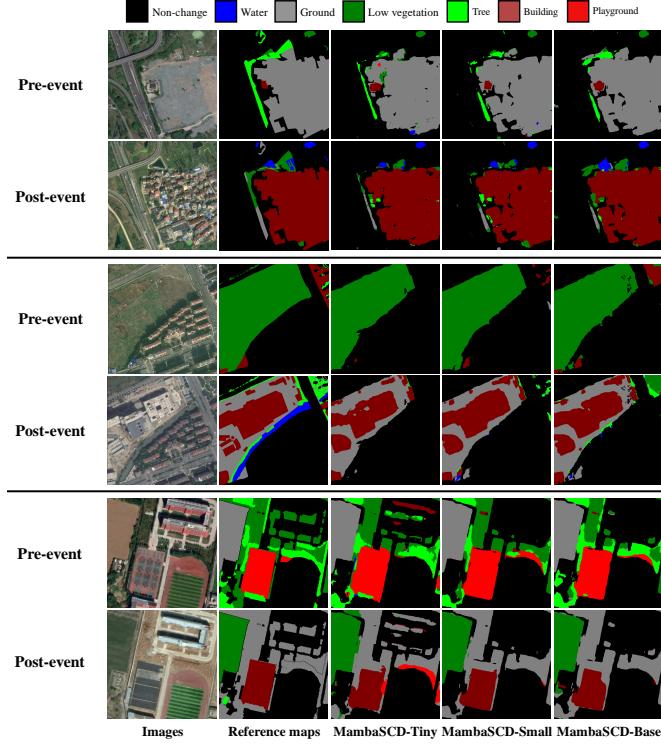


Fig. 11: Some semantic change maps obtained by the proposed MamabaSCD architectures on the test set of the SECONd dataset.

accurately identifying the specific semantic categories of these changes.

3) *BDA*: Table IX reports the accuracy of the MambaBDA architecture and some Transformer and CNN-based architectures on the xBD dataset. The performance of the MambaBDA architecture significantly outperforms these representative BDA approaches. Compared to the current Transformer-based SOTA method, DamFormer [31], our three MambaBDA architectures have a 2.64%, 4.12%, and 4.39% higher $F_1^{overall}$,

respectively. Moreover, it can be noticed that the accuracy improvement compared to DamFormer mainly comes from the more difficult damage classification task rather than the building localization task, which demonstrates that the proposed MambaBDA architecture can adequately learn the spatio-temporal relationships between the multi-temporal images so as to effectively distinguish different building damage levels.

In order to further show the misclassification between the different damage levels in more detail, we show their confusion matrices in Fig. 12. As can be seen from the confusion matrices, the categories “no damage” and “destroyed” are classified with high accuracy and are not easily misclassified into the remaining two categories. On the contrary, the categories of “minor damage” and “major damage” are easily misclassified as other categories, especially the “minor damage” category. A large portion of the “minor damage” category is misclassified as “no damage” and “major damage”. There are mainly the following reasons. First, the optical images in the xBD dataset were acquired with a relatively small viewing angle of the sensor. In many cases, “minor damage” is nuance damages to the structure of the building. Little relevant information can be obtained from the nearly orthorectified optical image for such cases. Furthermore, there is a semantic ambiguity in the definition of the categories of minor damage and major damage. We argue that it is difficult to address this problem by starting from the model architecture perspective. It can be considered to introduce auxiliary information, such as height information or multi-view imagery [21], [69], etc. to help the model make a more accurate classification.

Table X lists the number of parameters and GFLOPs of these BDA approaches. For MambaBDA-Small, our architecture yields more accurate results on building damage assessment compared to Siamese-UNet, ChangeOS-101, and MTF, with an improvement in $F_1^{overall}$ from 5.64 % to 9.46 % for a similar number of parameters and lower GFLOPs.

Fig. 13 shows the visualization results of the building damage caused by different disaster events obtained by Mam-

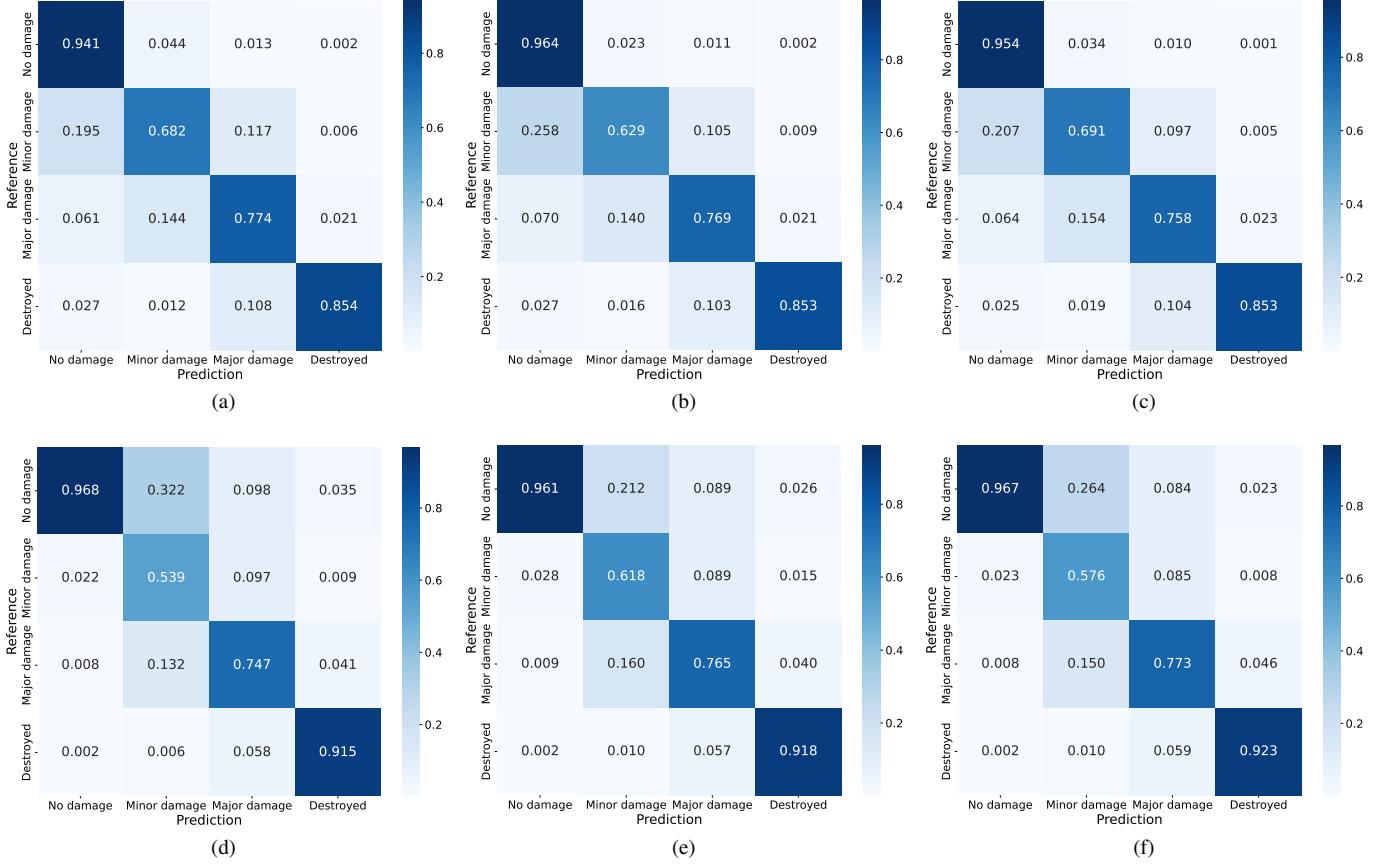


Fig. 12: Confusion matrices of MambaBDA-Tiny, MambaBDA-Small, and MambaBDA-Base. (a)-(c) are matrices normalized by rows. (d)-(f) are matrices normalized by columns.

TABLE X: Comparison of different BDA models in computational cost, where \mathcal{C} indicates CNN-based models, \mathcal{T} indicates Transformer-based models, and \mathcal{M} indicates Mamba-based models. Note that since ChangeOS [3] does not have open source code for the model implementation, we use our own implementation here to calculate the number of parameters and FLOPs.

| Type | Method | Params | GFLOPs |
|---------------|-------------------|--------|--------|
| \mathcal{C} | Siamese-UNet [51] | 51.46 | 138.88 |
| | MTF [52] | 44.40 | 268.76 |
| | ChangeOS-18 [3] | 23.49 | 229.22 |
| | ChangeOS-34 [3] | 33.60 | 255.85 |
| | ChangeOS-50 [3] | 52.62 | 469.41 |
| | ChangeOS-101 [3] | 71.61 | 508.41 |
| \mathcal{T} | DamFormer [31] | 32.52 | 169.34 |
| \mathcal{M} | MambabDAD-Tiny | 19.74 | 59.57 |
| | MambabDAD-Small | 52.11 | 130.80 |
| | MambabDAD-Base | 87.76 | 195.43 |

baBDA on the xBD dataset. MambabDAD can accurately localize and then differentiate the damage levels of buildings despite the varying type of building and the disaster event, implying the potential of the MambabDAD architecture for practical disaster response applications.

E. Different Spatio-Temporal Modeling Methods

Modeling spatio-temporal relationships is significant for CD tasks [17], [24], [46]. To demonstrate the effectiveness of our proposed modeling mechanisms, we compare them with several commonly used spatial-temporal modeling methods in CD. They are concatenation operations with feature pyramid network (FPN) [24] (baseline), temporal-symmetric transformation based on 3D convolutional layers (TST) [24], RNN-based method [17], and Transformer-based method [28]. Table XI lists the performance of these modeling methods on the three CD subtasks. On the simpler BCD tasks, Transformer does not yield significant advantages in performance compared to RNN. However, on the more difficult SCD and BDA tasks, the advantages of the Transformer architecture are illustrated. Compared with these current representative spatio-temporal relationship modeling approaches, our proposed modeling approach based on the Mamba architecture, which inherits the advantages of RNN and Transformer, shows superior performance on the three CD tasks of BCD, SCD, and BDA. Compared to the Transformer-based spatio-temporal modeling approach, our method improves the F1 value by 0.77% on BCD, SeK by 0.6% on the SCD task and $F_1^{overall}$ by 1.10% on the BDA task.

We further visualize the features extracted by the MambabCD architecture in Fig. 14. In the features extracted by

TABLE XI: Comparison of different approaches for modeling spatio-temporal relationships on three tasks. The table highlights the highest values in **red**, and the second-highest results in **blue**.

| Method | SYSU | | | | | SECOND | | | | | xBD | | |
|------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-----------------|--|--|
| | OA | F1 | IoU | KC | OA | F1 | mIoU | SeK | F_1^{loc} | F_1^{clf} | $F_1^{overall}$ | | |
| Baseline | 91.71 | 81.11 | 68.22 | 75.84 | 87.85 | 62.71 | 72.33 | 22.14 | 84.85 | 74.74 | 77.77 | | |
| STT [24] | 92.09 | 82.22 | 69.80 | 77.16 | 88.09 | 63.36 | 73.01 | 23.11 | 86.74 | 75.92 | 79.17 | | |
| RNN [17] | 92.97 | 82.02 | 69.52 | 76.94 | 88.20 | 63.63 | 72.86 | 23.09 | 85.74 | 76.48 | 79.26 | | |
| Transformer [28] | 92.14 | 82.17 | 69.74 | 77.17 | 88.05 | 64.05 | 73.02 | 23.44 | 86.09 | 77.45 | 80.04 | | |
| Ours | 92.35 | 82.83 | 70.70 | 77.94 | 88.38 | 64.10 | 73.61 | 24.04 | 86.61 | 78.80 | 81.14 | | |

TABLE XII: Comparison of different commonly used backbones in BCD on the SYSU dataset. The table highlights the highest values in **red**, and the second-highest results in **blue**.

| Encoder | OA | F1 | IoU | KC | Param (M) | GFLOPs |
|----------------------|--------------|--------------|--------------|--------------|-----------|--------|
| ResNet-101 [67] | 91.10 | 81.10 | 68.22 | 75.28 | 46.90 | 131.36 |
| EfficientNet-B5 [70] | 91.60 | 82.26 | 69.86 | 76.84 | 31.63 | 76.88 |
| MixFormer-v3 [35] | 87.70 | 75.00 | 60.00 | 66.87 | 47.03 | 82.98 |
| Swin-Small [34] | 92.09 | 82.22 | 69.80 | 77.16 | 37.75 | 97.99 |
| VMamba-Small [38] | 92.35 | 82.83 | 70.70 | 77.94 | 49.94 | 114.82 |

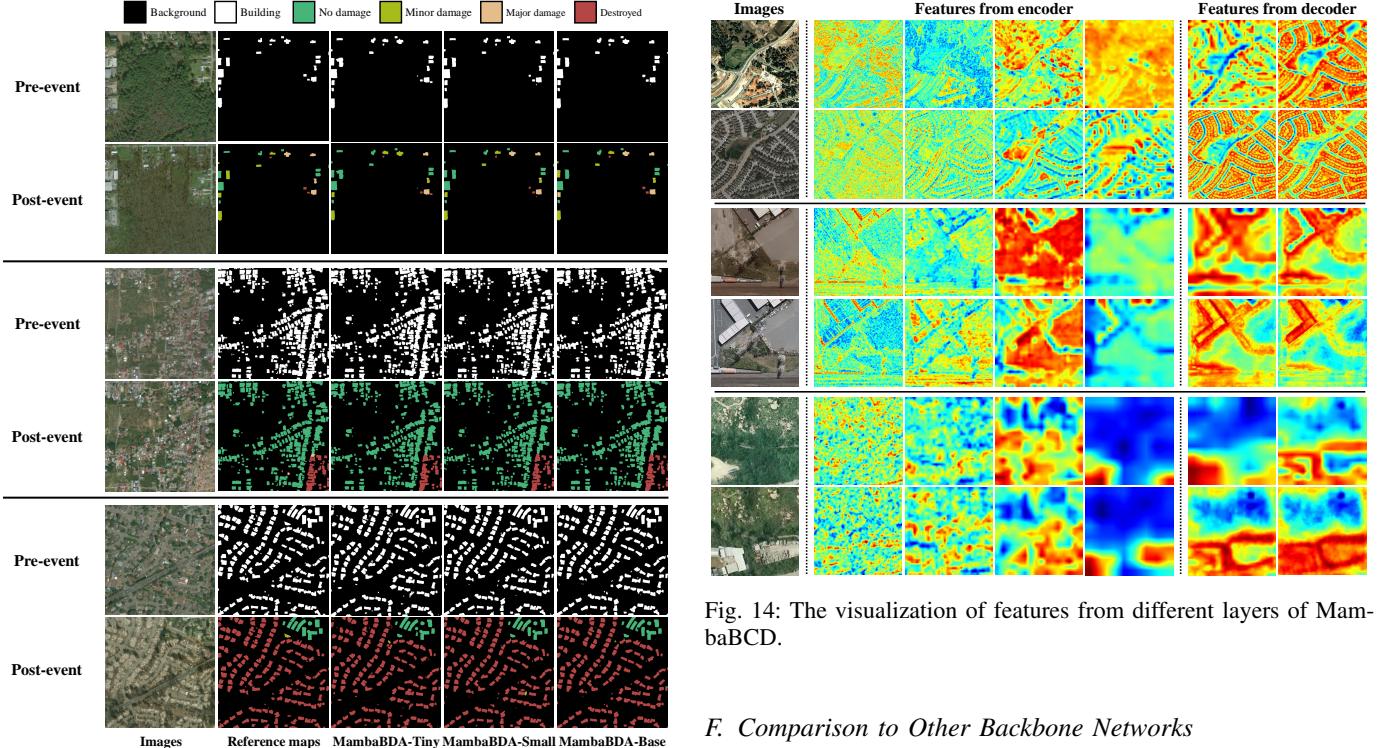


Fig. 13: Some building damage assessment results obtained by our methods on the test set of the xBD dataset. From top to bottom, the disaster events are typhoons, tsunamis, and wildfires.

the encoder, the features in the region of high response values are not necessarily regions of interest, i.e., changed regions. Whereas, in the decoder, after the spatio-temporal relationships are gradually modeled by the proposed spatio-temporal modeling mechanisms, changed regions gradually exhibit high response values.

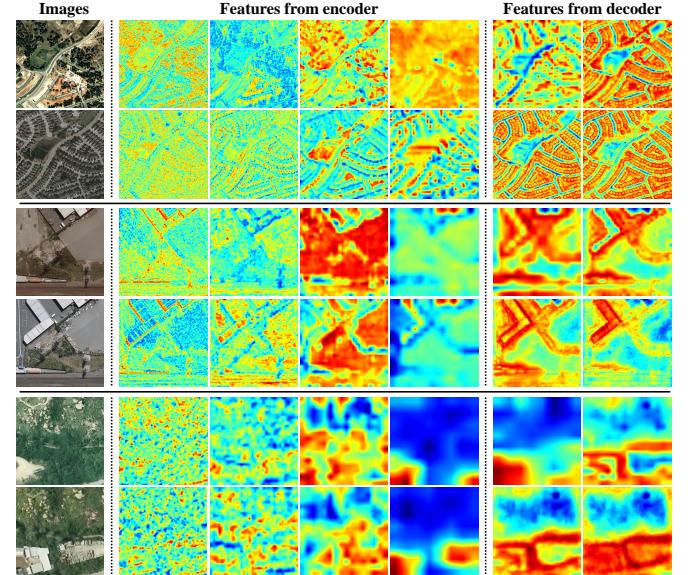


Fig. 14: The visualization of features from different layers of MambaBCD.

F. Comparison to Other Backbone Networks

Finally, we simply compare the Mamba with some representative backbone networks in CD tasks on the SYSU dataset in Table XII. It can be seen that Mamba outperforms the two Transformer backbone networks, Swin-Transformer and MixFormer. This is because, in order to reduce the computational overhead of self-attention operation, these two Transformer backbone networks either adopt a local attention mechanism (Swin-Transformer) [34] or a strided convolutional layer (MixFormer) [35] to reduce the size of the feature maps, which will negatively affect the learning of global contextual information. The adopted VMamba architecture, on the other hand, can reduce its computational consumption to a linear relationship with the number of tokens, thus eliminating the

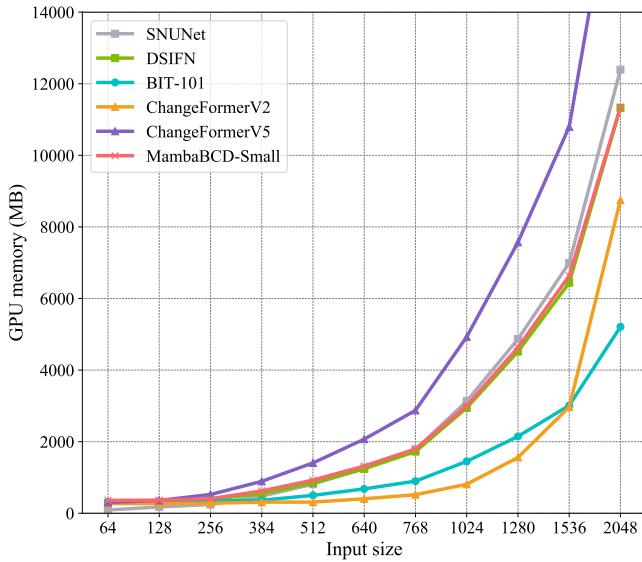


Fig. 15: GPU memory footprint of different models with varying input sizes.

need for these methods, which cause information loss, and allowing for the fully learning of global context information. As for the remote sensing change detection task, due to the multi-scale phenomenon of land-cover features and the differences in spatial resolution of different sensors, the sizes of changed objects often vary greatly. Therefore, the features extracted by Mamba through fully learning the global contextual information are particularly important for the subsequent detection of these changed objects with different sizes.

G. GPU Memory Footprint

In Section IV-D, we list the number of parameters and GFLOPs of different models. In this subsection, we further compare the GPU memory footprint of some models. Fig. 15 compares the GPU memory footprint of some CNN-based and Transformer-based methods with our MambaBCD architecture for varying input sizes. We can see that the trend of GPU memory footprint with input size for MambaBCD-Small is similar to some of the advanced CNN architectures (DSIFN and SNUNet). However, our MambaBCD is able to learn global spatial contextual information as well as adequately model spatio-temporal relationships, thereby exhibiting better detection performance. The GPU memory footprint of MambaBCD-Small is significantly lower compared to ChangeFormerV5, which employs the Transformer architecture as its backbone. Note that ChangeFormerV5 itself has employed strided convolutional layers to reduce the computational overhead of self-attention operations. Also, we can find that the lightweight architecture ChangeFormerV2 has surpassed the Mamba architecture and the two CNN architectures, DSIFN and SNUNet, in terms of the increasing trend of GPU memory footprint when the input size is increased from 1536 to 2048. These results imply that Transformer to be applicable to large-scale scenarios of remote sensing images often comes at the

cost of information loss. In comparison, Mamba architecture can avoid this problem.

H. Robustness Against Degraded Input Data

Finally, we test the robustness of our method to degraded input data. In the inference phase, the data input to the model is subjected to different levels of Gaussian blurring, Gaussian noise, and scaling of the data by different proportions, respectively. Fig. 16 shows the degradation of the F1 score of different methods with degraded data on the LEVIR-CD+ dataset. First, we can see that the CNN-based methods are very poorly robust to Gaussian blurred data. F1 scores of SiamCRNN-101 and DSIFN drop by nearly 60% when increasing the standard deviation of the Gaussian blur from 0 to 2. BIT-101 also drops its F1 score by more than 34% due to the adoption of ResNet to extract features. Whereas the two methods using Transformer as the backbone, SwinTransUNet and ChangeFormerV5 are relatively robust to Gaussian blurred data, with F1 scores decreasing by 14.83 % and 18.43 %, respectively. In contrast, the F1 score for our MambaBCD architecture drops by only 8.73%.

Fig. 16-(b) shows the effect of adding different levels of Gaussian noise to the input data. With the exception of ChangeFormerV5 and our MambaBCD, the remaining methods have significant F1 decreases ranging from 40.83% to 15.07%. Our MambaBCD architecture, on the other hand, shows robustness to Gaussian noise, with only an 8.26% drop in the F1 score for Gaussian noise with a standard deviation of 20.

Fig. 16-(c) compares the robustness of models trained at a single scale to test data at different scales. The F1 value of MambaBCD decreases by only 2.88% when the scale increases from 1 to 1.5. When the scaling ratio decreases from 1 to 0.75, the F1 value of MambaBCD decreases only by 1.18%. When the scaling factor is reduced to 0.5, all methods, including MambaBCD, show a larger decrease in the F1 score. However, compared to the decline of 14.99 % of SwinTransUNet and the decline of 16.41 % of ChangeFormerV5, the decline of the MambaBCD decline is smaller, at 12.03 %. All the above results again demonstrate the superiority of our proposed Mamba-based CD architecture.

V. CONCLUSION

In this paper, we explore the potential of the emerging Mamba architecture for remote sensing image change detection tasks. Three architectures, MambaBCD, MambaSCD, and MambaBDA are developed for binary change detection, semantic change detection, and building damage assessment tasks, respectively. The encoders of all three architectures are cutting-edge VMamba architecture, which can adequately learn the global contextual information of the input image with linear complexity. Then, for the change decoder, in order to adequately learn spatio-temporal features, we propose three ways of modeling spatio-temporal relationships that can adequately exploiting the attributes and advantages of Mamba architecture. Experiments on five benchmark datasets have fully revealed the potential of Mamba for multi-temporal

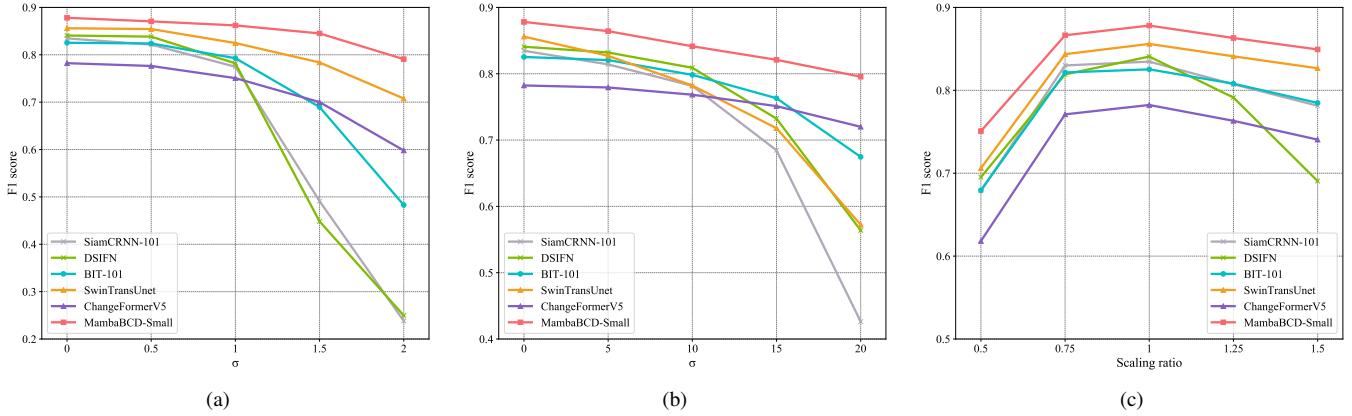


Fig. 16: Robustness of different models against degraded test images on the LEVIR-CD+ dataset. (a) Gaussian blur. (b) Gaussian noise. (c) Scaling.

remote sensing image data processing tasks. Compared to the CNN-based and Transformer-based approaches, the proposed three architectures can achieve SOTA performance on their respective tasks. Compared to some commonly used backbone networks, the Mamba architecture can extract features that are more suitable for downstream CD tasks. The combination of our proposed three spatio-temporal relationship modeling mechanisms and the Mamba architecture provides a new perspective on modeling spatio-temporal relationships, showing much better detection results on all three subtasks. Additional experiments show that our Mamba-based architecture is robust to degraded input data.

Our future work includes, but is not limited to, developing Mamba architectures more suited to the characteristics of remote sensing data and exploring the potential of Mamba for multimodal remote sensing tasks.

REFERENCES

- [1] D. Lu, P. Mausel, E. Brondizio, and E. Moran, “Change detection techniques,” *Int. J. Remote Sens.*, vol. 25, no. 12, pp. 2365–2407, 2004.
- [2] P. Coppin, I. Jonckheere, K. Nackaerts, B. Muys, and E. Lambin, “Digital change detection methods in ecosystem monitoring: A review,” *Int. J. Remote Sens.*, vol. 25, no. 9, pp. 1565–1596, 2004.
- [3] Z. Zheng, Y. Zhong, J. Wang, A. Ma, and L. Zhang, “Building damage assessment for rapid disaster response with a deep object-based semantic change detection framework: From natural disasters to man-made disasters,” *Remote Sens. Environ.*, vol. 265, p. 112636, 2021.
- [4] H. Guo, Q. Shi, A. Marinoni, B. Du, and L. Zhang, “Deep building footprint update network: A semi-supervised method for updating existing building footprint from bi-temporal remote sensing images,” *Remote Sens. Environ.*, vol. 264, p. 112589, 2021.
- [5] H. Chen, N. Yokoya, C. Wu, and B. Du, “Unsupervised Multimodal Change Detection Based on Structural Relationship Graph Representation Learning,” *IEEE Trans. Geosci. Remote Sens.*, pp. 1–18, 2022.
- [6] H. Chen, C. Lan, J. Song, C. Broni-Bediako, J. Xia, and N. Yokoya, “Land-cover change detection using paired openstreetmap data and optical high-resolution imagery via object-guided transformer,” *arXiv preprint arXiv:2310.02674*, 2023.
- [7] Y. Xiao, Q. Yuan, K. Jiang, J. He, C.-W. Lin, and L. Zhang, “Ttst: A top-k token selective transformer for remote sensing image super-resolution,” *IEEE Trans. Image Process.*, vol. 33, pp. 738–752, 2024.
- [8] C. Wu, B. Du, and L. Zhang, “Slow feature analysis for change detection in multispectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2858–2874, 2014.
- [9] 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, 2007.
- [10] L. Bruzzone and Diego Fernández Prieto, “Automatic Analysis of the Difference Image for Unsupervised Change Detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1171–1182, 2000.
- [11] M. Hussain, D. Chen, A. Cheng, H. Wei, and D. Stanley, “Change detection from remotely sensed images: From pixel-based to object-based approaches,” *ISPRS J. Photogramm. Remote Sens.*, vol. 80, pp. 91–106, 2013.
- [12] Y. Sun, L. Lei, D. Guan, and G. Kuang, “Iterative Robust Graph for Unsupervised Change Detection of Heterogeneous Remote Sensing Images,” *IEEE Trans. Image Process.*, vol. 30, pp. 6277–6291, 2021.
- [13] H. Chen, N. Yokoya, and M. Chini, “Fourier domain structural relationship analysis for unsupervised multimodal change detection,” *ISPRS J. Photogramm. Remote Sens.*, vol. 198, pp. 99–114, 2023.
- [14] C. Wu, H. Chen, B. Du, and L. Zhang, “Unsupervised change detection in multitemporal vhr images based on deep kernel pca convolutional mapping network,” *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12084–12098, 2022.
- [15] Y. Xiao, Q. Yuan, K. Jiang, J. He, X. Jin, and L. Zhang, “Edifssr: An efficient diffusion probabilistic model for remote sensing image super-resolution,” *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–14, 2024.
- [16] M. Gong, T. Zhan, P. Zhang, and Q. Miao, “Superpixel-based difference representation learning for change detection in multispectral remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2658–2673, 2017.
- [17] 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, 2020.
- [18] H. Chen, C. Wu, B. Du, and L. Zhang, “Deep Siamese Multi-scale Convolutional Network for Change Detection in Multi-Temporal VHR Images,” in *2019 10th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp)*, 2019, pp. 1–4.
- [19] Q. Shi, M. Liu, S. Li, X. Liu, F. Wang, and L. Zhang, “A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.
- [20] K. Yang, G.-S. Xia, Z. Liu, B. Du, W. Yang, M. Pelillo, and L. Zhang, “Asymmetric siamese networks for semantic change detection in aerial images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–18, 2022.
- [21] J. Song, H. Chen, and N. Yokoya, “Syntheworld: A large-scale synthetic dataset for land cover mapping and building change detection,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2024, pp. 8287–8296.
- [22] R. Caye Daudt, B. Le Saux, and A. Boulch, “Fully convolutional siamese networks for change detection,” in *Proceedings - International Conference on Image Processing, ICIP*, 2018, pp. 4063–4067.
- [23] C. Zhang, P. Yue, D. Tapete, L. Jiang, B. Shangguan, L. Huang, and G. Liu, “A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images,” *ISPRS J. Photogramm. Remote Sens.*, vol. 166, no. June, pp. 183–200, 2020.
- [24] Z. Zheng, Y. Zhong, S. Tian, A. Ma, and L. Zhang, “ChangeMask: Deep multi-task encoder-transformer-decoder architecture for semantic change

- detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 183, no. March 2021, pp. 228–239, 2022.
- [25] Y. Cao and X. Huang, "A full-level fused cross-task transfer learning method for building change detection using noise-robust pretrained networks on crowdsourced labels," *Remote Sens. Environ.*, vol. 284, p. 113371, 2023.
- [26] Z. Lv, J. Liu, W. Sun, T. Lei, J. A. Benediktsson, and X. Jia, "Hierarchical attention feature fusion-based network for land cover change detection with homogeneous and heterogeneous remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–15, 2023.
- [27] T. Liu, M. Gong, D. Lu, Q. Zhang, H. Zheng, F. Jiang, and M. Zhang, "Building change detection for vhr remote sensing images via local-global pyramid network and cross-task transfer learning strategy," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022.
- [28] H. Chen, Z. Qi, and Z. Shi, "Remote sensing image change detection with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.
- [29] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
- [30] W. G. C. Bandara and V. M. Patel, "A transformer-based siamese network for change detection," in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 207–210.
- [31] H. Chen, E. Nemni, S. Vallecorsa, X. Li, C. Wu, and L. Bromley, "Dual-tasks siamese transformer framework for building damage assessment," in *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2022, pp. 1600–1603.
- [32] C. Zhang, L. Wang, S. Cheng, and Y. Li, "Swinsunet: Pure transformer network for remote sensing image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022.
- [33] Q. Li, R. Zhong, X. Du, and Y. Du, "Transunetcd: A hybrid transformer network for change detection in optical remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–19, 2022.
- [34] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10012–10022.
- [35] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 12077–12090.
- [36] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint arXiv:2111.00396*, 2021.
- [37] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.
- [38] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, and Y. Liu, "Vmamba: Visual state space model," *arXiv preprint arXiv:2401.10166*, 2024.
- [39] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," *arXiv preprint arXiv:2401.09417*, 2024.
- [40] S. Fang, K. Li, J. Shao, and Z. Li, "Snunet-cd: A densely connected siamese network for change detection of vhr images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
- [41] C. Han, C. Wu, H. Guo, M. Hu, and H. Chen, "Hanet: A hierarchical attention network for change detection with bitemporal very-high-resolution remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 3867–3878, 2023.
- [42] C. Han, C. Wu, H. Guo, M. Hu, J. Li, and H. Chen, "Change guiding network: Incorporating change prior to guide change detection in remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 8395–8407, 2023.
- [43] S. Zhao, X. Zhang, P. Xiao, and G. He, "Exchanging dual-encoder-decoder: A new strategy for change detection with semantic guidance and spatial localization," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–16, 2023.
- [44] H. Chen, J. Song, C. Wu, B. Du, and N. Yokoya, "Exchange means change: An unsupervised single-temporal change detection framework based on intra- and inter-image patch exchange," *ISPRS J. Photogramm. Remote Sens.*, vol. 206, pp. 87–105, 2023.
- [45] R. Caye Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, "Multitask learning for large-scale semantic change detection," *Computer Vision and Image Understanding*, vol. 187, p. 102783, 2019.
- [46] L. Mou, L. Bruzzone, and X. X. Zhu, "Learning spectral-spatialoral features via a recurrent convolutional neural network for change detection in multispectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 924–935, 2019.
- [47] L. Ding, H. Guo, S. Liu, L. Mou, J. Zhang, and L. Bruzzone, "Bitemporal semantic reasoning for the semantic change detection in hr remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.
- [48] D. Peng, L. Bruzzone, Y. Zhang, H. Guan, and P. He, "Scdnet: A novel convolutional network for semantic change detection in high resolution optical remote sensing imagery," *International Journal of Applied Earth Observation and Geoinformation*, vol. 103, p. 102465, 2021.
- [49] S. Tian, X. Tan, A. Ma, Z. Zheng, L. Zhang, and Y. Zhong, "Temporal-agnostic change region proposal for semantic change detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 204, pp. 306–320, 2023.
- [50] 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, 2019.
- [51] R. Gupta, B. Goodman, N. Patel, R. Hosfelt, S. Sajeev, E. Heim, J. Doshi, K. Lucas, H. Choset, and M. Gaston, "Creating xbd: A dataset for assessing building damage from satellite imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.
- [52] E. Weber and H. Kané, "Building disaster damage assessment in satellite imagery with multi-temporal fusion," *arXiv preprint arXiv:2004.05525*, 2020.
- [53] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.
- [54] Y. Niu, H. Guo, J. Lu, L. Ding, and D. Yu, "Smnet: Symmetric multi-task network for semantic change detection in remote sensing images based on cnn and transformer," *Remote Sensing*, vol. 15, no. 4, 2023.
- [55] L. Ding, J. Zhang, H. Guo, K. Zhang, B. Liu, and L. Bruzzone, "Joint spatio-temporal modeling for semantic change detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–14, 2024.
- [56] J. T. Smith, A. Warrington, and S. W. Linderman, "Simplified state space layers for sequence modeling," *arXiv preprint arXiv:2208.04933*, 2022.
- [57] D. Y. Fu, T. Dao, K. K. Saab, A. W. Thomas, A. Rudra, and C. Ré, "Hungry hungry hippos: Towards language modeling with state space models," *arXiv preprint arXiv:2212.14052*, 2022.
- [58] X. He, K. Cao, K. Yan, R. Li, C. Xie, J. Zhang, and M. Zhou, "Pan-mamba: Effective pan-sharpening with state space model," *arXiv preprint arXiv:2402.12192*, 2024.
- [59] K. Chen, B. Chen, C. Liu, W. Li, Z. Zou, and Z. Shi, "Rsmamba: Remote sensing image classification with state space model," *arXiv preprint arXiv:2403.19654*, 2024.
- [60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, 2017.
- [61] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural networks*, vol. 107, pp. 3–11, 2018.
- [62] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [63] M. Berman, A. R. Triki, and M. B. Blaschko, "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [64] 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, 2018.
- [65] 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.
- [66] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
- [67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [68] S. Tian, Y. Zhong, Z. Zheng, A. Ma, X. Tan, and L. Zhang, "Large-scale deep learning based binary and semantic change detection in ultra high resolution remote sensing imagery: From benchmark datasets to urban application," *ISPRS J. Photogramm. Remote Sens.*, vol. 193, pp. 164–186, 2022.

- [69] Y. Cao and X. Huang, "A deep learning method for building height estimation using high-resolution multi-view imagery over urban areas: A case study of 42 chinese cities," *Remote Sensing of Environment*, vol. 264, p. 112590, 2021.
- [70] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97, 2019, pp. 6105–6114.