scientific reports

Check for updates

OPEN

Enhanced cloud removal via temporal U-Net and cloud cover evolution simulation

Qingwei Tong[1], Leiguang Wang[2,3]✉, Qinling Dai[4], Chen Zheng[5,6] & Fangrong Zhou[7]

Remote sensing images are indispensable for continuous environmental monitoring and Earth observations. However, cloud occlusion can severely degrade image quality, posing a significant challenge for the accurate extraction of ground information. Existing cloud removal techniques often suffer from incomplete cloud removal, artifacts, and color distortions. Owing to the scarcity of sequential data, the effective utilization of temporal information to enhance cloud removal performance poses a challenge. Therefore, we propose a cloud removal method based on cloud evolution simulation. This method is applicable to all paired cloud datasets, enabling the construction of cloud evolution time-series in the absence of actual temporal information. We embed temporal information from the sequence into the Temporal U-Net to achieve more accurate cloud predictions. We conducted extensive experiments on RICE and T-CLOUD datasets. The results demonstrate that our approach significantly improves the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) compared with existing methods.

Keywords Remote sensing image, Cloud removal, Cloud cover evolution (CCE) module, Temporal U-Net, Residual learning

Remote sensing imaging technology plays a pivotal role in Earth observation and environmental science through periodic acquisition of ground object information. However, the presence of cloud cover poses a significant challenge to the quality of remote sensing images. Clouds and haze obscure land surfaces and diminish image clarity, leading to information loss that hampers subsequent image processing tasks. Consequently, there is an urgent need for effective cloud removal techniques.

Remote sensing image cloud removal techniques can be broadly categorized into two groups: those relying on traditional image processing and those utilizing deep learning methods. Traditional image processing-based approaches primarily depend on simplified models or prior knowledge to remove clouds from images[1–4]. However, due to the complexity and variability of real-world environments, methods based on statistical priors cannot effectively tackle all challenges in cloud removal, and their robustness tends to decrease significantly.

In the field of deep learning, cloud removal methods are generally classified into two categories: single-temporal and multi-temporal approaches. Single-temporal deep learning methods primarily rely on atmospheric scattering models, Convolutional Neural Networks (CNNs), ResNet, and other architectures to build cloud removal frameworks. Representative methods include AOD-Net[5], GridDehazeNet[6], DANet[7], two-stage scheme[8], and others. Among multi-temporal methodologies, dual-temporal approaches are the most prevalent, such as pix2pix[9], MLD[10], SpA-GAN[11], Cloud-EGAN[12], MSDA-CR[13], etc. The advantage of these methods lies in the ease of acquiring dual-temporal image pairs, and the relatively simple model architecture, which generally leads to satisfactory cloud removal results. However, their limitations are also apparent, as they rely solely on data from two time points. This restricts their ability to fully capture the long-term trends and dynamic characteristics of cloud changes. As a result, cloud removal performance may be suboptimal in multi-cloud regions or complex scenarios. In recent years, deep learning methods based on true multi-temporal data have gradually gained

[1]College of Big Data and Intelligent Engineering, Southwest Forestry University, Kunming, Yunnan, China. [2]College of Landscape Architecture and Horticulture, Southwest Forestry University, Kunming, Yunnan, China. [3]Key Laboratory of National Forestry and Grassland Administration on Forestry and Ecological Big Data, Southwest Forestry University, Kunming, China. [4]College of Art and Design, Southwest Forestry University, Kunming, Yunnan, China. [5]School of Mathematics and Statistics, Henan University , Kaifeng, China. [6]Henan Engineering Research Center for Artificial Intelligence Theory and Algorithms, Institute of Applied Mathematics, Henan University, Kaifeng, China. [7]Joint Laboratory of power remote sensing technology(Electric Power Research Institute, Yunnan Power Grid Company ltd., China Southern Power Grid), Kunming, Yunnan, China. ✉email: leiguangwang@swfu.edu.cn

attention as a research hotspot. These methods typically utilize a series of temporal images from the same region, aiming to reconstruct multi-cloud areas by leveraging changes in cloud cover over time and across different seasons. Representative methods include STGAN[14], SEN12MS-CR-TS[15], DP-LRTSVD[16], ARRC[17], and others. However, the practical application of these methods remains constrained by the high cost of acquiring real-time series data and the complex design and training requirements of the models.

How to combine the strengths of both dual-temporal and multi-temporal methods to design a cloud removal approach that is both efficient and capable of fully capturing the dynamic features of cloud changes over time is a question worth exploring.

To tackle this challenge, this paper proposes a cloud removal method based on cloud cover evolution simulation. While the method is built on a dual-temporal dataset, it integrates temporal information by constructing a time series, thereby enhancing the model's capacity to capture the dynamic features of cloud changes. Specifically, we introduce the Cloud Cover Evolution (CCE) module and the Temporal U-Net network. The CCE module constructs a sequence of images simulating the temporal changes in cloud cover based on paired datasets. Subsequently, the temporal information from this sequence is embedded into the Temporal U-Net along with the corresponding images. Feature extraction is performed using the T-Res-blocks to achieve precise cloud prediction. Experimental results demonstrate the superior performance of this method in terms of peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM).

Methodology

The whole process of cloud removal is shown in Fig. 1. The core of our method is the Cloud Cover Evolution (CCE) module (as shown in Fig. 1(b)) and the Temporal U-Net network (as shown in Fig. 1(c)).

A. Cloud Cover Evolution (CCE) module

We designate the cloudy images as I_{cloud} and the cloud-free images as I_{clear}. Subsequently, both sets of images are fed into the CCE module. The CCE module enhances cloud removal by simulating the temporal evolution of cloud cover. It generates a series of images reflecting cloud cover changes over time based on paired cloudy and cloud-free images. This process includes image normalization, Mean Square Error (MSE) calculation, and MSE temporal dimension mapping. The MSE between corresponding images is calculated as follows:

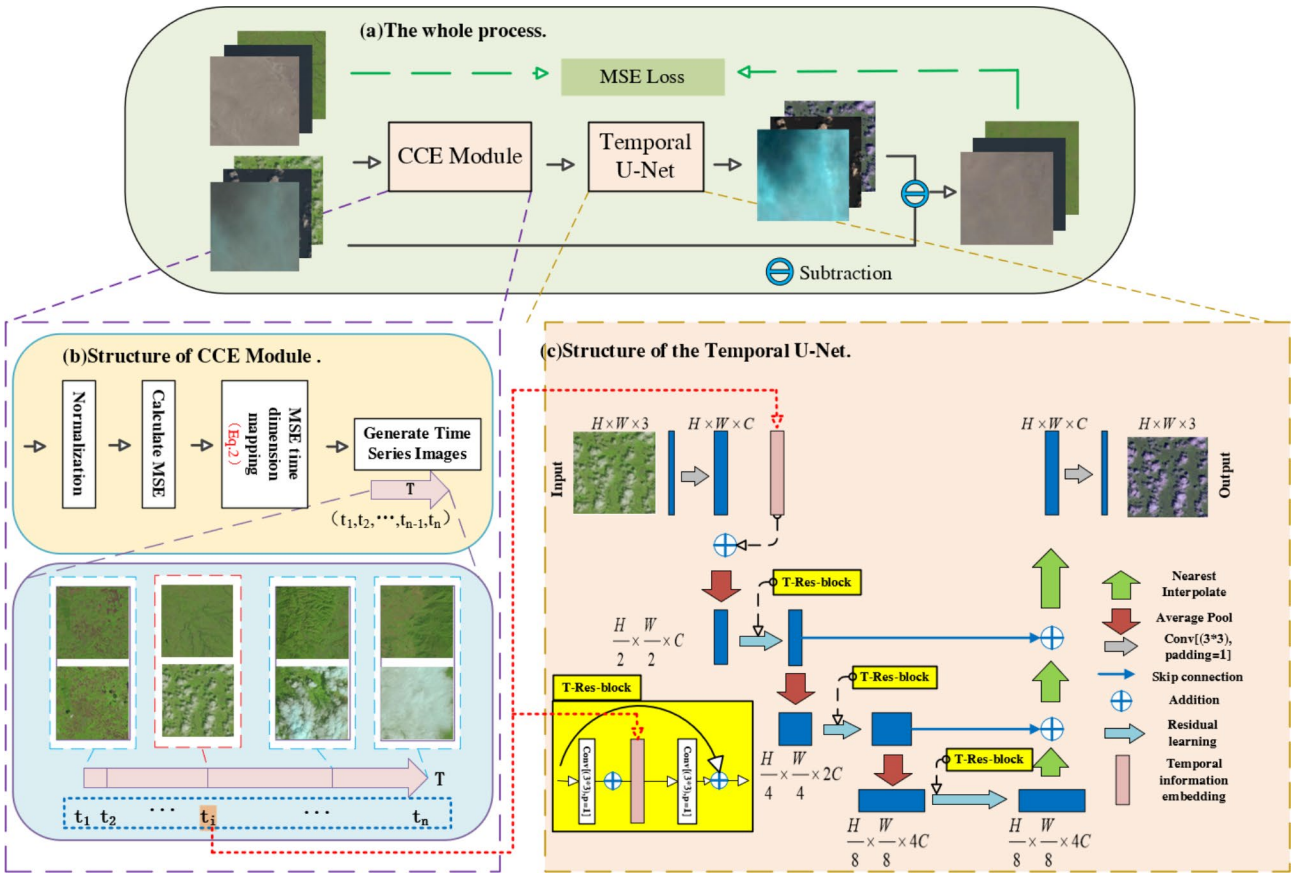

Fig. 1. The process of cloud removal in remote sensing image. (**a**) The whole process. (**b**) Structure of CCE Module. (**c**) Structure of the Temporal U-Net. The red dashed lines describe the position where temporal information is embedded in the model.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (I_{\text{cloud}}(i) - I_{\text{clear}}(i))^2 \tag{1}$$

where N is the number of pixels in the image.

MSE is chosen for its simplicity and effectiveness in quantifying pixel-wise differences between paired cloudy and cloud-free images[18,19]. As a measure of the discrepancy, MSE is sensitive to variations in cloud cover, making it particularly suitable for tracking the temporal dynamics of cloud changes.

MSE has been demonstrated to be an effective metric in many studies of temporal image data[20,21], particularly due to its ability to quantify pixel-wise differences. This advantage makes it highly suitable for modeling the temporal evolution of cloud cover, capturing the transition from partial to full cloud coverage. Such gradual changes are crucial for accurately representing the dynamic characteristics of cloud cover, as they provide essential insights into cloud behavior over time.

After calculating the MSEs of all images, we obtain the MSE value interval $[\,\mathrm{MSE}_{\min},\ \mathrm{MSE}_{\max}]$. We then linearly map all MSE values using Eq. 2, ensuring that the relative differences between MSE values are preserved in the temporal dimension.

This approach helps the model better perceive the temporal dynamics of cloud cover changes during the training process:

$$t = \frac{(\mathrm{MSE} - \mathrm{MSE}_{\min})}{\left(\frac{(\mathrm{MSE}_{\max} - \mathrm{MSE}_{\min})}{\mathrm{timesteps} - 1}\right)} + 1 \tag{2}$$

where $\mathrm{timesteps}$ is the hyperparameter we define to determine the length of the time dimension, which can be selected according to the different data sets. The t represents the temporal information corresponding to each pair of images, which will be embedded in the training process of the images.

B. Temporal U-Net

Temporal U-Net integrates temporal information into its training process to enhance accuracy and robustness. A notable feature is the embedding of temporal information from the CCE module into the initial convolutional layer and three T-Res-blocks. This strategic integration enables the model to better perceive and adapt to variations in cloud cover during feature extraction after down-sampling. The decoder then reconstructs these adapted features into images of the original size. This approach not only facilitates more precise cloud prediction but also enhances overall image quality by effectively reducing cloud artifacts.

The temporal information integration process is designed to inject temporal data into the model, enabling the network to adjust its feature extraction based on different stages of cloud evolution. This process is as follows:

Temporal embedding generation For each pair of input images, we introduce a corresponding temporal input, representing the temporal information of cloud cover evolution. The function transforms to an embedding vector. This embedding allows the model to adjust its processing of image features according to the temporal information (i.e., the cloud coverage at each moment).

Frequency calculation The embedding is constructed using a frequency-based encoding scheme. First, we compute the frequency for each embedding dimension as.

$$\mathrm{freq}_i = exp\left(\frac{-\log\left(\mathrm{max}_{\mathrm{period}}\right) \cdot i}{\frac{dim}{2}}\right),\ for\ i = 0, 1, \ldots, \frac{dim}{2} \tag{3}$$

Here, $\mathrm{max}_{\mathrm{period}}$ is a hyperparameter that controls the maximum frequency, and dim is the dimensionality of the embedding vector. The value of freq_i represents the frequency of each dimension's sinusoidal component.

Temporal information embedding Using the calculated frequencies, we create the temporal embedding vector for each. The embedding is constructed by alternating between cosine and sine functions at the corresponding frequencies.

$$\mathrm{emb}_t = [\cos(\mathrm{freq}_0 \cdot t), \sin(\mathrm{freq}_0 \cdot t), \cos(\mathrm{freq}_1 \cdot t), \sin(\mathrm{freq}_1 \cdot t), \ldots] \tag{4}$$

This vector encodes periodic information that represents the evolution of cloud cover at time t. By combining cosine and sine waves, the embedding captures both the positive and negative cycles, ensuring that the temporal information is effectively represented.

Embedding the feature map Once the temporal embedding is generated, it will be added to the input feature map. The input feature map has the shape of (B, C, H, W), where B is the batch size, C is the number of channels, and H and W are the height and width of the images, respectively. To perform the addition, we first expand the embedding vector to the same shape as the feature map using broadcasting. After this, the result is obtained by adding the two.

$$h = X + \mathrm{emb}_t \tag{5}$$

Here, h is the embedded feature map that now contains temporal information. This addition operation allows the model to dynamically adjust its feature learning based on the different temporal information, thus helping the network effectively track the evolution of cloud cover changes in the cloud removal task.

The model continues training after embedding the temporal information, and its final output is:

$$I_{\text{predicted\_cloud}} = \text{Model}\left(I_{\text{cloud}}, \text{t}\right) \tag{6}$$

where $I_{\text{predicted\_cloud}}$ represents the cloud occlusion map predicted by the model. Consequently, the final recovered cloudless image $I_{\text{recovered\_clear}}$ can be expressed as:

$$I_{\text{recovered\_clear}} = I_{\text{cloud}} - I_{\text{predicted\_cloud}} \tag{7}$$

C. Loss function

We employ the Mean Squared Error (MSE) loss as our loss function to minimize the disparity between the original cloud-free image (I_{clear}) and the final cloud removal image ($I_{\text{recovered\_clear}}$):

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}\left(I_{\text{clear}}(i) - I_{\text{recovered\_clear}}(i)\right)^2 \tag{8}$$

where N is the number of pixels in the image.

Experimental results and discussion

A. Data sets and settings

We choose to utilize the open-source Remote Sensing Image Cloud Removal (RICE)[22]dataset and the T-CLOUD[23] dataset for cloud removal research. The RICE dataset includes two subsets: RICE1 and RICE2. The RICE1 dataset, collected from Google Earth, consists of 500 pairs of 512 × 512 images depicting scenes with thin clouds and cloud-free conditions. The RICE2 dataset, acquired from the Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS) on the Landsat 8 satellite, contains 736 pairs of 512 × 512 images depicting scenes with thick clouds, cloud-free conditions, and cloud mask images. The average cloud cover in the RICE2 dataset images is 24.04%. The T-CLOUD dataset, also acquired from the Landsat 8 satellite, includes 2939 pairs of 256 × 256 images with cloud cover and their corresponding clear images. Sample data from the RICE and T-CLOUD datasets are illustrated in Fig. 2.

We partitioned the dataset into training and test sets using an 8:2 ratio. Specifically, for the RICE1 dataset, 400 pairs of images were randomly selected for training, with the remaining 100 pairs reserved for testing. For the T-CLOUD dataset, 2351 pairs of images were randomly selected for training, leaving 588 pairs for testing. Similarly, for the RICE2 dataset, 588 images were allocated for training, leaving 148 images for testing. To maintain consistency across training, testing, and evaluation phases, all data were resized to 256 × 256 pixels. The proposed method is implemented with PyTorch and trains the model on an NVIDIA RTX A4000 GPU. During training, a batch size of 4 and a learning rate of 1e-4 were employed. Training was conducted for 200 epochs on the RICE1, 300 epochs on the T-CLOUD, and 350 epochs on the RICE2 dataset.

To measure the quality of the generated cloud-removed images and evaluate the cloud removal ability of the proposed method, we utilize two widely used image quality assessment metrics: Peak Signal-to-Noise Ratio (PSNR)[24]and Structural Similarity Index (SSIM)[25].

B. Sensitivity Analysis of 'Timesteps'

In this section, we explore the impact of the timesteps hyperparameter on model performance. This hyperparameter is crucial for defining the temporal context of the model, influencing its ability to perceive and model temporal variations.

We conducted a series of experiments where the model was trained using different timesteps values, while keeping other hyperparameters constant, and evaluated their impact on performance metrics. Table 1 presents

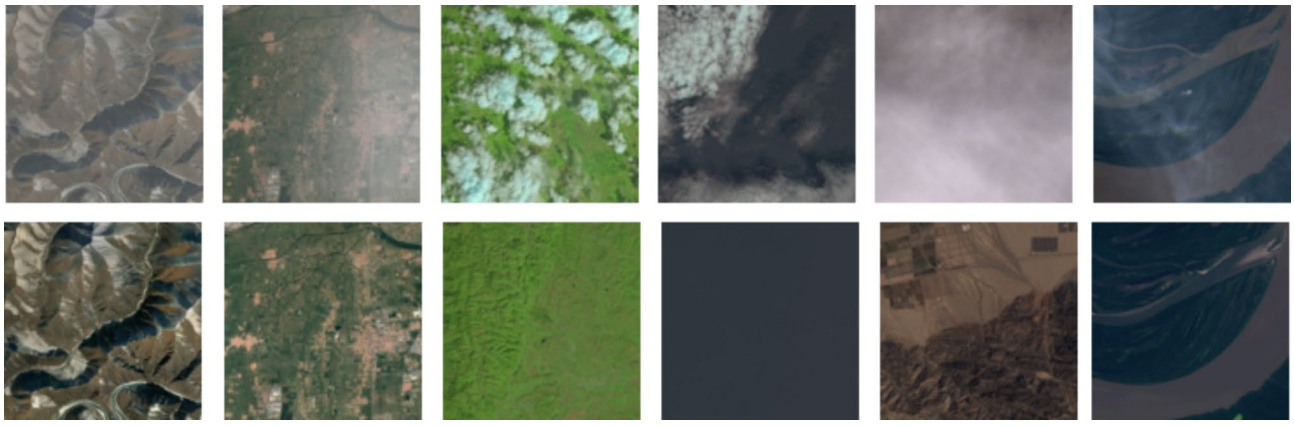

Fig. 2. Typical image samples from RICE and T-CLOUD. The first line shows images with cloud coverage, and the second line shows images without clouds. The left two columns are from RICE1, the middle two are from RICE2, and the right two are from T-CLOUD.

Timesteps	RICE1		RICE2		T-CLOUD	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
50	33.917	0.9689	35.517	0.9536	29.195	0.9296
200	33.832	0.9691	35.346	0.9535	28.994	0.9287
400	**34.277**	**0.9693**	35.532	0.9549	29.251	0.9301
600	34.183	0.9686	**35.587**	**0.9558**	**29.325**	**0.9312**
800	34.104	0.9691	35.562	0.9537	28.972	0.9286

Table 1. Impact of timesteps hyperparameter variation on model performance. Bold indicates the best result.

Variants	PSNR			SSIM		
	RICE1	T-CLOUD	RICE2	RICE1	T-CLOUD	RICE2
Standard U-Net	25.51	24.04	27.282	0.9381	0.8235	0.8913
Standard U-Net with Res-blocks	33.443	26.9	32.947	0.9528	0.907	0.9392
Temporal U-Net with T-Res-blocks	**34.277**	**29.325**	**35.587**	**0.9693**	**0.9312**	**0.9558**

Table 2. Results of the ablation experiment on RICE1, RICE2 and T-CLOUD. Bold indicates the best result.

Variants	Training time (h)	Inference time(s)	FLOPs(G)	Parameters(M)
Standard U-Net	3.1	0.0107	24.64	1.09
Standard U-Net with Res-blocks	33.4	0.0793	361.5	22
Temporal U-Net with T-Res-blocks	36.7	0.0824	362.31	24.26

Table 3. Comparison of running time and parameter complexity in ablation experiments. Training time and inference time describe the runtime; FLOPs and parameters describe the model's complexity.

the results of these experiments, showing the impact of different timesteps values on model performance. We observed that the timesteps parameter significantly affects the PSNR metric, while its impact on the SSIM metric is relatively minor. This suggests that selecting an appropriate timesteps value can enhance the model's ability to generate high-quality reconstructed images. Based on our findings, choosing a timestep value of 400 or 600 appears to be a better option because too small a value prevents the model from capturing sufficient temporal information, while too large a value introduces information redundancy and misses important details.

C. Ablation experiment

To investigate the contribution of temporal information embedding to the model, we designed the following variants: standard U-Net, standard U-Net with conventional res-blocks, and Temporal U-Net with T-Res-blocks. Evaluation of the ablation results continues to be based on PSNR and SSIM metrics.

Table 2 presents the quantitative results of ablation experiments on the three datasets. It is evident that our proposed method achieves the best performance in terms of PSNR and SSIM. This is attributed to the effective utilization of temporal information by the model, which enhances its capability in cloud prediction and removal. Table 3 shows a comparison of the running time and parameter complexity of different variants in the ablation study on the T-CLOUD dataset. Although our method increases the runtime and computational overhead, further analysis shows that the contribution of the temporal information embedding to FLOPs is minimal, at only 0.81 FLOPs (G). This is because the temporal information, input as a scalar, is used to generate the embedding vector, which is then added to the feature map. This results in very low computational complexity, almost negligible when compared to the overall FLOPs(G). However, the embedding of temporal information significantly improves the model's accuracy, demonstrating its effectiveness.

Figure 3 provides a qualitative comparison of the ablation results, clearly demonstrating that embedding temporal information enhances the detail and color fidelity of the cloud removal images, making them highly similar to the reference cloud-free images. To further visually demonstrate the impact of temporal information embedding on thick cloud removal, we present the model's predicted cloud-free images, as shown in Fig. 4. Clearly, before embedding the temporal information, the model could only remove partial cloud cover, resulting in a still blurry image. However, after embedding the temporal information, the model can identify and remove almost all cloud cover and its shadows, producing nearly perfect cloud-free images. Additionally, we observe that the inclusion of temporal information has a more pronounced effect on thick cloud removal compared to thin clouds. Thin clouds are often partially removed even without the temporal layer due to their semi-transparent nature, but their complete removal becomes more robust with temporal information embedding. For thick clouds, which are opaquer and challenging to handle, temporal information provides critical context,

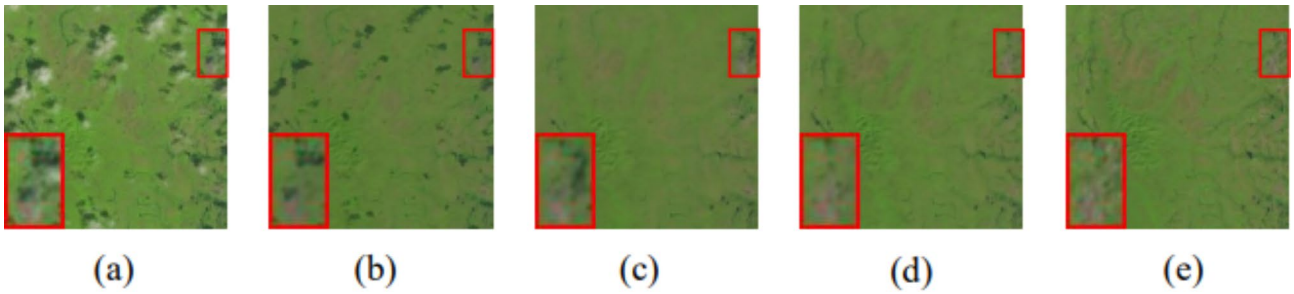

Fig. 3. Visual comparison of ablation experimental results. (**a**)Cloudy image. (**b**) Standard U-Net model only. (**c**) Standard U-Net model with res-blocks. (**d**) Temporal U-Net with T-Res-blocks. (**e**) Cloud-free image. The red bounding boxes highlight the magnified detailed features.

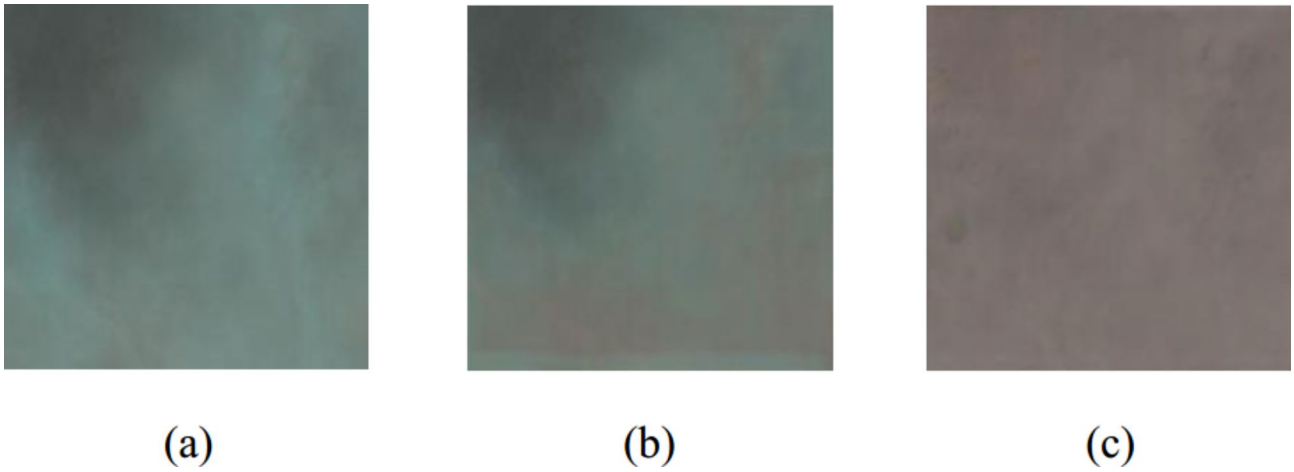

Fig. 4. Comparison of temporal information embeddings in thick cloud occlusion removal. (**a**) Thick cloud image. (**b**) Standard U-Net model with res-blocks. (**c**) Temporal U-Net with T-Res-blocks.

enabling the model to distinguish between clouds and the underlying surface more effectively. This highlights the importance of the temporal structure in improving the model's ability to handle various cloud types.

D. Qualitative and quantitative evaluations

Our method was compared with seven other approaches on the three datasets: RICE1, RICE2, and T-CLOUD. Among the comparison methods, DCP[3]and IDeRS[4]are traditional approaches, while GridDehazeNet[6], SpA-GAN[11], pix2pix[9], CVAE[23], and STGAN[14] are deep learning-based methods. The quantitative results for STGAN are quoted from other scholars' studies on the same dataset.

The qualitative results for the RICE1 and T-CLOUD test sets are shown in Fig. 5. Traditional methods fail to remove all clouds, resulting in unclear images and significant color distortion. Although SpA-GAN, pix2pix, GridDehazeNet, and CVAE can remove thin clouds, they still exhibit noticeable color distortion. In contrast, our method achieves lower spectral distortion and higher structural similarity.

Since traditional methods struggle with thick cloud removal, we focus on comparing the deep learning methods. The qualitative results for the RICE2 test set are shown in Fig. 6. Removing thick clouds is challenging, and pix2pix and SpA-GAN show limited success in removing large clouds and accurately reconstructing ground objects. CVAE effectively removes most thick clouds but leaves behind remnants and artifacts. GridDehazeNet performs well in eliminating thick clouds, but its reconstructions are overly smooth. Our proposed method, however, provides more realistic reconstructions while completely removing thick clouds.

Table 4 presents the quantitative results for all eight methods on the RICE1, RICE2, and T-CLOUD datasets. It is worth noting that the images in the RICE1 and RICE2 datasets mainly cover geographical scenes such as grasslands, oceans, and deserts, with relatively simple geographic features and uniform lighting conditions. As a result, the performance of models on these two datasets is generally good, with high metric values. On the other hand, in the T-CLOUD dataset, the images have darker lighting conditions and more diverse geographical scenes, which increases the complexity of cloud removal. As a result, the performance of various models is relatively poor on this dataset. Nevertheless, our method outperforms all other methods across all three datasets, highlighting the effectiveness of utilizing temporal information and the robustness of our approach.

Table 5 compares the runtime and parameter complexity of several deep learning methods on the T-CLOUD dataset. Although our method does not have a significant advantage in training time, FLOPs, or parameter count

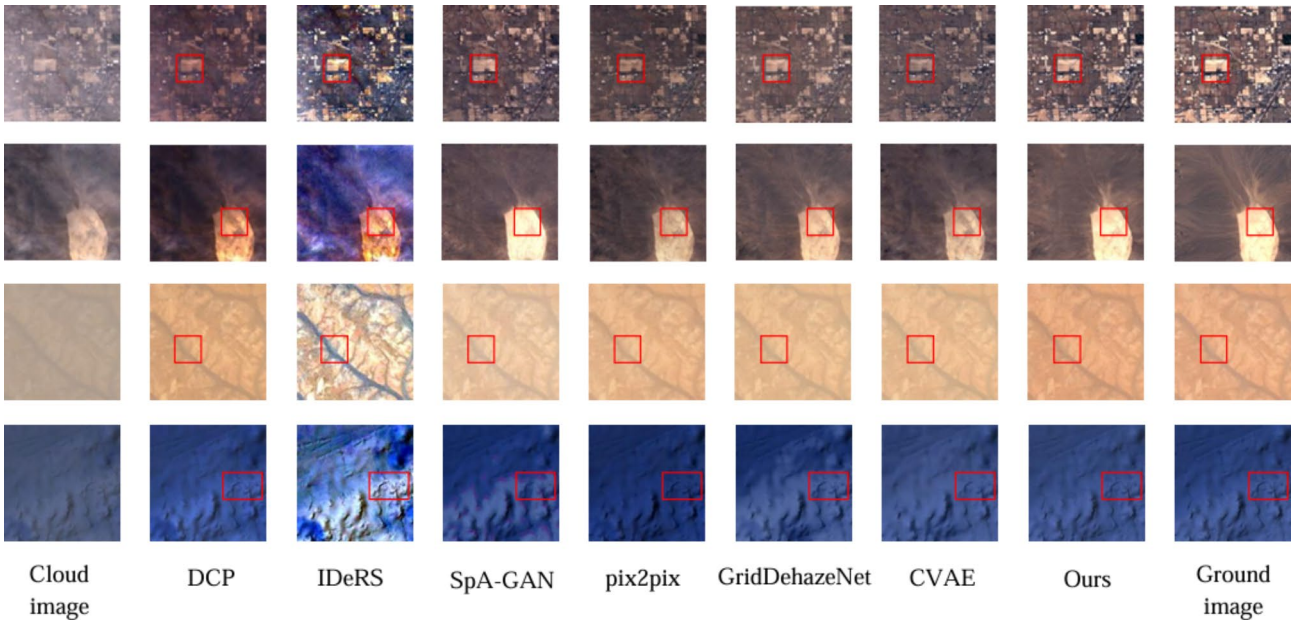

Fig. 5. Cloud removal effects of different methods on RICE1 and T-CLOUD test sets. The top two rows are from T-CLOUD, and the bottom two rows are from RICE1. The red bounding boxes highlight areas with significant differences, making it easier for readers to observe the differences between methods.

compared to other methods, it achieves a shorter inference time. Overall, our method trades longer training time and higher parameter complexity for improved cloud removal performance. Additionally, we calculated the computational cost of incorporating temporal information and found that the additional FLOPs (G) are only 0.81. This indicates that embedding temporal information can be easily extended to other datasets and models without introducing significant computational overhead.

E. Discussion

Based on the comparative experimental results mentioned above, our model demonstrates superior overall performance compared to current state-of-the-art methods. Clearly, it achieves the highest PSNR and SSIM values across all three datasets. Furthermore, the results from ablation experiments show that embedding temporal information into the model significantly enhances cloud removal and improves the quality of the resulting cloud-free images. By incorporating temporal information, the model accurately captures the dynamic features of cloud evolution over time, which enhances its ability to detect complex cloud cover patterns.

However, our method has certain limitations. First, it is currently only applicable to a bi-temporal cloud removal dataset and does not incorporate data from more than two time points. Therefore, future work could explore the inclusion of additional temporal data to further improve cloud removal performance. Second, while our method uses MSE-based temporal mapping to simulate the temporal evolution of cloud coverage, which is simple and effective, it may not fully capture all image differences, particularly those related to visual appearance. Thus, future research could consider using other metrics to describe image differences for temporal mapping, such as SSIM (Structural Similarity Index) or NCC (Normalized Cross-Correlation). Additionally, our current study is based on pixel-level analysis, but investigating the frequency domain could also provide valuable insights for further improving the method. Moreover, our model does not currently have an advantage in terms of complexity, with relatively higher training time and parameter count compared to other methods. Future work could explore lightweight optimization strategies to reduce computational cost and parameter complexity, making the method more efficient and scalable for practical applications.

Conclusion

In this article, we propose a cloud removal method based on cloud cover evolution simulation. This method constructs time-series images through the CCE module and embeds temporal information into the Temporal U-Net, thereby enhancing the model's ability to accurately estimate and generate cloud cover images. Finally, the predicted cloud cover image is subtracted from the original cloud image to obtain a clear restored image. Extensive experimental results demonstrate that our method is highly effective in removing both thin and thick clouds.

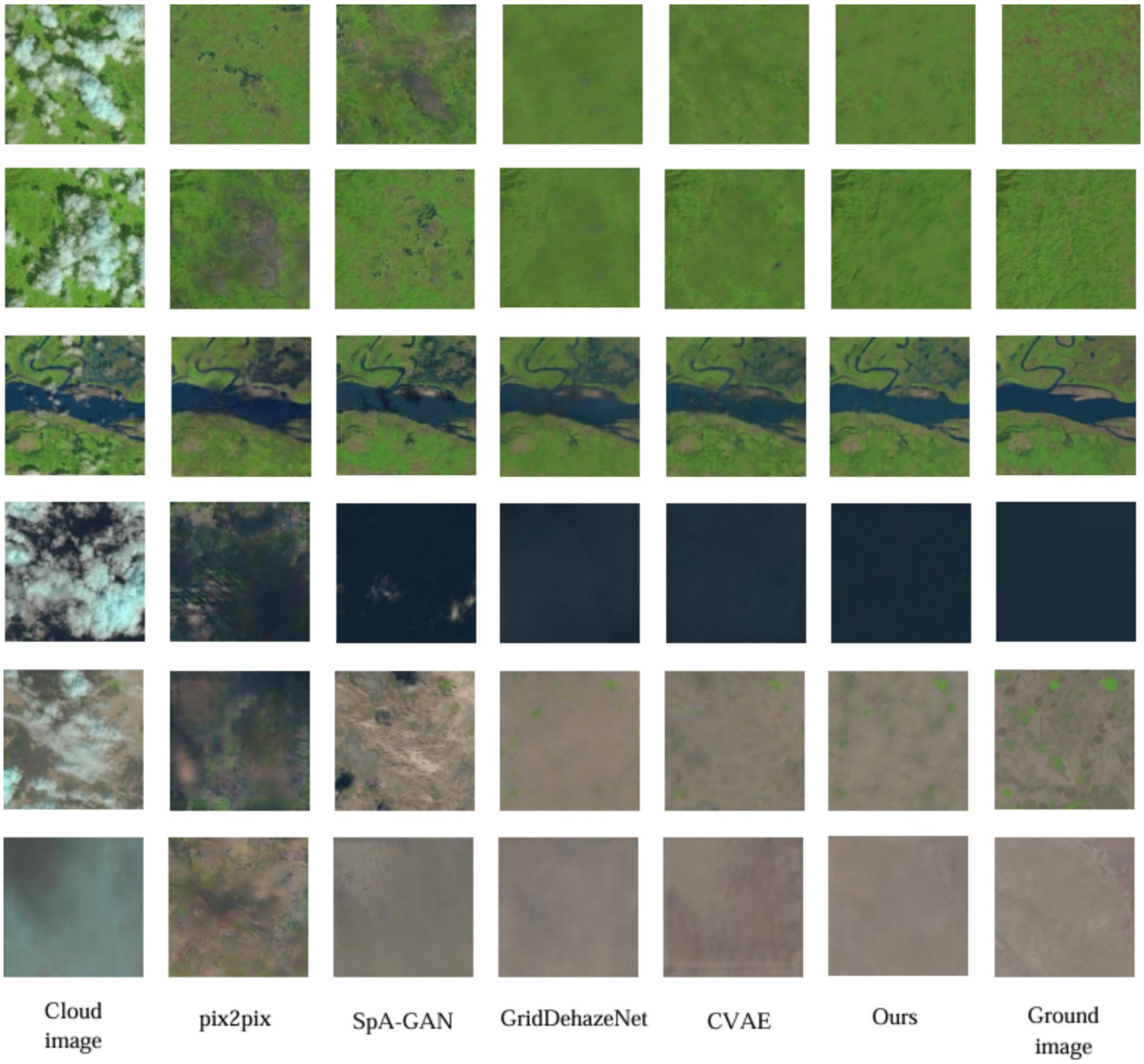

Fig. 6. Cloud removal effects of different methods on the RICE2 test set.

Method	RICE1		RICE2		T-CLOUD	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
IDeRS	14.28	0.5324	-	-	13.042	0.4302
DCP	20.563	0.8927	-	-	20.5	0.7976
SpA-GAN	29.749	0.9541	28.144	0.9035	25.371	0.8604
STGAN [14]	31.565	0.961	29.868	0.928	-	-
CVAE	*33.898*	0.9581	34.691	0.9008	27.82	0.8394
pix2pix	33.069	0.9633	25.177	0.8864	25.097	0.8859
GridDehazeNet	33.067	*0.967*	*34.557*	*0.9546*	*28.006*	*0.9203*
Proposed	**34.277**	**0.9693**	**35.587**	**0.9558**	**29.325**	**0.9312**

Table 4. Quantitative comparison of RICE1, RICE2 and T-CLOUD by different methods. Use bold and underline for best and sub-best performance, respectively.

Method	Training time (h)	Inference time(s)	FLOPs(G)	Parameters(M)
SpA-GAN	17.1	0.178	67.94	1.22
CVAE	28.6	0.283	185.736	15.28
pix2pix	13.3	0.072	18.15	54.41
GridDehazeNet	7.5	0.191	97.28	0.956
Proposed	36.7	0.0824	362.31	24.26

Table 5. The comparison of runtime and parameter complexity of different methods on the T-CLOUD dataset. Training time and inference time describe the runtime; FLOPs and parameters describe the model's complexity.

Data availability

The codes are available at https://github.com/Tongqingw/Temporal-U-net.git.

Received: 20 September 2024; Accepted: 17 January 2025
Published online: 06 February 2025

References

1. Mitchell, O. R., Delp, E. J. & Chen, P. L. Filtering to remove cloud cover in satellite imagery. *IEEE Trans. Geoscience Electron.* **15**, 137–141. https://doi.org/10.1109/TGE.1977.6498971 (1977).
2. Liu, J. et al. Thin cloud removal from single satellite images. *Opt. Express.* **22**, 618–632. https://doi.org/10.1364/OE.22.000618 (2014).
3. He, K., Sun, J. & Tang, X. Single image haze removal using Dark Channel Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 2341–2353. https://doi.org/10.1109/TPAMI.2010.168 (2011).
4. Xu, L. et al. IDeRs: iterative dehazing method for single remote sensing image. *Inf. Sci.* **489**, 50–62. https://doi.org/10.1016/j.ins.2019.02.058 (2019).
5. Li, B., Peng, X., Wang, Z., Xu, J. & Feng, D. in *IEEE International Conference on Computer Vision (ICCV)*. 4780–4788. (2017).
6. Liu, X., Ma, Y., Shi, Z. & Chen, J. in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 7313–7322.
7. Chen, Y., Cai, Z., Yuan, J. & Wu, L. A novel dense-attention network for thick cloud removal by reconstructing semantic information. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* **16**, 2339–2351 (2023).
8. Zheng, J., Liu, X. Y. & Wang, X. Single image cloud removal using U-Net and generative adversarial networks. *IEEE Trans. Geosci. Remote Sens.* **59**, 6371–6385 (2020).
9. Isola, P., Zhu, J. Y., Zhou, T. & Efros, A. A. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.
10. Xu, M., Jia, X., Pickering, M. & Plaza, A. J. Cloud removal based on sparse representation via multitemporal dictionary learning. *IEEE Trans. Geosci. Remote Sens.* **54**, 2998–3006 (2016).
11. Pan, H. Cloud Removal for Remote Sensing Imagery via Spatial Attention Generative Adversarial Network. *arXiv e-prints*, arXiv:13015, (2009). https://doi.org/10.48550/arXiv.2009.13015 (2020).
12. Ma, X., Huang, Y., Zhang, X., Pun, M. O. & Huang, B. Cloud-egan: rethinking cyclegan from a feature enhancement perspective for cloud removal by combining cnn and transformer. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* **16**, 4999–5012 (2023).
13. Yu, W., Zhang, X. & Pun, M. O. Cloud removal in optical remote sensing imagery using multiscale distortion-aware networks. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5 (2022).
14. Sarukkai, V., Jain, A., Uzkent, B. & Ermon, S. Cloud Removal in Satellite Images Using Spatiotemporal Generative Networks. arXiv *arXiv preprint arXiv:1912.06838*. (2019).
15. Ebel, P., Xu, Y., Schmitt, M. & Zhu, X. X. SEN12MS-CR-TS: a remote-sensing data set for multimodal multitemporal cloud removal. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–14 (2022).
16. Zhang, Q., Yuan, Q., Li, Z., Sun, F. & Zhang, L. Combined deep prior with low-rank tensor SVD for thick cloud removal in multitemporal images. *ISPRS J. Photogrammetry Remote Sens.* **177**, 161–173 (2021).
17. Cao, R., Chen, Y., Chen, J., Zhu, X. & Shen, M. Thick cloud removal in landsat images based on autoregression of Landsat time-series data. *Remote Sens. Environ.* **249**, 112001 (2020).
18. Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**, 600–612 (2004).
19. Tan, H. L., Li, Z., Tan, Y. H., Rahardja, S. & Yeo, C. A perceptually relevant MSE-based image quality metric. *IEEE Trans. Image Process.* **22**, 4447–4459 (2013).
20. Ren, Y., Ye, J., Wang, X., Xiao, F. & Liu, R. SAM-Net: spatio-temporal sequence typhoon cloud image prediction net with self-attention memory. *Remote Sens.* **16**, 4213 (2024).
21. Hoque, M. R. U. et al. An arithmetic deep model for temporal remote sensing image fusion. *Remote Sens.* **14**, 6160 (2022).
22. Lin, D. et al. A Remote Sensing Image Dataset for Cloud Removal. *arXiv e-prints*, arXiv:00600, (1901). https://doi.org/10.48550/arXiv.1901.00600 (2019).
23. Ding, H., Zi, Y. & Xie, F. in *Computer Vision – ACCV 2022.* (eds Lei Wang et al.) 52–68 (Springer Nature Switzerland).
24. Huynh-Thu, Q. & Ghanbari, M. Scope of validity of PSNR in image/video quality assessment. *Electron. Lett.* **44**, 800–801 (2008).
25. Horé, A. & Ziou, D. in *20th International Conference on Pattern Recognition*. 2366–2369. (2010).

Acknowledgements

This work was supported in part by the Major scientific and technological projects of Yun-nan Province under Grant 202202AD080010; in part by the National Natural Science Foundation of China under Grants 32160369, 41961053 and 31860182; and in part by Ten Thousand Talents Program" Special Project for Young Top-notch Talents of Yunnan Province under grant YNWR-QNBJ-2019-026.

Author contributions

Conceptualization, Q.T and L.W; methodology, Q.T and L.W; formal analysis, Q.T and L.W; writing—original draft, Q.T; article review, L.W, C.Z, F.Z and Q.D; visualization, Q.T.; All authors have read and agreed to the

published version of the manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to L.W.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.