TempFormer: Temporally Consistent Transformer for Video Denoising

Mingyang Song[1,2], Yang Zhang[2], and Tunç O. Aydın[2]

[1] ETH Zurich, Switzerland
[2] DisneyResearch|Studios, Switzerland
misong@student.ethz.ch, {yang.zhang,tunc}@disneyresearch.com

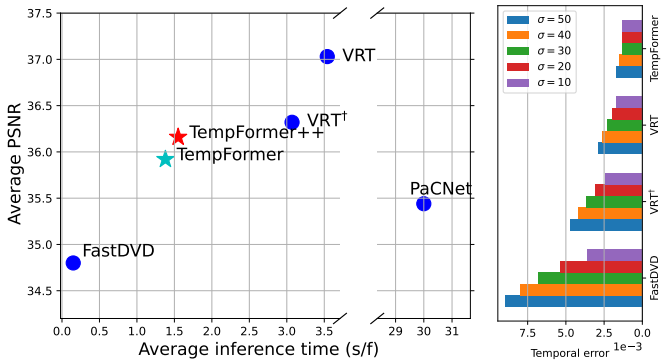


Fig. 1: Left: PSNR (averaged results on noise levels $\sigma = \{10, 20, 30, 40, 50\}$) Vs. Inference time on 480P video sequences. Right: Temporal consistency, the lower the better.

Abstract. Video denoising is a low-level vision task that aims to restore high quality videos from noisy content. Vision Transformer (ViT) is a new machine learning architecture that has shown promising performance on both high-level and low-level image tasks. In this paper, we propose a modified ViT architecture for video processing tasks, introducing a new training strategy and loss function to enhance temporal consistency without compromising spatial quality. Specifically, we propose an efficient hybrid Transformer-based model, *TempFormer*, which composes Spatio-Temporal Transformer Blocks (STTB) and 3D convolutional layers. The proposed STTB learns the temporal information between neighboring frames implicitly by utilizing the proposed *Joint Spatio-Temporal Mixer* module for attention calculation and feature aggregation in each ViT block. Moreover, existing methods suffer from temporal inconsistency artifacts that are problematic in practical cases and distracting to the viewers. We propose a sliding block strategy with recurrent architecture, and use a new loss term, *Overlap* Loss, to alleviate the flickering between adjacent frames. Our method produces state-of-the-art spatio-temporal denoising quality with significantly improved temporal coherency, and requires less computational resources to achieve comparable denoising quality with competing methods (Figure 1).

Keywords: Video denoising, Transformer, Temporal consistency

1 Introduction

A major challenge in video processing is efficiently utilizing temporal redundancy. Early methods utilize filtering by explicitly computing spatial and temporal similarity between pixels [17]. Since the emergence of deep learning convolutional neural networks (CNN) have replaced traditional patch-based non-local filtering. One approach for matching pixels temporally is to explicitly align the pixels through optical flow or deformable convolutional networks (DCN) [6, 14, 29]. There are also works [18, 21, 24] that avoid flow estimation and only use the capability of CNNs to extract temporal information implicitly.

Transformer networks were initially used in natural language processing [23], and more recently have shown promising performance in vision tasks due to the mechanism of global attention (GA). GA is affordable in some high-level vision tasks, such as object detection and classification [5, 10]. However, GA is a severe burden to GPU memory in video denoising tasks, especially when processing high-resolution videos, and the inference speed is unreasonable in practical applications. Swin Transformer [16] computes attention inside non-overlapping spatial windows, and uses shifted windows to extract different patches in each stage to introduce interactions between adjacent windows. Recently, a Transformer-based vision restoration model [14] extrapolates the spatial self attention mechanism within a single image to temporal mutual attention mechanism between adjacent frames. The mutual attention mechanism introduces many additional matrix multiplications, and therefore is inefficient during inference. While ViT needs much fewer parameters compared with CNNs thanks to the content-dependent attention map, it requires larger GPU memory during training.

To introduce interaction between frames inside models, existing methods mainly use temporal sliding windows, that divide a video sequence into blocks with or without overlappings [21,24]. While two neighboring blocks share several common input frames, the denoised frames still contain temporal inconsistency artifacts. Another strategy is the recurrent architecture [18], which has to load a large number of frames in one training step and is inefficient during training.

In this work we propose a model that we call *TempFormer*, which builds upon the Swin Transformer [16]. Our model does not require optical flow and uses the capacity of content-dependent mixer, attention mechanism, with MLP layers to integrate temporal information implicitly. TempFormer only contains explicit spatial attention, and has good efficiency during training and inference. Moreover, we combine the sliding block strategy with recurrent architecture to strengthen the interaction between temporal blocks, and introduce a new loss term to alleviate the incoherency artifacts.

2 Related Work

2.1 Image Denoising

Classical image denoising methods utilize the spatial self-similarity of images. The similarities serve as weights of a filter, and the denoising process is a

weighted average of all pixel values within a patch centered at the reference pixel. The Non-Local-Means filter [3] is a famous implementation of such an idea. Recently, the deep-learning based image denoising methods bypass the explicit similarity computation and use the neural network's powerful representation ability to integrate the reference pixel and its spatial neighborhoods [4, 15, 32–34].

2.2 Video Denoising

In video denoising, sequences are treated as volumes, and the non-linear functions are applied to the noisy pixels and their spatio-temporal neighborhoods. The traditional method VBM4D [17] groups similar volumes together and filters the volumes along four dimensions. Vaksman et al. [22] re-explores the patch-based method, uses K-Nearest Neighbor (KNN) to find all similar patches in the volume for each reference patch, then stacks them together as a kind of data augmentation. In modern deep learning based models, the temporal alignment is performed by optical flow or DCN [29], or their combination [14]. There are also some works that avoid the expensive flow estimation and warping, and use the powerful representation ability of CNNs to perform end-to-end training [21, 24].

2.3 Temporal Consistency

The removal of temporal flickering is a common challenge in artistic vision tasks, e.g. colorization, enhancement and style transfer, etc. Some blind methods use an extra model as post-processing on the outputs which were processed frame by frame. Lai et al. [11] proposed a recurrent model that takes frames before and after the processing as inputs and use Temporal Loss to enforce consistency. Lei et al. [12] divide the temporal inconsistency into two types, unimodal and multimodal inconsistency, and proposed a solution to each of them. These existing methods heavily rely on post-processing and lack of efficiency during practical applications. Besides, few attentions were focused on the inconsistency artifacts in the video denoising tasks. The temporal flickering is distracting to the viewers when the noise level is high, especially in the static contents of the video.

2.4 Vision Transformer

Vision Transformer (ViT) has shown promising performance on both high-level vision tasks, such us object detection [8, 16, 25, 28], classification [9, 31], and low-level vision tasks, such as image restoration [7, 15]. Liu et al. [16] proposed a new backbone for vision tasks, SWin Transformer, that divides image into non-overlapping spatial windows to solve the problem of quadratic computation complexity and uses shifted windows strategy to introduce the connection between neighboring windows. Based on SWin Transformer, Liang et al. [15] applied this new backbone on image restoration tasks. Yang et al. [26] introduced a multi-scale architecture and mix the features with multiple granularities to realize long-range attention. A variation of self-attention (SA) calculation was

proposed in [1, 30], where a Cross-Covariance attention operation was applied along the feature dimension instead of token dimension in conventional transformers. This modification resulted in linear complexity w.r.t the number of tokens, allowing efficient processing of high-resolution images. Recently, some methods transfer the attention mechanism from the spatial domain to the temporal domain on some video recognition tasks [2, 13]. Liang et al. [14] use temporal mutual attention on video restoration tasks as a type of soft warping after motion estimation. The extension from spatial attention to temporal attention is a natural extrapolation, but the boost in the number of attention maps makes training prohibitively expensive when the memory of GPU is limited, and this high computational complexity makes it less practical.

3 Method

Our model is a one-stage model and performs spatial and temporal denoisng simultaneously. For efficiency and temporal coherency reasons our model outputs more than one neighboring frames, namely takes $2 \times m + 1$ frames as inputs and outputs $2 \times n + 1$ frames. This strategy can be described in the following form:

$$\{\hat{I}^t_{-n}, \hat{I}^t_{-n+1}, ..., \hat{I}^t_0, ..., \hat{I}^t_{n-1}, \hat{I}^t_n\} = \Phi(\{\tilde{I}^t_{-m}, \tilde{I}^t_{-m+1}, ..., \tilde{I}^t_0, ..., \tilde{I}^t_{m-1}, \tilde{I}^t_m\}), \quad (1)$$

where $\tilde{I}$ represents the noisy frame of the temporal window $Block^t$, Φ is our video denoising model, and $\hat{I}$ represents the denoised frame of $Block^t$. To introduce communications between neighboring temporal blocks, we set m strictly larger than n so that they share multiple common input frames. We use the setting $m = 2, n = 1$ throughout the rest formulas and visualizations in this paper.

Our training pipeline contains two phases. Firstly, we use TempFormer to perform denoising within each temporal $Block^t$, where the input frames of $Block^t$ can extract information in the Spatio-Temporal Transformer Blocks (STTB). Secondly, to solve the flickering between adjacent temporal blocks, we finetune our network to the recurrent architecture, and propose a new loss term to strengthen stability further. See Section 3.2 for more detailed information.

3.1 Spatio-Temporal Video Denoising Phase

Overall Architecture. Figure 2 shows the architecture of our model, which is mainly composed of four modules: Wavelet Transform, shallow feature extraction, deep feature extraction, and the image reconstruction module. Firstly, we use Wavelet Transform to decompose the input frames and concatenate all sub-bands along the channel dimension. Secondly, a 3D convolutional layer converts the frequency channels of all sub-bands into shallow features. Next, the deep feature extraction module, which consist of a sequence of Spatial-Temporal Transformer Blocks (STTB), mixes the features of each token spatially and temporally. Following the STTBs, another 3D convolutional layer transforms the features back into the wavelet frequency space. Finally, we use the Inverse Wavelet Transform to convert the frequency sub-bands into high-quality images with the original resolution.

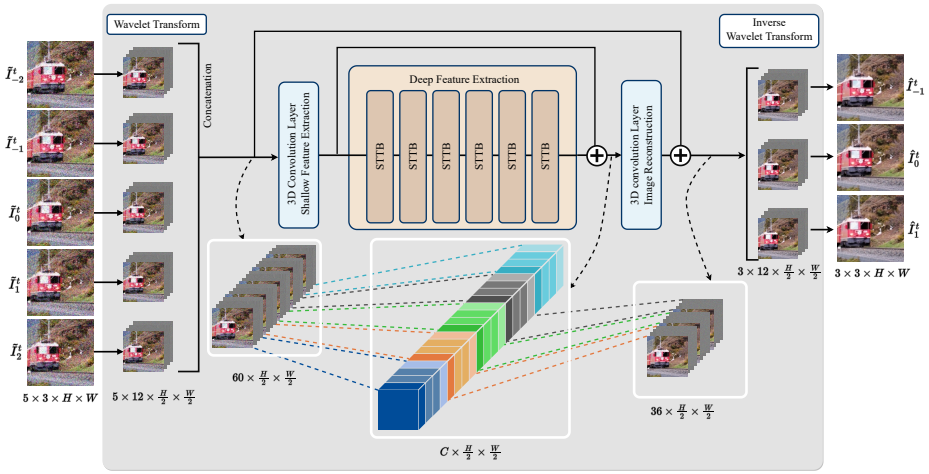


Fig. 2: The architecture of the proposed TempFormer.

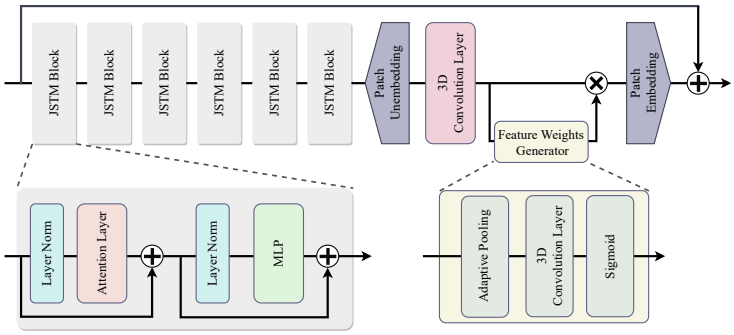


Fig. 3: The architecture of the Spatial-Temporal Transformer Block (STTB).

Spatio-Temporal Transformer Block. The architecture of the proposed STTB module is shown in Figure 3. In Liu et al. [16], the attention layers in SWin Transformer blocks perform spatial mixing followed by feature mixing. In our model, the attention layers perform spatial and temporal mixing jointly, which we call Joint Spatial-Temporal Mixer (JSTM). Inspired by the Residual SWin Transformer Block (RSTB) [15], we use a sequence of JSTMs followed by a convolutional layer at the end to extract deep features. A 3D convolutional layer has been employed to further enhance the temporal feature fusion between neighboring frames. Ghosting artifacts mitigation is challenging for all video processing tasks. We introduce a Feature Weights Generator module within STTB, which consists of an Adaptive pooling, 3D convolutional layer and Sigmoid activation for learning the weight of each feature in channel dimension.

Joint Spatio-Temporal Mixer. Computing GA between all pixels of the video sequence is impractical and unnecessary. Since the channel dimension contains the features from different frames, we follow the method described in SWin Transformer [16], by dividing the input images into several non-overlapping spatial windows with the size $w \times w$. The attention layer of ViT can be interpreted as a spatial tokens mixer, where weights for each token are content-dependent [15]. Moreover, as described in [27], the attention layers can also mix channels. As such, in our method temporal mixing is performed when generating the Queries, Keys and Values from the feature of the tokens, as follows:

$$Q = XP_Q, \quad K = XP_K, \quad V = XP_V, \tag{2}$$

where c is the number of feature channels of a frame, $X \in \mathbb{R}^{w^2 \times 5c}$ is the features of all frames before mixing, $\{P_Q, P_K, P_V\} \in \mathbb{R}^{5c \times 5d}$ are the linear projections that project the features into $\{Q, K, V\} \in \mathbb{R}^{w^2 \times 5d}$. Because we concatenate all input frames along the channel dimension, each $\{\mathbf{q}_{i,j}, \mathbf{k}_{i,j}, \mathbf{v}_{i,j}\} \in \mathbb{R}^{5d}$ integrates the features of all frames at spatial position (i, j), namely $\mathbf{x}_{i,j} \in \mathbb{R}^{5c}$. This process can be described as:

$$\mathbf{q}_{i,j} = \mathbf{x}_{i,j}P_Q, \ \mathbf{k}_{i,j} = \mathbf{x}_{i,j}P_K, \ \mathbf{v}_{i,j} = \mathbf{x}_{i,j}P_V, \tag{3}$$

$$\mathbf{q}_{i,j} = cat[\mathbf{q}_{i,j}^{I_{-2,\dots,2}}], \ \mathbf{k}_{i,j} = cat[\mathbf{k}_{i,j}^{I_{-2,\dots,2}}], \ \mathbf{v}_{i,j} = cat[\mathbf{v}_{i,j}^{I_{-2,\dots,2}}], \tag{4}$$

where $n \in \{-2, -1, 0, 1, 2\}$, and $\{\mathbf{q}_{i,j}^{I_n}, \mathbf{k}_{i,j}^{I_n}, \mathbf{v}_{i,j}^{I_n}\} \in \mathbb{R}^c$ is the *query*, *key* and *value* of the token in frame n with spatial position (i, j).

Since the motions introduce offsets between pairing pixels in different frames, resulting that the mixing only along the channel dimension is far from enough to integrate the temporal information (as described in Eq 3). We apply the following spatial mixing which aggregates all the spatial and temporal information to a reference token $\mathbf{y}_{i',j'}^{I_n}$ at spatial location (i', j') of frame I_n ($\langle \cdot\ , \cdot \rangle : V \times V \to \mathbb{R}$):

$$\mathbf{y}_{i',j'}^{I_n} = \sum_{i=1,j=1}^{i=w,j=w} \frac{\langle \mathbf{q}_{i',j'}^{I_n}, \mathbf{k}_{i,j}^{I_n} \rangle}{norm_{i',j'}^{I_n}} \mathbf{v}_{i,j}^{I_n}, \quad norm_{i',j'}^{I_n} = \sum_{i=1,j=1}^{i=w,j=w} \langle \mathbf{q}_{i',j'}^{I_n}, \mathbf{k}_{i,j}^{I_n} \rangle. \tag{5}$$

The spatio-temporal mixing function (Eq 3-4) of attention layer can be visualized in Figure 4. Above formulas written in matrix form is one of the computations of the attention mechanism in ViT, but we expand the calculation for spatio-temporal feature fusion:

$$\text{Attention}(Q, K, V) = \text{SoftMax}(\frac{QK^T}{\sqrt{D}} + bias)V, \tag{6}$$

where D is the features length of each token, in our case $D = 5d$, and a trainable relative position *bias*, which can increase the capacity of the model [16]. The following MLP layers in each JSTM act as a temporal mixer. Before feeding the tokens to the next STTB, we use a 3D convolutional layer followed by Feature Weights Generator module to extract features further. The end-to-end connection of the STTBs aggregates multiple spatial and temporal mixers together.

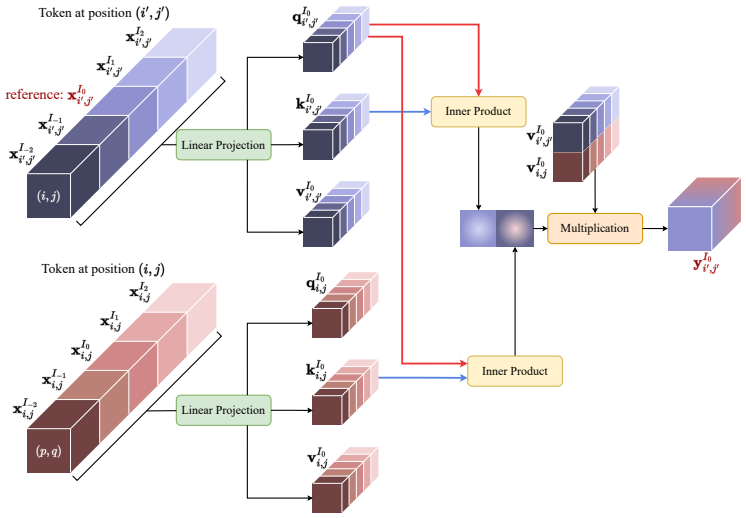


Fig. 4: The visual description of attention layer as implicit temporal mixer. The query ($\mathbf{q}^{I_0}_{i',j'}$), key ($\mathbf{k}^{I_0}_{i',j'}$) and value ($\mathbf{v}^{I_0}_{i',j'}$) of the reference token $\mathbf{x}^{I_0}_{i',j'}$ integrate the features of all frames at position (i', j'). In like manner, the query ($\mathbf{q}^{I_0}_{i,j}$), key ($\mathbf{k}^{I_0}_{i,j}$) and value ($\mathbf{v}^{I_0}_{i,j}$) of the token $\mathbf{x}^{I_0}_{i,j}$ integrate the features of all frames at position (i, j). The attention between $\mathbf{x}^{I_0}_{i',j'}$ and $\mathbf{x}^{I_0}_{i,j}$ fuses the features of all frames at both position (i', j') and (i, j), which performs the spatio-temporal fusion.

Wavelet Decomposition. The size of the attention map, SoftMax($QK^T/\sqrt{D}$+ $bias$) is $w^2 \times w^2$, and is the bottleneck of the inference speed. Inspired by Maggioni et al. [18], we use Wavelet Transform to halve the resolution to make training and inference more efficient. The reduced resolution enables much longer feature embeddings, which is beneficial for the performance of our network. See more comparisons and discussions in the ablation studies in Section 4.3.

With the proposed STTB and JSTM module, the spatio-temporal attention can be calculated and learned efficiently. Our proposed model achieved good spatial quality, and the output frames from one $Block^t$ are temporally stable. However, the temporal coherency between adjacent frames that come from neighboring blocks is not as good as those from the same block. As shown in an example Figure 5, we compute residual images of three consecutive denoised frames from $Block^t$ and $Block^{t+1}$, where larger value means higher difference between consecutive frames. We describe the solution in Section 3.2.

3.2 Temporal Coherency Enhancement (TCE) Phase

Recurrent Architecture. To improve the coherency between temporal blocks, we propose a recurrent architecture for fine-tuning the network and add a new loss term to alleviate flickering further. Despite the $2(m - n)$ common input frames shared in two adjacent blocks, the noise in the remaining $2n + 1$ input

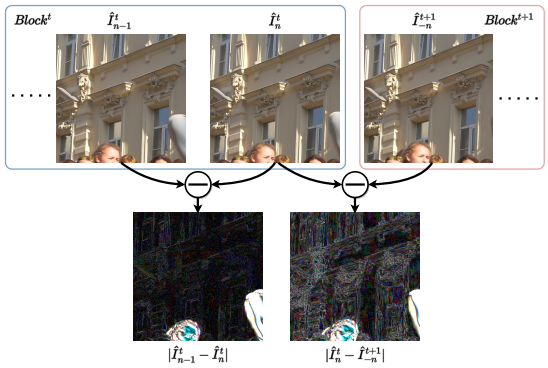


Fig. 5: The visualization of the inconsistency between adjacent frames from one temporal block (left) and from two neighboring temporal blocks (right).

frames vary in each block, which is the root cause of the incoherency. We modify our model into the recurrent architecture to enforce the connection between two adjacent blocks, namely the first input frame of the $Block^{t+1}$ is the last output frame of the $Block^t$, which can be described as:

$$Block^{t+1} : \{\hat{I}^{t+1}_{-1}, \hat{I}^{t+1}_0, \hat{I}^{t+1}_1\} = \Phi(\{\hat{I}^t_1, \tilde{I}^{t+1}_{-1}, \tilde{I}^{t+1}_0, \tilde{I}^{t+1}_1, \tilde{I}^{t+1}_2\}). \tag{7}$$

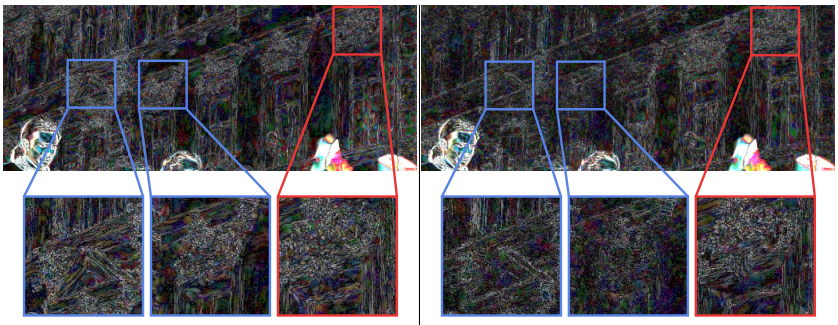
The recurrent architecture spreads the information of all frames from the current $Block^t$ to the $Block^{t+1}$ by propagating one denoised frame of $Block^t$ to the first input frame of the $Block^{t+1}$. Such recurrent architecture enhances the connection between neighboring temporal blocks and achieves better temporal consistency, as is shown in Figure 6.

The substitution of the first noisy input frame with the denoised one provides a solid prior knowledge to each block. However, the reconstruction errors can also propagate to the following blocks. Moreover, the dynamic contents and the static contents with periodical occlusions (e.g., the reliefs which the legs of the dancer sweep over, shown in the blue rectangles of Figure 6) are still temporally inconsistent. We describe the solution in the next section.

Overlap Loss. To solve the inconsistency of the dynamic contents, we modify the stride when dividing the video sequence so that the neighboring temporal blocks share $2(m-n)+1$ common input frames. Most importantly, an overlapping exists in the output frames of the neighboring blocks. The last output frame of $Block^t$, namely $\hat{I}^t_n$, and the first output frame of $Block^{t+1}$, namely $\hat{I}^{t+1}_{-n}$, should be the same image. Following this idea, we introduce a new loss term as follows:

$$\mathcal{L}^t_{overlap} = |\hat{I}^t_n - \hat{I}^{t+1}_{-n}|, \tag{8}$$

where $\mathcal{L}^t_{overlap}$ is the $l1$ loss between the last output frame of $Block_t$ and the first output frame of $Block_{t+1}$. The total loss $\mathcal{L}_{total}$ is composed of two parts, the



(a) before the recurrent fine-tune (b) after the recurrent fine-tune

Fig. 6: Comparison of the residual figures ($|I^t_{-n} - I^{t+1}_n|$) before and after the recurrent fine-tuning. The contents on the top left corner (blue squares) are static throughout the whole video sequence. For these contents, the temporal consistency is enhanced compared with those without the recurrent fine-tuning. However, the static parts with periodical occlusions (red square) and dynamic regions have limited improvement.

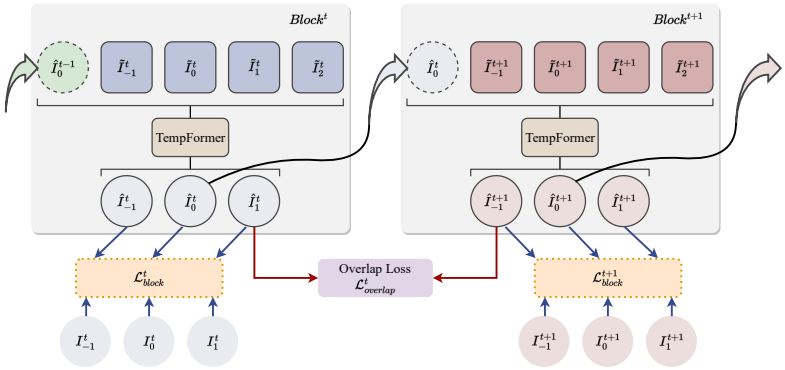


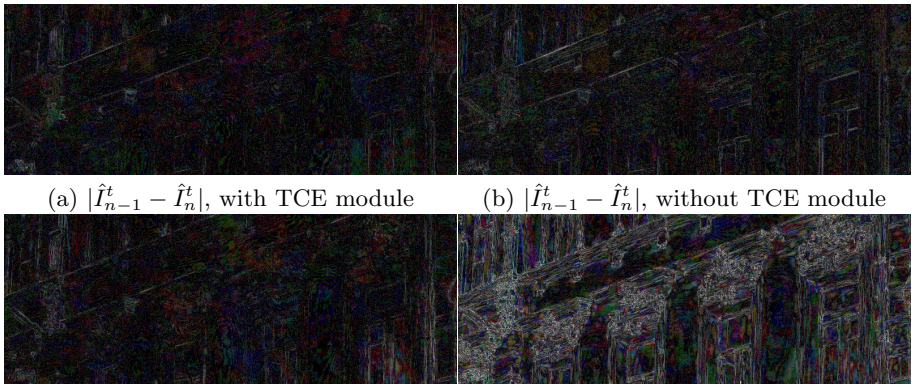
Fig. 7: Illustration of the Recurrent architecture and the Overlap Loss.

first part $\mathcal{L}^t_{block}$ is the loss between the denoised frames $\hat{I}$ and the ground truth I for each temporal block, and the second part is the Overlap Loss $\mathcal{L}^t_{overlap}$. We use a hyper parameter α to balance the spatial loss and the temporal loss, which is shown in the following formula:

$$\mathcal{L}^t_{block} = \frac{1}{2n+1} \sum_{i=-n}^{n} |\hat{I}^t_i - I^t_i|, \tag{9}$$

$$\mathcal{L}_{total} = \frac{1}{T} \sum_{t=0}^{T} \mathcal{L}^t_{block} + \alpha \frac{1}{T-1} \sum_{t=0}^{T-1} \mathcal{L}^t_{overlap}, \tag{10}$$

where T is the index of the temporal blocks in the sequence. Figure 7 shows overview of the recurrent architecture and loss functions. The fine-tuned model



(a) $|\hat{I}^t_{n-1} - \hat{I}^t_n|$, with TCE module (b) $|\hat{I}^t_{n-1} - \hat{I}^t_n|$, without TCE module

(c) $|\hat{I}^t_n - \hat{I}^{t+1}_{-n}|$, with TCE module (d) $|\hat{I}^t_n - \hat{I}^{t+1}_{-n}|$, without TCE module

Fig. 8: Comparison of the residual figures with and without the TCE module. Top: residual figures between adjacent frames in one temporal $Block^t$. Bottom: residual figures between adjacent frames from two neighboring $Block^t$ and $Block^{t+1}$.

achieved promising temporal stability, which is shown in Figure 8. The temporal consistency between neighboring blocks has significant improvement (Fig. 8(c), (d)), but the coherency between neighboring frames within each block also becomes better (Fig. 8(a), (b)). Compared with the recurrent model without $\mathcal{L}_{overlap}$, the dynamic contents and the static contents with periodical occlusions are also as stable as the ones that are static throughout the sequence.

4 Experiments

4.1 Experimental setup

Training Strategy. For the temporal sliding windows strategy, we input five neighboring frames and let the model predict three neighboring frames in the middle. During Spatio-Temporal Video Denoising Phase, we process one temporal block in each training step. During Temporal Coherency Enhancement Phase, instead of loading several blocks in one training step, we only load two neighboring blocks($Block^0$ and $Block^1$). For the first temporal block, we substitute the first noisy input frame with the corresponding ground truth to simulate the recurrent architecture. Following our design, we replace the second temporal block's first input frame ($\tilde{I}^1_{-2}$) with the second output frame of the first temporal block ($\hat{I}^0_0$), and add the Overlap Loss to the common output frames ($\hat{I}^0_1$ and $\hat{I}^1_{-1}$).

Datasets. Following the previous works [14,21,22], we use DAVIS 2017 dataset [19] (480P) as training and testing set for qualitative and quantitative evaluations. We train a non-blind model on five noise levels ($\sigma = \{10, 20, 30, 40, 50\}$).

	σ	FastDVD [21]	PaCNet [22]	VRT [14](†)	TempFormer	TempFormer++
DAVIS	10	39.07	39.97	40.82(40.42)	39.97	40.17
	20	35.95	36.82	38.15(37.49)	37.10	37.36
	30	34.16	34.79	36.52(35.73)	35.40	35.66
	40	32.90	33.34	35.32(34.47)	34.16	34.42
	50	31.92	32.30	34.36(33.47)	33.20	33.44
Set8	10	36.27	37.06	37.88(37.62)	36.97	37.15
	20	33.51	33.94	35.02(34.59)	34.55	34.74
	30	31.88	32.05	33.35(32.82)	33.01	33.20
	40	30.73	30.70	32.15(31.57)	31.86	32.06
	50	29.83	29.66	31.22(30.61)	30.96	31.16
Time* (s/f)		0.15	30	3.54(3.07)	1.38	1.55
Time§ (s/f)		0.68	-	17.67(16.16)	5.88	6.78

Table 1: Quantitative comparison with existing methods on DAVIS [19] and Set8 [20]. The best and second best methods are written in red and blue colors separately. Comparison of inference time per frame (s/f) on resolution of 480P (*) and 1080P videos (§) respectively. The fastest and second fastest methods are in red and blue respectively. VRT [14] uses temporal block size 12. VRT† [14] uses temporal block size 5.

4.2 Results

Spatial Accuracy. To compare quantitative results, we use PSNR as the evaluation metric. Inspired by [14], we propose another version of TempFormer with optical flow and warping on RGB space, which we call TempFormer++. Instead of warping both images and features, we only warp RGB images as data augmentation. In this way, there is no change to the architecture of our model. Table 1 reports the average PSNR for each noise level on the Test-Dev 2017 [19] 480P and Set8 [20]. We use the spatial tiling size 128×128 for both VRT [14] and our model, and adjust the other configurations so that both models fully utilize the GPU memory. We use RTX 3090ti as the testing device for evaluating the inference time on 480P and 1080P videos from DAVIS 2017 dataset respectively, and the comparison is shown in last two rows of Table 1.

Since the computation of the attention in ViT is the most expensive module of inference time. With our proposed method, we avoid the time consuming temporal mutual attention calculation and use the proposed STTB and JSTM modules to integrate temporal information implicitly. As a result, our model required approximately 40% inference time compared with VRT [14] with comparable spatial denosing quality on PSNR evaluation, as shown in Table 1. We report temporal consistency evaluation in Section 4.2 and Table 2.

Figure 9 shown the qualitative comparison of the results with the existing methods. As shown in the examples, our method produced comparable and even better visual quality to the state-of-the-art (SOTA) method. Note that, on noise level $\sigma = 30$ and $\sigma = 50$ (top and bottom row of Figure 9), TempFormer restored more detailed pattern and sharper edges than VRT [14] and TempFormer++,

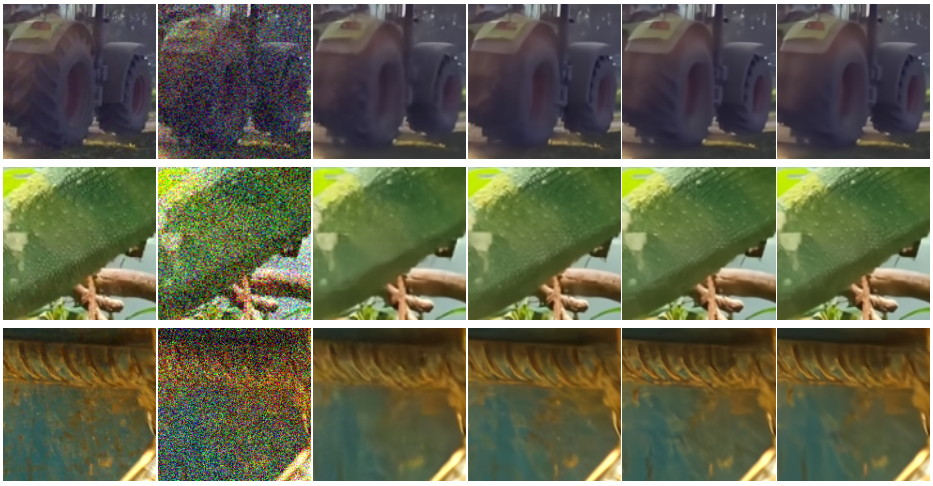


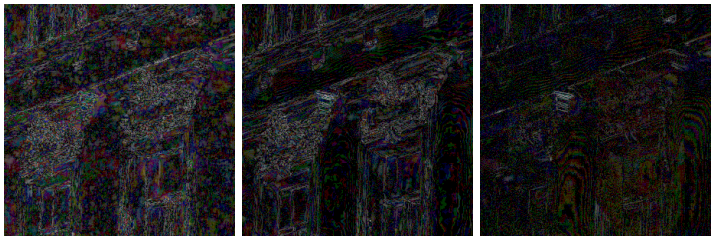
Fig. 9: Visual comparison with other methods. Top row: $\sigma = 30$, middle row: $\sigma = 40$, bottom row: $\sigma = 50$. (a) ground truth. (b) noisy. (c) FastDVD [21]. (d) VRT [14]. (e) TempFormer. (f) TempFormer++.

which indicated that the failure of optical flow estimation on higher noise level content could produce negative impact on the spatial accuracy.

Temporal Consistency. We qualitatively demonstrate the performance of the temporal consistency by visualizing the residual images of adjacent denoised frames' static region, which is shown in Figure 10. For a better quantitative comparison (without being influenced by the temporal artifacts in the original dataset), we create a toy video sequence where each frame is identical, and add noise with different random seed for each frame. In this toy sequence, the ground truth of the residual image between adjacent denoised frames is zero everywhere. We estimate the mean absolute error (MAE) between the adjacent output frames denoised by different methods respectively, as shown in Table 2. As long as the whole video sequence is processed block by block, our TCE strategy can alleviate the temporal flickering between frames and neighboring blocks, which resultant significantly better temporal consistency performance than the existing methods. As illustrated in Figure 10 and Table 2, the temporal consistency of our model outperforms all other state-of-the-art methods, especially under high noise levels. More visual results are provided in the videos contained in the supplementary.

4.3 Ablation Studies

Impact of the channel length and Wavelet Decomposition. In Transformer models, the length of channel is the main factor that effects the performance. We trained our model with two types of configurations where the



FastDVD [21] VRT [14] TempFormer

Fig. 10: Visualization of the residual figure on the static contents from video *berakdance* in DAVIS 2017 [19] 1080P (the reliefs in frame 39 and frame 40). The noise level is $\sigma = 30$. The residual figure of VRT [14] is computed on the junction of two temporal blocks.

σ	FastDVD [21]	VRT† [14]	VRT [14]	TempFormer
10	3.6×10^{-3}	2.4×10^{-3}	1.7×10^{-3}	1.3×10^{-3}
20	5.4×10^{-3}	3.1×10^{-3}	2.0×10^{-3}	1.3×10^{-3}
30	6.8×10^{-3}	3.7×10^{-3}	2.3×10^{-3}	1.3×10^{-3}
40	8.0×10^{-3}	4.2×10^{-3}	2.6×10^{-3}	1.5×10^{-3}
50	9.0×10^{-3}	4.7×10^{-3}	2.9×10^{-3}	1.7×10^{-3}

Table 2: Quantitative comparison of temporal consistency. We use one frame from video *skatejump* in DAVIS 2017 [19] 480P to create the toy sequence. VRT [14] uses temporal block size 12. VRT† [14] uses temporal block size 5.

channel length is 40 (small model) and 120 (large model) per frame respectively. We also trained a TempFormer without Wavelet decomposition to evaluate its impact. Table 3 shows the performance of the models and inference time, which demonstrates the effectiveness of the hyperparameter of the channel length and Wavelet decomposition in our model.

With the Wavelet decomposition, the number of the attention maps is reduced to 1/4 compared with the model without decomposition. As demonstrated in Table 3, in spite of the negative impact on the spatial performance, its boost in inference speed is evident. On the other hand, the halving in spatial resolution allows us to boost in channel length of the model, which achieves good balance between model capacity and efficiency, as shown in the last column of the Table 3. The temporal constraints requires larger capacity of the model, so the model with longer channel has better temporal consistency than the shorter ones, as demonstrated in the comparison at the last row of this table.

Wavelet Transform VS. Pixel Shuffle. Other than the Wavelet Transform, there are some other kinds of decomposition methods that can halve the input resolution. We tested Pixel Shuffle, and Table 4 shows the comparison. Since Wavelet Transform separates low and high frequency sub-bands (horizontal and vertical edges) of the images, and preserves image information better than Pixel

	w/o Wavelet	w/ Wavelet	
model size	S	S	L
PSNR	36.51	36.34	36.59
Inference time(s/f)	4.21	1.43	3.75
Temporal Consistency	✓	✓	✓✓

Table 3: Impact of the channel length and Wavelet Decomposition, tested on *breakdance* 1080p in DAVIS 2017 [19] with noise level $\sigma = 30$. S(small model): the length of channel is 40 per frame. L(large model): the length of channel is 120 per frame. Settings: the size of each tile is 128×128, process 8 tiles per batch. (For the PSNR and inference time comparison we use the models before the Temporal Coherency Enhancement Phase.)

σ	Wavelet	Pixel Shuffle
10	39.82	39.78
20	36.85	36.81
30	35.10	35.06
40	33.85	33.81
50	32.89	32.84

Table 4: Quantitative comparison between different decomposition methods: Wavelet Transform and Pixel Shuffle. (For this comparison we use the models before the Temporal Coherency Enhancement Phase.)

Shuffle. The experimental results showed that the model trained with Wavelet transform achieved better results. In our methods, we only utilize the Wavelet Transform to reduce the resolution and improve the efficiency. Different from [18], the weights of the kernel in our model are fixed.

5 Conclusions

This paper proposed an effective and efficient SWin Transformer-based video denoising network, TempFormer, which has outperformed most existing methods on additive Gaussian noise, achieved the best temporal coherent denoising results and lower computational complexity than the SOTA video denoiser. Specifically, we introduced the Wavelet Transform as pre-processing to halve the resolution of the video to improve efficiency. For utilizing the temporal information effectively, the spatial and temporal attention has been learned by the proposed Spatial-Temporal Transformer Block and the Joint Spatio-Temporal Mixer modules. Our model achieved both comparable quantitative and qualitative results with approximately 40% inference time requirement of the SOTA method. Moreover, the long-standing temporal inconsistency issue has been solved by the proposed recurrent strategy, together with the Overlap Loss function. The experimental results indicate that the proposed method dramatically enhanced the temporal coherency of the denoised video and almost exterminates the flicker between adjacent frames.

References

1. Ali, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al.: Xcit: Cross-covariance image transformers. Advances in neural information processing systems **34** (2021) 3
2. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding. arXiv preprint arXiv:2102.05095 **2**(3), 4 (2021) 4

3. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 2, pp. 60–65. IEEE (2005) 3
4. Cai, Z., Zhang, Y., Manzi, M., Oztireli, C., Gross, M., Aydin, T.O.: Robust image denoising using kernel predicting networks (2021) 3
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020) 2
6. Chan, K.C., Zhou, S., Xu, X., Loy, C.C.: Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. arXiv preprint arXiv:2104.13371 (2021) 2
7. Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W.: Pre-trained image processing transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12299–12310 (2021) 3
8. Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., Zhang, L.: Dynamic head: Unifying object detection heads with attentions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7373–7382 (2021) 3
9. Dai, Z., Liu, H., Le, Q., Tan, M.: Coatnet: Marrying convolution and attention for all data sizes. Advances in Neural Information Processing Systems **34** (2021) 3
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 2
11. Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: Proceedings of the European conference on computer vision (ECCV). pp. 170–185 (2018) 3
12. Lei, C., Xing, Y., Chen, Q.: Blind video temporal consistency via deep video prior. Advances in Neural Information Processing Systems **33** (2020) 3
13. Li, K., Wang, Y., Gao, P., Song, G., Liu, Y., Li, H., Qiao, Y.: Uniformer: Unified transformer for efficient spatiotemporal representation learning. arXiv preprint arXiv:2201.04676 (2022) 4
14. Liang, J., Cao, J., Fan, Y., Zhang, K., Ranjan, R., Li, Y., Timofte, R., Van Gool, L.: Vrt: A video restoration transformer. arXiv preprint arXiv:2201.12288 (2022) 2, 3, 4, 10, 11, 12, 13
15. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1833–1844 (2021) 3, 5, 6
16. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021) 2, 3, 5, 6
17. Maggioni, M., Boracchi, G., Foi, A., Egiazarian, K.: Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. IEEE Transactions on image processing **21**(9), 3952–3966 (2012) 2, 3
18. Maggioni, M., Huang, Y., Li, C., Xiao, S., Fu, Z., Song, F.: Efficient multi-stage video denoising with recurrent spatio-temporal fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3466–3475 (2021) 2, 7, 14
19. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object

segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 724–732 (2016) 10, 11, 13, 14
20. Tassano, M., Delon, J., Veit, T.: Dvdnet: A fast network for deep video denoising. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 1805–1809. IEEE (2019) 11
21. Tassano, M., Delon, J., Veit, T.: Fastdvdnet: Towards real-time deep video denoising without flow estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1354–1363 (2020) 2, 3, 10, 11, 12, 13
22. Vaksman, G., Elad, M., Milanfar, P.: Patch craft: Video denoising by deep modeling and patch matching. arXiv preprint arXiv:2103.13767 (2021) 3, 10, 11
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017) 2
24. Wang, C., Zhou, S.K., Cheng, Z.: First image then video: A two-stage network for spatiotemporal video denoising. arXiv preprint arXiv:2001.00346 (2020) 2, 3
25. Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., Bai, X., Liu, Z.: End-to-end semi-supervised object detection with soft teacher. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3060–3069 (2021) 3
26. Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., Gao, J.: Focal self-attention for local-global interactions in vision transformers. arxiv 2021. arXiv preprint arXiv:2107.00641 3
27. Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., Yan, S.: Metaformer is actually what you need for vision. arXiv preprint arXiv:2111.11418 (2021) 6
28. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432 (2021) 3
29. Yue, H., Cao, C., Liao, L., Chu, R., Yang, J.: Supervised raw video denoising with a benchmark dataset on dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2301–2310 (2020) 2, 3
30. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. arXiv preprint arXiv:2111.09881 (2021) 3
31. Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L.: Scaling vision transformers (2021) 3
32. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE transactions on image processing **26**(7), 3142–3155 (2017) 3
33. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3929–3938 (2017) 3
34. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. IEEE Transactions on Image Processing **27**(9), 4608–4622 (2018) 3