Continuous Urban Change Detection From Satellite Image Time Series With Temporal Feature Refinement and Multitask Integration

Sebastian Hafner, Heng Fang, Hossein Azizpour, and Yifang Ban, *Senior Member, IEEE*

***Abstract*— Urbanization advances at unprecedented rates, leading to negative environmental and societal impacts. Remote sensing can help mitigate these effects by supporting sustainable development strategies with accurate information on urban growth. Deep learning-based methods have achieved promising urban change detection results from optical satellite image pairs using convolutional neural networks (ConvNets), transformers, and a multitask learning setup. However, bi-temporal methods are limited for continuous urban change detection, i.e., the detection of changes in consecutive image pairs of satellite image time series (SITS), as they fail to fully exploit multitemporal data (>2 images). Existing multitemporal change detection methods, on the other hand, collapse the temporal dimension, restricting their ability to capture continuous urban changes. In addition, multitask learning methods lack integration approaches that combine change and segmentation outputs. To address these challenges, we propose a continuous urban change detection framework incorporating two key modules. The temporal feature refinement (TFR) module employs self-attention to improve ConvNet-based multitemporal building representations. The temporal dimension is preserved in the TFR module, enabling the detection of continuous changes. The multitask integration (MTI) module utilizes Markov networks to find an optimal building map time series based on segmentation and dense change outputs. The proposed framework effectively identifies urban changes based on high-resolution SITS acquired by the PlanetScope constellation ($F1$ score 0.551), Gaofen-2 ($F1$ score 0.440), and WorldView-2 ($F1$ score 0.543). Moreover, our experiments on three challenging datasets demonstrate the effectiveness of the proposed framework compared to bi-temporal and multitemporal urban change detection and segmentation methods. The code is available on GitHub: https://github.com/SebastianHafner/ContUrbanCD**

***Index Terms*— Earth observation, multitask learning, multitemporal, remote sensing, transformers.**

I. Introduction

URBANIZATION is progressing at unprecedented rates [1]. Thus, the global amount of urban land is projected to increase by a factor of 2–6 over the 21st century [2]. The rapid expansion of urban land, i.e., urban sprawl, is associated with multiple negative effects on the environment and human well-being [3], [4]. To mitigate urban sprawl, informed and sustainable urban development strategies are crucial [5]. However, these strategies are currently hampered by a lack of timely information on the extent of urban land.

Remote sensing is an efficient tool to monitor the Earth's surface [6]. Urban changes are commonly detected from two satellite images acquired at different times over the same geographical area. Traditional change detection methods use arithmetic operations to derive change features (CFs) from bi-temporal image pairs. For example, various arithmetic methods have been developed to derive CFs from optical images, such as image differencing, image regression, and change vector analysis [7]. These features are then classified into changed/unchanged pixels or objects using different classification algorithms, including machine learning algorithms [6], [7].

In recent years, deep learning has been continuously replacing traditional change detection methods [8], [9], [10], [11]. Specifically, deep convolutional neural networks (ConvNets) have been used extensively for change detection in bi-temporal optical satellite image pairs [see Fig. 1(a)]. The simplest way of adapting common ConvNets such as U-Net [12] for change detection is with an input-level fusion (or early fusion [13]) strategy, referring to the concatenation of image pairs before passing them to a ConvNet. Contrarily, late fusion strategies typically process images separately in a Siamese network consisting of two ConvNets with shared weights. Extracted bi-temporal features are then fused using concatenation or absolute differencing [13], [14]. Since Siamese networks are generally considered preferable to input-level fusion strategies, multiple studies developed modules that are incorporated into Siamese network architectures to improve feature representations [15], [16], [17]. For example, Chen et al. [18], [19] proposed to refine features extracted by ConvNets from very high-resolution (VHR) imagery using a transformer-based module, alleviating the limited long-range context modeling capability of convolutions with self-attention. Since then, self-attention has become a popular mechanism for capturing long-range spatial dependencies in VHR change detection [17], [20], [21], [22].

In recent years, high-resolution (i.e., 1–10 m) satellite image time series (SITS) have become increasingly available [23]. Those data have enabled a shift from detecting land cover changes in image pairs acquired years apart to continuous

Received 17 January 2025; revised 4 April 2025 and 16 May 2025; accepted 7 June 2025. Date of publication 11 June 2025; date of current version 19 June 2025. This work was supported in part by the EU Horizon 2020 Project HARMONIA under Agreement 101003517, in part by the Digital Futures under the Grant for the EO-AI4GlobalChange Project, and in part by the EO-AI4Urban Project within the ESA-MOST Dragon 5 Program. *(Corresponding author: Yifang Ban.)*

Sebastian Hafner and Yifang Ban are with the Division of Geoinformatics, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden (e-mail: yifang@kth.se).

Heng Fang and Hossein Azizpour are with the Division of Robotics, Perception and Learning, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden.

Digital Object Identifier 10.1109/TGRS.2025.3578866

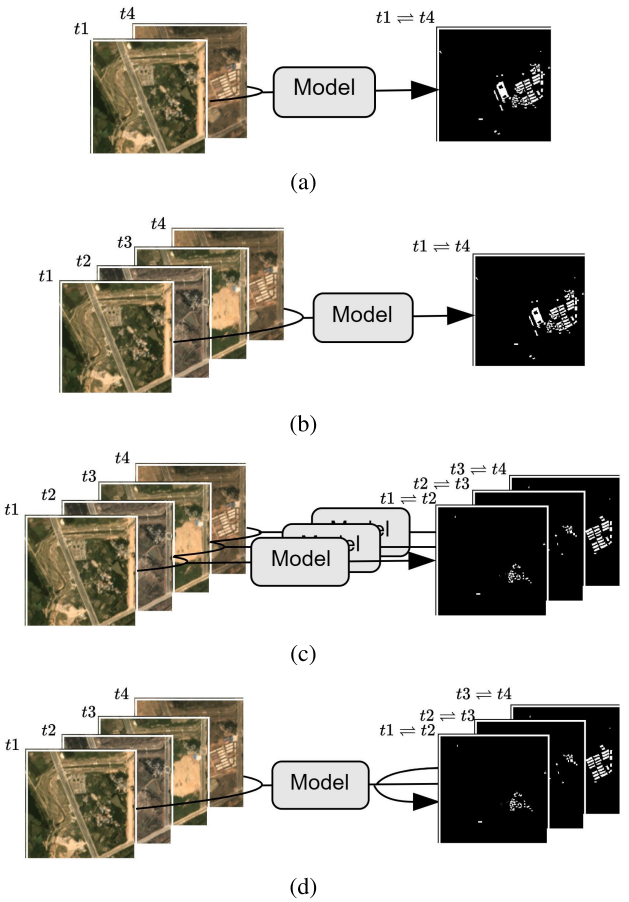


Fig. 1. Overview of standard urban change detection frameworks and the proposed method. (a) Bi-temporal urban change detection is typically performed on image pairs acquired multiple years apart. (b) Multitemporal change detection methods leverage image time series but only predict changes between the first and last image. (c) Bi-temporal model is applied to consecutive image pairs of a time series to perform continuous urban change detection; however, this method fails to incorporate multitemporal (i.e., >2 images) information (c). On the other hand, we propose (d) continuous urban change detection method that incorporates multitemporal information.

annual and subannual change detection [24]. In contrast, urban change detection methods are predominantly designed for bi-temporal change detection from image pairs acquired multiple years apart [25]. However, considering the unprecedented rate of global urbanization [26], it is essential to develop a new suite of methods that detect urban changes continuously. While continuous change detection can be achieved by applying a bi-temporal model to consecutive image pairs in SITS [see Fig. 1(c)], this approach fails to exploit multitemporal, i.e., >2 images, information. Furthermore, recent studies [27], [28] have demonstrated the effectiveness of multitemporal change detection models that predict changes between the first and last images of an SITS [see Fig. 1(b)]. For example, multitemporal information can help to reduce commission errors from registration errors, illumination differences, or other types of change unrelated to the problem of interest [28]. In addition, it can mitigate the effect of cloud artifacts in single images [27]. Existing multitemporal change detection methods employ either recurrent layers, such as long short-term memory (LSTM) layers [29], or 3-D convolutional layers to model temporal information [27], [28], [30], [31]. While these layers effectively model short-range temporal dependencies in time series data, the self-attention mechanism can explicitly model temporal dependencies across all timestamps of a time series [32]. Thus, several recent segmentation methods for SITS employ the self-attention mechanism to explicitly model temporal dependencies across all timestamps of a time series [33], [34]. However, the temporal models in these methods collapse the temporal dimension, resulting in a single output feature. Therefore, they do not facilitate continuous urban change detection, which requires the full temporal information to produce change maps between each consecutive image pair in the SITS.

Another promising avenue of research for change detection is multitask learning [35], where a related semantic segmentation task is trained parallel to the change detection task using a shared feature representation. The change detection task is typically combined with building segmentation for urban change detection [28], [36], [37], [38], [39]. To that end, Siamese networks are extended with an additional decoder for the semantic segmentation task. The feature maps extracted by the encoder are then shared between the change decoder and the segmentation decoder. However, despite the attention multitask learning has attracted in change detection, effective methods to integrate segmentation and change outputs have been largely unaddressed. For example, most multitask urban change detection studies consider building and change predictions independent outputs of the network [28], [36], [37], [38], [39]. Therefore, these studies do not account for inconsistencies between the building segmentation and urban change predictions. Moreover, they fail to exploit the complementary information produced by multitask predictions.

In this article, we propose a continuous urban change detection method [see Fig. 1(d)] and explore two research gaps in the current literature, namely, 1) the modeling of multitemporal information using self-attention for continuous urban change detection and 2) the integration of segmentation and change predictions in multitask learning setups. Specifically, we propose a new network architecture that relies on convolutions to extract multitemporal building representations and employs self-attention to model temporal dependencies in feature space while preserving the temporal dimension. We also propose a novel integration approach that determines the optimal building segmentation for each image in a time series based on the multitask network outputs. The effectiveness of the proposed architecture and integration approach is demonstrated on three urban change detection datasets featuring high-resolution optical SITS.

The following summarizes the main contributions of this article.

1) We introduce a continuous urban change detection model that produces change outputs for each consecutive image pair in SITS while leveraging the full temporal dimension of the time series.
2) To enable continuous change detection, we present a transformer-based feature refinement module that effectively models temporal information in SITS. Importantly, our module preserves the temporal dimension of the representations, in contrast to existing temporal

modules that collapse multitemporal representations into a single one.

3) We propose a new multitask integration (MTI) approach that represents segmentation and change outputs in Markov networks to find the optimal building maps time series based on the network outputs.
4) Experiments on three datasets, namely, SpaceNet 7 (SN7), Wuhan Urban Semantic Understanding (WUSU), and Time Series Change Detection (TSCD), show that the proposed continuous urban change detection method is more effective than related methods.

II. Related Work

A. Bi-Temporal Change Detection

In recent years, a plethora of deep learning-based bi-temporal change detection methods have been proposed. Most of these works focus on developing new Siamese network architectures and/or training strategies. Initially, Daudt et al. [13], [14] proposed two Siamese ConvNet architectures for change detection. The Siam-Diff and Siam-Conc architectures employ encoders with shared weights for feature extraction from bi-temporal high-resolution image pairs and combine the corresponding feature maps using a subtraction and concatenation strategy, respectively. While encoders and decoders in these models follow the U-Net architecture [12], Fang et al. [15] incorporated a nested U-Net (i.e., UNet++ [40]) into a Siamese network to maintain high-resolution, fine-grained representations through dense skip connections. Many works also improved Siamese networks by incorporating different modules into the architecture. For example, an ensemble channel attention module was proposed for feature refinement in [15], and a new spatial pyramid pooling block was utilized in [16] to preserve shapes of change areas.

However, most recent methods are developed for bi-temporal change detection from VHR image pairs. Consequently, many methods employ the self-attention mechanism to improve the modeling of long-range dependencies in VHR imagery [17], [18], [19], [20]. Both [18] and [19] extract image features with ConvNets and employ self-attention modules to learn more discriminative features. Other works combined ConvNets and transformers with attention modules and multiscale processing [17], [41]. Bandara and Patel [20], on the other hand, proposed a fully transformer-based change detection method. Specifically, ChangeFormer combines two hierarchically structured transformer encoders with shared weights and a multilayer perception decoder in a Siamese network architecture. Since transformer-based methods strongly rely on pretraining, Noman et al. [22] recently proposed ScratchFormer which is a transformer-based change detection method that is trained from scratch but achieves SOTA performance. The ScrachFormer architecture utilizes shuffled sparse attention layers that enable faster convergence due to their sparse structure. Although these transformer-based methods are considered SOTA for urban change detection, their effectiveness has been predominately demonstrated on bi-temporal VHR datasets such as LEVIR-CD [18] and WHU-CD [42]. In comparison, high-resolution imagery is acquired much more frequently by satellite constellations such as PlanetScope, making it an invaluable data source for change detection applications. Therefore, developing methods that effectively leverage transformers for change detection from high-resolution imagery is crucial.

B. Change Detection and Segmentation From Time Series Data

Few studies have developed deep learning methods for urban change detection from high-resolution SITS. For example, Papadomanolaki et al. [27] proposed to incorporate LSTM networks into a U-Net model to leverage optical SITS for change detection. Their L-UNet outperformed bi-temporal ConvNet-based methods on a bi-temporal dataset enriched with intermediate satellite images [27]. Others proposed an encoder–decoder LSTM model that is trained to rearrange temporally shuffled time series [31]. The core assumption of this unsupervised method is that the model fails to correctly rearrange shuffled data for changed pixels. On the other hand, Meshkini et al. [30] proposed a weakly supervised change detection method that employs 3-D convolutional layers to capture spatial–temporal information in SITS. Recently, He et al. [43] presented a deep learning method for time series land cover change detection. However, since their model only uses 1-D convolutions along the temporal dimension, it does not consider the spatial dimension, which is a limiting factor for high-resolution data.

Due to the limited number of change detection methods for SITS, we also expand this review to the semantic segmentation of SITS. Several recent semantic segmentation methods for SITS employed the self-attention mechanism for temporal modeling of multitemporal features [33], [34], [44]. Garnot and Landrieu [33] employed a lightweight-temporal attention encoder [45] for the temporal modeling of multitemporal feature maps extracted using a shared ConvNet encoder. Similarly, Cai et al. [44] employed an attention bidirectional LSTM module for temporal modeling of ConvNet-based feature maps time series. The modules in both studies collapse the temporal dimension, resulting in a single feature map obtained using a ConvNet decoder. While Tarasiou et al. [34] used a vision transformer to learn feature representation from SITS, their model also outputs a single feature map for semantic segmentation.

In summary, existing multitemporal change detection methods rely on recurrent and 3-D convolutional layers for temporal modeling. While multitemporal semantic segmentation methods frequently employ temporal attention-based modules, they collapse the temporal dimension similar to multitemporal change detection methods. While these methods can be adapted for change detection, collapsing the temporal dimension limits them to the detection of changes between the first and the last image of an SITS [i.e., multitemporal change detection in Fig. 1(b)].

C. Multitask Learning

Multitask learning has been investigated by several studies for urban change detection over the past years. Liu et al. [36]

proposed a dual-task Siamese ConvNet to learn more discriminative feature representations for building change detection from bi-temporal image pairs. The proposed dual-task constrained deep Siamese convolutional network (DTCDSCN) consists of three main components: a shared ResNet-based encoder, a shared decoder for building segmentation, and a separate decoder for change detection. On the other hand, Papadomanolaki et al. [28] proposed a multitask learning framework for urban change detection from image time series by adding building segmentation tasks for the first and last images of a time series to the urban change detection task. While L-UNet [27] is employed to extract changes, the segmentation is performed with a separate decoder that directly uses the feature maps extracted for the image pair by the shared encoder.

Some urban change detection studies also combined multitask learning with semi-supervised learning [37], [38]. In [37], the Siam-Diff network [14] was extended with an additional shared decoder for building segmentation, and an unsupervised term was introduced to encourage consistency between the changes derived from the building predictions and those predicted by the change decoder. Shu et al. [38], on the other hand, proposed to learn consistency between two building predictions corresponding to the prechange image. The first prediction is obtained by segmenting the prechange image and the second one by combining segmentation features of the postchange image with changes features.

In general, these multitask studies demonstrate that learning a segmentation task in parallel to the change detection task improves the latter. However, none of these studies investigate combining the change and segmentation network outputs to improve performance. Consequently, inconsistencies between the network outputs are also not accounted for.

III. Proposed Method

A. Overview

As illustrated in Fig. 2, the main components of the proposed method are a ConvNet-based encoder, transformer-based temporal feature refinement (TFR) modules, a CF module, two task-specific ConvNet-based decoders, and a Markovian MTI module. The following summarizes the urbanization monitoring process of the proposed method for a time series of satellite images.

1) First, for each image in the time series, multiscale feature maps are extracted using an encoder with shared weights.
2) Next, the above time series of feature maps are grouped by scale and fed to separate TFR modules consisting of transformer encoder layers. The temporally refined feature maps are regrouped according to their timestamp.
3) Then, the CF module obtains CF maps from the temporally refined segmentation feature maps. The module considers changes between all possible combinations of temporal pairs.
4) Two task-specific decoders are deployed to obtain building segmentation outputs for each image in the time series from the temporally refined segmentation features maps and change outputs from the CF maps.
5) Finally, the building and urban change outputs are combined using the MTI module. The module uses pixel-wise Markov networks to obtain optimal building states for the SITS.

Detailed descriptions of the components comprising the proposed method, as well as the loss function, are given in Sections III-B and III-G.

B. SITS Encoding

We consider a time series of T satellite images, represented as $\mathbf{X} \in \mathbb{R}^{T\times C\times H\times W}$, where C, H, and W denote the channel, height, and width dimensions, respectively. A ConvNet encoder with shared weights is utilized to separately extract feature maps $\mathbf{F}_{\text{seg}}$ from each image in the time series, as follows:

$$\mathbf{F}_{\text{seg}} = e(\mathbf{X}) \tag{1}$$

where $e(\cdot)$ represents the encoder, and subscript seg indicates that the feature maps contain representations for building segmentation.

The architecture of the encoder is based on the U-Net encoder [12]. Specifically, after an initial convolution block, the combination of a max-pooling layer and a consecutive convolution block is applied four times. Each of these four steps halves the spatial dimensions H and W due to the pooling operation, whereas the number of features D is doubled with the convolution block. Importantly, U-Net achieves precise localization by leveraging skip connections that forward the feature maps before each pooling operation to the decoder. Therefore, the output of the encoder consists of five feature map time series with different scales. To denote the scale of these time series, we introduce superscript s in the notation: $\mathbf{F}^s_{\text{seg}}$, where $s \in \{0, 1, 2, 3, 4\}$. For a given feature map time series $\mathbf{F}^s_{\text{seg}}$, the sizes of the height and width dimensions, as well as the feature dimension, are dependent on s, as follows:

$$H^s = \frac{H}{2^s}, \quad W^s = \frac{W}{2^s}, \quad D^s = 64 \cdot 2^s. \tag{2}$$

It should be noted that for brevity, Fig. 2 illustrates the proposed method for $s \in \{0, 1, 2\}$.

C. Temporal Feature Refinement

The TFR module, illustrated in Fig. 3, creates temporally refined feature maps using the self-attention mechanism along the temporal dimension [32]. Unlike the temporal modules in existing change detection and segmentation methods for SITS, our module preserves the temporal dimension.

The module takes as input a time series of feature maps at the same scale s and reshapes this 4-D tensor to a 3-D tensor $\mathbf{T}^s \in \mathbb{R}^{T\times D^s\times P^s}$ by flattening the spatial dimensions H^s and W^s. Consequently, T, D, and P represent the temporal, feature embedding, and spatial dimensions, respectively. After reshaping the feature map time series, self-attention is applied along the temporal dimension T for each cell in the spatial dimension

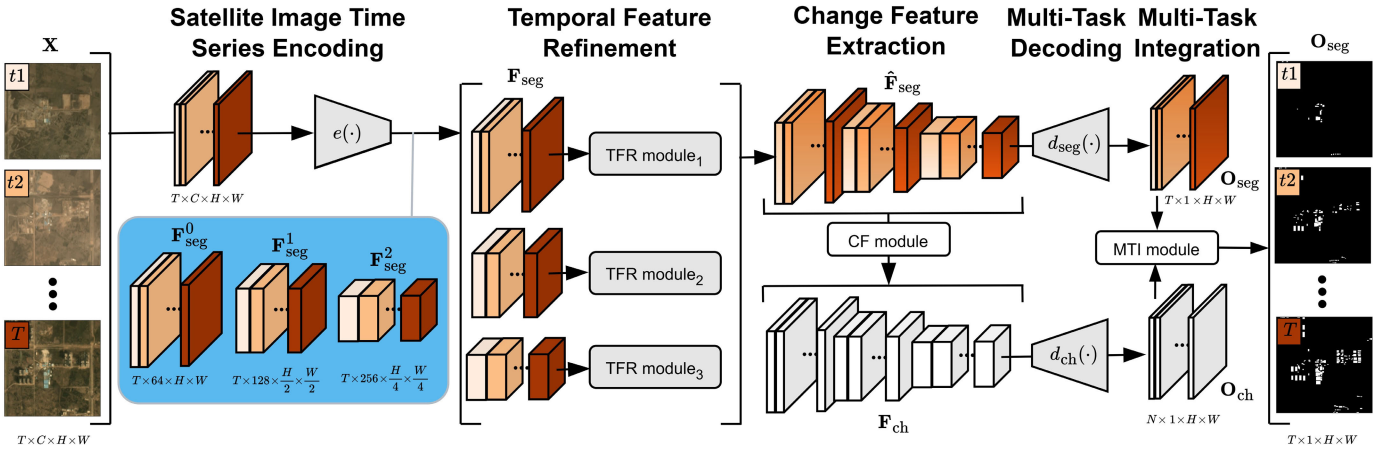


Fig. 2. Overview of the proposed method. First, an encoder extracts multiscale feature maps from an SITS. Next, transformer-based TFR modules enrich feature maps at each scale with multitemporal information, and the CF module generates bi-temporal difference feature maps from the temporally refined feature maps. Then, two separate decoders are used to obtain segmentation and change predictions from the respective feature maps. Finally, predictions for the two tasks are combined using an MTI module. For brevity, only three out of the five scales of the feature maps are shown.

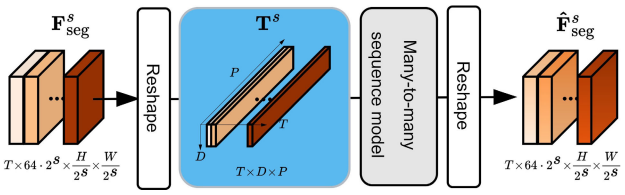


Fig. 3. Illustration of the TFR module, preserving the temporal dimension of the input feature maps.

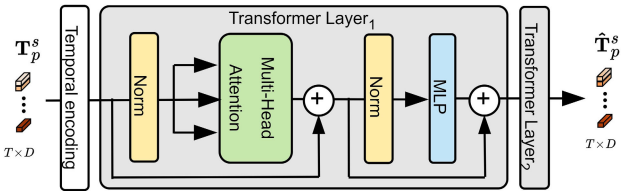


Fig. 4. Illustration of the transformer encoder layer applying multiheaded self-attention.

P. However, since the self-attention mechanism contains no recurrence, it is necessary to first inject information about the temporal position of the feature vectors in the time series. Specifically, temporal encodings having the same dimension as the feature vectors are generated based on sine and cosine functions of different frequencies [32]. These relative temporal encodings are then added to the feature vectors.

The tensor, enriched with relative temporal position information, is passed through two transformer encoder layers (see Fig. 4). The key component of the transformer encoder layer is the multihead attention block, which performs self-attention defined as follows:

$$\text{Att}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D}}\right)\mathbf{V} \tag{3}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are referred to as query, key, and value, respectively. The query–key–value triplet is computed with three linear projection layers with parameter matrices $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{D\times D}$ that are separately applied to a given cell of the 3-D tensor $\mathbf{T}^s_p$, where p denotes the cell index in $\mathbf{T}^s$.

The core idea of multihead attention is, however, that self-attention is performed multiple times in parallel using h attention heads, as follows:

$$\begin{aligned}\text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) &= \text{Concat}(\text{head}_1,\ldots,\text{head}_h)\mathbf{W}^O\\ \text{where head}_i &= \text{Att}\left(\mathbf{Q}\mathbf{W}^Q_i, \mathbf{K}\mathbf{W}^K_i, \mathbf{V}\mathbf{W}^V_i\right).\end{aligned} \tag{4}$$

Each head$_i$ performs self-attention on different projections of the keys, values, and queries obtained from linear layers with parameter matrices $\mathbf{W}^Q_i, \mathbf{W}^K_i, \mathbf{W}^V_i \in \mathbb{R}^{D\times D_{\text{head}}}$. Finally, the concatenated outputs of the heads are reprojected using parameter matrix $\mathbf{W}^O \in \mathbb{R}^{hD_{\text{head}}\times D}$. We employ $h = 2$ attention heads, where the head dimension is given by $D_{\text{head}} = D/h$.

After applying self-attention to each multitemporal feature vector, we obtain a 3-D tensor containing temporally refined building representations. In practice, however, all cells of tensor $\mathbf{T}^s$ are processed in parallel by incorporating them into the batch dimension which is omitted for clarity. Finally, the 3-D tensor is reshaped to the dimensions of the feature map time series by unflattering dimension P. We denote this temporally refined feature map time series with $\hat{\mathbf{F}}^s_{\text{seg}}$.

D. CF Extraction

The CF module is used to convert the temporally refined segmentation feature maps to CF maps. Specifically, we consider the temporally refined feature maps $\hat{\mathbf{F}}^t_{\text{seg}}$ and $\hat{\mathbf{F}}^k_{\text{seg}}$, corresponding to two images acquired at time t and time k, where $1 \le t < k \le T$. Then, CF map $\mathbf{F}^n_{\text{ch}}$ corresponding to the urban changes between the bi-temporal image pair is constructed as follows:

$$\mathbf{F}^n_{\text{ch}} = \hat{\mathbf{F}}^k_{\text{seg}} - \hat{\mathbf{F}}^t_{\text{seg}} \tag{5}$$

where n denotes a change edge between timestamps t and k. It should be noted that this is done for each scale s of the feature maps.

This operation is identical to the CF computation in the Siam-Diff method [14]. However, instead of only considering changes between the first and the last images of a time series,

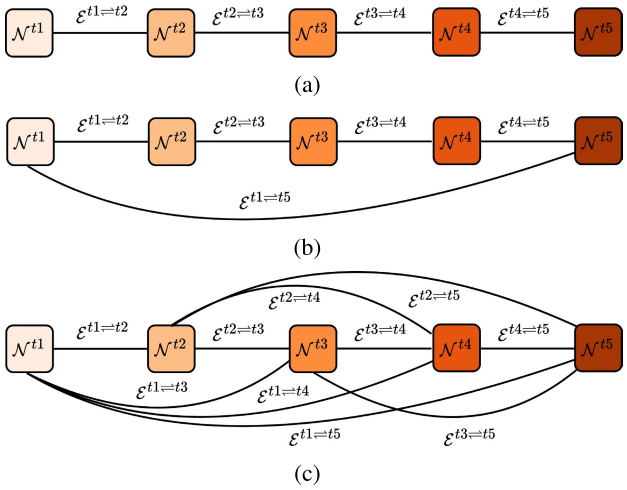


Fig. 5. Markov networks with different edge settings (exemplified for a time series with length $T = 5$). Nodes and edges are denoted by $\mathcal{N}$ and $\mathcal{E}$, respectively. (a) Adjacent. (b) Cyclic. (c) Dense.

the CF module computes CF maps for all possible combinations of image pairs. The total number of combinations N_{dense} is given by the length of the time series T, defined as follows:

$$N_{\text{dense}} = \frac{T(T-1)}{2}. \tag{6}$$

We refer to the scenario when all possible combinations of image pairs are considered as *dense*. However, we also investigate sparser settings such as *adjacent*, considering only changes between temporally adjacent images ($N_{\text{adjacent}} = T-1$), and *cyclic*, adding the changes between the first and last image to the adjacent setting ($N_{\text{cyclic}} = T$) (see Fig. 5(a)–(c) for visualizations of these settings in Markov networks with $T = 5$). We provide ablation results on the different edge settings in Section V-D. Finally, it should be noted that the CF module does not have any trainable parameters.

E. Multitask Decoding

Two separate decoders are deployed to convert the temporally enriched segmentation feature maps and the CF maps to building outputs and urban change outputs, respectively. Formally, we obtain T built-up area segmentation outputs $\mathbf{O}_{\text{seg}} \in (0,1)^{T\times H\times W}$ with the segmentation decoder $d_{\text{seg}}(\cdot)$ as follows:

$$\mathbf{O}_{\text{seg}} = d_{\text{seg}}(\hat{\mathbf{F}}_{\text{seg}}). \tag{7}$$

Furthermore, we obtain N change detection outputs $\mathbf{O}_{\text{ch}} \in (0,1)^{N\times H\times W}$ with the change decoder $d_{\text{ch}}(\cdot)$ as follows:

$$\mathbf{O}_{\text{ch}} = d_{\text{ch}}(\mathbf{F}_{\text{ch}}). \tag{8}$$

Both decoders follow the architecture of the U-Net expansive path consisting of four upsampling blocks followed by a 1×1 convolution layer and a sigmoid activation function. Upsampling blocks double the height and width of feature maps via a transpose conv 2×2 layer. Upsampled feature maps are then concatenated along the channel dimension with the temporally refined feature maps matching their scale (skip connection). Subsequently, the layer triplet 3×3 convolution, batch normalization, and ReLu activation is applied twice.

F. Loss Function

The network is trained using a loss function composed of two terms, namely, for the urban change detection task ($\mathcal{L}_{\text{ch}}$) and the building segmentation task ($\mathcal{L}_{\text{seg}}$). For both loss terms, a Jaccard metric measuring the similarity between continuous network outputs $\mathbf{O} \in (0,1)$ and binary labels $\mathbf{Y} \in \{0,1\}$ is used [46]. We denote the Jaccard metric by $J(,)$ and define the loss function as follows:

$$\mathcal{L} = \sum_{t=1}^{T} J\left(\mathbf{O}_{\text{seg}}^{t}, \mathbf{Y}_{\text{seg}}^{t}\right) + \sum_{n=1}^{N} J\left(\mathbf{O}_{\text{ch}}^{n}, \mathbf{Y}_{\text{ch}}^{n}\right) \tag{9}$$

where T denotes the length of the time series and N denotes the number of edges (i.e., combinations of bi-temporal image pairs) considered. Segmentation and change labels are denoted by $\mathbf{Y}_{\text{seg}} \in \{0,1\}^{T\times H\times W}$ and $\mathbf{Y}_{\text{ch}} \in \{0,1\}^{N\times H\times W}$, respectively. Specifically, we assume that pixel-wise building annotations, $\mathbf{Y}_{\text{seg}}$, are available and derive pixel-wise built-up changes, $\mathbf{Y}_{\text{ch}}$, according to the considered edges.

G. Multitask Integration

To combine segmentation and change predictions, we propose the MTI module which determines the optimal building segmentation output for each image in a time series. Since this is a pixel-based approach, we represent the location of a specific pixel in the segmentation and change output notations by introducing superscript coordinates i and j. Following that, $\mathbf{O}_{\text{seg}}^{(i,j),t}$ denotes the segmentation output for a specific pixel i and j at timestamp t, where $i \in \{1,\ldots,H\}$, $j \in \{1,\ldots,W\}$, and $t \in \{1,\ldots,T\}$. Likewise, the change output for a specific pixel is denoted by $\mathbf{O}_{\text{ch}}^{(i,j),n}$, where n denotes the change edge connecting timestamps t and k.

The core idea of the MTI module is to represent segmentation and change outputs in a pairwise Markov network. This subclass of Markov networks is associated with an undirected graph $G = (\mathcal{N}, \mathcal{E})$ in which the nodes $\mathcal{N}$ correspond to random variables and the edges $\mathcal{E}$ represent pairwise relationships between the nodes (see [47]). For a given pixel, we construct a Markov network with T nodes corresponding to the timestamps in an image time series. Specifically, all nodes in the network correspond to a binary variable representing the presence of buildings (i.e., $\mathcal{N}^t \in \{\text{true}, \text{false}\}$). We use state 1 to denote true, representing the presence of a building, and state 0 to denote false, representing the absence of a building. Adjacent nodes in the network are connected with $N-1$ edges, where we use $\mathcal{E}^{t\rightleftharpoons k}$ to denote an edge connecting timestamps t and k. We refer to this Markov network structure as an adjacent network [see Fig. 5(a)].

To represent the segmentation and change outputs, the graph structure needs to be associated with a set of parameters that capture the relationships between nodes. The parameterization in a pairwise Markov network is achieved by assigning *factors* over nodes or edges, where a factor ϕ, also referred to as *potential*, is a function from value assignments of random variables to real positive numbers $\mathbb{R}^+$. Thus, a pairwise Markov

TABLE I
OVERVIEW OF DATASET CHARACTERISTICS FOR SN7, THE WUSU DATASET, AND THE TSCD DATASET

Dataset	Location	Satellite	Resolution		
			Spectral	Spatial	Temporal
SpaceNet 7	Global (60 sites)	PlanetScope	3 bands (RGB)	4 m	5 images*(2017 to 2020)
WUSU	Wuhan, China	Gaofen-2	4 bands (RGB + NIR)	1 m	3 images (2015, 2016, 2018)
TSCD	Chengdu, China	WorldView-2	3 bands (RGB)	0.5 m	4 images (2016, 2018, 2020, 2022)

* Approximately 24 monthly timestamps acquired between 2017 and 2020 are available for each site in the dataset.

network is associated with a set of node potentials $\{\phi(\mathcal{N}_t) : t = 1, \ldots, T\}$ and a set of edge potentials $\{\phi(\mathcal{N}_t, \mathcal{N}_k) : (\mathcal{N}_t, \mathcal{N}_k) \in G\}$. The overall distribution represented by the network is then the normalized product of all the node and edge potentials.

The segmentation outputs for a specific pixel are incorporated into the Markov network by assigning a factor ϕ_t over each node $\mathcal{N}_t$. Then, the segmentation outputs are encoded as node potentials, as follows:

$$\begin{aligned}\phi_t(\mathcal{N}^t = 1) &= \mathbf{O}_{\text{seg}}^{(i,j),t} \\ \phi_t(\mathcal{N}^t = 0) &= 1 - \mathbf{O}_{\text{seg}}^{(i,j),t}.\end{aligned} \tag{10}$$

These node potentials characterize the bias of nodes toward state 1 or 0, representing the presence or absence of a building, respectively. We refer to this Markov network as *degenenrate* network, characterized by the absence of functions that capture the interactions between nodes.

To incorporate the change outputs for a specific pixel, we first include additional edges for the edge settings cyclic and dense. Specifically, for the cyclic case [see Fig. 5(b)], edge $\mathcal{E}^{t1 \rightleftharpoons T}$, connecting the first node $\mathcal{N}^{t1}$ and the last node $\mathcal{N}^T$, is added. On the other hand, for the dense case [see Fig. 5(c)], all possible nonadjacent edges are added to the graph. Then, we define factors over the edges in the Markov network to add pairwise interactions of connected nodes. Specifically, we define pairwise potentials ϕ_n for each edge $\mathcal{E}^n$, connecting two nodes $\mathcal{N}^t$ and $\mathcal{N}^k$. Since all variables in the network are binary, each factor over an edge has four parameters. The change outputs are encoded as edge potentials for the four combinations of states, as follows:

$$\begin{aligned}\phi_n(\mathcal{N}^t = 0, \mathcal{N}^k = 1) &= \mathbf{O}_{\text{ch}}^{(i,j),n} \\ \phi_n(\mathcal{N}^t = 1, \mathcal{N}^k = 0) &= \mathbf{O}_{\text{ch}}^{(i,j),n} \\ \phi_n(\mathcal{N}^t = 0, \mathcal{N}^k = 0) &= 1 - \mathbf{O}_{\text{ch}}^{(i,j),n} \\ \phi_n(\mathcal{N}^t = 1, \mathcal{N}^k = 1) &= 1 - \mathbf{O}_{\text{ch}}^{(i,j),n}\end{aligned} \tag{11}$$

where edge n is connecting timestamps t and k.

The value associated with each particular assignment of states denotes the affinity between the two states. Consequently, the higher the value assigned to the edge potential for a particular combination of states, the more compatible these two states are.

To define a global model from the local interactions defined in the parameterization of the Markov network, we take the product of the local factors and then normalize it. Once the distribution is defined, we perform a maximum a posteriori query to find the optimal state assignment for each node in the graph. The optimal state assignment corresponds to the configuration that minimizes the overall energy determined by the node and edge potentials assigned to the graph, as defined in (10) and (11). Therefore, the resulting state assignment for the nodes is optimal with respect to the potentials obtained from the network outputs but not necessarily with respect to the segmentation and change labels. We perform inference using the belief propagation algorithm (see [47, Algorithm 10.4]). Due to the absence of loops in the graph, belief propagation provides an exact solution. Finally, it should be noted that the MTI module does not contain any trainable parameters and is only deployed during inference.

IV. EXPERIMENTAL SETTING

A. Datasets

A diverse set of multitemporal datasets is used to evaluate the proposed method. Table I compares key characteristics of these datasets, and detailed descriptions are provided in the following paragraphs.

1) SpaceNet 7: The SN7 dataset features time series of satellite images acquired by the PlanetScope constellation between 2017 and 2020 for 60 sites spread across the globe [48]. Each time series consists of about 24 monthly mosaics with a spatial resolution of 4 m (approximately 1024 × 1024 pixels). Furthermore, the SN7 dataset provides manually annotated building footprints, whereas annotations are missing for image parts affected by clouds. While the task of the original SN7 challenge was to track these building footprints (i.e., vector data), the SN7 dataset has been leveraged for a diverse set of tasks such as urban mapping [49], building counting [50], and urban change forecasting [51]. To split the SN7 dataset into training, validation, and test areas, we apply the within-scene splits recommended in [50]. Specifically, approximately 80% of a scene (top part) is used as a source of training patches, and the remaining 20% (bottom part) is divided evenly into validation (bottom-left part) and test (bottom-right part) areas. Within-scene splits minimize the occurrence of out-of-distribution data during testing while simultaneously avoiding data leakage between the training and test set by utilizing spatially disjoint areas for the different dataset splits. During training, samples from the training areas are generated by randomly selecting T timestamps from the time series of a site. The rasterized building labels (see [37]) for these timestamps, were used to compute the change label. We draw 100 samples from each site during an epoch to reach

an adequate number of steps before model evaluation. For model evaluation (validation and testing), the first and the last cloud-free images of a time series, in addition to evenly spaced intermediate images, were selected.

2) Wuhan Urban Semantic Understanding Dataset: The WUSU dataset features tri-temporal high-resolution Gaofen-2 images covering two districts in Wuhan (Hubei Province, central China) in 2015, 2016, and 2018 [52]. The preprocessing workflow of the satellite images includes orthographic correction and multitemporal image registration. Furthermore, the four multispectral bands acquired at 4 m spatial resolution are pansharpened to a spatial resolution of 1 m, resulting in images of size 6358 × 6382 and 7025 × 5500 pixels for Hongshan District and Jiang'an District, respectively. In addition to the Gaofen-2 images, the WUSU dataset provides corresponding land-use/land-cover (LULC) labels, including manually refined building annotations (Class 2 Low building and Class 3 high building). Since the proposed method requires binary building labels, Class 2 and Class 3 were remapped to the foreground class, whereas all other classes were remapped to the background class. We follow the within-scene split recommended by the authors, using the top halves of the six images for the test set and the bottom halves for the training set that was further divided into training (90%) and validation (10%) tiles.

3) Time Series Change Detection Dataset: The TSCD dataset features bi-annual WorldView-2 satellite images acquired over Chengdu (Sichuan Province) between 2016 and 2022 [53]. The images have a resolution of approximately 0.5 m and are split into 512 × 512 pixel tiles. The tiles are divided into a training, validation, and test set. The TSCD dataset provides building change labels for each adjacent image pair (2016–2018, 2018–2020, and 2020–2022). We derived change labels for an arbitrary image pair from the time series by considering all adjacent change labels connecting this pair and computing the number of changes. An odd number of adjacent changes indicates a change between the image pair, whereas an even number indicates no change.

B. Baseline and Benchmark Methods

We selected a comprehensive set of baseline and benchmark methods for quantitative and qualitative comparisons with the proposed methods. These selected methods are grouped into two categories, which are listed below.

1) Bi-Temporal Change Detection Methods:

1) Siam-Diff [14] employs a Siamese encoder to extract feature maps from bi-temporal images. A decoder produces the change prediction from the subtracted feature maps. The encoders and decoder follow the U-Net architecture.
2) SNUNet [15] replaces the architecture in the Siam-Diff network with a Nested U-Net (UNet++ [40]). In addition, a channel attention module is incorporated into the architecture.
3) DTCDSCN [36] combines a typical Siamese ConvNet for bi-temporal change detection with a dual attention module and two additional decoders with shared weights for building segmentation.
4) BIT [19] employs a bi-temporal image transformer module that operates in a compact token space to refine features extracted by a Siamese ConvNet.
5) AMTNet [17] also extracts features using a Siamese ConvNet, and combines attention mechanisms and multiscale processing techniques to model contextual information in bi-temporal images.
6) ScratchFormer [22] introduces shuffled sparse attention layers in a Siamese ConvNet encoder to effectively capture semantic changes when training from scratch.

2) Multitemporal Change Detection/Segmentation Methods:

1) L-UNet [27] employs a shared U-Net for multiscale feature extraction in SITS and uses LSTM modules [29] for temporal modeling. The LSTM modules produce a single multiscale feature map which is transformed into the output feature map using a U-Net decoder.
2) Multitask L-UNet [28] adds a semantic U-Net decoder to L-UNet to segment buildings in the first and last images of a time series.
3) U-TAE [33] uses a shared U-Net encoder to extract multiscale feature maps for the SITS. A temporal attention encoder (L-TAE [45] is then used to collapse the temporal dimension, before using the U-Net decoder to produce a single output feature map.
4) TSViT [34] splits an SITS into nonoverlapping patches in space and time which are tokenized and subsequently processed by a temporo-spatial encoder. A segmentation head reassembles the encoded features into a single output feature map.
5) U-TempoNet [44] uses a shared ConvNet encoder to extract multiscale feature maps for all images. Subsequently, a single multiscale feature map, obtained through temporal modeling with a bidirectional LSTM, is processed with a decoder to produce the output feature map.

The Siamese ConvNets Siam-Diff, SNUNet, and DTCDSCN are commonly used as change detection baselines, whereas BIT, AMTNet, and ScratchFormer represent recent methods combining Siamese ConvNets with transformers. On the other hand, L-UNet and Multitask L-UNet are benchmark methods for multitemporal change detection. It should be noted that Multitask L-UNet and DTCDSCN are multitask methods that perform change detection and building segmentation. Finally, U-TAE, TSViT, and U-TempoNet are recent segmentation methods for SITS inputs that can be adopted for multitemporal change detection without architectural changes.

C. Model Evaluation

Three accuracy metrics were used for the quantitative assessment of model predictions: $F1$ score, intersection over union (IoU), and overall accuracy (OA). Formulas for the metrics are given below [see (12)–(14)], where TP, TN, FP, and FN represent the number of true positive, true negative, false positive, and false negative pixels, respectively,

$$F1 \text{ score} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})} \tag{12}$$

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{13}$$

$$\text{OA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{14}$$

Using these two accuracy metrics, we assessed model performance across three tasks to accommodate the large variety of baseline and benchmark methods. These tasks are described in detail in the following.

1) *Bi-Temporal Change Detection:* Measures the accuracy of the predicted changes between the first and last image of a time series.
2) *Continuous Change Detection:* Measures the average accuracy of the predicted changes between consecutive image pairs in a time series.
3) *Segmentation:* Measures the accuracy of the building predictions corresponding to the last image of a time series.

The first task focuses on urban change detection from image pairs acquired multiple years apart. This task is considered by most urban change detection methods. For bi-temporal change detection methods, changes were directly predicted based on the first-last image pair, ignoring intermediate images in a time series. On the other hand, the second task focuses on assessing change predictions between consecutive image pairs in a time series. Consequently, the continuous urban change detection task focuses on image pairs with periods between acquisition dates that are considerably shorter (i.e., annual and subannual). The last task assesses the auxiliary segmentation task of multitask methods and segmentation models. It should be noted that the change detection performance of segmentation models is not assessed because postclassification comparison suffers from the accumulation of classification errors [54].

D. Implementation Details

We implement the proposed method using the deep learning framework PyTorch [55]. In addition, the einops package [56] was used to efficiently reshape feature maps, and the pgmpy package [57] to implement the Markov network and perform belief propagation. Models were trained for a maximum duration of 100 epochs on NVIDIA GeForce RTX 3080 graphics cards, using early stopping with patience 10 to prevent models from overfitting to the training set. AdamW was used as optimizer [58] with a linear learning rate scheduler. The remainder of this section describes the training setup in detail.

1) Augmentations: To enhance the training dataset, we applied four data augmentation operations, namely, rotations ($k \cdot 90°$, where k is randomly selected from $\{0, 1, 2, 3\}$), flips (horizontal and vertical with a probability of 50%), Gaussian blur, and random color jittering. The parameters that determine how much to jitter the brightness, contrast, saturation, and hue of an image were set to 0.3 [20]. For validation and testing, on the other hand, no data augmentation was applied.

2) Oversampling: To account for the fact that the occurrence of change is usually considerably less frequent than no change [59], change areas were oversampled during network training. For a given site, twenty patches of size 64×64 pixels were randomly cropped from the change label, before assigning each patch a probability according to its change pixel percentage, including a base probability for patches with no change pixels. A single patch was chosen based on those probabilities. For transformer-based methods (BIT, AMTNet, and Scratchformer), the patch size was increased to 128×128 pixels to include more long-range spatial context.

3) Hyper-Parameter Tuning: For each model, hyper-parameters were tuned empirically on the validation set using grid search. Specifically, an exhaustive search with three learning rates ($1 \cdot 10^{-5}$, $5 \cdot 10^{-5}$, $1 \cdot 10^{-4}$) and two batch sizes (8, 16) was performed to determine the optimum values of hyper-parameters. Then, five models were trained with the best hyper-parameters but different seeds for weight initialization and data shuffling. Consequently, reported values correspond to the average of five runs.

Multitask L-UNet requires additional hyper-parameters to balance the contribution of the segmentation and change loss terms, which we adopted from the article [28]. All bi-temporal urban change detection methods were trained on the cyclic edges setting [see Fig. 5(b)].

V. Experiment Results

In this section, we present the quantitative and qualitative results on the SN7 and WUSU datasets, and the ablation study results. It should be noted that all accuracy values are reported on the respective test sets and correspond to the mean values obtained from five models. These were trained with different seeds using the best hyper-parameters determined with a grid search (see Section IV-D). On the other hand, the median model is used for the qualitative results, which are only shown for competitive methods selected based on the quantitative results.

A. SpaceNet 7

The quantitative results for the SN7 dataset are listed in Table II. The proposed method achieved the highest $F1$ scores and IoU values for both urban change detection tasks (i.e., bi-temporal and continuous). Several multitemporal models (L-UNet, Multitask L-UNet, and U-TempoNet) outperform bi-temporal change detection methods on the bi-temporal task while others are less effective (U-TAE and TSViT). Among the bi-temporal methods, ScartchFormer and the ConvNet-based methods Siam-Diff and SNUNet achieved the highest accuracy values. For building segmentation, the proposed method also outperformed the other multitask methods including DTCD-SCN and Multitask L-UNet.

Figs. 6 and 7 show qualitative change detection and building segmentation results for two SN7 test sites located in Australia and USA, respectively. For both sites, the proposed method detects urban changes more accurately than competing methods (DTCDSCN, ScratchFormer, and Multitask L-UNet). In particular, the continuous change detection results for consecutive image pairs (rows two to five) show a better agreement with the label than those of the other methods. In addition, the change outputs of the proposed method show a high level of consistency, meaning that the aggregated continuous changes correspond to the bi-temporal

TABLE II

QUANTITATIVE RESULTS ON THE SN7 TEST AREAS. THE BEST AND SECOND-BEST PERFORMANCES ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY. "-" DENOTES THAT THE ACCURACY METRIC DOES NOT APPLY TO A SPECIFIC METHOD SINCE THE CORRESPONDING VARIABLE IS NOT PREDICTED

Method	Change detection		Segmentation
	Bi-temporal (F1 / IoU / OA)	Continuous (F1 / IoU / OA)	(F1 / IoU / OA)
Siam-Diff	0.453 / 0.293 / 98.8	0.273 / 0.158 / 99.5	-
SNUNet	0.454 / 0.294 / 98.8	0.300 / 0.177 / 99.6	-
DTCDSCN	0.413 / 0.260 / 98.7	0.250 / 0.143 / 99.6	0.488 / 0.323 / 92.3
BIT	0.386 / 0.239 / 99.0	0.275 / 0.160 / 99.6	-
AMTNet	0.424 / 0.269 / 98.7	0.282 / 0.164 / 99.6	-
ScratchFormer	0.468 / 0.305 / 98.9	0.328 / 0.196 / 99.6	-
L-UNet	0.519 / 0.350 / 98.9	-	-
MT L-UNet	0.515 / 0.347 / 98.9	-	0.512 / 0.344 / 92.7
U-TAE	0.366 / 0.225 / 97.5	-	-
TSViT	0.168 / 0.092 / 97.2	-	-
U-TempoNet	0.494 / 0.328 / 98.8	-	-
Proposed	0.551 / 0.381 / 99.0	0.414 / 0.261 / 99.7	0.596 / 0.424 / 94.3

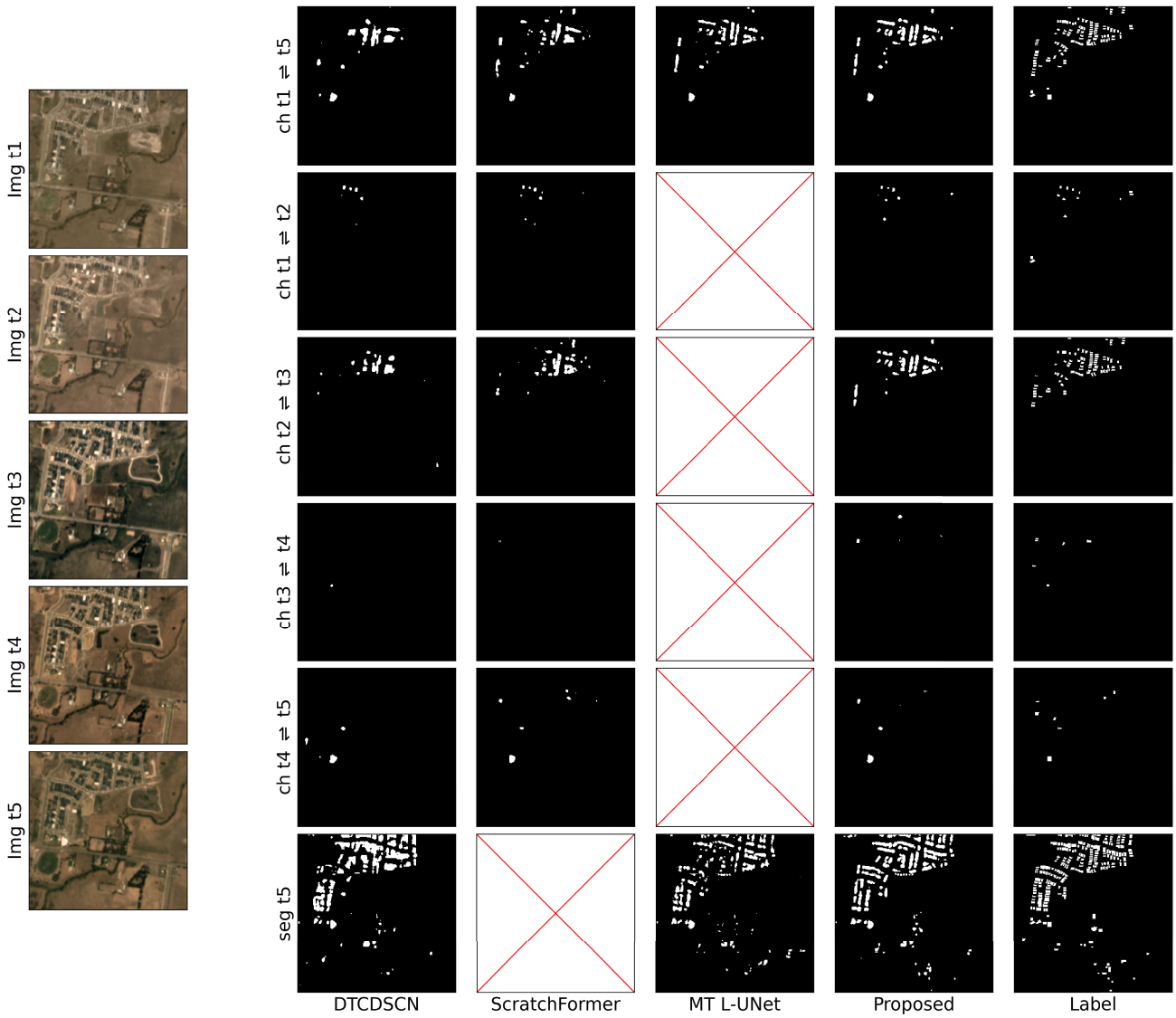


Fig. 6. Qualitative urban change and building segmentation results for an SN7 test site located in Australia. The PlanetScope SITS is shown on the left, and the model outputs and the label are shown on the right. The top row shows the changes between the first and last image of the time series, rows two to five show the continuous changes between consecutive image pairs, and the bottom row the buildings segmentation corresponding to the last image.

changes between the first and the last image (top row). Finally, the proposed method maps buildings with more detail than the competing methods, as shown in the bottom row of Figs. 6 and 7.

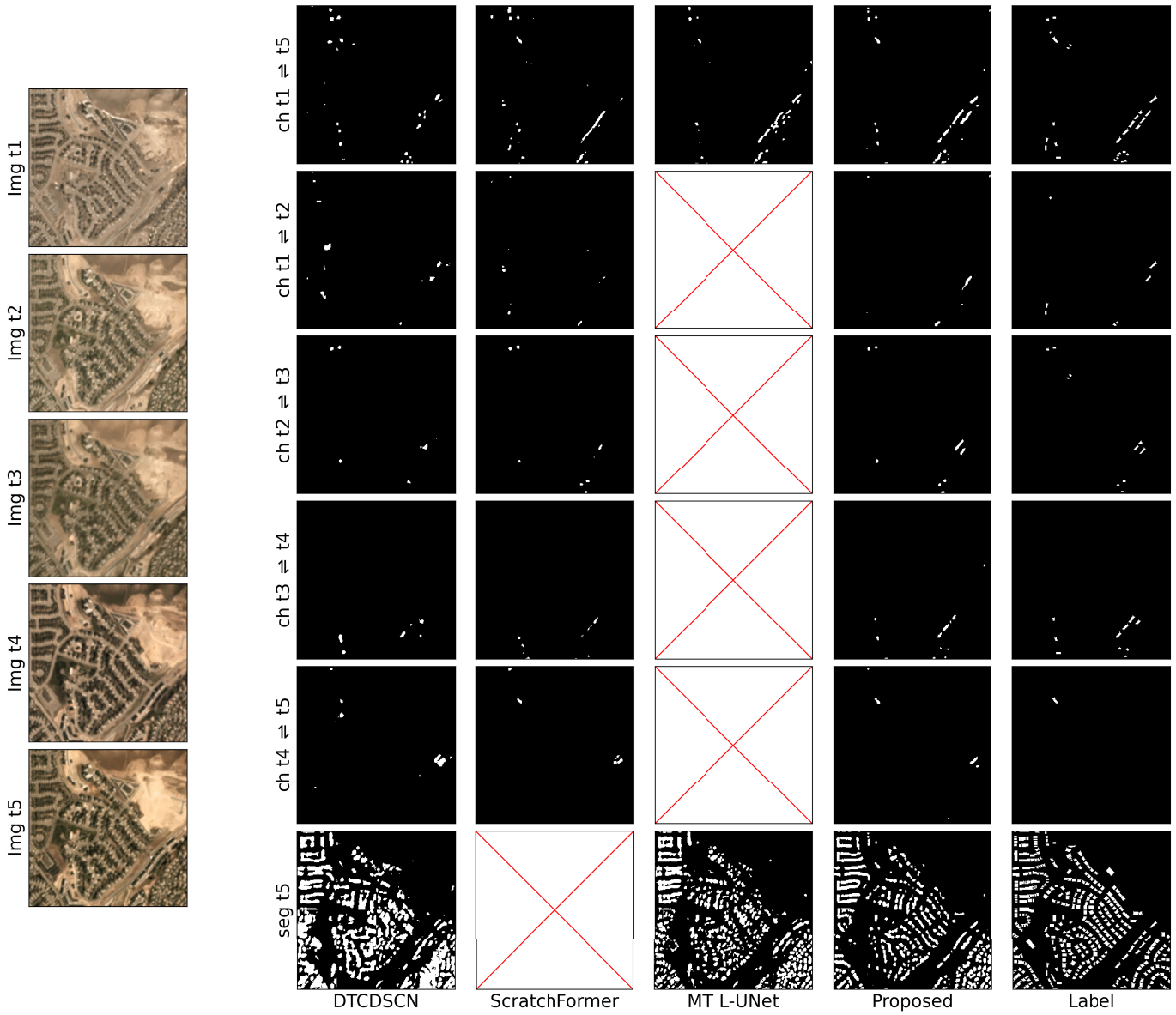


Fig. 7. Qualitative urban change and building segmentation results for an SN7 test site located in USA. The PlanetScope SITS is shown on the left, and the model outputs and the label are shown on the right. The top row shows the changes between the first and last image of the time series, rows two to five show the continuous changes between consecutive image pairs, and the bottom row the buildings segmentation corresponding to the last image.

B. Wuhan Urban Semantic Understanding Dataset

Change detection performance on the WUSU dataset is lower than on the SN7 dataset (see Table III). For bi-temporal change detection, only ScratchFormer, L-UNet, and Multitask L-UNet exceed an $F1$ score of 0.275 and an IoU value of 0.160. In comparison, accuracy values for the continuous change detection task are slightly higher, except for the proposed method. Overall, our method outperformed all other methods on both change detection tasks. This also applied to the building segmentation task, where the proposed method achieved an $F1$ score of 0.663 and an IoU value of 0.496.

Figs. 8 and 9 show qualitative change detection and building segmentation results for two WUSU test sites located in Wuhan's Jiang'an District in China. The change detection results produced by the proposed method show good agreement with the label, especially in comparison with the competing methods. As for SN7, the proposed method achieved a high level of consistency between the continuous change detection outputs (rows two and three) and the change output between the first and last image (top row). In contrast, the bi-temporal change detection methods identified changes between the first and last images (ch $t1 \rightleftharpoons t3$) that are present in neither of the continuous change rows (ch $t1 \rightleftharpoons t2$ and ch $t2 \rightleftharpoons t3$). Finally, the bottom row demonstrates that all methods accurately map buildings, but the proposed method achieved more detailed building delineations than its competitors. It should also be noted that the label does not distinguish individual buildings in very dense built-up areas (e.g., the bottom left area in Fig. 8).

C. Time Series Change Detection Dataset

The quantitative results for the TSCD dataset are listed in Table IV. Multitemporal methods outperformed bi-temporal

TABLE III

QUANTITATIVE RESULTS ON THE WUSU TEST AREAS. THE BEST AND SECOND-BEST PERFORMANCES ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY. "-" DENOTES THAT THE ACCURACY METRIC DOES NOT APPLY TO A SPECIFIC METHOD SINCE THE CORRESPONDING VARIABLE IS NOT PREDICTED

Method	Change detection		Segmentation
	Bi-temporal (F1 / IoU / OA)	Continuous (F1 / IoU / OA)	(F1 / IoU / OA)
Siam-Diff	0.175 / 0.096 / 96.1	0.236 / 0.134 / 97.4	-
SNUNet	0.188 / 0.104 / 94.9	0.211 / 0.118 / 96.5	-
DTCDSCN	0.278 / 0.162 / 96.2	0.318 / 0.189 / 97.6	0.539 / 0.369 / 84.2
BIT	0.213 / 0.120 / 96.5	0.314 / 0.187 / 97.9	-
AMTNet	0.187 / 0.104 / 95.9	0.264 / 0.152 / 97.6	-
ScratchFormer	0.324 / 0.193 / 96.6	0.352 / 0.214 / 97.8	-
L-UNet	0.279 / 0.162 / 96.0	-	-
MT L-UNet	0.276 / 0.161 / 96.1	-	0.479 / 0.315 / 83.1
U-TAE	0.267 / 0.154 / 92.1	-	-
TSViT	0.219 / 0.123 / 92.1	-	-
U-TempoNet	0.246 / 0.141 / 95.0	-	-
Proposed	0.440 / 0.282 / 97.0	0.389 / 0.242 / 98.3	0.663 / 0.496 / 88.9

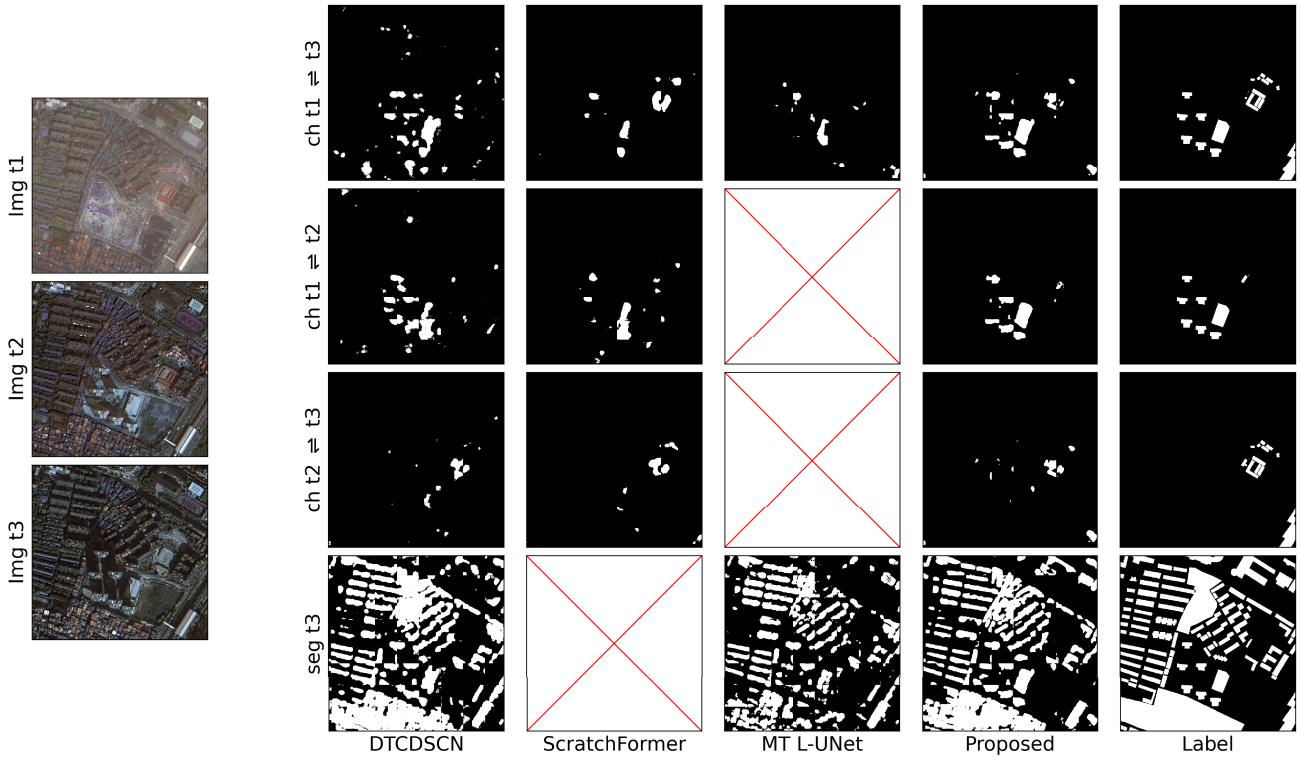


Fig. 8. Qualitative urban change and building segmentation results for a WUSU test site located in Wuhan's Jiang'an District, China. The Gaofen-2 SITS is shown on the left, and the model outputs and the label are shown on the right. The top row shows the changes between the first and last image of the time series, rows two and three show the continuous changes between consecutive image pairs, and the bottom row the buildings segmentation corresponding to the last image.

methods on detecting changes between the first and last timestamps of SITS. On the continuous task, bi-temporal methods capable of modeling long-range spatial dependencies in VHR imagery (BIT, AMTNet, and ScratchFormer) outperformed Siam-Diff and SNUNet. However, our method achieved the highest performance on both tasks despite not leveraging spatial attention.

D. Ablation Study

1) Loss Function: Table V shows the ablation results for different change loss edge settings (first-last, adjacent, cyclic, and dense) and the segmentation loss. MTI was disabled for this experiment to isolate the effect of the loss function on network performance. It should also be noted that the settings adjacent, cyclic, and dense all require building annotations for each image in the time series as labels (see Section III-F), whereas first-last only requires building annotations for the first and last images. Considering additional change edges in the loss function generally improves continuous change detection performance. In contrast, a single change loss term suffices for the bi-temporal change detection task. The segmentation loss term improves performance for both change

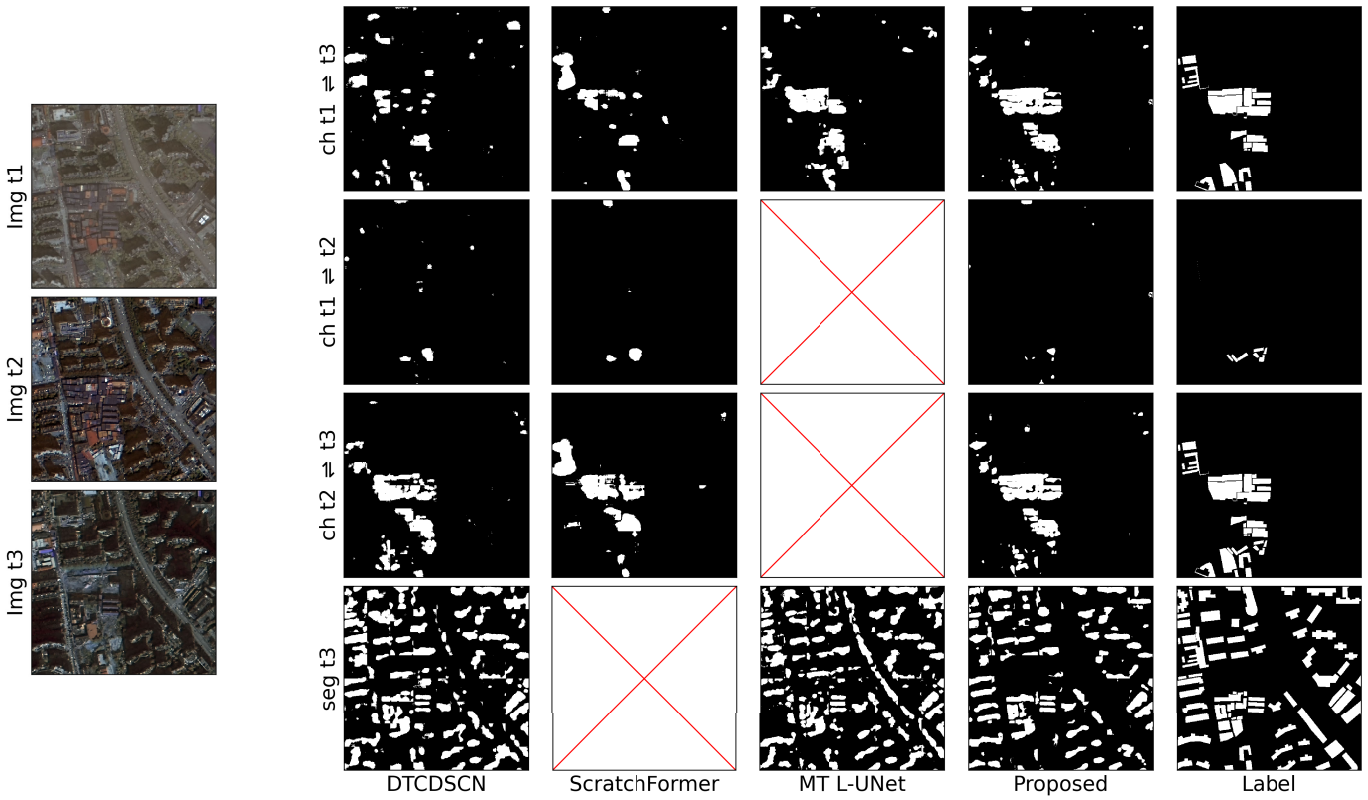


Fig. 9. Qualitative urban change and building segmentation results for a WUSU test site located in Wuhan's Jiang'an District, China. The Gaofen-2 SITS is shown on the left, and the model outputs and the label are shown on the right. The top row shows the changes between the first and last image of the time series, rows two and three show the continuous changes between consecutive image pairs, and the bottom row the buildings segmentation corresponding to the last image.

TABLE IV

QUANTITATIVE RESULTS ON THE TSCD TEST AREAS. THE BEST AND SECOND-BEST PERFORMANCES ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY. "-" DENOTES THAT THE ACCURACY METRIC DOES NOT APPLY TO A SPECIFIC METHOD SINCE THE CORRESPONDING VARIABLE IS NOT PREDICTED

Method	Change detection	
	Bi-temporal (F1 / IoU / OA)	Continuous (F1 / IoU / OA)
Siam-Diff	0.211 / 0.118 / 89.3	0.284 / 0.166 / 93.2
SNUNet	0.220 / 0.123 / 89.4	0.328 / 0.197 / 92.6
BIT	0.203 / 0.114 / 90.0	0.363 / 0.222 / 94.5
AMTNet	0.298 / 0.178 / 91.1	0.398 / 0.249 / 94.9
ScratchFormer	0.270 / 0.157 / 91.7	0.504 / 0.340 / 95.9
L-UNet	0.401 / 0.251 / 84.6	-
U-TAE	0.415 / 0.262 / 81.6	-
TSViT	0.355 / 0.216 / 79.3	-
U-TempoNet	0.435 / 0.279 / 85.1	-
Proposed	0.543 / 0.373 / 93.7	0.573 / 0.402 / 96.6

detection tasks, which holds for all change loss settings. Therefore, the optimal loss setting consists of a change loss with dense edges combined with a segmentation loss.

2) TFR Module: We perform an additional ablation experiment investigating the contribution of the TFR module and testing if recurrent sequence models, particularly recurrent neural networks (RNNs) [60] and LSTMs [29], can be considered as alternatives to self-attention. We run all settings with and without the MTI module due to the complementary nature of the modules. Table VI shows that adding the TFR module to our framework achieves large performance gains across all tasks and datasets. Among the sequence models, self-attention outperformed the recurrent sequence models on the segmentation task across all datasets and the continuous urban change detection task on SN7 and the TSCD dataset. It only fell short of LSTMs on the WUSU dataset. Finally, for bi-temporal change detection, self-attention outperformed RNNs on all datasets but LSTMs achieved better results before MTI. In general, self-attention is the most effective sequence model, especially in combination with the MTI module. However, LSTMs should be considered for multitemporal change detection and datasets with few timestamps.

3) MTI Module: We perform a third ablation experiment investigating performance gains from leveraging additional edges in the MTI module. Here, we apply MTI with different settings to the outputs of our method trained using the change loss with the maximum number of edges and the segmentation loss. The TFR module using self-attention was enabled. A degenerate Markov network that uses no change information (i.e., only the segmentation information represented as nodes) is added as a baseline. Table VII shows that introducing change information in the MTI module results in considerable change detection performance gains compared to the degenerate setting, as well as minor segmentation performance gains. For example, using the adjacent setting improves change detection performance (first-last) by 44.8% and 61.6% in terms of

TABLE V

ABLATION RESULTS FOR THE LOSS FUNCTION WITH DIFFERENT EDGE SETTINGS FOR THE CHANGE LOSS TERM. THE BEST AND SECOND-BEST PERFORMANCES ON THE SN7, WUSU, AND TSCD DATASETS ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY

Dataset	Change loss	Seg loss	Change detection		Segmentation
			Bi-temporal (F1 / IoU / OA)	Continuous (F1 / IoU / OA)	(F1 / IoU / OA)
SN7	first-last	✗	0.453 / 0.294 / 98.7	0.253 / 0.145 / 99.6	-
	first-last	✓	0.519 / 0.350 / 98.8	0.323 / 0.193 / 99.7	0.565 / 0.394 / 93.8
	adjacent	✗	0.479 / 0.315 / 98.8	0.357 / 0.217 / 99.6	-
	adjacent	✓	0.520 / 0.352 / 98.9	0.384 / 0.238 / 99.6	0.584 / 0.413 / 94.1
	cyclic	✗	0.516 / 0.348 / 98.8	0.361 / 0.220 / 99.6	-
	cyclic	✓	0.532 / 0.363 / 98.9	0.377 / 0.233 / 99.7	0.581 / 0.409 / 94.0
	dense	✗	0.511 / 0.343 / 98.8	0.364 / 0.223 / 99.6	-
	dense	✓	0.537 / 0.367 / 98.8	0.397 / 0.248 / 99.7	0.593 / 0.422 / 94.3
WUSU	first-last	✗	0.274 / 0.159 / 95.7	0.234 / 0.133 / 96.3	-
	first-last	✓	0.401 / 0.251 / 96.5	0.297 / 0.175 / 98.3	0.650 / 0.482 / 88.4
	adjacent	✗	0.251 / 0.144 / 96.4	0.297 / 0.175 / 97.9	-
	adjacent	✓	0.356 / 0.217 / 95.3	0.373 / 0.229 / 98.2	0.650 / 0.482 / 88.6
	cyclic	✗	0.328 / 0.196 / 96.0	0.289 / 0.169 / 98.2	-
	cyclic	✓	0.420 / 0.266 / 96.6	0.391 / 0.243 / 98.3	0.660 / 0.493 / 88.8
TSCD	first-last	✗	0.270 / 0.157 / 84.9	0.157 / 0.086 / 62.8	
	adjacent	✗	0.255 / 0.147 / 89.6	0.453 / 0.296 / 95.1	-
	cyclic	✗	0.512 / 0.344 / 92.9	0.553 / 0.383 / 96.4	-
	dense	✗	0.543 / 0.373 / 93.7	0.573 / 0.402 / 96.6	-

TABLE VI

ABLATION RESULTS FOR THE TFR MODULE WITH DIFFERENT SEQUENCE MODELS. THE MAXIMUM NUMBER OF EDGES WAS USED FOR EACH EXPERIMENT IN THE CHANGE LOSS AND MTI MODULE. THE BEST AND SECOND-BEST PERFORMANCES ON THE SN7, WUSU, AND TSCD DATASETS ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY

Dataset	TFR module	Sequence model	MTI module	Change detection		Segmentation
				Bi-temporal (F1 / IoU / OA)	Continuous (F1 / IoU / OA)	(F1 / IoU / OA)
SN7	✗	-	✗	0.490 / 0.324 / 98.8	0.314 / 0.186 / 99.6	0.549 / 0.379 / 93.6
	✗	-	✓	0.511 / 0.343 / 98.9	0.344 / 0.208 / 99.6	0.576 / 0.404 / 94.2
	✓	Recurrent (RNN)	✗	0.523 / 0.354 / 98.8	0.333 / 0.200 / 99.6	0.556 / 0.386 / 93.5
	✓	Recurrent (RNN)	✓	0.525 / 0.356 / 98.9	0.377 / 0.232 / 99.6	0.572 / 0.401 / 93.8
	✓	Recurrent (LSTM)	✗	0.549 / 0.378 / 98.9	0.351 / 0.213 / 99.7	0.571 / 0.400 / 94.1
	✓	Recurrent (LSTM)	✓	0.547 / 0.377 / 99.0	0.402 / 0.252 / 99.6	0.588 / 0.417 / 94.4
	✓	Self-attention	✗	0.537 / 0.367 / 98.8	0.397 / 0.248 / 99.7	0.593 / 0.422 / 94.3
	✓	Self-attention	✓	0.551 / 0.381 / 99.0	0.414 / 0.261 / 99.7	0.596 / 0.424 / 94.3
WUSU	✗	-	✗	0.342 / 0.208 / 96.8	0.364 / 0.223 / 97.9	0.583 / 0.412 / 86.2
	✗	-	✓	0.392 / 0.245 / 96.6	0.339 / 0.205 / 97.6	0.649 / 0.480 / 88.4
	✓	Recurrent (RNN)	✗	0.423 / 0.268 / 96.7	0.391 / 0.243 / 98.2	0.613 / 0.443 / 87.1
	✓	Recurrent (RNN)	✓	0.440 / 0.282 / 96.9	0.390 / 0.243 / 98.2	0.658 / 0.491 / 88.4
	✓	Recurrent (LSTM)	✗	0.426 / 0.271 / 96.8	0.394 / 0.245 / 98.3	0.615 / 0.444 / 87.3
	✓	Recurrent (LSTM)	✓	0.441 / 0.283 / 96.9	0.387 / 0.240 / 98.2	0.658 / 0.490 / 88.5
	✓	Self-attention	✗	0.420 / 0.266 / 96.6	0.391 / 0.243 / 98.3	0.660 / 0.493 / 88.8
	✓	Self-attention	✓	0.440 / 0.282 / 97.0	0.389 / 0.242 / 98.3	0.663 / 0.496 / 88.9
TSCD	✗	-	✗	0.278 / 0.163 / 89.7	0.295 / 0.173 / 93.7	-
	✓	Recurrent (RNN)	✗	0.445 / 0.293 / 91.8	0.497 / 0.334 / 95.5	-
	✓	Recurrent (LSTM)	✗	0.561 / 0.391 / 93.8	0.565 / 0.394 / 96.6	-
	✓	Self-attention	✗	0.543 / 0.373 / 93.7	0.573 / 0.402 / 96.6	-

$F1$ score and IoU, respectively, compared to the degenerate setting on the SN7 dataset. On the WUSU dataset, the corresponding performance improvements are 57.5% and 73.6% for the $F1$ score and IoU, respectively (cyclic loss scenario). Table VII also shows that introducing change information beyond adjacent edges results in further change detection performance gains, albeit to a much lesser extent. For example, compared to only using adjacent change information, dense information improved the $F1$ score and IoU values by 2.0% and 3.0%, respectively. In the context of the WUSU dataset, additional change information (i.e., adjacent versus cyclic) improved the $F1$ score and IoU values by 1.6% and 2.2%, respectively. Therefore, the ablation study demonstrates that the proposed MTI module effectively integrates the outputs

TABLE VII

ABLATION RESULTS FOR THE MTI MODULE ON THE SN7 AND WUSU DATASETS. THE EDGE SETTINGS CYCLIC AND DENSE ARE EQUIVALENT FOR THE WUSU DATASET SINCE ITS TIME SERIES CONSISTS OF THREE IMAGES. THE BEST AND SECOND-BEST PERFORMANCES ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY

Dataset	MTI module	Edge setting	Change detection		Segmentation
			Bi-temporal (F1 / IoU / OA)	Continuous (F1 / IoU / OA)	(F1 / IoU / OA)
SN7	✗	-	0.537 / 0.367 / 98.8	0.397 / 0.248 / 99.7	0.593 / 0.422 / 94.3
	✓	degenerate	0.373 / 0.229 / 97.8	0.173 / 0.095 / 98.8	0.593 / 0.422 / 94.3
	✓	adjacent	0.540 / 0.370 / 99.0	0.410 / 0.258 / 99.7	0.595 / 0.424 / 94.4
	✓	cyclic	0.547 / 0.377 / 99.0	0.412 / 0.260 / 99.7	0.596 / 0.424 / 94.3
	✓	dense	0.551 / 0.381 / 99.0	0.414 / 0.261 / 99.7	0.596 / 0.424 / 94.3
WUSU	✗	-	0.420 / 0.266 / 96.6	0.391 / 0.243 / 98.3	0.660 / 0.493 / 88.8
	✓	degenerate	0.275 / 0.159 / 93.8	0.203 / 0.113 / 95.5	0.660 / 0.493 / 88.8
	✓	adjacent	0.433 / 0.276 / 97.0	0.389 / 0.242 / 98.3	0.663 / 0.497 / 88.9
	✓	cyclic	0.440 / 0.282 / 97.0	0.389 / 0.242 / 98.3	0.663 / 0.496 / 88.9

of the segmentation and change detection tasks at inference time.

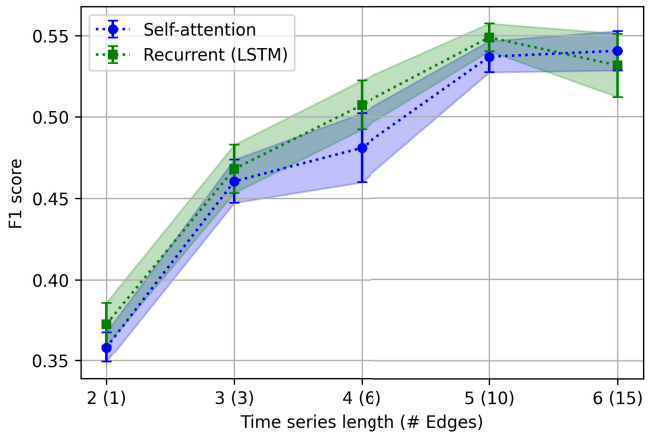
4) Time Series Length: To investigate the effect of SITS length on performance, we tested the proposed network with different settings for T on SN7. In addition, we compared self-attention and LSTM for sequence modeling in the TFR module. The maximum number of edges (i.e., dense edge setting) was used across all T settings in the change loss [see (6)]. The segmentation loss was enabled, whereas the MTI module was disabled. It should be noted that for lengths $T = 3$ and $T = 2$, the edge settings dense are equivalent to cyclic and adjacent, respectively. Fig. 10 shows the results of this experiment for bi-temporal change detection [see Fig. 10(a)], continuous change detection [see Fig. 10(b)], and segmentation [see Fig. 10(c)]. The bi-temporal results show that adding intermediate images improves change detection performance. However, performance saturates at $T = 5$ and even decreases for the recurrent sequence model at $T = 6$. Segmentation performance generally also increases with time series length until $T = 5$. In contrast, the continuous change detection task increases in difficulty with time series length, since the temporal gap between adjacent image pairs in the time series decreases. Consequently, continuous change detection performance tends to decrease at longer time series lengths. Regarding the sequence model comparison, our network achieved better bi-temporal change detection performance using the recurrent model in the TFR module, except for the longest time series length ($T = 6$). However, performance differences are below 5.5% across all-time series lengths. On the other hand, self-attention outperformed the recurrent model for the continuous change detection and segmentation tasks. The largest performance gains in both cases were observed for the longest time series lengths (i.e., $T = 5$ and $T = 6$).

VI. DISCUSSION

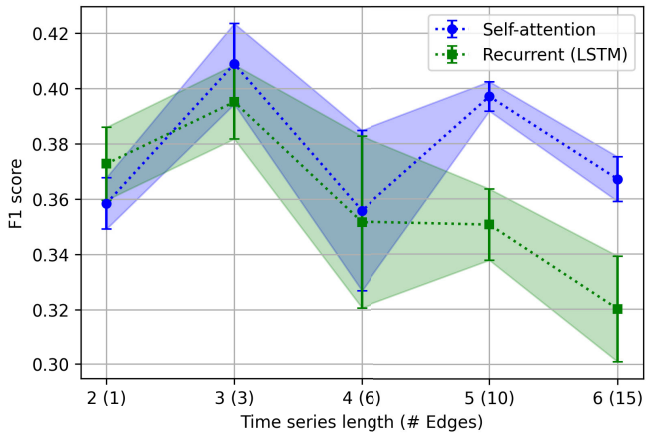
Our results highlight the challenging nature of continuously detecting urban changes from high-resolution SITS. Indeed, most change detection studies focus on bi-temporal urban change detection from VHR imagery. The suite of methods considered SOTA for bi-temporal change detection typically employ the self-attention mechanism popularized by the transformer architecture to capture long-range contextual information in VHR imagery [17], [18], [19], [20]. On popular benchmark datasets such as LEVIR-CD [18] and WHU-CD [42], these SOTA methods have achieved remarkable results with $F1$ scores exceeding 0.9. Although transformer-based methods generally outperformed their bi-temporal ConvNet-based counterparts in our experiments, they rarely achieved $F1$ scores higher than 0.4 on the continuous urban change detection task (see Tables II–IV). Therefore, our results indicate that bi-temporal change detection methods generally face limitations for continuous urban change detection.

In comparison to the SOTA methods for bi-temporal change detection, our method leverages the self-attention mechanism to model multitemporal information in SITS. We showed that the proposed TFR module contributes to the network's representation learning capability, resulting in improved continuous change detection performance (see Table VI). We also compared our method against multitemporal methods using self-attention or recurrent models for temporal modeling. However, these methods collapse the temporal dimension of the SITS, limiting them to the detection between the first and last image. Although multitemporal methods generally outperformed bi-temporal methods on this task, they fell short of the proposed method (see Tables II–IV). Overall, we deem self-attention an effective mechanism for the temporal modeling of SITS. Despite that, our ablation results show that recurrent sequence models such as LSTMs could be considered as an alternative, especially at shorter time series lengths (see Table VI).

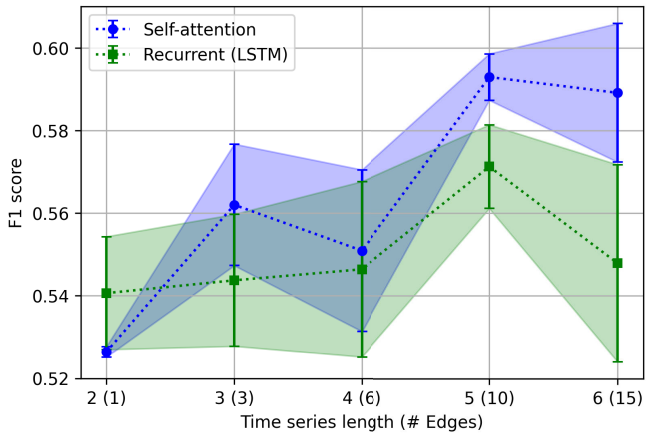
Our work also highlights the need for effective integration approaches in multitask learning schemes. Specifically, although multitask learning is commonly applied for change detection [28], [36], existing multitask studies do not address the integration of the semantic segmentation and change detection outputs. To fill this research gap, we proposed the MTI module that represents segmentation and change predictions using Markov networks to find the optimal built-up area state for each timestamp in a pixel time series. Our results demonstrate that the proposed integration approach improves



(a)



(b)



(c)

Fig. 10. Effect of time series length and sequence model in the TFR module on network performance (without MTI module) for the SN7 dataset. Network performance is evaluated in terms of (a) bi-temporal change detection, (b) continuous change detection, and (c) segmentation. The maximum number of edges was used for each time series length in the change loss. Values represent the mean ±1 standard deviation of 5 runs.

both tasks, namely, the change detection and building segmentation task (see Table VII). Furthermore, we demonstrate that the proposed approach benefits from integrating dense change information, obtained from predicting changes between all possible combinations of satellite image pairs in a time series, compared to using only adjacent change information (see Table VII).

Although we demonstrated the effectiveness of our method across three datasets, performances vary significantly across them. Overall, the lowest change detection performances were obtained for the WUSU dataset. The visualizations of the Gaofen-2 SITS highlight several challenging aspects for change detection in this dataset (see Figs. 8 and 9). In particular, the images were acquired under different atmospheric conditions, and they have large off-nadir angles and contain shadows. In comparison, these artifacts are not apparent in the SN7 images (see Figs. 6 and 7), reducing the complexity of the change detection task. However, building segmentation performance on the SN7 dataset is lower than on the WUSU dataset. We attribute this to the fact that individual buildings are more difficult to distinguish in PlanetScope imagery due to its lower spatial resolution (see Table I). The TSCD dataset stands out for its strong continuous urban change detection performances, while performances on the bi-temporal task are lower, especially for bi-temporal change detection methods. Here, it should be considered that the dataset only provides labels between adjacent images. Therefore, we had to derive change labels for nonadjacent images (see Section IV-A), which could affect the reference data quality.

Despite the improvements our method achieves over existing methods, we also identified several limitations related to our work. First of all, our integration approach relies on meaningful potentials extracted from the multitask network outputs. However, the outputs of deep networks may not be well-calibrated [61]. Furthermore, we assumed that our networks do not encounter out-of-distribution data during deployment due to the within-scene splits. In practice, however, urban mapping and change detection methods may encounter domain shifts when deployed to unseen geographic areas [39], [49]. Therefore, future work will test the susceptibility of our MTI approach to out-of-distribution data. For example, the effectiveness of the MTI module could be improved by explicitly calibrating the segmentation and change outputs of the model, using calibration techniques such as temperature scaling [61].

Another limitation of the proposed method is that it requires continuous building labels for training. Most popular urban change detection datasets are bi-temporal and feature VHR imagery. On the other hand, few urban change detection datasets featuring SITS and corresponding building labels for each image are available. Therefore, weakly supervised methods, using partial annotation or less accurate labeling, should be investigated for continuous urban change detection from SITS (e.g., [30]).

VII. Conclusion

This study introduces a continuous urban change detection framework for optical SITS. The proposed method incorporates a transformer-based module to temporally refine feature representations extracted from image time series using a shared ConvNet. Unlike existing temporal modules in multitemporal change detection methods, our module preserves the temporal dimension, enabling the detection of continuous

changes. In addition, we propose a novel MTI approach based on pairwise Markov networks, effectively combining building segmentation and dense urban change information. We evaluated our method on three SITS change detection datasets: SN7, the WUSU dataset, and the TSCD dataset. The proposed method outperformed existing bi-temporal and multitemporal change detection methods and segmentation methods. In particular, our findings show the limitations of bi-temporal methods in continuous change detection, as they cannot fully exploit multitemporal information in SITS. While multitemporal change detection methods overcome this limitation, they remain constrained to detecting changes between the first and last images in SITS. Our ablation study further demonstrates the effectiveness of the TFR module in modeling multitemporal information and the benefits of incorporating dense change information during training. Moreover, it confirms that the MTI module successfully integrates segmentation and change outputs, leading to improved accuracy across both tasks. In summary, this research underscores the potential of high-resolution SITS for continuous urban change detection. Future work will explore weakly supervised and self-supervised change detection methods for SITS to reduce dependence on annotations.

REFERENCES

[1] X. Liu et al., "High-spatiotemporal-resolution mapping of global urban change from 1985 to 2015," *Nature Sustainability*, vol. 3, no. 7, pp. 564–570, May 2020.

[2] J. Gao and B. C. O'Neill, "Mapping global urban land for the 21st century with data-driven simulations and shared socioeconomic pathways," *Nature Commun.*, vol. 11, no. 1, p. 2302, May 2020.

[3] M. P. Johnson, "Environmental impacts of urban sprawl: A survey of the literature and proposed research agenda," *Environ. Planning A, Economy Space*, vol. 33, no. 4, pp. 717–735, Apr. 2001.

[4] S. A. Sarkodie, P. A. Owusu, and T. Leirvik, "Global effect of urban sprawl, industrialization, trade and economic development on carbon dioxide emissions," *Environ. Res. Lett.*, vol. 15, no. 3, Mar. 2020, Art. no. 034049.

[5] S. Arshad, S. R. Ahmad, S. Abbas, A. Asharf, N. A. Siddiqui, and Z. U. Islam, "Quantifying the contribution of diminishing green spaces and urban sprawl to urban heat island effect in a rapidly urbanizing metropolitan city of Pakistan," *Land Use Policy*, vol. 113, Feb. 2022, Art. no. 105874.

[6] Y. Ban and O. Yousif, "Change detection techniques: A review," in *Multitemporal Remote Sensing*. Cham, Switzerland: Springer, Jan. 2016, pp. 19–43, doi: 10.1007/978-3-319-47037-5_2.

[7] D. Lu, P. Mausel, E. Brondízio, and E. Moran, "Change detection techniques," *Int. J. Remote Sens.*, vol. 25, no. 12, pp. 2365–2401, 2004.

[8] X. X. Zhu et al., "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[9] H. Jiang et al., "A survey on deep learning-based change detection from high-resolution remote sensing images," *Remote Sens.*, vol. 14, no. 7, p. 1552, Mar. 2022.

[10] Z. Lv, Z. Lei, L. Xie, N. Falco, C. Shi, and Z. You, "Novel distribution distance based on inconsistent adaptive region for change detection using hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 4404912.

[11] Z. Lv, T. Yang, T. Lei, W. Zhou, Z. Zhang, and Z. You, "Spatial–Spectral similarity based on adaptive region for landslide inventory mapping with remote-sensed images," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 4405111.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351. Cham, Switzerland: Springer, 2015, pp. 234–241.

[13] R. C. Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, "Urban change detection for multispectral Earth observation using convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2018, pp. 2115–2118.

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

[15] 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, 2021.

[16] K. S. Basavaraju, N. Sravya, S. Lal, J. Nalini, C. S. Reddy, and F. Dell'Acqua, "UCDNet: A deep learning model for urban change detection from bi-temporal multispectral Sentinel-2 satellite images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5408110.

[17] W. Liu, Y. Lin, W. Liu, Y. Yu, and J. Li, "An attention-based multiscale transformer network for remote sensing image change detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 202, pp. 599–609, Aug. 2023.

[18] 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, p. 1662, 2020.

[19] H. Chen, Z. Qi, and Z. Shi, "Remote sensing image change detection with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2021.

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

[21] V. Marsocci, V. Coletta, R. Ravanelli, S. Scardapane, and M. Crespi, "Inferring 3D change detection from bitemporal optical images," *ISPRS J. Photogramm. Remote Sens.*, vol. 196, pp. 325–339, Feb. 2023.

[22] M. Noman et al., "Remote sensing change detection with transformers trained from scratch," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 4704214.

[23] D. P. Roy, H. Huang, R. Houborg, and V. S. Martins, "A global analysis of the temporal availability of PlanetScope high spatial resolution multi-spectral imagery," *Remote Sens. Environ.*, vol. 264, Oct. 2021, Art. no. 112586.

[24] Z. Zhu, S. Qiu, and S. Ye, "Remote sensing of land change: A multifaceted perspective," *Remote Sens. Environ.*, vol. 282, Dec. 2022, Art. no. 113266.

[25] A. Shafique, G. Cao, Z. Khan, M. Asad, and M. Aslam, "Deep learning-based change detection in remote sensing images: A review," *Remote Sens.*, vol. 14, no. 4, p. 871, Feb. 2022.

[26] X. Liu et al., "High-resolution multi-temporal mapping of global urban land using Landsat images based on the Google Earth engine platform," *Remote Sens. Environ.*, vol. 209, pp. 227–239, May 2018.

[27] M. Papadomanolaki, S. Verma, M. Vakalopoulou, S. Gupta, and K. Karantzalos, "Detecting urban changes with recurrent neural networks from multitemporal Sentinel-2 data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 214–217.

[28] M. Papadomanolaki, M. Vakalopoulou, and K. Karantzalos, "A deep multitask learning framework coupling semantic segmentation and fully convolutional LSTM networks for urban change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7651–7668, Sep. 2021.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] K. Meshkini, F. Bovolo, and L. Bruzzone, "Multiannual change detection using a weakly supervised 3-D CNN in HR SITS," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, 2024, Art. no. 5001405.

[31] S. Saha, F. Bovolo, and L. Bruzzone, "Change detection in image time-series using unsupervised LSTM," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[32] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.

[33] V. S. Fare Garnot and L. Landrieu, "Panoptic segmentation of satellite image time series with convolutional temporal attention networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4872–4881.

[34] M. Tarasiou, E. Chavez, and S. Zafeiriou, "ViTs for SITS: Vision transformers for satellite image time series," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 10418–10428.

[35] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

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

[37] S. Hafner, Y. Ban, and A. Nascetti, "Urban change detection using a dual-task Siamese network and semi-supervised learning," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Kuala Lumpur, Malaysia, Jul. 2022, pp. 1071–1074.

[38] Q. Shu, J. Pan, Z. Zhang, and M. Wang, "MTCNet: Multitask consistency network with single temporal supervision for semi-supervised building change detection," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 115, Dec. 2022, Art. no. 103110.

[39] S. Hafner, Y. Ban, and A. Nascetti, "Semi-supervised urban change detection using multi-modal Sentinel-1 SAR and Sentinel-2 MSI data," *Remote Sens.*, vol. 15, no. 21, p. 5135, Oct. 2023.

[40] Z. Zhou, M. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-net architecture for medical image segmentation," in *Proc. Int. Workshop Deep Learn. Med. Image Anal.*, vol. 11045, 2018, pp. 3–11.

[41] L. Wang, J. Zhang, Q. Guo, and D. Chen, "IFTSDNet: An interact-feature transformer network with spatial detail enhancement module for change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[42] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 574–586, Jan. 2018.

[43] H. He, J. Yan, D. Liang, Z. Sun, J. Li, and L. Wang, "Time-series land cover change detection using deep learning-based temporal semantic segmentation," *Remote Sens. Environ.*, vol. 305, May 2024, Art. no. 114101.

[44] Z. Cai et al., "A cost-effective and robust mapping method for diverse crop types using weakly supervised semantic segmentation with sparse point samples," *ISPRS J. Photogramm. Remote Sens.*, vol. 218, pp. 260–276, Dec. 2024.

[45] V. S. F. Garnot and L. Landrieu, "Lightweight temporal self-attention for classifying satellite images time series," in *Proc. Int. Workshop Adv. Anal. Learn. Temporal Data*. Cham, Switzerland: Springer, 2020, pp. 171–181.

[46] D. Duque-Arias et al., "On power Jaccard losses for semantic segmentation," in *Proc. 16th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2021, pp. 1–8.

[47] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[48] A. Van Etten, D. Hogan, J. Martinez-Manso, J. Shermeyer, N. Weir, and R. Lewis, "The multi-temporal urban development SpaceNet dataset," 2021, *arXiv:2102.04420*.

[49] S. Hafner, Y. Ban, and A. Nascetti, "Unsupervised domain adaptation for global urban extraction using Sentinel-1 SAR and Sentinel-2 MSI data," *Remote Sens. Environ.*, vol. 280, Oct. 2022, Art. no. 113192.

[50] M. T. Razzak, G. Mateo-García, G. Lecuyer, L. Gómez-Chova, Y. Gal, and F. Kalaitzis, "Multi-spectral multi-image super-resolution of Sentinel-2 with radiometric consistency losses and its effect on building delineation," *ISPRS J. Photogramm. Remote Sens.*, vol. 195, pp. 1–13, Jan. 2023.

[51] N. Metzger, M. Ö. Türkoglu, R. C. Daudt, J. D. Wegner, and K. Schindler, "Urban change forecasting from satellite images," *PFG-J. Photogramm., Remote Sens. Geoinformation Sci.*, vol. 91, no. 6, pp. 443–452, Dec. 2023.

[52] S. Shi et al., "Multi-temporal urban semantic understanding based on GF-2 remote sensing imagery: From tri-temporal datasets to multi-task mapping," *Int. J. Digit. Earth*, vol. 16, no. 1, pp. 3321–3347, Oct. 2023.

[53] Y. Zhao, H.-C. Li, S. Lei, N. Liu, J. Pan, and T. Celik, "COUD: Continual urbanization detector for time series building change detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 19601–19615, 2024.

[54] S. Liu, H. Su, G. Cao, S. Wang, and Q. Guan, "Learning from data: A post classification method for annual land cover analysis in urban areas," *ISPRS J. Photogramm. Remote Sens.*, vol. 154, pp. 202–215, Aug. 2019.

[55] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*.

[56] A. Rogozhnikov, "Einops: Clear and reliable tensor manipulations with einstein-like notation," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=oapKSVM2bcj

[57] A. Ankan and A. Panda, "Pgmpy: Probabilistic graphical models using Python," in *Proc. Python Sci. Conf.*, 2015, pp. 6–11, doi: 10.25080/majora-7b98e3ed-001.

[58] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in Adam," in *Proc. ICLR*, 2018, pp. 1–14.

[59] F. Bovolo and L. Bruzzone, "The time variable in data fusion: A change detection perspective," *IEEE Geosci. Remote Sens. Mag.*, vol. 3, no. 3, pp. 8–26, Sep. 2015.

[60] J. L. Elman, "Finding structure in time," *Cogn. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.

[61] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Intl. Conf. Mach. Learn.*, Sydney, NSW, Australia, 2017, pp. 1321–1330.