Supplementary Material
Feature Spectrum Learning for Remote Sensing Change Detection

Qi Zang[1*], Dong Zhao[1*], Shuang Wang[1✉], Dou Quan[1], Zhun Zhong[2]
[1] School of Artificial Intelligence, Xidian University, Shaanxi, China
[2] School of Computer Science and Information Engineering, Hefei University of Technology, China
shwang@mail.xidian.edu.cn

In the supplementary material, we further provide implementation details of the proposed modules, used datasets and experimental setup, an analysis of why spectral transform is effective in change detection, more visualization of modules, ablation studies on another benchmark, comparison of generalization, comparisons on more datasets or tasks, comparison and discussion of efficiency, a visual comparison of the detection results of our method with other methods, and a presentation of more generated data for existing transformation-based methods.

1. Implementation Details of Modules

We provide implementation details of the proposed EB-LS and EB-AA to aid in understanding our algorithms.

1.1. Details of EB-LS

The proposed EB-LS (*i.e.*, Eq. (7) in the original text) is implemented by the following pseudo-code. It outputs the **symmetric binary vector** $\mathrm{mask}^{\mathrm{h}}$, where the values in the intervals [0, $H/2 - HS_h/2$] and [$H/2 + HS_h/2$, H] are 0, while in the interval [$H/2 - HS_h/2$, $H/2 + HS_h/2$] are 1, as shown in Fig. 2(c)-*II* and Fig. 3 in the original text. The outer product between two symmetric binary vectors $\mathrm{mask}^{\mathrm{h}}$ and $\mathrm{mask}^{\mathrm{w}}$ in Eq. (5) of the original text makes only the center region ($WS_w \times HS_h$) of matrix $\acute{M}$ to be ones.

Algorithm 1 Details of EB-LS

$m \leftarrow$ torch.arange$(0, H)$
$\mathrm{mask}^{\mathrm{h}} \leftarrow$ torch.clamp(torch.ceil($H \times S_h/2 - (H/2 - m)$.abs()), 0, 1) # S_h is the strides with gradient.

1.2. Details of EB-AA

With the above-built binary $\acute{M}$, we use the re-parameterization trick to optimize the spectrum value as follows. When optimizing M^o with Eq. (3) in the original text, $\grave{M}^o$ will be optimized as a floating-point number. We simply binarize it (use the derivable *torch.round()*) and make the gradient have enough influence.

Algorithm 2 Details of EB-AA

$\grave{M}^o \leftarrow$ torch.nn.Parameter($\acute{M}$.clone().detach())
$\grave{M}^o \leftarrow$ Update history optimized spectral values to $\grave{M}^o$
$M^o \leftarrow \acute{M} \cdot \grave{M}^o$ #optimize value and size of spectrum.

2. Why is spectrum transformation effective in siamese-dominated change detection?

Here, we further analyze why spectrum transformation is effective in siamese-dominated change detection. In the change detection, we are provided with two geographically registered images x^{t_1} and x^{t_2} from different phases. The goal is to generate a change map $\mathbf{P}_{out}$ that indicates where the change occurs between x^{t_1} and x^{t_2}. The above process is a pixel-by-pixel binary classification of the bitemporal image, and can be defined as Eq. (1) in the original text. x^{t_1} and x^{t_2} are first analyzed by the feature extractor f, then their features are extracted and compared, and finally the different parts are classified by the classifier h.

With the deepening of research, scholars found that the models are prone to detect the variant of the imaging conditions (seasons or acquisition sensors) as change, which is called the pseudo-changes. These pseudo-changes directly affect the calculation of the difference feature in the interactive operation, leading to the misjudgment of the h. Existing work regards this pseudo-changes as a style shift. Thus, an image x can be decoupled into content (scene structure) x_c and style (imaging condition) x_s,

$$\begin{aligned} x^{t_1} &= x_c^{t_1} \oplus x_s^{t_1}, \\ x^{t_2} &= x_c^{t_2} \oplus x_s^{t_2}, \end{aligned} \tag{1}$$

where $\oplus$ is the nonlinear superposition operation. Existing GAN-based methods operate on pixels (*i.e.*, on the x) in the spatial domain and transform image styles to eliminate

shifts, which makes them prone to excessively destroying content x_c. Then, it is necessary to find a style proxy to achieve alignment that can avoid the loss of content x_c by providing explicit physical guidance (*i.e.*, only operate on x_s). Traditional spectrum transformation (ST) achieve this goal in image space,

$$\mathcal{F}^{-1}(x) = \mathcal{F}_A^x + \mathcal{F}_P^x. \tag{2}$$

As shown in Eq. (1) in the original text, the interactive operation is performed on the encoded features. If the style part (amplitude spectrum $\mathcal{F}_A^x$ obtained by ST) of the image is processed before encoding, the discriminability of the obtained features for content $\mathcal{F}_P^x$ may be weakened. Because operations on $\mathcal{F}_A^x$ only indicate visual consistency and may cause over-alignment. The stylized new image mapped back to the spatial domain is prone to distortion, which negatively affects the extraction of rich nonlinear features so that h loses the key basis for analyzing the images. That is, once the input is damaged, the loss will be irreversible. To this end, we formulate ST into feature space and achieve alignment directly based on the encoded features $f(x)$,

$$\mathcal{F}^{-1}(f(x)) = \mathcal{F}_A^{f(x)} + \mathcal{F}_P^{f(x)}. \tag{3}$$

Note that the object to be processed is $\mathcal{F}_A^{f(x)}$ and not $\mathcal{F}_A^x$. In the high-dimensional feature space, we first ensure the acquisition of rich nonlinear representations based on the original input images. Then, ST is only applied to the shallow layers of the neural network which mainly contain style information. This significantly preserves the discriminative power of the features while precisely eliminating style shifts.

3. Datasets and Experimental Setup

3.1. Datasets

In the original text, we use two datasets Season-Varying Change Detection Dataset (SVCD) [11] and DSIFN Change Detection Dataset (DSIFN) [17] to conduct experiments. The SVCD has 11 image pairs obtained by Google Earth, including 7 pairs with the original size of 4725×2700 and 4 pairs with the original size of 1900×1000. The spatial resolution is from 3 cm to 100 cm per pixel. All images are further cropped to 256×256 patches. We use 10,000 patches as the training set, 3,000 patches as the validation set, and 3,000 patches as the testing set. The DSIFN is collected from Google Earth and consists of 6 pairs covering six cities in China, Beijing, Chengdu, Shenzhen, Chongqing, Wuhan, and Xian. Each image has a high resolution of 2 m. The five original image pairs (Beijing, Chengdu, Shenzhen, Chongqing, and Wuhan) are further cropped into 394 pairs of size 512×512. After applied data augmentation on them, 3,940 image pairs are obtained, 90% of which are used for training and the remaining 10% for validation. The original image pairs from Xian are also cropped into patches of the same size for testing. Since the images for training and testing the model come from different places, the generalization ability of the model has great challenges in this dataset.

3.2. Implementation Details

The batch size is set to 16. The SGD is adopted as our optimizer to optimize the network parameters, where the weight decay rate is set to 1×10^{-8} and momentum is set to 0.9. The initial learning rates of the network and EB-AA/LS are set to 1×10^{-4} and 1×10^{-3} respectively, which are decayed following a polynomial learning rate scheduling with a power of 0.9 during training. To evaluate the performance of our method, we utilize four standard evaluation metrics, *i.e.*, precision (Prec.), recall (Rec.), F1-score (F1), and intersection over union (IoU). All experiments are implemented in PyTorch 1.8 on an Nvidia Tesla RTX3090 GPU with 24GB of memory.

3.3. Network

We adopt the widely used FC-Siam-diff-res [5] as base network to analyze the effectiveness of our method FeaSpect. The proposed FST strategy is only embedded in the first two layers of the FC-Siam-diff-res.

4. Learned Strides on Different Datasets

We visualize the learning trajectory of strides on the SVCD dataset in the original text. To further analyze the trend of the learned strides on different data, we first provide the learning trajectory of the stride on the DSIFN dataset in Fig. 1. We then provide the distribution of learned strides on both datasets, as shown in Fig. 2. In Fig. 2, we can see that the distribution of learned strides (by EB-LS) are different on SVCD and DSIFN, showing that our method can adaptively handle different style variations.

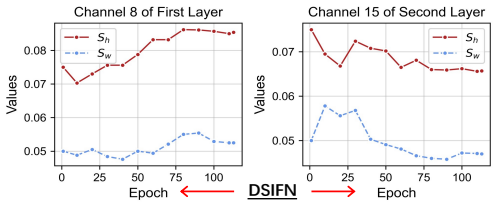

Figure 1. Visualization of different channels of the learned strides on DSIFN dataset.

5. Ablation Studies

In addition to the ablation study on the SVCD dataset in the original text, we additionally provide an ablation study

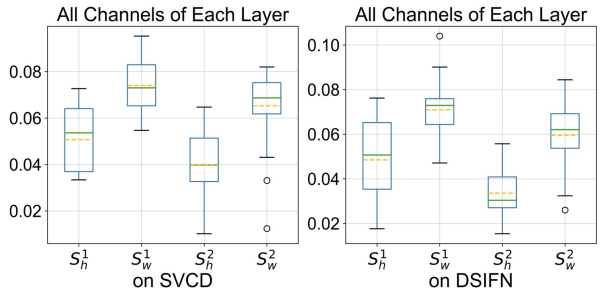

Figure 2. Visualization of distribution of the learned strides on different datasets.

Model	FST	EB-AA	EB-LS	Prec.(%)	Rec.(%)	F1(%)	IoU(%)
Baseline [5]				57.47	67.23	61.97	44.90
FeaSpect-fixed	✓			65.08	68.11	66.56	49.88
FeaSpect-AA	✓	✓		67.76	69.01	68.38	51.95
FeaSpect-LS	✓		✓	68.95	69.83	69.39	53.12
FeaSpect (ours)	✓	✓	✓	**69.47**	**70.27**	**69.87**	**53.69**

Table 1. Ablation studies for each component of our method on DSIFN dataset.

on the DSIFN dataset to verify the effectiveness of the proposed method FeaSpect.

As shown in Table 1, the base model (Baseline) achieves 57.47%, 67.23%, 61.97%, and 44.90% on the four evaluation metrics. Adding the FST with fixed content and stride to the base model (FeaSpect-fixed) surpasses Baseline on four evaluation metrics by 7.61%, 0.88%, 4.59%, and 4.98%, respectively. By adding the extraction box with adaptive attention (EB-AA) based on the FST, FeaSpect-AA further boosts the performance of the model, which improved by 2.68%, 0.90%, 1.82%, and 2.07%. By adding the extraction box with learnable strides (EB-LS), FeaSpect-LS improves the performance to a higher extent than adding the EB-AA, *i.e.*, 3.87%, 1.72%, 2.83%, and 3.24%. Finally, combining the FST with the EB-AA and EB-LS (FeaSpect) achieves the best performance on the four evaluation metrics, *i.e.*, 69.47%, 70.27%, 69.87%, and 53.69%. The above experimental results further prove that the mechanisms proposed in our method are effective and complementary to each other.

6. Comparison of Generalization

To explore our method's ability to learn domain-agnostic features, we conduct experiments in a domain generalization setting. The model trained on the SVCD dataset is directly tested on the SZADA dataset [2] to evaluate its performance. The results are reported in Table 2. Existing methods suffer from severe performance degradation due to domain shift caused by cross-geographic regions. In contrast, our method is relatively less degenerate on domain generalization setting. Compared to existing methods, our method finally achieves the best results on the two evaluation metrics. We consider this to occur because the absence of domain shift between features during training allows the model to focus on learning the essential characteristics (texture and shape, etc.) of the object. This acquired ability greatly assists the model in accurately identifying changed pixels. In addition, the performance degradation of methods based on the transformer backbone is relatively alleviated. We analyze that the powerful ability of large models to capture relationships between features is more conducive to the model learning robust representations.

	FC-EF		FC-Siam-conc		STANet		DSAMNet		ESCNet		Ours (U-Net)	
ConvNet -based	F1(%)	IoU(%)	F1(%)	IoU(%)	F1(%)	IoU(%)	F1(%)	IoU(%)	F1(%)	IoU(%)	F1(%)	IoU(%)
	24.3	13.8	15.4	8.3	14.7	7.9	11.8	6.3	17.5	9.6	**43.2**	**27.6**
	BIT		SwinsUNet		ChangeFormer		-		-		Ours (MiT-B1)	
Transformer -based	F1(%)	IoU(%)	F1(%)	IoU(%)	F1(%)	IoU(%)	-	-	-	-	F1(%)	IoU(%)
	28.1	16.4	29.4	17.2	33.2	19.9	-	-	-	-	**47.0**	**30.7**

Table 2. Comparison results of testing on SZADA dataset using the model trained on SVCD dataset.

7. Comparison on More Datasets

For binary change detection, in addition to the datasets used in the original text, we also provide comparative results on other datasets to fully evaluate our method, including the SYSU-CD [14] and LEVIR-CD [3] datasets. The results are reported in Tables 3 and 4, respectively. As shown in the two tables, our method achieves better performance than existing methods on multiple datasets. We analyze that this significant improvement is contributed by adaptively capturing domain information so that features can be well aligned.

8. Comparisons on More Tasks

To verify the scalability of our method, we compare our method with existing methods on the benchmarks of semantic change detection (SECOND [16] and Hi-UCD [15] datasets) and building damage assessment (xBD dataset [7]), respectively. The results are reported in Table 5. As shown in Table 5, our method significantly improves the baseline (BDANet [13]) and even outperforms the SOTA method (SCanNet [6]), confirming our method's scalability in more tasks. The above analysis and results demonstrate that our method can achieve accurate detection in diverse and challenging scenarios by carefully learning object characteristics without domain shift interference.

9. Comparison and Discussion of Efficiency

To compare the computational efficiency, we record the number of parameters (Params.), floating-point operations per second (FLOPs), training time per epoch (T/E), and inference time (Time) of each method. These metrics reflect the computational efficiency of each method from multiple

Method	Backbone	Pre.(%)	Rec.(%)	F1(%)	IoU(%)
• *ConvNet-based:*					
FC-EF [5]	U-Net*	74.21	79.41	76.72	62.23
FC-Siam-conc [5]	U-Net*	76.47	76.24	76.35	61.75
FC-Siam-diff-res [5]	U-Net*	76.92	79.01	77.96	63.87
FCN-PP [12]	U-Net*	69.81	76.90	73.18	57.71
W-Net [8]	U-Net*	71.08	78.42	74.57	59.45
CDGAN [8]	U-Net*	70.51	79.03	74.53	59.40
STANet [3]	ResNet-18*	70.98	81.21	75.75	60.97
DSAMNet [14]	ResNet-18*	73.93	78.31	76.06	61.36
ESCNet [19]	ResNet-18⋆	80.06	79.15	79.60	66.12
SEIFNet [9]	ResNet-18*	84.02	79.16	81.52	68.80
ChangeSTAR [20]	ResNet-101⋆	82.73	79.61	81.14	68.27
FeaSpect (Ours)	U-Net⋆	**85.70**	**80.64**	**83.09**	**71.08**
FeaSpect (Ours)	ResNet-18⋆	**86.01**	**80.84**	**83.34**	**71.45**
FeaSpect (Ours)	ResNet-101⋆	**86.53**	**81.21**	**83.79**	**72.10**
○ *Transformer-based:*					
BIT [4]	ViTAEv2-S⋆	79.24	76.55	77.87	63.76
VcT [10]	ViTAEv2-S*	84.15	71.76	77.46	63.22
SwinsUNet [18]	Swin-Trans⋆	83.91	72.58	77.83	63.71
ChangeFormer [1]	MiT-B2⋆	84.87	71.05	77.35	63.06
FeaSpect (Ours)	ViTAEv2-S⋆	**86.33**	**81.09**	**83.63**	**71.86**
FeaSpect (Ours)	MiT-B1	**86.41**	**81.17**	**83.71**	**71.98**
FeaSpect (Ours)	MiT-B2⋆	**86.89**	**81.65**	**84.19**	**72.69**
FeaSpect (Ours)	Swin-Trans⋆	**87.10**	**81.94**	**84.44**	**73.07**

Table 3. Comparison results on SYSU-CD dataset. */⋆ defines the backbone model modified in different/same ways.

Method	Backbone	Pre.(%)	Rec.(%)	F1(%)	IoU(%)
• *ConvNet-based:*					
FC-EF [5]	U-Net*	85.83	79.96	82.79	70.64
FC-Siam-conc [5]	U-Net*	85.72	77.15	81.21	68.36
FC-Siam-diff-res [5]	U-Net⋆	89.91	80.20	84.78	73.58
FCN-PP [12]	U-Net*	84.01	75.42	79.48	65.95
W-Net [8]	U-Net*	88.49	85.17	86.80	76.68
CDGAN [8]	U-Net*	89.68	86.01	87.81	78.26
STANet [3]	ResNet-18*	83.92	90.01	86.86	76.77
DSAMNet [14]	ResNet-18*	84.60	89.19	86.83	76.73
ESCNet [19]	ResNet-18⋆	86.12	88.53	87.31	77.48
SEIFNet [9]	ResNet-18*	91.67	88.84	90.23	82.20
ChangeSTAR [20]	ResNet-101⋆	90.72	89.20	89.95	81.74
FeaSpect (Ours)	U-Net⋆	**92.16**	**90.04**	**91.09**	**83.63**
FeaSpect (Ours)	ResNet-18⋆	**92.30**	**90.11**	**91.19**	**83.81**
FeaSpect (Ours)	ResNet-101⋆	**92.59**	**90.28**	**91.42**	**84.20**
○ *Transformer-based:*					
BIT [4]	ViTAEv2-S⋆	89.92	89.06	89.49	80.98
VcT [10]	ViTAEv2-S*	91.89	87.94	89.87	81.61
SwinsUNet [18]	Swin-Trans⋆	90.01	89.61	89.81	81.50
ChangeFormer [1]	MiT-B2⋆	91.56	88.74	90.13	82.03
FeaSpect (Ours)	ViTAEv2-S⋆	**92.43**	**90.19**	**91.30**	**83.99**
FeaSpect (Ours)	MiT-B1	**92.50**	**90.26**	**91.37**	**84.10**
FeaSpect (Ours)	MiT-B2⋆	**92.61**	**90.30**	**91.44**	**84.23**
FeaSpect (Ours)	Swin-Trans⋆	**92.94**	**90.38**	**91.64**	**84.57**

Table 4. Comparison results on LEVIR-CD dataset. */⋆ defines the backbone model modified in different/same ways.

Method	Year	SECOND Dataset			Hi-UCD Dataset			xBD Dataset		
		OA(%)	mIoU(%)	Sek(%)	OA(%)	mIoU(%)	Sek(%)	$F1_b$(%)	$F1_d$(%)	$F1_o$(%)
BDANet	2022	85.7	68.9	18.3	87.9	56.5	22.3	86.4	78.2	80.6
SCanNet	2024	87.6	73.3	23.8	91.7	61.8	27.5	87.3	78.7	81.3
Ours	-	**89.3**	**74.6**	**23.9**	**92.1**	**62.8**	**28.1**	**89.0**	**80.7**	**83.2**

Table 5. $F1_b$: F1-score of building localization. $F1_d$: F1-score of damage classification. $F1_o$: Overall F1-score, $0.3F1_b+0.7F1_d$.

perspectives in time and space. We also calculate the "parameters/time per epoch" (P/T) to represent the efficiency of the model [21], where lower values indicate higher model efficiency. For a fair comparison, all methods are reproduced and tested on the same server, which equipped with an Nvidia Tesla RTX3090 GPU with 24G memory. As shown in Table 6, FC-EF, FC-Siam-conc, and FC-Siam-diff-res have lower trainable parameters and computational cost while their detection accuracy is not impressive. Recent methods STANet, DSAMNet, and ESCNet achieve better accuracy but have higher time and space complexity. In comparison, our method not only achieves the best accuracy but also has the smallest trainable parameters and acceptable computational cost. Importantly, our FeaSpect only costs about 13ms to generate a change map for each image, which is crucial for practical applications.

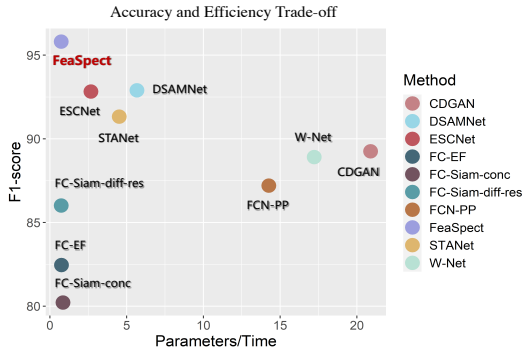

Figure 3. Trade-off between accuracy and efficiency for each method on SVCD dataset.

We plot Fig. 3 to illustrate the trade-off between accuracy and efficiency of the model. P/T is taken as the abscissa and F1-score as the ordinate. A superior accuracy/efficiency trade-off implies a model with higher accuracy and lower efficiency. FC-Siam-diff-res, FC-EF, and FC-Siam-conc methods are close to the left-bottom area in the figure, signifying good efficiency but struggling to achieve sufficient accuracy. ESCNet, STANet, and DSAMNet methods are close to the left-top corner, showcasing better accuracy than other methods but with less efficiency. The proposed FeaSpect stands out in the upper-left corner, attaining the best accuracy/efficiency trade-off among existing methods.

10. Visual Comparison of Detection Results

To further demonstrate the superiority of the proposed method FeaSpect, we provide a visual comparison of detection results. The visual comparisons on the SVCD dataset are demonstrated in Fig. 4 and Fig. 5. As shown in Fig. 4(d)-(i) and Fig. 5(d)-(i), for large objects, small objects and complex scenes, the detection results of existing methods are scattered and their boundaries are not smooth. Among these methods, false detections often occur, and the detected changed objects do not have basic contours. For example, the edges of each car in Fig. 4(d)-(i) are indistinct, *i.e.*, truncated or stitched with other changed areas. In Fig. 5(d)-(i),

	Method	FC-EF	FC-Siam-conc	FC-Siam-diff-res	FCN-PP	W-Net	CDGAN	STANet	DSAMNet	ESCNet	FeaSpect
Train	Params. (MB)	1.35	1.55	1.35	27.81	40.49	115.12	16.93	17.00	5.12	**1.35**
	FLOPs (GB)	**2.68**	4.06	3.50	34.81	94.89	164.74	14.40	37.02	11.65	4.40
	T/E (s)	**180.00**	181.80	182.40	195.00	235.20	551.08	374.40	300.00	192.02	183.00
	P/T (MB/10^2s)	0.75	0.85	0.74	14.26	17.22	20.89	4.52	5.67	2.67	**0.74**
Test	Time (ms)	13.04	13.58	13.56	30.49	16.28	51.91	41.28	58.03	130.97	**13.00**
	F1-score (%)	82.46	80.22	86.02	87.21	88.91	89.26	91.33	92.90	92.83	**95.81**

Table 6. Comparison of computational efficiency of different methods on SVCD dataset. The image input into the model has a size of $256 \times 256 \times 3$. Time is reported by computing the average inference time on 100 randomly selected images.

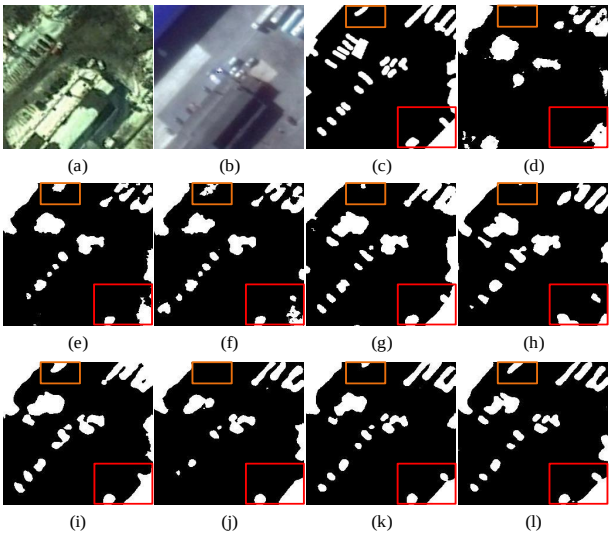

Figure 4. Quantitative comparison results of different methods on the SVCD dataset. (a) Image x_{t_1}. (b) Image x_{t_2}. (c) Ground truth. (d) FC-EF. (e) FC-Siam-conc.(f) FCN-PP. (g) STANet. (h) DSAMNet. (i) ESCNet. (j) Baseline (FC-Siam-diff-res). (k) FeaSpect-fixed. (l) FeaSpect.

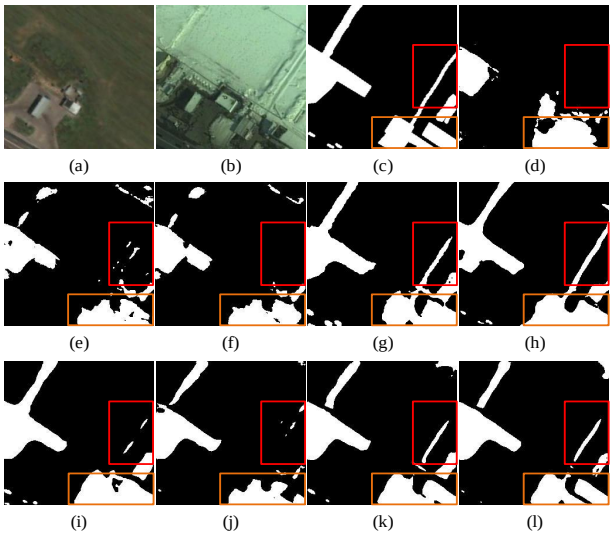

Figure 5. Quantitative comparison results of different methods on the SVCD dataset. (a) Image x_{t_1}. (b) Image x_{t_2}. (c) Ground truth. (d) FC-EF. (e) FC-Siam-conc.(f) FCN-PP. (g) STANet. (h) DSAMNet. (i) ESCNet. (j) Baseline (FC-Siam-diff-res). (k) FeaSpect-fixed. (l) FeaSpect.

the same false detection also occurs for detecting the target object ditch. Furthermore, due to style differences caused by illumination and shadows, the existing methods produce overdetection for those pixels with style shifts. From Fig. 4(d)-(i) and Fig. 5(d)-(i), we can see that all existing methods detect road or grass areas as changed pixels to varying degrees. In contrast, our methods FeaSpect-fixed and FeaSpect achieve the most superior visual results and avoid overdetection of pseudo-changed pixels. Although FeaSpect-fixed produces some missed detections, by further making the stride of the extraction box learnable, FeaSpect accurately detects almost all detailed changes.

11. Presentation of More Generated Data

As mentioned in the original text, existing transformation-based methods utilize generative adversarial networks (GANs) to align the styles of bitemporal images, and these efforts are limited by the complexity of optimizing GANs and the absence of guidance from physical properties, leading to transformed images susceptible to distortion from artifacts. Therefore, in addition to Fig. 1(b) in the original text, we additionally provide more transformed images generated by GANs in Figs. 6-7. As shown in the orange rectangular boxes in Fig. 6-7, there is obvious distortion in the transformed images. This seriously reduces the discriminability of features, thereby affecting the model's accurate detection of changed areas.

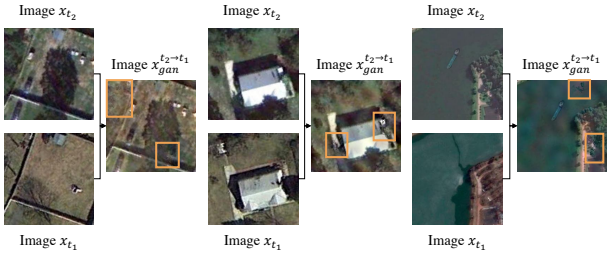

Figure 6. Presentation of data generated by transformation-based methods utilizing generative adversarial networks (GANs).

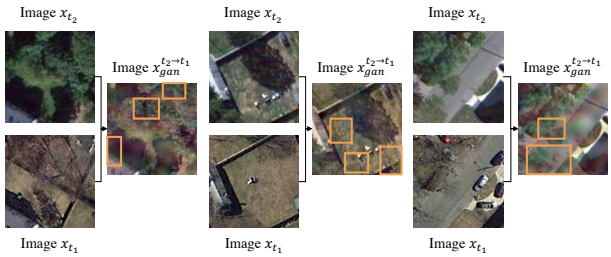

Figure 7. Presentation of data generated by transformation-based methods utilizing generative adversarial networks (GANs).

References

[1] Wele Gedara Chaminda Bandara and Vishal M Patel. A transformer-based siamese network for change detection. In *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 207–210. IEEE, 2022. 4

[2] Csaba Benedek and Tamás Szirányi. Change detection in optical aerial images by a multilayer conditional mixed markov model. *IEEE Transactions on Geoscience and Remote Sensing*, 47(10):3416–3430, 2009. 3

[3] Hao Chen and Zhenwei Shi. A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. *Remote Sensing*, 12(10):1662, 2020. 3, 4

[4] Hao Chen, Zipeng Qi, and Zhenwei Shi. Remote sensing image change detection with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021. 4

[5] Rodrigo Caye Daudt, Bertr Le Saux, and Alexandre Boulch. Fully convolutional siamese networks for change detection. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 4063–4067. IEEE, 2018. 2, 3, 4

[6] Lei Ding, Jing Zhang, Haitao Guo, Kai Zhang, Bing Liu, and Lorenzo Bruzzone. Joint spatio-temporal modeling for semantic change detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 2024. 3

[7] Ritwik Gupta, Bryce Goodman, Nirav Patel, Ricky Hosfelt, Sandra Sajeev, Eric Heim, Jigar Doshi, Keane Lucas, Howie Choset, and Matthew Gaston. Creating xbd: A dataset for assessing building damage from satellite imagery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 10–17, 2019. 3

[8] Bin Hou, Qingjie Liu, Heng Wang, and Yunhong Wang. From w-net to cdgan: Bitemporal change detection via deep learning techniques. *IEEE Transactions on Geoscience and Remote Sensing*, 58(3):1790–1802, 2019. 4

[9] Yanyuan Huang, Xinghua Li, Zhengshun Du, and Huanfeng Shen. Spatiotemporal enhancement and interlevel fusion network for remote sensing images change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2024. 4

[10] Bo Jiang, Zitian Wang, Xixi Wang, Ziyan Zhang, Lan Chen, Xiao Wang, and Bin Luo. Vct: Visual change transformer for remote sensing image change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2023. 4

[11] MA Lebedev, Yu V Vizilter, OV Vygolov, VA Knyaz, and A Yu Rubis. Change detection in remote sensing images using conditional adversarial networks. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42(2), 2018. 2

[12] Tao Lei, Yuxiao Zhang, Zhiyong Lv, Shuying Li, Shigang Liu, and Asoke K Nandi. Landslide inventory mapping from bitemporal images using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 16(6): 982–986, 2019. 4

[13] Yu Shen, Sijie Zhu, Taojiannan Yang, Chen Chen, Delu Pan, Jianyu Chen, Liang Xiao, and Qian Du. Bdanet: Multiscale convolutional neural network with cross-directional attention for building damage assessment from satellite images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021. 3

[14] Qian Shi, Mengxi Liu, Shengchen Li, Xiaoping Liu, Fei Wang, and Liangpei Zhang. A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 2021. 3, 4

[15] Shiqi Tian, Yanfei Zhong, Zhuo Zheng, Ailong Ma, Xicheng Tan, and Liangpei Zhang. Large-scale deep learning based binary and semantic change detection in ultra high resolution remote sensing imagery: From benchmark datasets to urban application. *ISPRS Journal of Photogrammetry and Remote Sensing*, 193:164–186, 2022. 3

[16] Kunping Yang, Gui-Song Xia, Zicheng Liu, Bo Du, Wen Yang, Marcello Pelillo, and Liangpei Zhang. Asymmetric siamese networks for semantic change detection in aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–18, 2021. 3

[17] Chenxiao Zhang, Peng Yue, Deodato Tapete, Liangcun Jiang, Boyi Shangguan, Li Huang, and Guangchao Liu. A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:183–200, 2020. 2

[18] Cui Zhang, Liejun Wang, Shuli Cheng, and Yongming Li. Swinsunet: Pure transformer network for remote sensing image change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022. 4

[19] Hongyan Zhang, Manhui Lin, Guangyi Yang, and Liangpei Zhang. Escnet: An end-to-end superpixel-enhanced change detection network for very-high-resolution remote sensing images. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 4

[20] Zhuo Zheng, Ailong Ma, Liangpei Zhang, and Yanfei Zhong. Change is everywhere: Single-temporal supervised object change detection in remote sensing imagery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15193–15202, 2021. 4

[21] Zhi Zheng, Yi Wan, Yongjun Zhang, Sizhe Xiang, Daifeng Peng, and Bin Zhang. Clnet: Cross-layer convolutional neural network for change detection in optical remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 175:247–267, 2021. 4