A CNN-Transformer Combined Remote Sensing Imagery Spatiotemporal Fusion Model

Mingyu Jiang and Hua Shao

***Abstract*—Remote sensing images (RSIs) spatiotemporal fusion (STF) make a significant contribution to acquisition of RSIs sequence with simultaneously high temporal and spatial resolution, which broadens its application fields. However, the existing RSIs STF methods lack effective strategies for extracting global information and fusion features between different images. Conversely, the existing state-of-the-art (SOTA) methods generally require more than two RSIs from different satellites as reference, which increases the difficulty of data collection and limits the application in practice. To address these problems, this article proposed an end-to-end CNN and Transformer combined RSIs STF model (CTSTFM) based on two reference RSIs. Specifically, the proposed CTSTFM consists of three basic modules: multikernel convolutional transformer encoder (MKCE), cross fusion module (CFM), and convolutional-based compression decoder (CCD). The MKCE combines multikernel channel attention block and multikernel spatial attention block to extract shadow features and long-term interdependencies in reference RSIs. The CFM uses the unique cross exchange transformer block and combine fusion transformer block to enhance the feature fusion results. Due to the powerful encoder and fusion module, in the CCD part we only use a simple design convolution module to save the consumption of computational resources. Experiments on two well-known open access datasets show that CTSTFM achieves competitive results in both qualitative and quantitative comparisons compared to the SOTA methods. Meanwhile, we conduct experiment to analyze the image Tessellation effects and its solution. The effectiveness of the proposed module will be demonstrated through ablation experiments.**

***Index Terms*—Convolutional neural network (CNN), multisource satellite data, remote sensing, spatiotemporal fusion (STF), transformer.**

I. Introduction

BENEFIT from the high efficiency, low cost, and wide coverage advantages of remote sensing images (RSIs), RSIs are widely used in many fields (e.g., agriculture [1], [2], environment monitoring [3], [4], target detection [5], [6], and large-scale urban 3-D modeling [7], [8]) [9], [10]. However, limited by the satellite technology and manufacturing budget, it is difficult for existing satellite RSIs to have both high spatial (HS) resolution and high temporal (HT) resolution [11]. It is not difficult to consider that combining HS and HT, RSIs will be a great solution, and therefore, RSIs spatiotemporal fusion techniques have received a lot of attention since 1990s [12], [13]. Spatiotemporal fusion of RSIs can be summarized as a technique to generating sequences of RSIs with both HT and spatial resolution using multiple pairs or pair of HS resolution but lowtemporal resolution RSIs and low spatial resolution but HT resolution RSIs as references, all of these reference RSIs have similar spectral response functions. For example, a HT resolution RSIs sequences (spatial resolution of 30 m and temporal resolution of 1 day) can be generated by fusing Landsat-8 (with HS resolution of 30 m but low temporal resolution of 16 days, hereafter referred to fine imagery) with moderate-resolution imaging spectroradiometer (MODIS, with low spatial resolution of 500 m but HT resolution of 1 day, hereafter referred to coarse imagery) imagery [14]. Specifically, remote sensing imagery spatiotemporal fusion methods can be classified into following four categories:

1) Unmixing-based methods;
2) Weight function-based methods;
3) Learning-based methods;
4) Hybrid-based methods [15].

The unmixing-based spatiotemporal fusion methods considers that the pixels in the coarse imagery are compressed and combined from the corresponding pixels in the fine imagery, the purpose is to decode the pixels in the coarse imagery to the pixels in the fine imagery. The multisensor multiresolution technique (MMT) method [16] is the first unmixing-based spatiotemporal fusion (STF) method for RSIs. Based on the quantity ratio between different categories in the fine image, MMT uses the moving window mode to unmixing pixels in the coarse imagery to reconstruct the fine imagery of the target date. However, the assumption of MMT that the land cover information does not change with time is not reasonable, and therefore, the predicted fine images by MMT are far from the real situation. To this end, Lopez et al. [17] processed both fine imagery and coarse imagery pixels as mixed pixels. And Shi et al. [18] proposed the reliable and adaptive spatiotemporal data fusion (RASDF) method, which collaboratively uses adaptive global and local unmixing models to enhance the model's ability to mine spatial information changes. In order to solve the block effect that commonly exists in the unmixing-based spatiotemporal fusion methods, Wang et al. [19] proposed blocks-removed spatial unmixing (SU-BR). The SU-BR remove the block effect by using the spatial continuity construction constraints, in addition, the SU-BR is applicable to the existing RSIs unmixing-based STF methods. Liu et al. [20] used spectral mixing analysis and

Manuscript received 20 May 2024; revised 30 June 2024 and 22 July 2024; accepted 26 July 2024. Date of publication 30 July 2024; date of current version 15 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 42271420, and Grant 42301420, and in part by Postgraduate Research & Practice Innovation Program of Jiangsu Province. *(Corresponding author: Hua Shao.)*

The authors are with the School of Geomatics Science and Technology, Nanjing Tech University, Nanjing 211816, China (e-mail: 202261223014@njtech.edu.cn; shaohua@njtech.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3435739

optimized moving window to estimate the end-element spectra and their corresponding abundance maps to reconstruct the fused images, effectively eliminating the block effect.

The weight function-based STF methods assign different weight coefficients to the input reference imagery and combines them to obtain the fine imagery of the target date. The spatial and temporal adaptive reflectance fusion model (STARFM) [21] is the first weight function-based spatiotemporal fusion method. STARFM calculates the weight function based on spectral and structural information between reference images, and assigns higher weights to the purer rough pixels. Inspired by STARFM, numerous weight function-based STF methods have emerged over the past decade. The improvement of these methods mainly focuses on the weight function selection strategy [22], [23], deals with sensor differences [24], [25], and additional technologies [26].

Both the unmixing-based and weight function-based STF methods can be summarized as traditional methods, and have already achieved great results. However, the effectiveness of these methods relies heavily on the artificially designed relationships and assumptions, these choices are empirical and not generalized, which means that they have to be greatly modified to suit different problems [27]. In addition, these developed traditional algorithms inevitably meet the performance bottleneck due to the lack of the ability to comprehensively analyze and interpret strongly heterogeneous data [28]. Learning-based methods are able to adaptively extract more important features from input information (e.g., images) than traditional methods, and therefore, learning-based methods can achieve better results than traditional methods. Shallow learning-based RSIs spatiotemporal fusion methods, according to the underlying technique used, can be classified as Bayesian learning-based [29], [30], [31], sparse representation learning-based [32], [33], [34], and paired dictionary-based [35] STF methods, these methods can achieve better performance than traditional methods.

The deep learning-based RSIs STF methods mostly build up base on convolutional neural network (CNN), due to its strong nonlinear representation ability [36]. Inspired by SRCNN [37], Song et al. [38] proposed the spatial temporal fusion convolutional network (STFCNN), the first CNN-based RSIs STF method. STFCNN uses three convolution layers to learn the nonlinear mapping between coarse imagery and fine imagery, and obtains the fine imagery at target date by weighted summation. After that, numerous CNN-based STF models appeared, e.g., deep convolutional spatiotemporal fusion network (DCSTFN) [39], a two-stream CNN for spatiotemporal image fusion (STFNet) [40], bias-driven spatiotemporal fusion model [41], multicooperative deep CNN [42], progressive spatiotemporal image fusion with deep neural networks and spatial, sensor, and temporal spatiotemporal fusion [43]. In recent years, generative adversarial networks (GANs) [44] have gained great advantages in many fields, and it have also been introduced into RSIs STF. Zhang et al. [45] proposed an end-to-end STF model of RSIs based on generative adversarial networks (STFGAN), which processed coarse imagery and fine imagery with super-resolution and high-frequency features extraction, respectively, and then combined the features of the two to generate fine imagery at the target date. Tan et al. [13] proposed the GAN-based spatiotemporal fusion model (GAN-STFM). GAN-STFM uses only one coarse imagery at target date and one fine imagery at any date as reference imagery. Ma et al. [43] separated the spatiotemporal fusion task into three subtasks of temporal, spatial, and sensor disparity and modeled them separately. Song et al. [14] devised an encoder-decoder structure for GAN frames to learn image fusion multilevel features (MLFs). However, GANs suffer from their inherent problems such as mode collapse [46] and optimization instability [47], which makes the neural network difficult to train and sometimes the results generated by GAN may even collapse. Yang et al. [48] design a cross-stage adaptive fusion module, which adaptively integrate the features of different scales according to the temporal and spatial features. STF-Trans [49] is an encoder-decoder transformer architecture based STF method, which aims to fuse coarse image and HS resolution Google Earth image to produce both HS and temporal fine image at target date.

The hybrid-based RSIs STF method combines the advantages of the three methods mentioned above with each other to produce more accurate fusion results. Zhu et al. [50] proposed the flexible spatiotemporal data fusion (FSDAF) method, which is based on both unmixing-based and weight function-based methods. FSDAF predicts the spectral and spatial information change characteristics of surface, and uses the weighted sum of the spectral and spatial change characteristics to obtain the fine image at target date. Cui et al. [51] proposed an STF method combining linear pixel decomposition and STARFM, the coarse imagery were downscaled by a linear spectral mixture model and then used as input of STARFM for prediction. In order to greatly improve the efficiency of the algorithm, Hou et al. [52] modified the spectral unmixing in FSDAF to adaptively select spectral unmixing method. Guo et al. [53] proposed a retrieving land-cover changes and preserving spatial details performance improved FSDAF (FSDAF 2.0), FSDAF 2.0 is an improved FSDAF method that combines change detection technology and an optimized model for changed-type areas. Wang et al. [54] proposed an unmixing-based and weight function-based STF method, namely, cross-stage adaptive fusion module (CSAFM), used for evaluation of remotely sensed evapotranspiration in an irrigated agricultural area with a complex planting structure. Li et al. [55] proposed a multistream fusion network (MSNet), which uses a CNN and Transformer mixed structure to extract features, and fusion these features by average weighting method. Chen et al. [56] combined the feature extraction advantages of Swin-Transformer [57] and the unmixing theories, significant improve the quality of fusion results.

The existing STF methods designed for multisource satellite data have achieved great results, but still have some shortcomings. First of all, the premise assumptions and prior knowledge of traditional methods are not enough to reflect the real situation, and the performance has reached the bottleneck. Second, the existing learning-based methods are mostly based on CNN, therefore, these CNN-based models are limited by the size of the convolution kernel, lack of global information usage, it makes the model pay excessive attention to local information

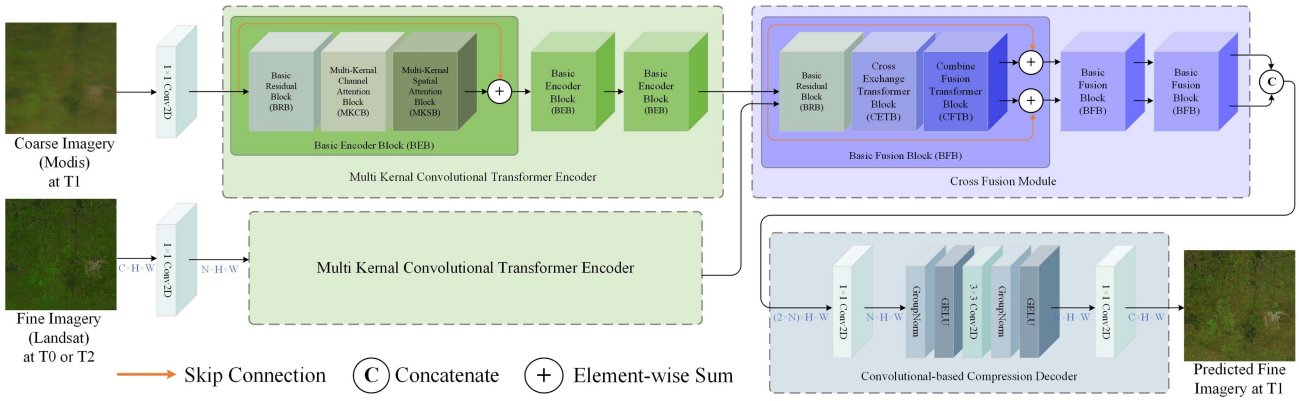

Fig. 1. Overall model structure of the proposed CTSTFM.

and neglect the connection of the whole image, which leads to a decrease in spatiotemporal fusion accuracy in the region where the spatiotemporal information changes drastically. Meanwhile, the existing methods mostly use weighted summation to deal with the information between multiple images and lack effective fusion strategies. In addition, the existing learning-based methods that achieve the best results require more than two RSIs as reference, which increases the difficulty of data collection in practical applications. To solve the problems mentioned above, we propose an end-to-end remote sensing imagery spatiotemporal fusion model based on a pair of reference images (two images), named CNN-Transformer combined spatiotemporal fusion model (CTSTFM). The main works of this article can be summarized as follows.

1) In this article, we propose a novel end-to-end spatiotemporal fusion model which use two RSIs (one coarse imagery at target date, one fine imagery at any date) as reference only, named CNN-Transformer combined spatial-temporal fusion model (CTSTFM).The network structure of CTSTFM is mainly composed of three parts: multikernel convolutional transformer encoder (MKCE), cross fusion module (CFM), and convolutional-based compression decoder (CCD).
2) In MKCE, multi kernel channel attention block (MKCB) and multikernel channel spatial block (MKSB) are used for feature extraction and correlation encoding between coarse imagery and fine imagery. By combining convolution modules of different convolution kernel sizes and attention mechanisms, MKCB and MKSB solve the problem that common convolution is limited by the receptive field and reduces the overspending computing resources problem causing by Transformer.
3) In CFM, the specially designed cross exchange transformer block (CETB) and combine fusion transformer block (CFTB) are used to fuse the coarse image's surface cover change information with the fine image's surface texture information.
4) Experiments on two open access Landsat-MODIS STF datasets, Coleambally Irrigation Area (CIA) dataset and Daxing (DX) dataset show that compared with the SOTA model, CTSTFM achieves better performance with only a few model parameters. In addition, a series of ablation experiments demonstrate the effectiveness of the proposed modules.

The rest of this article is organized as follows. In Section II, the proposed CTSTFM will be described in detail. Section III and Section IV present the experimental settings and comparative results. Finally, Section V concludes this article.

II. Methodology

A. Overall Structure

The overall model structure of CTSTFM is shown in Fig. 1, it can be divided into three parts—MKCE, CFM, and CCD. Specifically, the CTSTFM predict target date (at date T1) fine imagery by using a coarse imagery at date T1 and a fine imagery at date before date T1 (at date T0) or after date T1 (at date T2). Since the coarse imagery and the fine imagery provide surface cover change information and surface texture information for CTSTFM, respectively, the date of the fine imagery should be close to the date of coarse imagery. The inputs of CTSTFM first through one 2-D convolution layer with kernel size of 1×1 to expend channel dimension and extract shadow feature F_{C_0} and F_{F_0}.

$$F_{C_0} = f^{1\times1}_{\text{Conv2-D}}(I_{\text{Coarse}}) \tag{1}$$

$$F_{F_0} = f^{1\times1}_{\text{Conv2-D}}(I_{\text{Fine}}) \tag{2}$$

where $f^{1\times1}_{\text{Conv2-D}}$ means the 1×1 2-D convolution operation, I_{Coarse} and I_{Fine} are the inputs of CTSTFM, the subscripts C and F represent features from the coarse imagery and the fine imagery, respectively. After that, the shadow feature F_{C_0} and F_{F_0} are transmitted to the MKCE for further in-depth feature extraction and enhancement. In MKCE, the basic encoder block (BEB) is the basic unit and it also consists of three submodules: Basic residual block (BRB), MKCB, and MKSB. It is worth noting that the I_{Coarse} and I_{Fine} have separate encoders and do not share network parameters with each other.

$$F_{i\sim E} = f_{\text{BEB}_e}\left(f_{\text{BEB}_{E-1}}(\ldots(f_{\text{BEB}_1}(F_{i\sim0}))\ldots)\right)$$

$$e = 1, \ldots, E; \quad i = C, F \tag{3}$$

$$F_{\text{BEB}_e} = f_{\text{MKSB}_e}\left(f_{\text{MKCB}_e}(f_{\text{BRB}_e}(F_{\text{BEB}_{e-1}}))\right) + F_{\text{BEB}_{e-1}} \tag{4}$$

where $F_{i\sim E}$ and $F_{i\sim 0}$ represent the input and output of MKCE. The F_{BEB_e}, f_{MKSB_e}, f_{MKCB_e}, and f_{BRB_e} denote the eth BEB output and MKSB, MKCB, and BRB mapping operation. The CFM is designed as a network structure where information is exchanged between different feature maps, it consists of three submodules: BRB, CETB, and CFTB and the basic fusion block (BFB) is its basic unit.

$$F_{i\sim F} = f_{\text{BFB}_m}\left(f_{\text{BFB}_{m-1}}(\ldots(f_{\text{BFB}_1}(F_{i\sim E}))\ldots)\right) \quad i = C, F \tag{5}$$

$$F_{\text{BFB}_m} = f_{\text{CFTB}_m}\left(f_{\text{CETB}_m}\left(f_{\text{BRB}_m}(F_{\text{BFB}_{m-1}})\right)\right) + F_{\text{BFB}_{m-1}}, \quad m = 1, \ldots, M \tag{6}$$

where $F_{i\sim F}$ and $F_{i\sim E}$ represent the input and output of CFM. The F_{BFB_m}, f_{CETB_m}, f_{CFTB_m}, and f_{BRB_m} denote the mth BFB output and CETB, CFTB, and BRB mapping operation. Finally, the $F_{C\sim F}$ and $F_{F\sim F}$ are concatenated together at channel dimension and then sent to the CCD, which have a series of convolution layers, group normalization and GELU activation, for feature compression and achieve fine image at date T1.

B. Multi Kernel Channel Attention Block

A large number of studies spot that attention mechanism and Transformer have great effectiveness for global information modeling, however, it requires a lot of computing resources. At the same time, convolution neural networks are limited by the size of convolution kernel, which makes it difficult to use global information effectively. In addition, increasing the convolution kernel size will increase the demand for computing resources. To solve these problems, we designed a convolutional projection module with multiple convolution kernel sizes to model features at different sizes, and apply attention mechanisms at the channel dimension.

As shown in Fig. 2(c), the input feature maps of MKCB first modeled locally on feature information of different scales through three convolution layers with convolution kernel sizes of 3×3, 5×5, and 7×7, and then concatenate at channel dimension.

$$F_{\text{concated}} = \text{Concatenate}(f^{3\times 3}_{\text{Conv2-D}}(F) \; f^{5\times 5}_{\text{Conv2-D}}(F) \; f^{7\times 7}_{\text{Conv2-D}}(F)). \tag{7}$$

The concatenated feature maps F_{concated} contains feature information at different scales. The fusion of these different scales feature can weaken the problem that the convolution network is limited by the size of the receptive field, and strengthen the ability of the neural network to mine information of different scales, which is conducive to the evaluation of the importance of the subsequent channel attention mechanism for different layers of the feature maps. After that, F_{concated} go through average pooling, maximum pooling, convolution layers with kernel size of 1×1 and 3×3, GELU activation, and finally, Sigmoid operation output the importance coefficient of each feature layer.

$$F_{\text{Pool}} = f_{\text{AvgPool}}(F_{\text{concated}}) + f_{\text{MaxPool}}(F_{\text{concated}}) \tag{8}$$

$$F_{\text{weight}} = f_{\text{Sigmoid}}(f^{1\times 1}_{\text{Conv2-D}}(f_{\text{GELU}}(f^{3\times 3}_{\text{Conv2-D}}(F_{\text{Pool}})))) \tag{9}$$

$$F_{\text{MKCB}} = F_{\text{weight}} \times F \tag{10}$$

Where, f_{AvgPool} and f_{MaxPool} denote average pooling and maximum pooling, f_{Sigmoid} represents Sigmoid operation, F_{MKCB} means the output of MKCB and $\times$ denotes elementwise multiplication.

C. Multi Kernel Spatial Attention Block

MKCB enhanced the ability of channel attention mechanism to express features at different scales by using convolution layers with different convolution kernel sizes. Similar to MKCB, MKSB combines a convolution projection module with multiple convolution kernel sizes with a spatial attention mechanism. As shown in Fig. 2(d), the multiscale feature information extracted by the multikernel convolution projection module is added, and the added feature group is further screened and modeled by the BRB module. Finally, the importance of each pixel in the feature maps is evaluated by Softmax operation.

$$F_{\text{sumed}} = f^{3\times 3}_{\text{Conv2-D}}(F) + f^{5\times 5}_{\text{Conv2-D}}(F) + f^{7\times 7}_{\text{Conv2-D}}(F) \tag{11}$$

$$F_{\text{MKSB}} = f_{\text{Softmax}}(f_{\text{BRB}}(F_{\text{sumed}})) \times F \tag{12}$$

where f_{Softmax} means Softmax operation, F_{MKSB} represents output of MKSB mapping operation and $\times$ denotes elementwise multiplication.

D. Cross Exchange Transformer Block

As shown in Fig. 2(e), CETB can be regarded as a cross-attention module with two branches. The coarse feature maps and fine feature maps are projected into three subvectors Queen (Q), Key (K), and Value (V) after a group normalization operation and three convolution layers with kernel size of 1×1. The coarse and fine feature maps exchange each other's Q-vectors to achieve the effect of feature information exchange before the matrix multiplication of the attention mechanism. Overall, CETB is a two-branch structure. In the upper half of the branch, the coarse feature maps fuses the surface texture information in the fine feature maps, and in the lower half of the branch, the Fine feature maps fuses the surface cover change information in the coarse feature maps.

$$F_{i\sim\text{norm}} = f_{GN}(F_i), \quad i = C, F \tag{13}$$

$$Q_i, K_i, V_i = f^{1\times 1}_{Q\sim\text{Conv2-D}}(F_{i\sim\text{norm}}) \; f^{1\times 1}_{K\sim\text{Conv2-D}}(F_{i\sim\text{norm}}) \; f^{1\times 1}_{V\sim\text{Conv2-D}}(F_{i\sim\text{norm}}), \quad i = C, F \tag{14}$$

$$F_{i\sim\text{CETB}} = f_{\text{BRB}}\left(f^{1\times 1}_{\text{Conv2-D}}(f_{\text{Softmax}}\left(\frac{Q_i K_j^T}{\sqrt{d}}\right)) * V_i\right)$$

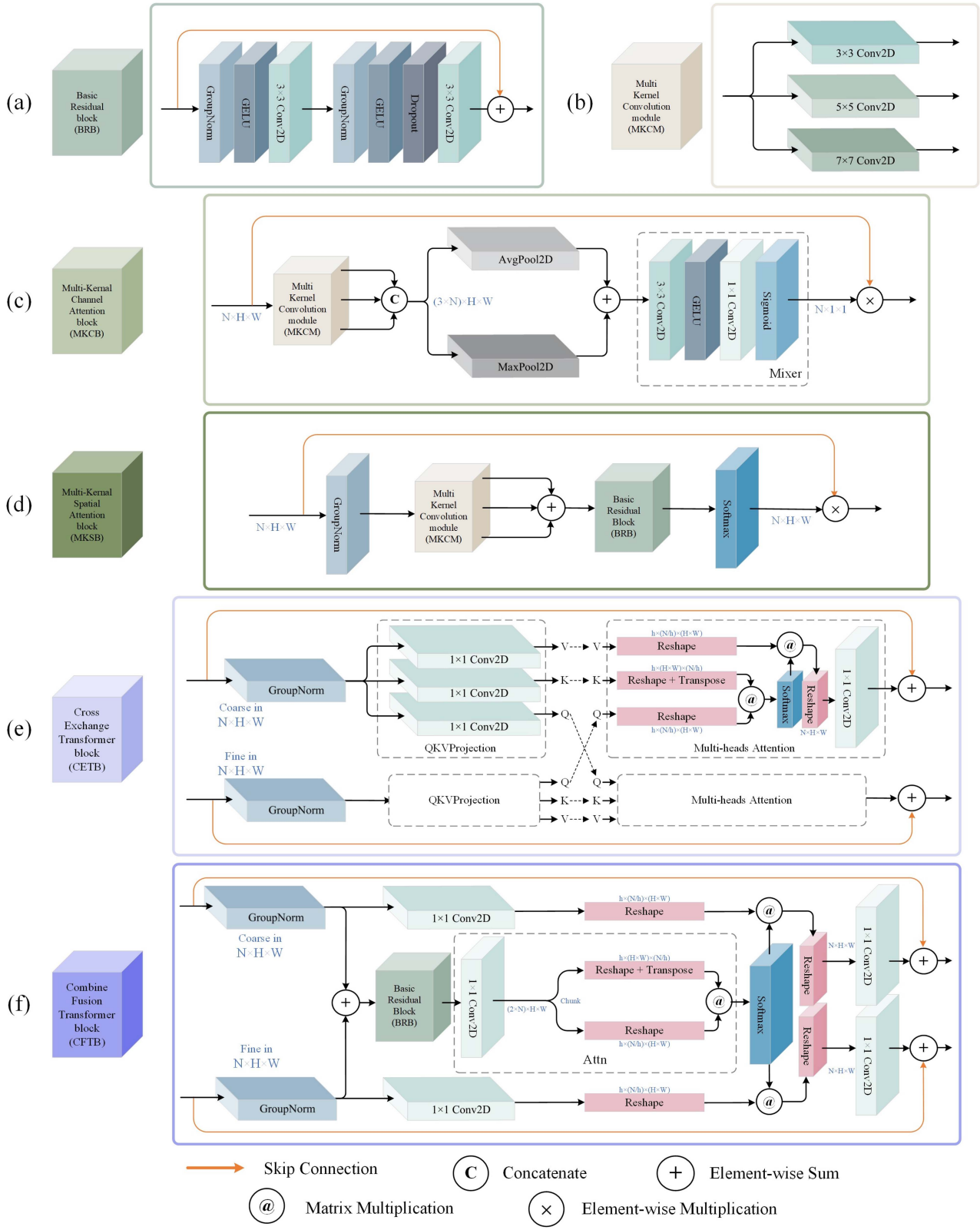

Fig. 2. Structure of the basic module in CTSTFM. (a) BRB. (b) MKCM. (c) MKCB. (d) MKSB. (e) CETB. (f) CFTB.

$$+ F_{i\sim\text{norm}}, \quad i = C, F;\ j = F, C \tag{15}$$

where $f^{1\times1}_{Q\sim\text{Conv2-D}}$, $f^{1\times1}_{K\sim\text{Conv2-D}}$, and $f^{1\times1}_{V\sim\text{Conv2-D}}$ are three convolution mapping layers for Q, K, and V, $F_{i\sim\text{CETB}}$ represents output of CETB operation for coarse or fine feature maps, and $*$ denotes matrix multiplication. For subscript i and j, if $i = C$ so $j = F$, else $j = C$.

E. Combine Fusion Transformer Block

To further enhance the information exchange and fusion between the coarse feature maps and fine feature maps, we also designed the CFTB. As shown in the Fig. 2(f), CFTB is a strong association module, and the input feature maps, F_C and F_F, are summed together after a layer of Group Normalization, and further information is enhanced by BRB. After that, the weight matrix is output by the self-attention mechanism and Softmax operation, which directs the model to further focus on the critical feature information.

$$F_{i\sim\text{norm}} = f_{\text{GN}}(F_i) \tag{16}$$

$$F_{\text{weight}} = f_{\text{softmax}}(\text{Attn}(f_{\text{BRB}}(F_{C\sim\text{norm}} + F_{F\sim\text{norm}}))) \tag{17}$$

$$F_{i\sim\text{Atten}} = f^{1\times1}_{\text{Conv2-D}}\big(\text{reshape}(F_{i\sim\text{norm}})\big) * F_{\text{weight}} \tag{18}$$

$$F_{i\sim\text{CFTB}} = f_{\text{BRB}}(f^{1\times1}_{\text{Conv2-D}}(\text{reshape}(F_{i\sim\text{Atten}})) + F_{i\sim\text{norm}}) \quad i = C, F \tag{19}$$

where Attn is the operation in the gray dashed box in the Fig. 2(f), which includes convolution, reshape, dimension transpose, and matrix multiplication operations.

F. Loss Function

Our assumption is that if the model has stronger feature extraction and mapping capabilities, then absolute errors, such as mean absolute error (MAE) and square mean root error (RMSE), can effectively guide the model parameters update, and the additional losses are added as auxiliary roles. Therefore, in this article, we only use the L1 loss as the total loss function for CTSTFM.

$$\text{Loss}_{\text{total}} = \frac{1}{N \times M}\sum_{i=1}^{N}\sum_{j=1}^{M}\left|I^{\text{target}}_{i,j} - I'_{i,j}\right| \tag{20}$$

Where, I^{target} represents the fine imagery at prediction date and $I\prime$ denotes the fine imagery predicted by method, and N and M are the width and height of the evaluate imagery.

III. Experiment

A. Dataset

In this article, two open access RSIs STF datasets are selected to verify the performance of CTSTFM and other comparison methods, including Coleambally Irrigation Area (CIA) Dataset[1] and Daxing (DX) dataset [58].

The CIA dataset contains 17 cloud-free pairs of Landsat (Landsat-7 Enhanced Thematic Mapper Plus)-Modis (MODIS Collection 5) RSIs. These paired RSIs were taken of a rice-based irrigation system located in southern New South Wales (34.0034E, 145.0675S), Australia, during the summer growing season from 2001 to 2002. The image size of Landsat is 1720×2040 and with a spatial resolution of 25 m, the spatial resolution of MODIS image is 500 m and the image size is consistent with Landsat image by interpolation algorithm. These 17 paired Landsat-Modis RSIs were divided into training set (from 08 October 2001 to 13 February 2002, 50%), validation set (from 22 February 2002 to 17 March 2002, 20%) and test set (from 02 April 2002 to 04 May 2002, 30%).

The DX dataset provide a benchmark to assess the performance of the spatiotemporal fusion methods in the task of detecting land-cover change, it contains 29 cloud-free paired of Landsat (Landsat-8 OLI)-Modis (MOD02HKM) RSIs. These paired RSIs were available from September 2013 to November 2019 collected from the south of Beijing (39.0009 N, 115.0986 E, Daxing district), China. The image size of Landsat is 1640×1640, the spatial resolution of MODIS image is 500 m and the image size is consistent with Landsat image by interpolation algorithm. These 29 paired Landsat-Modis RSIs were divided into training set (from 01 September 2013 to 14 December 2016, 50%), validation set (from 07 May 2019 to 03 February 2018, 20%) and test set (from 08 April 2018 to 5 November 2019, 30%).

Both Landsat and MODIS imagery in CIA dataset and DX dataset have six bands with close spectral response functions, and these images were uniformly cropped into nonoverlapping small patches, each measuring 256 × 256 pixels. As a result, each image in the CIA and DX dataset were cropped into 38 and 36 patches, respectively.

B. Implementation Details

In this article, the comparison experiments are carried out on two datasets: CIA dataset and DX dataset. Eight STF methods are selected for performance comparison, including STARFM [21], FSDAF method [50], enhanced deep convolutional spatiotemporal fusion network (EDCSTFN) [59], GAN-based spatio-temporal fusion model (GAN-STFM) [13], fast variation-based spatiotemporal data fusion (FastVSDF) method [60], multilevel feature fusion with generative adversarial network (MLFF-GAN) [14], swin spatiotemporal fusion model (Swin-stfm) [56] and enhanced cross-paired wavelet based spatiotemporal fusion networks (ECPW-STFN) [61]. STARFM, FSDAF, and FastVDSF are nonlearning methods, among them, STARFM and FSDAF are the most frequently used classical methods in STF of RSIs, and FastVDSF is a new proposed nonlearning method in 2024. EDCSTFM, GAN-STFM, MLFF-GAN, Swin-stfm and ECPW-STFN are deep learning-based models. EDCSTFM is the CNNs based model, GAN-STFM and MLFF-GAN use the architecture of GAN, while Swin-stfm and ECPW-STFN are Transformer-based model. For STARFM, FSDAF, FastVDSF, GAN-STFM, ECPW-STFN and the proposed CTSTFM, only one fine imagery at any date and one coarse imagery at target date are required as input. However, EDCSTFM, MLFF-GAN, and Swin-stfm require a

[1][Online]. Available: https://doi.org/10.4225/08/5111AC0BF1229

TABLE I
FOUR EVALUATOR METRICS AND THESE EQUATION AND DESCRIPTION

Metrics	Equation	Description
MAE	$\text{MAE} = \frac{1}{N\times M}\sum_{i=1}^{N}\sum_{j=1}^{M}\lvert I_{i,j}^{\text{target}} - I'_{i,j}\rvert$	N and M are the width and height of the evaluate imagery.
SAM	$\text{SAM} = \frac{1}{N\times M}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\cos^{-1}\left(\frac{(I^{\text{target}})^{T}{}_{j,i}\times I'_{i,j}}{\lvert\lvert I_{i,j}^{\text{target}}\rvert\rvert\times\lvert\lvert I'_{i,j}\rvert\rvert}\right)\right)$	Superscript T denotes matrix transpose.
SSIM	$\text{SSIM}\ (I^{\text{target}}, I') = \frac{(2\mu_{I^{\text{target}}}\ \mu_{I'}+C_1)\times(2\sigma_{I^{\text{target}}{}_{,}I'}+C_2)}{\left(\mu^2_{I^{\text{target}}}+\mu^2_{I'}+C_1\right)\ \times\left(\sigma^2_{I^{\text{target}}}+\sigma^2_{I'}+C_2\right)}$	$\mu_{I^{\text{target}}}$, $\mu_{I'}$, $\sigma_{I^{\text{target}}}$ and $\sigma_{I'}$ represent the mean and variance of I^{target} and $I^{'}$, σ_{IJ} is the covariance between two images. C_1 and C_2 are constant.
PSNR	$\text{PSNR} = 20\times\log_{10}\left(\frac{\text{MAX}\ (I^{\text{target}})}{\sqrt{\text{MSE}\ (I^{\text{target}}, I')}}\right)$	Max represents the max pixel value in target Fine imagery I^{target}.

pair of coarse-fine images at any date and a coarse image of the target date as input. The best and second-best performance will be shown in **bold** and underlined, respectively.

The CTSTFM is implemented using PyTorch 1.12.1 framework with Python 3.9.16. For a fair comparison, we retrain all the methods on CIA and DX datasets, respectively, with RTX 3090 with 24 GB VRAM GPU, running on CUDA 11.3, cuDNN 8.3.2, and Ubuntu 20.04.5 LTS environment, and after each training epoch is completed, test is carried out on the validation set, and the model weight that gets the best MAE result on the validation set is selected. For the CTSTFM, the initial learning rate is set to 1e-4 and the batch size is 4 and the maximum number of training epochs is 150. The adaptive moment estimation (ADAM) optimizer is used to optimize the CTSTFM parameters with momentum hyperparameters $\beta_1 = 0.9$ and $\beta_1 = 0.999$. The model settings, training epochs, the initial learning rate, learning rate decay strategy, batch size and used optimizer for the comparison methods are all follow the information that mentioned in their article.

C. Evaluation Metrics

Four metrics were used to quantitatively evaluate the proposed CTSTFM and other comparison methods, including MAE, spectral angle mapping (SAM), peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM). The equation and description of these four metrics are shown in Table I. MAE and PSNR are used to evaluate the distance between I^{target} and model predict I'. For MAE, the value close to zero means the two images are closer to each other, for PSNR, the value higher is better. A higher SSIM value indicates reflects a higher structural similarity between the two images, the value range of SSIM is [0,1]. The SAM metric is used to evaluate the imagery quality of I^{target} and I', a smaller value means better imagery quality.

D. Model Efficiency Analysis

In order to measure the complexity of the model, we calculate the parameters and time cost of six learning-based STF models. All the models were tested on the RTX4060 Laptop and the input image size is 256×256. As shown in Table II, the parameters and time cost of CTSTFM are 0.701 M and 102.55 ms, respectively, the number of parameters is only one-eighth that of the second-best model MLFF-GAN. Compared with Swin-stfm, which is also Transformer-based model, CTSTFM has a significant advantage. The time cost of CTSTFM is the most of all the models, but considering the performance gains, an additional time cost of less than 100 ms is not unacceptable.

TABLE II
MODEL EFFICIENCY ANALYSIS OF THE SIX LEARNING-BASED SPATIOTEMPORAL FUSION MODELS

Model	Time Cost (ms)	Param. (M)
EDCSTFN	24.98	0.284
GAN-STFM	4.337	0.578 (Generator) 3.6 (Discriminator)
MLFF-GAN	21.178	5.925 (Generator) 2.8 (Discriminator)
Swin-stfm	46.963	37.466
ECPW-STFN	101.172	0.472
CTSTFM	102.455	0.701

IV. DISSCUSION

A. Comparisons With the SOTA Methods on CIA Dataset

Table III list out the MAE, SAM, SSIM, and PSNR results for all methods on the CIA test set. The proposed CTSTFM achieves the best results of 0.01487, 8.20825, 0.87985, and 34.14173 for the six bands average MAE, SAM, SSIM, and PSNR metrics, respectively, ahead of the second-best model, MLFF-GAN, by 0.00065, 0.29416, 0.01048, and 0.18144. Compared to the worst performing method, STARFM, CTSTFM is ahead in MAE, SAM, SSIM, and PSNR by 0.00286, 1.12725, 0.04424, and 1.21018, respectively. In the comparison of the different bands, CTSTFM slightly lags behind the best performing MLFF-GAN by a difference of 0.00159 in SAM metrics in the sixth band, while achieving the best performance in all metrics in other bands.

Fig. 3 shows the visualized comparison results of all the comparison methods on the CIA test set, where the colored images consist of bands 5 (Red), 4 (Green), and 3 (Blue), and the error map is the MAE difference between the ground

TABLE III
FUSION RESULTS QUANTITATIVE COMPARISON WITH SOTA METHODS ON CIA DATASET

Band	Metrics	STARFM	FSDAF	EDCSTFN	GAN-STFM	FastVSDF	MLFF-GAN	Swinstfm	ECPW-STFN	CTSTFM
1	MAE↓	0.00853	0.00849	0.00855	0.01277	0.00862	0.00738	0.00792	0.01035	**0.00706**
	SAM↓	12.06208	12.08997	11.01211	12.72445	12.26189	11.09054	11.38864	11.90873	**10.54813**
	SSIM↑	0.92949	0.92919	0.93931	0.91362	0.92695	0.94488	0.93929	0.92734	**0.9484**
	PSNR↑	38.77081	38.84569	39.0871	36.23073	38.73122	40.00443	39.6149	37.8281	**40.25564**
2	MAE↓	0.00945	0.00947	0.0101	0.0152	0.00956	0.00862	0.00878	0.01065	**0.00831**
	SAM↓	9.13261	9.12988	8.58923	10.97374	9.18106	8.74622	8.70574	8.79135	**8.39331**
	SSIM↑	0.92798	0.92643	0.93663	0.91085	0.92558	0.9339	0.93308	0.93195	**0.93948**
	PSNR↑	37.35034	37.4145	37.0896	34.2585	37.35572	38.06895	38.04235	36.83232	**38.18471**
3	MAE↓	0.01368	0.01323	0.01399	0.02473	0.01328	0.01309	0.01284	0.01376	**0.01193**
	SAM↓	9.94148	9.58602	9.07568	12.35664	9.56794	9.4129	9.10646	9.31789	**8.84984**
	SSIM↑	0.87611	0.88359	0.89779	0.85237	0.88384	0.89045	0.89019	0.8946	**0.90187**
	PSNR↑	34.11173	34.39626	34.1818	30.17545	34.40384	34.67998	34.81969	34.20835	**35.03789**
4	MAE↓	0.02333	0.02099	0.02017	0.02509	0.02084	0.02009	0.01895	0.01981	**0.01849**
	SAM↓	7.90519	6.99904	6.64903	7.83908	6.92429	6.80114	6.59794	6.686	**6.50728**
	SSIM↑	0.78418	0.84006	0.85978	0.8382	0.84489	0.85122	0.85031	0.85082	**0.86386**
	PSNR↑	29.60086	30.53136	30.77714	29.15345	30.61315	30.89403	31.20771	30.92476	**31.21227**
5	MAE↓	0.02628	0.02491	0.02421	0.03369	0.02486	0.02273	0.02357	0.02494	**0.02237**
	SAM↓	7.46719	7.12791	6.73063	8.11566	7.09681	6.7375	6.80992	6.89037	**6.72322**
	SSIM↑	0.74962	0.78597	0.81124	0.77879	0.79012	0.79337	0.79484	0.80372	**0.81155**
	PSNR↑	28.55082	28.94452	29.31882	26.81701	28.97242	29.5714	29.41023	29.05758	**29.584**
6	MAE↓	0.02508	0.02366	0.02298	0.02862	0.02362	0.02121	0.02183	0.02302	**0.02108**
	SAM↓	9.50444	8.94496	8.29287	9.94407	8.89797	**8.22614**	8.30571	8.48634	8.22773
	SSIM↑	0.74627	0.7821	0.80479	0.78238	0.78572	0.80239	0.80064	0.80739	**0.81394**
	PSNR↑	29.20474	29.66789	30.13	28.28227	29.70623	30.54293	30.38194	30.05502	**30.57587**
Mean	MAE↓	0.01773	0.01679	0.01667	0.02335	0.0168	0.01552	0.01565	0.01709	**0.01487**
	SAM↓	9.3355	8.97963	8.39159	10.32561	8.988327	8.50241	8.48574	8.68011	**8.20825**
	SSIM↑	0.83561	0.85789	0.87492	0.84604	0.859517	0.86937	0.86806	0.86930	**0.87985**
	PSNR↑	32.93155	33.30004	33.43074	30.81957	33.2971	33.96029	33.9128	33.15102	**34.14173**

Where the best and second-best performance are **bold** and underlined, the symbol of ↑ and ↓ indicate that higher and lower values are better, respectively.

truth fine image and the method predicted one, the black part indicates no data. Among the comparisons of colored images, the results of all models are nearly the same, but MLFF-GAN shows a clear continuous grid-like texture in the middle of the image. The MAE error map better reflects the gap between the method outputs and the ground truth, with brighter colors represent a wider gap from the ground truth. The errors of the nonlearning-based methods (STARFM, FSDAF, and FastVSDF) are scattered all over the image. The EDCSTFM, MLFF-GAN, and ECPW-STFN errors are mainly concentrated on the left side of the image, and there are some scattered error on the right side. The GAN-STFM has no prominence of errors, but the error map is the brightest, which indicates that the errors are evenly distributed in all parts of the image, and the gap with the ground truth is the largest. Due to the effective global information modeling by the Transformer structure, the error map of Swin-stfm and CTSTFM have no obvious highlights, but the error map of CTSTFM is darker than that of Swin-stfm, reflecting that the fusion results by CTSTFM is most close to the ground truth.

B. Comparisons With the SOTA Methods on DX Dataset

As a further different from the CIA dataset, in which the land-cover type is agricultural area, the DX dataset was taken in the outskirts of the city, and there is a large area of building complexes in the shooting area. The construction process of a large international airport (Daxing International Airport) was completely recorded in DX dataset, which is not covered in the CIA dataset. Meanwhile, the data storage methods of CIA and DX datasets are also different, the data range of RSIs in CIA dataset is [0, 9999], while the images in DX dataset are compressed and the data range is $[0, 255]$. Therefore, CIA dataset and DX dataset are two different types of datasets, in order to verify the influence of different land-cover change types on the model performance, we conducted experiments on the DX dataset.

As shown in Table IV, the trend of the experimental results on DX dataset is similar to that on the CIA dataset, the proposed CTSTFM maintained its advantage, achieves the best MAE, SAM, SSIM, and PSNR of 0.02302, 11.75844,

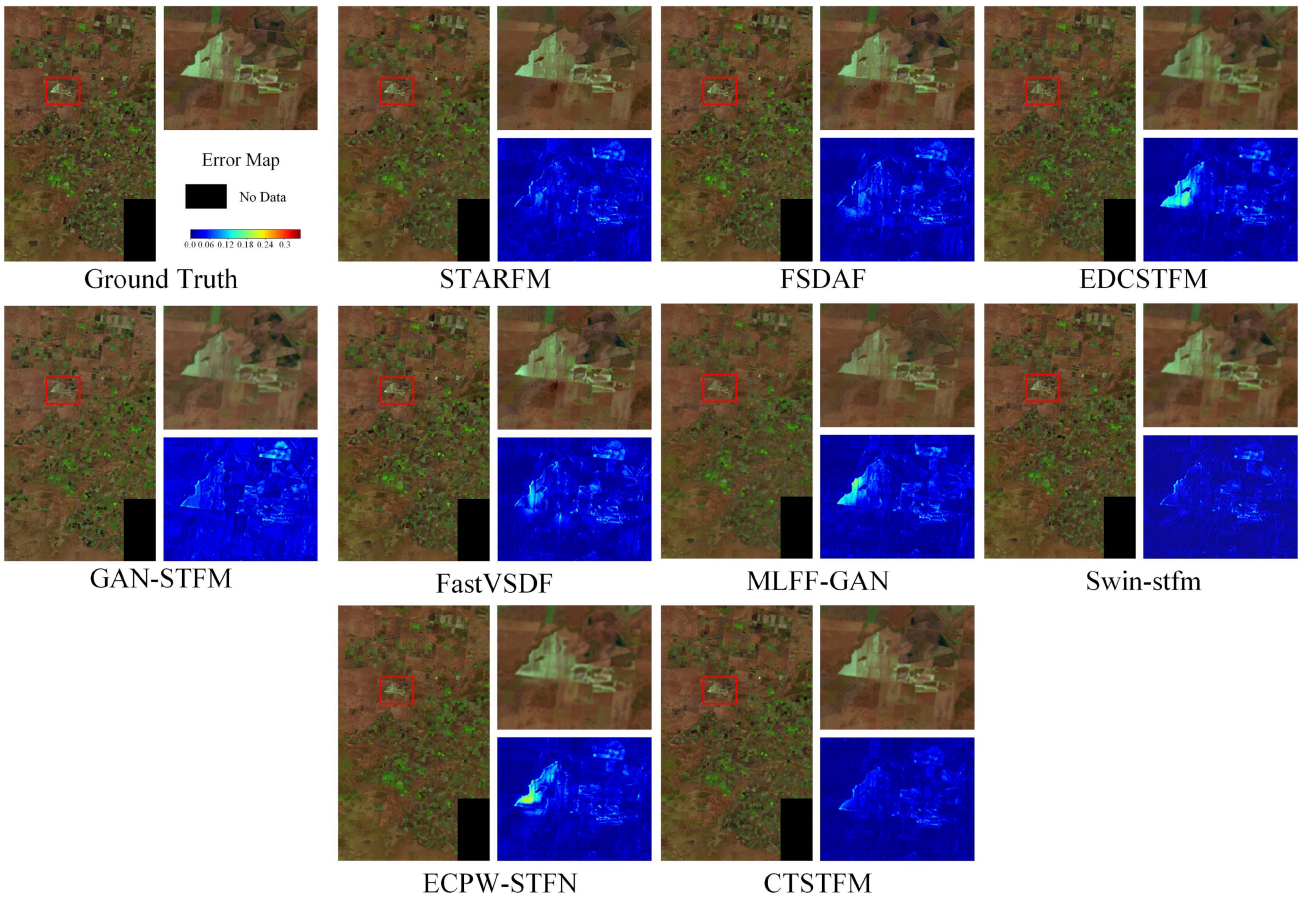

Fig. 3. Fusion results visualization comparison with SOTA methods on CIA dataset on 04 May 2002. The RGB colored images consist of bands 5 (Red), 4 (Green), and 3 (Blue). The error maps are the MAE difference between the ground truth fine image and the model predicted one, the black part indicates no data. Zoom-in for a better view.

0.79267, and 31.01882 for the six bands average, and improves the second-best MLFF-GAN by 0.00114, 0.78052, 0.0245, and 0.51494. CTSTFM also achieves the best performance in each six bands. Compare with the GAN-STFM, CTSTFM takes an outstanding lead of 0.02947, 5.93153, 0.16365, and 5.79821 in MAE, SAM, SSIM, and PSNR metrics, respectively.

Fig. 4 shows the visualized comparison results of all the comparison models on the DX test set. The results of STARFM are blurred and unable to distinguish ground objects, FastVSDF shows a blue error color block. GAN-STFM, EDCSTFM, ECPW-STFN, Swin-stfm, and MLFF-GAN have the phenomenon of color shift, especially the GAN-STFM which is far from the real situation. This result is better illustrated by the error map, with GAN-STFM, FastVSDF, Swin-stfm, and ECPW-STFN having prominent error regions. STARFM, EDCSTFM, and MLFF-GAN do not have prominent error regions, but the images are bright overall, suggest that the errors are distributed all over the place and far from the real situation. FSDAF results are closest to CTSTFM, but CTSTFM has better performance on buildings region.

Due to the extraction and enhancement of global information by MKCE and the interaction and fusion of coarse and fine RSIs features by CFM, the fusion results of CTSTFM are more accurate than other methods and sensitive to the areas with prominent spatial and temporal variations.

C. *Tessellation Effects*

The visualization result of CIA and DX datasets, we note that all methods show varying degrees of tessellation effects, with unnatural gaps appearing where small patches touch each other. As shown in the Fig. 5, this phenomenon is particularly obvious on colored images that consist of bands 1 (Red), 2 (Green), and 3 (Blue), and this problem is worse on models that perform well on the MAE metrics. Especially MLFF-GAN, Swin-stfm, and CTSTFM. In the edge area where the small pieces touch each other, MLFF-GAN presents a green color shift, Swin-stfm, and CTSTFM appear white and black holes, respectively. In addition, MLFF-GAN seems to be affected by either overfitting or all the inherent optimization instability and mode collapse, and the image shows a checkerboard artifacts (please zoom-in the PDF at least four times to observe this phenomenon). However, the tessellation effects is not insurmountable, on the contrary, this problem can be solved by a nonmodel modification approach.

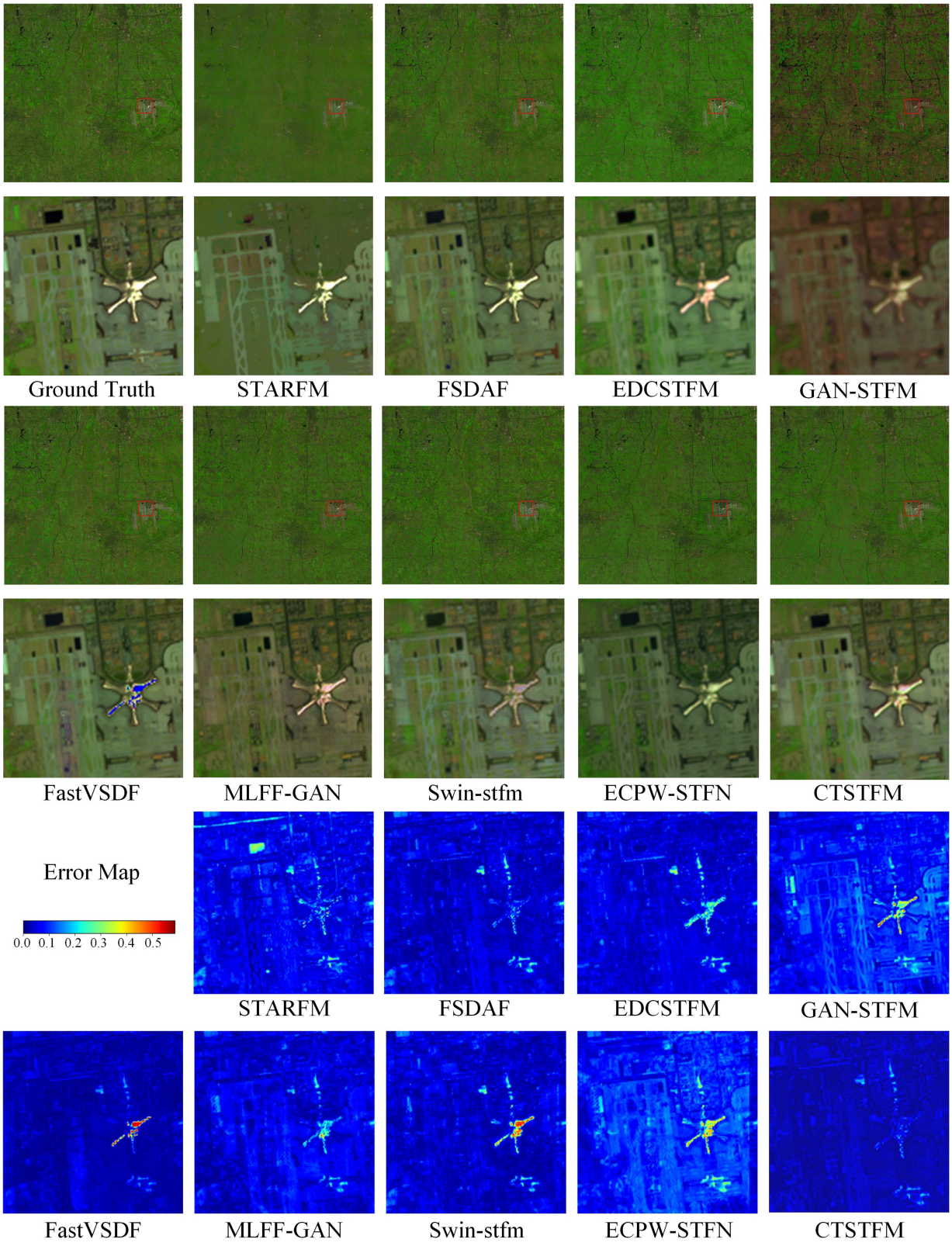

Fig. 4. Fusion results visualization comparison with SOTA methods on DX dataset on 17 August 2019. The colored images consist of bands 5 (Red), 4 (Green), and 3 (Blue). The error maps are the MAE difference between the ground truth fine image and the model predicted one. Zoom-in for a better view.

TABLE IV
FUSION RESULTS QUANTITATIVE COMPARISON WITH SOTA METHODS ON DX DATASET

Band	Metrics	STARFM	FSDAF	EDCSTFN	GAN-STFM	FastVSDF	MLFF-GAN	Swinstfm	ECPW-STFN	CTSTFM
1	MAE↓	0.01365	0.01293	0.0136	0.02401	0.01304	0.01268	0.01374	0.01409	**0.01192**
	SAM↓	15.97247	15.51164	15.60473	22.1476	15.43798	15.02656	16.18839	15.74097	**13.8736**
	SSIM↑	0.86051	0.87327	0.86789	0.74548	0.87638	0.8799	0.86183	0.86605	**0.89598**
	PSNR↑	34.93076	35.2187	34.92493	30.55088	35.14632	35.28817	34.64753	34.71439	**36.02182**
2	MAE↓	0.0163	0.01563	0.01619	0.02773	0.01599	0.01533	0.01671	0.01729	**0.01452**
	SAM↓	14.29665	13.97405	13.77537	18.00945	14.22019	13.53823	14.3926	14.07846	**12.5938**
	SSIM↑	0.83708	0.84526	0.84636	0.74248	0.84743	0.85413	0.83254	0.84204	**0.86973**
	PSNR↑	33.38777	33.64206	33.40152	29.47901	33.42079	33.59976	33.05852	32.92581	**34.32459**
3	MAE↓	0.02315	0.02102	0.02261	0.03991	0.02177	0.01999	0.02137	0.02318	**0.01867**
	SAM↓	17.34815	16.29738	15.84428	20.92885	16.7315	15.43567	16.02186	15.92186	**14.41816**
	SSIM↑	0.73488	0.77329	0.7795	0.64853	0.77395	0.79246	0.77152	0.77866	**0.81606**
	PSNR↑	30.52383	31.21234	30.68263	26.76447	30.86557	31.44827	31.05738	30.54123	**32.11887**
4	MAE↓	0.04728	0.03991	0.05225	0.08129	0.04131	0.03877	0.0415	0.04604	**0.03705**
	SAM↓	10.78548	9.25307	8.73164	12.7453	9.55595	8.85891	9.25756	8.96251	**8.3437**
	SSIM↑	0.40008	0.61261	0.60914	0.48021	0.61731	0.61846	0.59815	0.61123	**0.65446**
	PSNR↑	24.12548	25.55946	23.89845	20.30664	25.26324	25.82494	25.34299	24.7512	**26.14916**
5	MAE↓	0.04295	0.03598	0.03904	0.07689	0.03706	0.03148	0.03445	0.0354	**0.03073**
	SAM↓	12.29308	10.53514	9.86691	13.77591	10.79004	9.31388	9.79557	9.6331	**8.8464**
	SSIM↑	0.50264	0.67927	0.7161	0.5713	0.70217	0.72201	0.7	0.72423	**0.75234**
	PSNR↑	25.34994	26.82078	26.22425	21.39653	26.58678	27.76142	27.12525	26.97397	**27.97838**
6	MAE↓	0.03491	0.0303	0.03158	0.06511	0.03124	0.02668	0.02853	0.02938	**0.02524**
	SAM↓	16.3148	14.60436	13.79254	18.5327	14.92088	13.06049	13.73537	13.63162	**12.47496**
	SSIM↑	0.59505	0.69664	0.73217	0.58612	0.72046	0.74204	0.71778	0.74019	**0.76746**
	PSNR↑	27.13167	28.20843	27.92158	22.82615	27.93335	29.10073	28.61667	28.45888	**29.52011**
Mean	MAE↓	0.02971	0.02596	0.02921	0.05249	0.02674	0.02416	0.02605	0.02756	**0.02302**
	SAM↓	14.50177	13.36261	12.93591	17.68997	13.60942	12.53896	13.23189	12.99475	**11.75844**
	SSIM↑	0.65504	0.74672	0.75853	0.62902	0.75628	0.76817	0.74697	0.7604	**0.79267**
	PSNR↑	29.24158	30.1103	29.50889	25.22061	29.86934	30.50388	29.97472	29.72758	**31.01882**

Where the best and second-best performance are **bold** and underlined, the symbol of ↑ and ↓ indicate that higher and lower values are better, respectively.

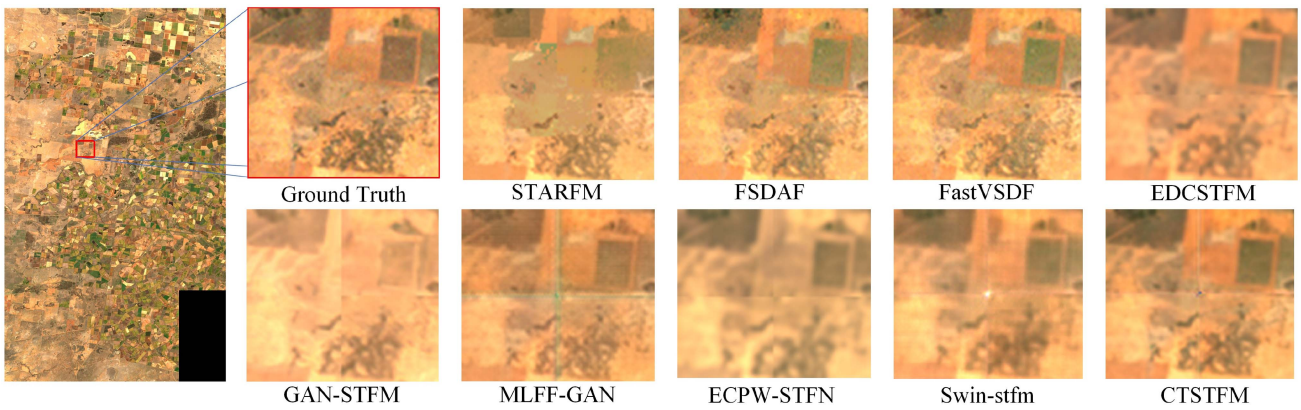

Fig. 5. Fusion results visualization comparison with SOTA methods on CIA dataset on 11 April 2006. The colored images consist of bands 1 (Red), 2 (Green) and 3 (Blue). Zoom-in for a better view.

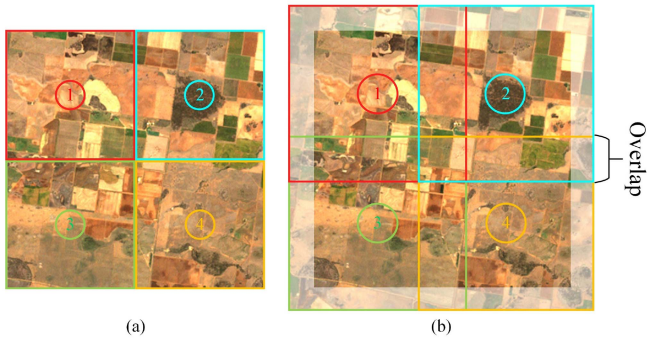

Fig. 6. Illustration of the data cropping process. (a) Nonoverlap cropped. (b) Cropped with overlap.

Before proposing a solution, we will describe the process of cropping images and explain the reason for doing so. As shown in Fig. 6(a), we crop the whole image uniformly into a number of small patches according to a certain size, and there is no overlap between these small patches, which are used in train, validation and test set of both CIA and DX dataset. This is a regular operation of data preprocessing because of the limitation of GPU memory, a complete RSIs often contain of 6000 × 6000 pixels or even higher, the conventional GPU cannot accommodate the whole image for computing, and the professional computing GPU with 24 GB or even higher memory are often sold at a high price. At the same time, as shown in the full frame row of the Tables V and VI, full frame indicates that the complete RSIs is directly used as input. Compared with the result of using the cropped image as input, all models show performance degradation on both datasets. This reflect that the resolution of the model's input is far from that of the training data, it will cause performance degradation.

One of the methods to solve the tessellation effects is to make the individual small patches have some overlap between each other. As shown in Fig. 6(b), unlike the previous approach as shown in Fig. 6(a), we artificially make the small patches have some overlap between each other during the image cropping process. These overlaped small patches are used as inputs to the methods, and finally the outputs are cropped to obtain the final result. In order to verify the effectiveness of this approach, we did experiments on the CIA test set. As shown in the Fig. 7, the tessellation effects is mitigated for both MLFF-GAN and CTSTFM when the overlap size is greater than 0. When the overlap size is equal to 32, the tessellation effects is almost unobservable, and continuing to increase the overlap size does not optimize the results. For the influence of fusion accuracy, as shown in the Tables V and VI, overlap sizes of 16 and 32 did not affect the results, while a small decrease was observed for sizes larger than 32. This result demonstrates that, slightly increasing the size of the input image will not significantly affect accuracy, and crop the whole RSIs into several small patches for processing is an efficient and economical solution. It is worth noting that this method does not solve the checkerboard artifacts of MLFF-GAN, and the unnatural texture still exists among the images. And Swin-stfm can only accept fixed-size image sizes, that is, the image sizes used during training stage, so the tessellation effects of Swin-stfm cannot be solved by this method.

TABLE V
FUSION RESULTS QUANTITATIVE COMPARISON WITH DIFFERENT OVERLAP SIZES ON CIA TEST SET

overlap size	Model	MAE↓	SAM↓	SSIM↑	PSNR↑
0	EDCSTFN	0.016667	8.391592	0.874923	33.43074
	GAN-STFM	0.02335	10.32561	0.846035	30.81957
	MLFF-GAN	0.01552	8.502407	0.869368	33.96029
	ECPW-STFN	0.017088	8.680113	0.869303	33.15102
	CTSTFM	**0.014873**	**8.208252**	**0.87985**	**34.14173**
16	EDCSTFN	0.016707	8.38369	0.875785	33.4201
	GAN-STFM	0.02312	10.25308	0.84704	30.8963
	MLFF-GAN	0.015555	8.556827	0.870678	33.95256
	ECPW-STFN	0.017182	8.691892	0.869427	33.11298
	CTSTFM	**0.014887**	**8.21638**	**0.879773**	**34.1263**
32	EDCSTFN	0.016707	8.38369	0.875785	33.4201
	GAN-STFM	0.022968	10.20208	0.847753	30.94745
	MLFF-GAN	0.015577	8.554507	0.870765	33.94974
	ECPW-STFN	0.017433	8.841083	0.867408	32.95293
	CTSTFM	**0.014887**	**8.208985**	**0.879833**	**34.12433**
64	EDCSTFN	0.016707	8.38369	0.875785	33.4201
	GAN-STFM	0.022795	10.14736	0.848475	30.99983
	MLFF-GAN	0.015592	8.551383	0.87085	33.94719
	ECPW-STFN	0.017537	8.885393	0.867115	32.88586
	CTSTFM	**0.014873**	**8.208252**	**0.87985**	**34.12506**
96	EDCSTFN	0.017087	8.555195	0.87403	33.22166
	GAN-STFM	0.023018	10.26253	0.847115	30.91062
	MLFF-GAN	0.015945	8.699457	0.869315	33.72663
	ECPW-STFN	0.017978	9.073562	0.865217	32.7107
	CTSTFM	**0.015287**	**8.403808**	**0.878008**	**33.86529**
Full Frame	EDCSTFN	0.017917	8.903605	0.842515	33.05644
	GAN-STFM	0.026597	11.01946	0.80058	30.17355
	MLFF-GAN	0.018343	9.233115	0.830622	32.93697
	ECPW-STFN	0.018735	9.366848	0.830253	32.6132
	CTSTFM	**0.016153**	**8.461378**	**0.852768**	**33.72874**

Where the best performance are **bold**, the symbol of ↑ and ↓ indicate that higher and lower values are better, respectively. "Full Frame" indicates that the complete rsis is directly used as input.

D. Ablation Study

The proposed CTSTFM consists of four basic modules, including MKCB, MKSB, CETB, and CFTB. In order to explore the importance of these four basic modules, we conducted ablation experiments. As shown in the Table VII, no matter which of the basic modules is removed, it will cause a decrease in accuracy, especially the CETB, compared to the complete CTSTFM, its MAE, RMSE, SSIM, and PSNR decreased by 0.00072, 0.00084, 0.0095, and 0.26689, respectively, which is the module with the greatest impact on accuracy among all the four modules, and even the overall accuracy is lower than the second best performing method MLFF-GAN, but still has a better performance than the other methods. Due to the synergistic work of these four basic modules, a higher accuracy than the existing SOTA RSIs spatiotemporal fusion model is achieved in CTSTFM.

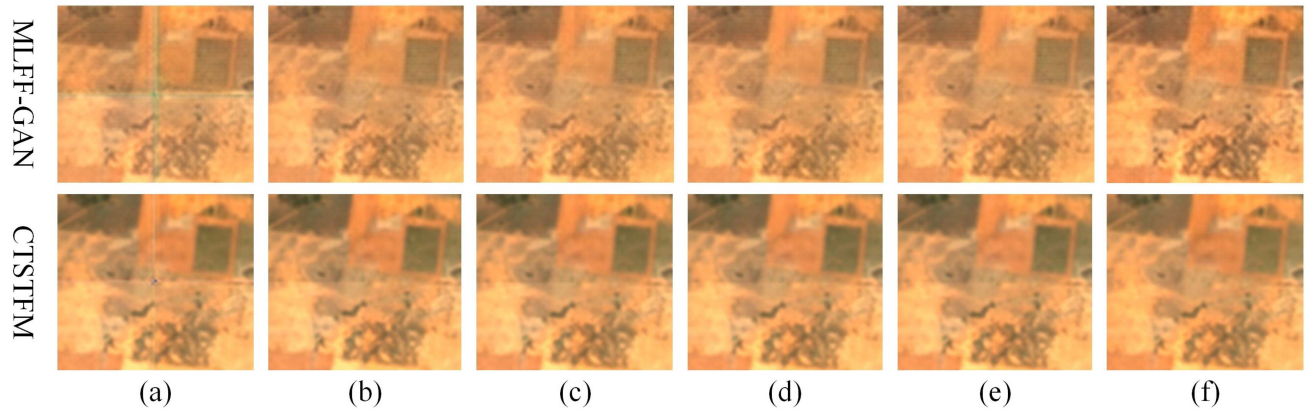

Fig. 7. Fusion results visualization comparison of CTSTFM and MLFF-GAN with five different overlap sizes on CIA dataset, 11 April 2006. The colored images consist of bands 1 (Red), 2 (Green), and 3 (Blue). Zoom-in for better view. Overlap size: (a) 0, (b) 16, (c) 32, (d) 64, (e) 96, (f) Full Frame. "Full frame" indicates that the complete RSIs is directly used as input.

TABLE VI
FUSION RESULTS QUANTITATIVE COMPARISON WITH DIFFERENT OVERLAP SIZES ON DX TEST SET

overlap size	Model	MAE↓	SAM↓	SSIM↑	PSNR↑
0	EDCSTFN	0.029212	12.93591	0.758527	29.50889
	GAN-STFM	0.05249	17.68997	0.62902	25.22061
	MLFF-GAN	0.024155	12.53896	0.768167	30.50388
	ECPW-STFN	0.027563	12.99475	0.7604	29.72758
	CTSTFM	**0.023022**	**11.75844**	**0.792672**	**31.01882**
16	EDCSTFN	0.029183	12.92076	0.759053	29.5183
	GAN-STFM	0.052748	16.75419	0.671993	25.51543
	MLFF-GAN	0.024258	12.59466	0.766725	30.46386
	ECPW-STFN	0.027707	12.99282	0.760855	29.69221
	CTSTFM	**0.022947**	**11.73029**	**0.793758**	**31.04219**
32	EDCSTFN	0.029183	12.92076	0.759053	29.5183
	GAN-STFM	0.05263	16.66701	0.672807	25.54158
	MLFF-GAN	0.024402	12.66313	0.764778	30.41667
	ECPW-STFN	0.027975	13.02328	0.760335	29.63146
	CTSTFM	**0.022917**	**11.72086**	**0.794302**	**31.04909**
64	EDCSTFN	0.029183	12.92076	0.759053	29.5183
	GAN-STFM	0.051867	16.39254	0.676147	25.69802
	MLFF-GAN	0.02457	12.742	0.762227	30.36153
	ECPW-STFN	0.028103	13.01416	0.760005	29.58809
	CTSTFM	**0.022898**	**11.70581**	**0.794717**	**31.05064**
96	EDCSTFN	0.029183	12.92076	0.759053	29.5183
	GAN-STFM	0.051497	16.21722	0.677542	25.78053
	MLFF-GAN	0.024672	12.7879	0.7604	30.32549
	ECPW-STFN	0.02812	13.0223	0.760018	29.59978
	CTSTFM	**0.022895**	**11.70807**	**0.795125**	**31.04306**
Full Frame	EDCSTFN	0.029187	12.92171	0.759047	29.5177
	GAN-STFM	0.051907	17.17116	0.629293	25.39628
	MLFF-GAN	0.025155	13.01774	0.75724	30.17321
	ECPW-STFN	0.035573	13.70798	0.74132	28.10164
	CTSTFM	**0.023682**	**12.07209**	**0.792852**	**30.6932**

Where the best performance are **bold**, the symbol of ↑ and ↓ indicate that higher and lower values are better, respectively. "Full Frame" indicates that the complete rsis is directly used as input.

TABLE VII
ABLATION STUDY OF IMPORTANCE OF THE FOUR BASIC MODULES OF CTSTFM

Dropped Module	MAE↓	RMSE↓	SSIM↑	PSNR↑
\	**0.01567**	**0.02444**	**0.96791**	**32.92489**
MKCB	0.01616	0.02461	0.96751	32.84447
MKSB	0.01599	<u>0.02451</u>	0.96754	<u>32.87878</u>
CETB	0.01639	0.02528	0.95841	32.6584
CFTB	<u>0.01585</u>	0.02459	<u>0.96768</u>	32.86951

Where the best and second-best performance are bold and underlined, the symbol of ↑ and ↓ indicate that higher and lower values are better, respectively.

V. CONCLUSION

In this article, we combine the CNN and Transformer architectures and apply them to the RSIs spatiotemporal fusion task, proposed two feature extraction basic modules (MKCB and MKSB) and two feature fusion basic modules (CETB and CFTB). In particular, the multikernel convolutional blocks, channel and spatial attention mechanisms in the two feature extraction modules solve the CNN limitation about the size of convolutional kernels, and the two feature fusion modules achieve information fusion between coarse and fine feature maps through feature information exchange. The quantitative and visualization comparisons on both CIA and DX datasets demonstrate that CTSTFM outperform the existing SOTA RSIs spatiotemporal fusion methods. Considering that CTSTFM requires only two images as input, which reduces the data requirements and allows practical applications to show better flexibility. Compared with GAN-STF, which also requires two images as input only, CTSTFM has better performance.

Nevertheless, we noticed that the four commonly used quality evaluation metrics of RSIs selected in this article could not comprehensively evaluate the fusion results. For example, the MLFF-GAN, which performance is second only to CTSTFM, shows great competitiveness in evaluation metrics, but the visualization results show obvious tessellation effects and

checkerboard artifacts. The image quality evaluation system of RSIs spatiotemporal fusion still needs to be further explored. In addition, for the two-branch structure combining coarse and fine feature maps in the CTSTFM is not efficient, there is still much room for improvement. Therefore, our subsequent works will mainly focus on the efficient design of deep network architecture for RSIs STF task to further improve the performance and robustness of the model.

ACKNOWLEDGMENT

This work was developed by the IEEE Publication Technology Department. This work is distributed under the Project Public License (LPPL) (http://www.latex-project.org/) version 1.3. A copy of the LPPL, version 1.3, is included in the base documentation of all distributions of released 2003/12/01 or later. The opinions expressed here are entirely that of the author. No warranty is expressed or implied. User assumes all risk.

REFERENCES

[1] P. Defourny et al., "Near real-time agriculture monitoring at national scale at parcel resolution: Performance assessment of the Sen2-Agri automated system in various cropping systems around the world," *Remote Sens. Environ.*, vol. 221, pp. 551–568, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425718305145

[2] A. N. French et al., "Surface energy fluxes with the advanced spaceborne thermal emission and reflection radiometer (ASTER) at the Iowa 2002 SMACEX site (USA)," *Remote Sens. Environ.*, vol. 99, no. 1/2, pp. 55–65, 2005.

[3] R. Padmanaban, A. K. Bhowmik, and P. Cabral, "A remote sensing approach to environmental monitoring in a reclaimed mine area," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 12, 2017, Art. no. 401.

[4] Q. Yuan et al., "Deep learning in environmental remote sensing: Achievements and challenges," *Remote Sens. Environ.*, vol. 241, 2020, Art. no. 111716.

[5] W. Liu, L. Ma, J. Wang, and H. xsChen, "Detection of multiclass objects in optical remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 5, pp. 791–795, May 2019.

[6] Z. Ren, Y. Tang, Z. He, L. Tian, Y. Yang, and W. Zhang, "Ship detection in high-resolution optical remote sensing images aided by saliency information," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5623616.

[7] M. Tancik et al., "Block-Nerf: Scalable large scene neural view synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8248–8258.

[8] Y. Xiangli et al., "BungeeNeRF: Progressive neural radiance field for extreme multi-scale scene rendering," in *Proc. Eur. Conf. Comput. Vis.*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., 2022, pp. 106–122.

[9] Y. Liu, X. Chen, Z. Wang, Z. J. Wang, R. K. Ward, and X. Wang, "Deep learning for pixel-level image fusion: Recent advances and future prospects," *Inf. Fusion*, vol. 42, pp. 158–173, 2018.

[10] F. Meng et al., "Single remote sensing image super-resolution via a generative adversarial network with stratified dense sampling and chain training," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5400822.

[11] S. Hou et al., "RFSDAF: A new spatiotemporal fusion method robust to registration errors," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5616018.

[12] X. Zhu, F. Cai, J. Tian, and T. K.-A. Williams, "Spatiotemporal fusion of multisource remote sensing data: Literature survey, taxonomy, principles, applications, and future directions," *Remote Sens.*, vol. 10, no. 4, 2018, Art. no. 527.

[13] Z. Tan, M. Gao, X. Li, and L. Jiang, "A flexible reference-insensitive spatiotemporal fusion model for remote sensing images using conditional generative adversarial network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5601413.

[14] B. Song et al., "MLFF-GAN: A multilevel feature fusion with GAN for spatiotemporal remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4410816.

[15] J. Xiao, A. K. Aggarwal, N. H. Duc, A. Arya, U. K. Rage, and R. Avtar, "A review of remote sensing image spatiotemporal fusion: Challenges, applications and recent trends," *Remote Sens. Appl.: Soc. Environ.*, vol. 32, 2023, Art. no. 101005.

[16] B. Zhukov, D. Oertel, F. Lanzl, and G. Reinhackel, "Unmixing-based multisensor multiresolution image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 3, pp. 1212–1226, May 1999.

[17] J. A.-Lopez, L. G.-Chova, L. Alonso, L. Guanter, J. Moreno, and G. C.-Valls, "Regularized multiresolution spatial unmixing for ENVISAT/MERIS and Landsat/TM image fusion," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 5, pp. 844–848, Sep. 2011.

[18] W. Shi, D. Guo, and H. Zhang, "A reliable and adaptive spatiotemporal data fusion method for blending multi-spatiotemporal-resolution satellite images," *Remote Sens. Environ.*, vol. 268, 2022, Art. no. 112770.

[19] Q. Wang, K. Peng, Y. Tang, X. Tong, and P. M. Atkinson, "Blocks-removed spatial unmixing for downscaling MODIS images," *Remote Sens. Environ.*, vol. 256, 2021, Art. no. 112325.

[20] W. Liu, Y. Zeng, S. Li, and W. Huang, "Spectral unmixing based spatiotemporal downscaling fusion approach," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 88, 2020, Art. no. 102054.

[21] F. Gao, J. Masek, M. Schwaller, and F. Hall, "On the blending of the Landsat and MODIS surface reflectance: Predicting daily Landsat surface reflectance," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 8, pp. 2207–2218, Aug. 2006.

[22] J. Wang and B. Huang, "A rigorously-weighted spatiotemporal fusion model with uncertainty analysis," *Remote Sens.*, vol. 9, no. 10, 2017, Art. no. 990.

[23] C. Liao, J. Wang, I. Pritchard, J. Liu, and J. Shang, "A spatio-temporal data fusion model for generating NDVI time series in heterogeneous regions," *Remote Sens.*, vol. 9, no. 11, 2017, Art. no. 1125.

[24] H. Shen, P. Wu, Y. Liu, T. Ai, Y. Wang, and X. Liu, "A spatial and temporal reflectance fusion model considering sensor observation differences," *Int. J. Remote Sens.*, vol. 34, no. 12, pp. 4367–4383, 2013.

[25] R. Ghosh, P. K. Gupta, V. Tolpekin, and S. Srivastav, "An enhanced spatiotemporal fusion method–implications for coal fire monitoring using satellite imagery," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 88, 2020, Art. no. 102056.

[26] R. Dong, L. Zhang, and H. Fu, "RRSGAN: Reference-based super-resolution for remote sensing image," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5601117.

[27] Z. Wang, Y. Ma, and Y. Zhang, "Review of pixel-level remote sensing image fusion based on deep learning," *Inf. Fusion*, vol. 90, pp. 36–58, 2023.

[28] J. Li et al., "Deep learning in multimodal remote sensing data fusion: A comprehensive review," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 112, 2022, Art. no. 102926.

[29] A. Li, Y. Bo, Y. Zhu, P. Guo, J. Bi, and Y. He, "Blending multi-resolution satellite sea surface temperature (SST) products using Bayesian maximum entropy method," *Remote Sens. Environ.*, vol. 135, pp. 52–63, 2013.

[30] Á. Moreno-Martínez et al., "Multispectral high resolution sensor fusion for smoothing and gap-filling in the cloud," *Remote Sens. Environ.*, vol. 247, 2020, Art. no. 111901.

[31] B. Huang, H. Zhang, H. Song, J. Wang, and C. Song, "Unified fusion of remote-sensing imagery: Generating simultaneously high-resolution synthetic spatial-temporal-spectral Earth observations," *Remote Sens. Lett.*, vol. 4, no. 6, pp. 561–569, 2013.

[32] B. Huang and H. Song, "Spatiotemporal reflectance fusion via sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3707–3716, Oct. 2012.

[33] J. Wei, L. Wang, P. Liu, X. Chen, W. Li, and A. Y. Zomaya, "Spatiotemporal fusion of MODIS and Landsat-7 reflectance images via compressed sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7126–7139, Dec. 2017.

[34] B. Chen, B. Huang, and B. Xu, "A hierarchical spatiotemporal adaptive fusion model using one image pair," *Int. J. Digit. Earth*, vol. 10, no. 6, pp. 639–655, 2017.

[35] H. Song and B. Huang, "Spatiotemporal satellite image fusion through one-pair image learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 1883–1896, Apr. 2013.

[36] Z. Liu, R. Feng, L. Wang, W. Han, and T. Zeng, "Dual learning-based graph neural network for remote sensing image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5628614.

[37] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.
[38] H. Song, Q. Liu, G. Wang, R. Hang, and B. Huang, "Spatiotemporal satellite image fusion using deep convolutional neural networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 821–829, Mar. 2018.
[39] Z. Tan, P. Yue, L. Di, and J. Tang, "Deriving high spatiotemporal remote sensing images using deep convolutional network," *Remote Sens.*, vol. 10, no. 7, 2018, Art. no. 1066.
[40] X. Liu, C. Deng, J. Chanussot, D. Hong, and B. Zhao, "STFNet: A two-stream convolutional neural network for spatiotemporal image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6552–6564, Sep. 2019.
[41] Y. Li, J. Li, L. He, J. Chen, and A. Plaza, "A new sensor bias-driven spatio-temporal fusion model based on convolutional neural networks," *Sci. China Inf. Sci.*, vol. 63, pp. 1–16, 2020.
[42] W. Li, C. Yang, Y. Peng, and X. Zhang, "A multi-cooperative deep convolutional neural network for spatiotemporal satellite image fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 10 174–10 188, 2021.
[43] Y. Ma, J. Wei, W. Tang, and R. Tang, "Explicit and stepwise models for spatiotemporal fusion of remote sensing images with deep neural networks," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 105, 2021, Art. no. 102611.
[44] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
[45] H. Zhang, Y. Song, C. Han, and L. Zhang, "Remote sensing image spatiotemporal fusion using a generative adversarial network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4273–4286, May 2020.
[46] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2017. [Online]. Available: https://openreview.net/
[47] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 5769–5779.
[48] G. Yang et al., "MSFusion: Multistage for remote sensing image spatiotemporal fusion based on texture transformer and convolutional neural network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4653–4666, 2022.
[49] T. Benzenati, A. Kallel, and Y. Kessentini, "STF-Trans: A two-stream spatiotemporal fusion transformer for very high resolution satellites images," *Neurocomputing*, vol. 563, 2024, Art. no. 126868. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231223009918
[50] X. Zhu, E. H. Helmer, F. Gao, D. Liu, J. Chen, and M. A. Lefsky, "A flexible spatiotemporal method for fusing satellite images with different resolutions," *Remote Sens. Environ.*, vol. 172, pp. 165–177, 2016.
[51] J. Cui, X. Zhang, and M. Luo, "Combining linear pixel unmixing and STARFM for spatiotemporal fusion of Gaofen-1 wide field of view imagery and modis imagery," *Remote Sens.*, vol. 10, no. 7, 2018, Art. no. 1047.
[52] S. Hou et al., "Adaptive-sfsdaf for spatiotemporal image fusion that selectively uses class abundance change information," *Remote Sens.*, vol. 12, no. 23, 2020, Art. no. 3979.
[53] D. Guo, W. Shi, M. Hao, and X. Zhu, "FSDAF 2.0: Improving the performance of retrieving land cover changes and preserving spatial details," *Remote Sens. Environ.*, vol. 248, 2020, Art. no. 111973.
[54] S. Wang et al., "A classification-based spatiotemporal adaptive fusion model for the evaluation of remotely sensed evapotranspiration in heterogeneous irrigated agricultural area," *Remote Sens. Environ.*, vol. 273, 2022, Art. no. 112962.
[55] W. Li, D. Cao, Y. Peng, and C. Yang, "MSNet: A multi-stream fusion network for remote sensing spatiotemporal fusion based on transformer and convolution," *Remote Sens.*, vol. 13, no. 18, 2021, Art. no. 3724. [Online]. Available: https://www.mdpi.com/2072-4292/13/18/3724
[56] G. Chen, P. Jiao, Q. Hu, L. Xiao, and Z. Ye, "SwinSTFM: Remote sensing spatiotemporal fusion using swin transformer," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5410618.
[57] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.
[58] J. Li, Y. Li, L. He, J. Chen, and A. Plaza, "Spatio-temporal fusion for remote sensing data: An overview and new benchmark," *Sci. China Inf. Sci.*, vol. 63, pp. 1–17, 2020.
[59] Z. Tan, L. Di, M. Zhang, L. Guo, and M. Gao, "An enhanced deep convolutional model for spatiotemporal image fusion," *Remote Sens.*, vol. 11, no. 24, 2019, Art. no. 2898.
[60] C. Xu et al., "FastVSDF: An efficient spatiotemporal data fusion method for seamless data cube," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5402022.
[61] X. Zhang, S. Li, Z. Tan, and X. Li, "Enhanced wavelet based spatiotemporal fusion networks using cross-paired remote sensing images," *ISPRS J. Photogrammetry Remote Sens.*, vol. 211, pp. 281–297, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092427162400176X

Mingyu Jiang received the B.E. degree in surveying and mapping engineering in 2022 from College of Geomatics Science and Technology, Nanjing Tech University, Nanjing, China, where he is currently working toward the M.S. degree in geography with College of Geomatics Science and Technology.

His research interests include deep learning, data fusion, and intelligent processing of remote sensing imagery.

Hua Shao received the B.S. degree in information and computing science from Nanjing University, Nanjing, China, in 2004, and the Ph.D. degree in cartography and geographic information system from Nanjing Normal University, Nanjing, in 2014.

From 2018 to 2019, he was a Visiting Scholar with George Mason University, USA. He is currently an Associate Professor with College of Geomatics Science and Technology, Nanjing Tech University, Nanjing, China. His research interests include intelligent interpretation of remote sensing image data, spatiotemporal data analysis and mining, and neural network based 3-D reconstruction.