VRT: A Video Restoration Transformer

Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, *Member, IEEE*, Radu Timofte, *Member, IEEE*, and Luc Van Gool

***Abstract*— Video restoration aims to restore high-quality frames from low-quality frames. Different from single image restoration, video restoration generally requires to utilize temporal information from multiple adjacent but usually misaligned video frames. Existing deep methods generally tackle with this by exploiting a sliding window strategy or a recurrent architecture, which are restricted by frame-by-frame restoration. In this paper, we propose a Video Restoration Transformer (VRT) with parallel frame prediction ability. More specifically, VRT is composed of multiple scales, each of which consists of two kinds of modules: temporal reciprocal self attention (TRSA) and parallel warping. TRSA divides the video into small clips, on which reciprocal attention is applied for joint motion estimation, feature alignment and feature fusion, while self attention is used for feature extraction. To enable cross-clip interactions, the video sequence is shifted for every other layer. Besides, parallel warping is used to further fuse information from neighboring frames by parallel feature warping. Experimental results on five tasks, including video super-resolution, video deblurring, video denoising, video frame interpolation and space-time video super-resolution, demonstrate that VRT outperforms the state-of-the-art methods by large margins (up to 2.16dB) on fourteen benchmark datasets. The codes are available at https://github.com/JingyunLiang/VRT.**

***Index Terms*— Video restoration, video super-resolution, video deblurring, video denoising, video frame interpolation, space-time video super-resolution.**

I. Introduction

VIDEO restoration, which reconstructs high-quality (HQ) frames from multiple low-quality (LQ) frames, has attracted much attention in recent years. Compared with single image restoration, the key challenge of video restoration lies in how to make full use of neighboring highly-related but misaligned supporting frames for the reconstruction of the reference frame.

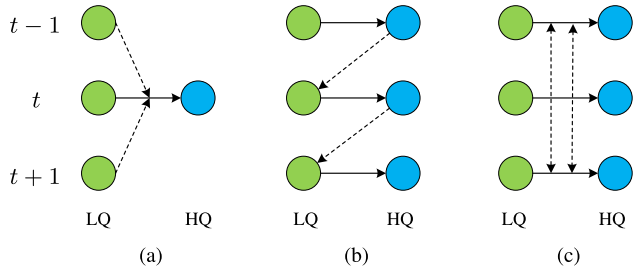


Fig. 1. Illustrative comparison of sliding window-based models (1a, *e.g.*, [2], [3]), recurrent models (1b, *e.g.*, [7], [9], [11], [12], [13]) and the proposed parallel VRT model (1c). Green and blue circles denote low-quality (LQ) input frames and high-quality (HQ) output frames, respectively. $t-1$, t and $t+1$ are frame serial numbers. Dashed lines represent information fusion among different frames.

Existing video restoration methods can be mainly divided into two categories: sliding window-based methods [1], [2], [3], [4], [5], [6] and recurrent methods [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. As shown in Fig. 1a, sliding window-based methods generally input multiple frames to generate a single HQ frame and process long video sequences in a sliding window fashion. Each input frame is processed for multiple times in inference, leading to inefficient feature utilization and increased computation cost.

Some other methods are based on a recurrent architecture. As shown in Fig. 1b, recurrent models mainly use previously reconstructed HQ frames for subsequent frame reconstruction. Due to the recurrent nature, they have three disadvantages. First, recurrent methods are limited in parallelization for efficient distributed training and inference. Second, they suffer from significant performance drops on few-frame videos [17]. Third, as will show in the experiments, recurrent models are less robust to noises, since information is accumulated frame by frame.

In this paper, we propose a **V**ideo **R**estoration **T**ransformer (VRT) that allows for parallel computation in video restoration. Based on a multi-scale framework, VRT divides the video sequence into non-overlapping clips and shifts it alternately to enable inter-clip interactions. Specifically, each scale of VRT has several temporal reciprocal self attention (TRSA) modules followed by a parallel warping module. In TRSA, reciprocal attention is focused on mutual alignment between neighboring two-frame clips, while self attention is used for feature extraction. At the end of each scale, we further use parallel warping to fuse neighboring frame information into the current frame. After multi-scale feature extraction, alignment and

Manuscript received 19 October 2022; revised 10 October 2023 and 3 January 2024; accepted 8 February 2024. Date of publication 7 March 2024; date of current version 25 March 2024. This work was supported in part by the Eidgenössische Technische Hochschule Zürich (ETH Zurich) Fund (OK), in part by Huawei Technologies Oy (Finland) Project, in part by China Scholarship Council, in part by the Amazon Web Services (AWS) Grant, and in part by the Alexander von Humboldt Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Giacomo Boracchi. *(Corresponding author: Kai Zhang.)*

Jingyun Liang, Jiezhang Cao, and Yawei Li are with the Computer Vision Laboratory, D-ITET, ETH Zürich, 8092 Zürich, Switzerland (e-mail: jingyun.liang@vision.ee.ethz.ch; jiezhang.cao@vision.ee.ethz.ch; yawei.li@vision.ee.ethz.ch).

Yuchen Fan and Rakesh Ranjan are with Meta Inc, Menlo Park, CA 94025 USA (e-mail: ycfan@meta.com; rakeshr@meta.com).

Kai Zhang was with the Computer Vision Laboratory, D-ITET, ETH Zürich, 8092 Zürich, Switzerland. He is now with the School of Intelligence Science and Technology, Nanjing University, Suzhou Campus, Suzhou 215163, China (e-mail: cskaizhang@gmail.com).

Radu Timofte is with the Computer Vision Laboratory, IFI & CAIDAS, University of Würzburg, 97070 Würzburg, Germany, and also with the Computer Vision Laboratory, D-ITET, ETH Zürich, 8092 Zürich, Switzerland (e-mail: radu.timofte@uni-wuerzburg.de).

Luc Van Gool is with the Computer Vision Laboratory, D-ITET, ETH Zürich, 8092 Zürich, Switzerland, and also with the Processing Speech and Images (PSI), Department of Electrical Engineering, KU Leuven, 3001 Leuven, Belgium (e-mail: vangool@vision.ee.ethz.ch).

Digital Object Identifier 10.1109/TIP.2024.3372454

fusion, the HQ frames are individually reconstructed from their corresponding frame features.

Compared with existing video restoration frameworks, VRT has several benefits. First, as shown in Fig. 1c, VRT is trained and tested on long video sequences in parallel. In contrast, both sliding window-based and recurrent methods are often tested frame by frame. Second, VRT has the ability to utilize information from multiple neighbouring frames during the reconstruction of each frame. By contrast, sliding window-based methods cannot be easily scaled up to long sequence modelling, while recurrent methods may forget distant information after several time steps. Third, VRT proposes to use reciprocal attention for joint feature alignment and fusion. It adaptively utilizes features from supporting frames and fuses them into the reference frame, which can be regarded as implicit motion estimation and feature warping.

Our contributions can be summarized as follows:

1) We propose a new framework named Video Restoration Transformer that is characterized by parallel computation. It jointly extracts, aligns, and fuses frame features at multiple scales.
2) We propose reciprocal attention for mutual alignment between frames. It is a generalized "soft" version of image warping after implicit motion estimation.
3) VRT achieves state-of-the-art performance on video restoration, including video super-resolution, deblurring, denoising, frame interpolation and space-time video super-resolution. It outperforms state-of-the-art methods by up to 2.16dB on benchmark datasets.

II. Related Work

A. Video Restoration

Similar to image restoration [18], [19], [20], learning-based methods, especially CNN-based methods, have become the primary workhorse for video restoration [2], [5], [12], [21], [22], [23], [24], [25], [26], [27] in recent years.

1) Framework Design: From the perspective of architecture design, existing methods can be roughly divided into two categories: sliding window-based and recurrent methods. Sliding window-based methods often takes a short sequence of frames as input and merely predict the center frame [2], [3], [5], [6], [28], [29]. Although some works [30] predict multiple frames, they still focus on the reconstruction of the center frame during training and testing. Recurrent framework is another popular choice [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. Huang et al. [7] propose a bidirectional recurrent convolutional neural network for SR. Sajjadi et al. [8] warp the previous frame prediction onto the current frame and feed it to a restoration network along with the current input frame. This idea is used by Chan et al. [12] for bidirectional recurrent network, and further extended as grid propagation in [13].

2) Temporal Alignment and Fusion: Since supporting frames are often highly-related but misaligned, temporal alignment plays an critical role in video restoration [2], [3], [12], [13], [31]. Early methods [1], [32] use traditional flow estimation methods to estimate optical flow and then warp the supporting frames towards the reference frame. To compensate occlusion and large motion, Xue et al. [31] utilize task-oriented flow by fine-tuning the pre-trained optical flow estimation model SpyNet [33] on different video restoration tasks. Jo et al. [34] use dynamic upsampling filters for implicit motion compensation. Tian et al. [3] propose TDAN that utilizes deformable convolution [35] for feature alignment. Based on TDAN, Wang et al. [2] extend it to multi-scale alignment, while Chan et al. [13] incorporate optical flow as a guidance for offsets learning.

3) Attention Mechanism: Attention mechanism has been exploited in video restoration in combination with CNN [2], [17], [32], [36]. Liu et al. [32] learn different weights for different temporal branches. Wang et al. [2] learn pixel-level attention maps for spatial and temporal feature fusion. To better incorporate temporal information, Suin and Rajagopalan [36] propose a reinforcement learning-based framework with factorized spatio-temporal attention. Cao et al. [17] propose to use self attention among local patches within a video.

B. Vision Transformer

In the last few years, transformer-based models [37], [38], [39] have achieved promising performance in various vision tasks, such as image recognition [40], object detection [41] and image restoration [20], [42], [43]. The key idea lies behind these methods is the self attention mechanism, which refines one feature according to the weighting of other features based on the calculated similarity. In fact, a related idea of self-similarity has been exploited in video restoration before the deep learning era. For example, [44] and [45] exploited the non-local means in video denoising, by searching the pixels that resemble the pixel in restoration. Concurrent to our paper, there are several modern transformer-based methods for various restoration tasks. For video super-resolution, Liu et al. [46] propose to use attention on relevant visual tokens along spatio-temporal trajectories. Song et al. [47] propose an efficient hybrid transformer-based model with spatio-temporal transformer blocks and 3D convolutional layers. The flickering artifact is also investigated and alleviated by the their proposed sliding block strategy and overlap loss. Liu et al. [48] design a spatio-temporal stability module to learn the self-alignment from inter-frames. For video deblurring and denoising, Li et al. [49] propose the grouped spatial-temporal shift to implicitly capture inter-frame correspondences for multi-frame aggregation. For video frame interpolation, Zhou et al. [50] develop a texture consistency loss and a guided cross-scale pyramid module for better alignment. For spatio-temporal video super-resolution, Zhang et al. [51] propose the cross-frame transformer module to refine the current feature maps through self-attention mechanism. For compressed video super-resolution, Qiu et al. [52] divide the video into different frequency bands and apply temporal attention to them before joint spatial attention. Compared with above methods, the proposed VRT is a general video restoration framework with parallel frame prediction ability and it achieves state-of-the-art performance on various restoration tasks.

III. Video Restoration Transformer

A. Overall Framework

Let $I^{LQ} \in \mathbb{R}^{T\times H\times W\times C_{in}}$ be a sequence of low-quality (LQ) input frames and $I^{HQ} \in \mathbb{R}^{T\times sH\times sW\times C_{out}}$ be a sequence of high-quality (HQ) target frames. T, H, W, C_{in} and C_{out} are the frame number, height, width and input channel number and output channel number, respectively. s is the upscaling factor, which is larger than 1 (*e.g.*, for video SR) or equal to 1 (*e.g.*, for video deblurring). The proposed Video Restoration Transformer (VRT) aims to restore T HQ frames from T LQ frames in parallel for various video restoration tasks, including video SR, deblurring, denoising, *etc*. As illustrated in Fig. 2, VRT can be divided into two parts: feature extraction and reconstruction.

1) Feature Extraction: At the beginning, we extract shallow features $I^{SF} \in \mathbb{R}^{T\times H\times W\times C}$ by a single spatial 2D convolution from the LQ sequence I^{LQ}. After that, based on [53], we propose a multi-scale network that aligns frames at different image resolutions. More specifically, when the total scale number is S, we downsample the feature for $S-1$ times by squeezing each 2×2 neighborhood to the channel dimension and reducing the channel number to the original number via a linear layer. Then, we upsample the feature gradually by unsqueezing the feature back to its original size. In such a way, we can extract features and deal with object or camera motions at different scales by two kinds of modules: temporal reciprocal self attention (TRSA, see III-B) and parallel warping (see III-C). Skip connections are added for features of same scales. Finally, after multi-scale feature extraction, alignment and fusion, we add several TRSA modules for further feature refinement and obtain the deep feature $I^{DF} \in \mathbb{R}^{T\times H\times W\times C}$.

2) Reconstruction: After feature extraction, we reconstruct the HQ frames from the addition of shallow feature I^{SF} and deep feature I^{DF}. Different frames are reconstructed independently based on their corresponding features. Besides, to ease the burden of feature learning, we employ global residual learning and only predict the residual between the bilinearly upsampled LQ sequence and the ground-truth HQ sequence. In practice, different reconstruction modules are used for different restoration tasks. For video SR, we use the sub-pixel convolution layer (pixel shuffle layers followed by convolution layers) [54] to upsample the feature by a scale factor of s. For video deblurring, a single convolution layer is enough for reconstruction. Apart from this, the architecture designs are kept the same for all tasks.

3) Loss Function: For fair comparison with existing methods, we use the commonly used Charbonnier loss between the reconstructed HQ sequence I^{RHQ} and the ground-truth HQ sequence I^{HQ} as

$$\mathcal{L} = \sqrt{\|I^{RHQ} - I^{HQ}\|^2 + \epsilon^2}, \tag{1}$$

where ϵ is a constant that is empirically set as 10^{-3}.

B. Temporal Reciprocal Self Attention

In this section, based on the attention mechanism [37], [39], we first introduce the reciprocal attention and then propose the temporal reciprocal self attention.

1) Reciprocal Attention: Given a reference frame feature $X^R \in \mathbb{R}^{N\times C}$ and a supporting frame feature $X^S \in \mathbb{R}^{N\times C}$, where N is the number of feature elements and C is the number of feature channels, we define the *query* Q^R, *key* K^S and *value* V^S as

$$Q^R = X^R P^Q, \quad K^S = X^S P^K, \quad V^S = X^S P^V, \tag{2}$$

where $P^Q, P^K, P^V \in \mathbb{R}^{C\times D}$ are linear projection matrices. D is the channel number of projected features. In particular, the *query* Q^R is the projected feature of the reference frame feature X^R, while the *key* K^S and *value* V^S are two corresponding features of the supporting frame X^S. Next, the *query* Q^R is used to query the *key* K^S in order to generate the attention map $A = \text{SoftMax}(Q^R(K^S)^T/\sqrt{D}) \in \mathbb{R}^{N\times N}$, which is then used for weighted sum of *value* V^S. This is formulated as

$$\text{MA}(Q^R, K^S, V^S) = \text{SoftMax}(Q^R(K^S)^T/\sqrt{D})V^S, \tag{3}$$

where SoftMax means the row softmax operation. When we regard the supporting frame feature as a dictionary with key-value pairs, this attention calculates the weighted sum of all values according to one input query from the reference frame.

Since Q^R and K^S come from X^R and X^S, respectively, A reflects the correlation between elements in the reference image and the supporting image. For clarity, we rewrite Eq. (3) for the i-th element of the reference image as

$$Y^R_{i,:} = \sum_{j=1}^{N} A_{i,j} V^S_{j,:}, \tag{4}$$

where $Y^R_{i,:}$ refers to the new feature of the i-th element in the reference frame. As shown in Fig. 3a, when $K^S_{k,:}$ (*e.g.*, the yellow square from the supporting frame) is the most similar element to $Q^R_{i,:}$ (*e.g.*, the orange square from the reference frame), $A_{i,k} > A_{i,j}$ holds for all $j \neq k$ ($j \leq N$). When all $K^S_{j,:} (j \neq k)$ are very dissimilar to Q^R_i, we have

$$\begin{cases} A_{i,k} \to 1, \\ A_{i,j} \to 0, \qquad\qquad for \quad j \neq k, j \leq N. \end{cases} \tag{5}$$

In this extreme case, by combining Eq. (4) and (5), we have $Y^R_{i,:} = V^S_{k,:}$, which moves the k-th element in the supporting frame to the position of the i-th element in the reference frame (see the dashed red line in Fig. 3a). This actually equals to image warping given an optical flow vector. When $A_{i,k} \to 1$ does not hold, Eq. (4) can be regarded as a "soft" version of image warping. In practice, the reference frame and supporting frame can be exchanged, allowing reciprocal alignment between two frames. Besides, similar to multi-head self attention, we can also perform the attention for h times and concatenate the results as multi-head reciprocal attention (MRA).

Particularly, reciprocal attention has several benefits over the combination of explicit motion estimation and image

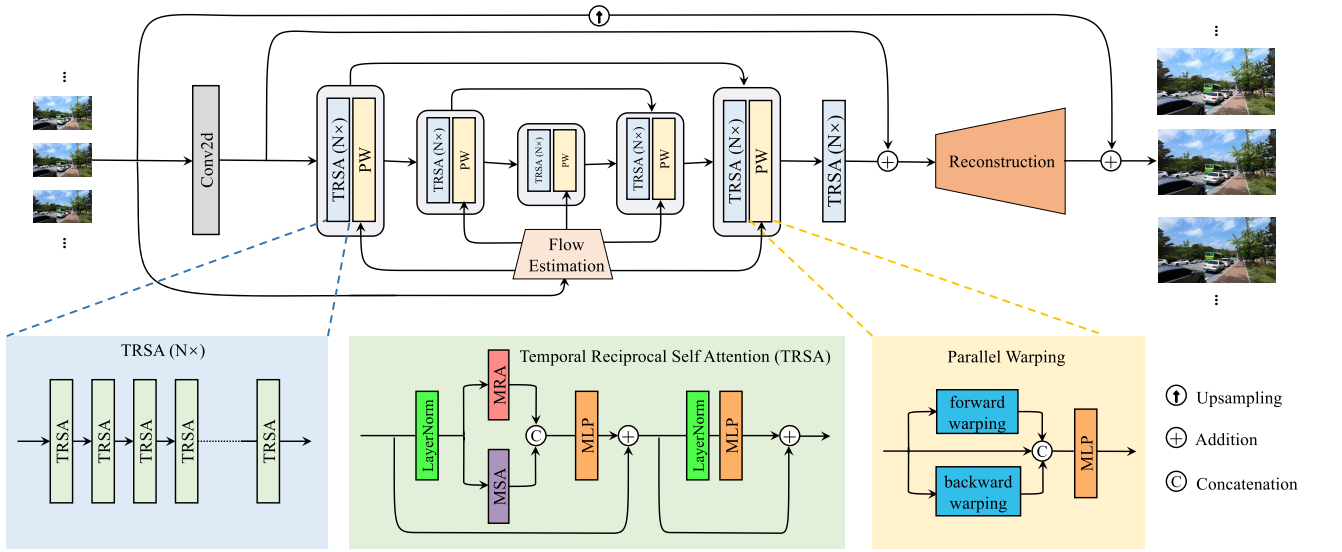
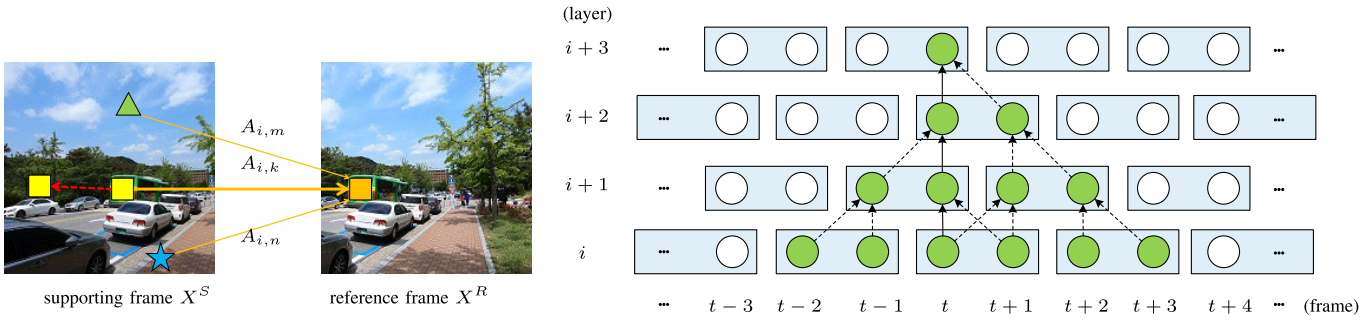


Fig. 2. The framework of the proposed Video Restoration Transformer (VRT). Given T low-quality input frames, VRT reconstructs T high-quality frames in parallel. It jointly extracts features, deals with misalignment, and fuses temporal information at multiple scales. On each scale, it has two kinds of modules: temporal reciprocal self attention (TRSA, see Sec. III-B) and parallel warping (PW, see Sec. III-C). The downsampling and upsampling operations between different scales are omitted for clarity. In the TRSA module, MRA, MSA and MLP represent multi-head reciprocal attention, multi-head self attention and multi-layer perception, respectively.



(a) Reciprocal attention (b) Stacking of temporal reciprocal self attention (TRSA)

Fig. 3. Illustrations for reciprocal attention and temporal reciprocal self attention (TRSA). In Fig. 3a, we let the orange square (the i-th element of the reference frame) query elements in the supporting frame and use their weighted features as a new representation for the orange square. The weights are shown around solid arrows (we only show three examples for clarity). When $A_{i,k} \to 1$ and the rest $A_{i,j} \to 0 (j \neq k)$, the reciprocal attention equals to warping the yellow square to the position of the orange square (illustrated as a dashed arrow). Fig. 3b shows a stack of temporal reciprocal self attention (TRSA) layers. The sequence is partitioned into 2-frame clips at each layer and shifted for every other layer to enable cross-clip interactions. Dashed lines represent information fusion among different frames.

warping. First, reciprocal attention can adaptively preserve information from the supporting frame than image warping, which only focuses on the target pixel. It also avoids black hole artifacts when there is no matched positions. Second, reciprocal attention does not have the inductive biases of locality, which is inherent to most CNN-based motion estimation methods [33], [55] and may lead to performance drop when two neighboring objects move towards different directions. Third, reciprocal attention equals to conducting motion estimation and warping on image features in a joint way. In contrast, optical flows are often estimated on the input RGB image and then used for warping on features [12], [13]. Besides, flow estimation on RGB images is often not robust to lighting variation, occlusion and blur [31].

2) Temporal Reciprocal Self Attention (TRSA): Reciprocal attention is proposed for joint feature alignment between two frames. To extract and preserve feature from the current frame, we use reciprocal attention together with self attention. Let $X \in \mathbb{R}^{2\times N\times C}$ represent two frames, which can be split into $X_1 \in \mathbb{R}^{1\times N\times C}$ and $X_2 \in \mathbb{R}^{1\times N\times C}$. We use multi-head reciprocal attention (MRA) on X_1 and X_2 for two times: warping X_1 towards X_2 and warping X_2 towards X_1. The warped features are combined and then concatenated with the result of multi-head self attention (MSA) within two frames, followed by a multi-layer perceptron (MLP) for the purpose of dimension reduction. After that, another MLP is added for further feature transformation. Two LayerNorm (LN) layers and two residual connections are also used as shown in the green box of Fig. 2. The whole process formulated as follows

$$\begin{aligned} X_1, X_2 &= \text{Split}_0(\text{LN}(X)) \\ Y_1, Y_2 &= \text{MRA}(X_1, X_2), \text{MRA}(X_2, X_1) \\ Y_3 &= \text{MSA}(\text{Concat}_0(X_1, X_2)) \\ X &= \text{MLP}(\text{Concat}_2(\text{Concat}_0(Y_1, Y_2), Y_3)) + X \\ X &= \text{MLP}(\text{LN}(X)) + X \end{aligned} \tag{6}$$

where the subscripts of Split and Concat refer to the specified dimensions. For example, $\text{Split}_0(\cdot)$ means splitting the tensor

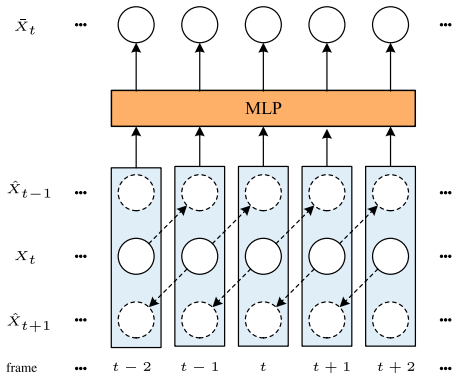


Fig. 4. Illustration of parallel warping. For every frame feature $X_t (t \leq T)$, frame X_{t-1} and X_{t+1} are warped towards X_t as $\hat{X}_{t-1}$ and $\hat{X}_{t+1}$, respectively. Then, X_t, $\hat{X}_{t-1}$ and $\hat{X}_{t+1}$ are concatenated together (denoted by blue boxes) for feature fusion and dimension reduction with a multi-layer perception (MLP). The final output is $\bar{X}_t$. The dashed arrows and circles denote warping operations and warped features, respectively.

along the 0-th dimension. However, due to the design of reciprocal attention, Eq. (6) can only deal with two frames at a time.

One naive way to extend Eq. (6) for T frames is to deal with frame-to-frame pairs exhaustively, resulting in the computational complexity of $\mathcal{O}(T^2)$. Inspired by the shifted window mechanism [40], [56], we propose the temporal reciprocal self attention (TRSA) to remedy the problem. TRSA first partitions the video sequence into non-overlapping 2-frame clips and then applies Eq. (6) to them in parallel. Next, as shown in Fig. 3b, it shifts the sequence temporally by 1 frame for every other layer to enable cross-clip connections, reducing the computational complexity to $\mathcal{O}(T)$. The temporal receptive field size is increased when multiple TRSA modules are stacked together. Specifically, at layer i ($i \geq 2$), one frame can utilize information from up to $2(i-1)$ frames.

3) Discussion: Video restoration tasks often need to process high-resolution frames. Since the complexity of attention is quadratic to the number of elements within the attention window, global attention on the full image is often impractical. Therefore, following [20], [40], we partition each frame spatially into non-overlapping $M \times M$ local windows, resulting in $\frac{HW}{M^2}$ windows. Shifted window mechanism (with the shift of $\lfloor \frac{M}{2} \rfloor \times \lfloor \frac{M}{2} \rfloor$ pixels) is also used spatially to enable cross-window connections. Besides, although stacking multiple TRSA modules allows for long-distance temporal modelling, distant frames are not directly connected. As will show in the ablation study, using only a small temporal window size cannot fully exploit the potential of the model. One naive idea is to directly use larger temporal attention window size. However, this could be computationally intensive and impractical for videos. To balance the performance and efficiency, we use the normal TRSA modules for the first three quarters of layers and use larger temporal window size for the last quarter of layers.

C. Parallel Warping

Due to spatial window partitioning, the reciprocal attention mechanism may not be able to deal with large motions well. Hence, as shown in the orange box of Fig. 2, we use feature warping at the end of each network stage to handle large motions. As shown in Fig. 4, for any frame feature X_t, we calculate the optical flows of its neighbouring frame features X_{t-1} and X_{t+1}, and warp them towards the frame X_t as $\hat{X}_{t-1}$ and $\hat{X}_{t+1}$ (*i.e.*, backward and forward warping). Then, we concatenate X_t, $\hat{X}_{t-1}$ and $\hat{X}_{t+1}$ along the channel dimension (denoted by the blue box). To keep the original channel size for later operations, we reduce its dimension by a multi-layer perception (MLP) and obtain $\bar{X}_t$. This mechanism can be generalized for four (*i.e.*, X_{t-2}, X_{t-1}, X_{t+1} and X_{t+2}) and six (*i.e.*, X_{t-3}, X_{t-2}, X_{t-1}, X_{t+1}, X_{t+2} and X_{t+3}) neighboring frames. Note that different frames are processed in parallel.

Specifically, following [13], we predict the optical flow by a flow estimation model and use deformable convolution [35] for deformable alignment. Given estimated optical flows $O_{t-1,t}$ and $O_{t+1,t}$, we first use them to warp X_{t-1} and X_{t+1}, respectively as

$$\begin{cases} X'_{t-1} = \mathcal{W}(X_{t-1}, O_{t-1,t}), \\ X'_{t+1} = \mathcal{W}(X_{t+1}, O_{t+1,t}), \end{cases} \tag{7}$$

where $\mathcal{W}$ represents the image warping function. X'_{t-1} and X'_{t+1} are the initial warped features. Then, we use several convolution layers (denoted as $\mathcal{C}$) to predict the optical flow residuals (offsets) $o_{t-1,t}$, $o_{t+1,t}$ and modulation masks $m_{t-1,t}$, $m_{t+1,t}$ from the concatenation of $O_{t-1,t}$, $O_{t+1,t}$, X'_{t-1} and X'_{t+1} as

$$\begin{aligned} o_{t-1,t}, o_{t+1,t}, m_{t-1,t}, m_{t+1,t} = \mathcal{C}(Concat(O_{t-1,t}, \\ O_{t+1,t}, X'_{t-1}, X'_{t+1})). \end{aligned} \tag{8}$$

Next, we warp X_{t-1} and X_{t+1} again as

$$\begin{cases} \hat{X}_{t-1} = \mathcal{D}(X_{t-1}, O_{t-1,t} + o_{t-1,t}, m_{t-1,t}), \\ \hat{X}_{t+1} = \mathcal{D}(X_{t+1}, O_{t+1,t} + o_{t+1,t}, m_{t+1,t}), \end{cases} \tag{9}$$

where $\mathcal{D}$ refers to the deformable convolution. Finally, the outputs $\hat{X}_{t-1}$ and $\hat{X}_{t+1}$ and concatenated with X_t as the new feature for the t-th frame.

IV. Experiments

A. Experimental Setup

For video SR, we use 4 scales for VRT. On each scale, we stack 8 TRSA modules, the last two of which use a temporal window size of 8. The spatial window size $M \times M$, head size h, and channel size C are set to 8×8, 6 and 120, respectively. After 7 multi-scale feature extraction stages, we add 24 TRSA modules (only with self attention) for further feature extraction before reconstruction. For flow estimation, we extract multi-scale flows from different layers of SpyNet [33], [57] and feed them into different scales of VRT. In video SR, for the additional TRSA modules in the 8-th stage, the channel number is set as 180. We use temporal window sizes of 8 and 2 for the first two thirds of modules and the rest ones, respectively. In video deblurring and denoising, we use a relatively smaller model. The channel sizes for the first 7 stages and the 8-th stage are 96 and 120, respectively. For the 8-th stage, we only use 16 TRSA modules. In addition,

TABLE I

QUANTITATIVE COMPARISON (AVERAGE PSNR/SSIM) WITH STATE-OF-THE-ART METHODS FOR **VIDEO SUPER-RESOLUTION** (×4) ON **REDS4** [60], **VIMEO-90K-T** [31], **VID4** [61] AND **UDM10** [21]. BEST AND SECOND BEST RESULTS ARE IN RED AND BLUE COLORS, RESPECTIVELY. †WE CURRENTLY DO NOT HAVE ENOUGH GPU MEMORY TO TRAIN THE FULLY PARALLEL MODEL VRT ON 30 FRAMES

Method	Training Frames (REDS/ Vimeo-90K)	Params(M)/ Runtime(ms)	BI degradation			BD degradation		
			REDS4 [60] (RGB channel)	Vimeo-90K-T [31] (Y channel)	Vid4 [61] (Y channel)	UDM10 [21] (Y channel)	Vimeo-90K-T [31] (Y channel)	Vid4 [61] (Y channel)
Bicubic	- / -	-	26.14/0.7292	31.32/0.8684	23.78/0.6347	28.47/0.8253	31.30/0.8687	21.80/0.5246
SwinIR [20]	-	11.9 / -	29.05/0.8269	35.67/0.9287	25.68/0.7491	35.42/0.9380	34.12/0.9167	25.25/0.7262
SwinIR-ft [20]	1/1	11.9 / -	29.24/0.8319	35.89/0.9301	25.69/0.7488	36.76/0.9467	35.70/0.9293	25.62/0.7498
TOFlow [31]	5/7	- / -	27.98/0.7990	33.08/0.9054	25.89/0.7651	36.26/0.9438	34.62/0.9212	25.85/0.7659
FRVSR [8]	10/7	5.1 / 137	-	-	-	37.09/0.9522	35.64/0.9319	26.69/0.8103
DUF [34]	7/7	5.8 / 974	28.63/0.8251	-	27.33/0.8319	38.48/0.9605	36.87/0.9447	27.38/0.8329
PFNL [21]	7/7	3.0 / 295	29.63/0.8502	36.14/0.9363	26.73/0.8029	38.74/0.9627	-	27.16/0.8355
RBPN [10]	7/7	12.2 / 1507	30.09/0.8590	37.07/0.9435	27.12/0.8180	38.66/0.9596	37.20/0.9458	27.17/0.8205
RLSP [9]	-/7	4.2 / 49	-	-	-	38.48/0.9606	36.49/0.9403	27.48/0.8388
RSDN [11]	-/7	6.2 / 94	-	-	-	39.35/0.9653	37.23/0.9471	27.92/0.8505
FDAN [14]	-/7	9.0 / -	-	-	-	39.91/0.9686	37.75/0.9522	27.88/0.8508
EDVR [2]	5/7	20.6 / 378	31.09/0.8800	37.61/0.9489	27.35/0.8264	39.89/0.9686	37.81/0.9523	27.85/0.8503
GOVSR [25]	-/7	7.1 / 81	-	-	-	40.14/0.9713	37.63/0.9503	28.41/0.8724
VSRT [17]	5/7	32.6 / -	31.19/0.8815	37.71/0.9494	27.36/0.8258	-	-	-
VRT (ours)	6/-	30.7 / 236	31.60/0.8888	-	-	-	-	-
BasicVSR [12]	15/14	6.3 / 63	31.42/0.8909	37.18/0.9450	27.24/0.8251	39.96/0.9694	37.53/0.9498	27.96/0.8553
IconVSR [12]	15/14	8.7 / 70	31.67/0.8948	37.47/0.9476	27.39/0.8279	40.03/0.9694	37.84/0.9524	28.04/0.8570
TCNet [48]	15/14	9.6 / 94	31.82/0.9002	37.84/0.9514	27.48/0.8380	-	-	-
TTSR [46]	50/14	6.8 / -	-	-	-	40.41/0.9712	37.92/0.9526	28.40/0.8643
BasicVSR++ [13]	30/14	7.3 / 77	32.39/0.9069†	37.79/0.9500	27.79/0.8400	40.72/0.9722	38.21/0.9550	29.04/0.8753
VRT-lightweight (ours)	30/14	7.3 / 95	32.01/0.8976	37.83/0.9505	27.82/0.8404	-	-	-
VRT (ours)	16/7	35.6 / 243	32.19/0.9006	38.20/0.9530	27.93/0.8425	41.05/0.9737	38.72/0.9584	29.42/0.8795

the gated variant GEGLU [38] is used to replace the plain feed-forward network.

We augment the input frames by random flipping, rotation and cropping, with a training batch size of 8 and 300K training iterations. The model is trained by the Adam optimizer [58] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The learning rate is initialized as $4e-4$ and decreased gradually according to the Cosine Annealing scheme [59]. To stabilize training, we use the pretrained model of SpyNet for initialization and fix the flow estimation part for the first 20K iterations. We also use a smaller initial learning rate (*i.e.*, $5e-5$) for it.

There are several training differences in different tasks. First, we set the training frame number as 6 for most tasks (7 for Vimeo-90K in video SR) and additionally provide experiments on 16 frames for REDS in video SR. For Vimeo-90K, following [13], we initialize the model with the REDS pretrained model. Second, the training patch size is 64×64 for video SR and 192×192 for other tasks. Third, for video denoising, we follow [28], [29] and train a non-blind denoising model using varying noise levels ($\sigma \sim \mathcal{U}(0, 50)$) by concatenating the noise level map with the noisy video along the channel dimension. All experiments are conducted on a server with 8 A100 GPUs. For video SR, it takes about 5 and 10 days for 6-frame and 16-frame experiments, respectively. For video deblurring and denoising, the training time is about 10 days.

1) Dataset: For video super-resolution, we train the model on two different training datasets for scale factor 4. First, we generate low-resolution images by the MATLAB `imresize` function (*i.e.*, bicubic degradation) and train the model on REDS [60]. REDS4 [2] is used as the test set. Second, we train the model on Vimeo-90K [31] with two different degradations: bicubic and blur downsampling (Gaussian blur with $\sigma = 1.6$ followed by subsampling). The testing datasets include Vimeo-90K-T [31], Vid4 [61] and UDM10 [21]. For video deblurring, we train the model on three different datasets (DVD [4], GoPro [62] and REDS [60]). We test it on their corresponding testing sets (for REDS, we use REDS4 [2]). For video denoising, we train the model on the DAVIS [63] and test it on the corresponding testing set and Set8 [28].

For evaluation, following [2], [6], [12], [28], [36], we calculate the metrics on RGB channel for REDS4 [2], DVD testing set [4], GoPro testing set [62], DAVIS testing set [63] as well as Set8 [28], and on the Y channel for Vimeo-90K-T [31], Vid4 [61] and UDM10 [21].

B. Video SR

1) Quantitative Results: As shown in Table I, we compare VRT with the state-of-the-art image and video SR methods [2], [8], [9], [10], [11], [12], [13], [14], [17], [21], [25], [31], [34]. VRT achieves best performance for both bicubic (BI) and blur-downsamplng (BD) degradations. Specifically, when trained on the REDS [60] dataset with short sequences, VRT outperforms VSRT by up to 0.57dB in PSNR. Compared with another representative sliding window-based model EDVR, VRT has an improvement of 0.50∼1.57dB on different datasets, showing its good ability to fuse information from multiple frames. Note that VRT outputs all frames simultaneously rather than predicting them frame by frame as EDVR does. On the Vimeo-90K [31] dataset, VRT surpasses BasicVSR++ by up to 0.38dB, although BasicVSR++ and other recurrent models may mirror the 7-frame video for training and testing. When VRT is trained on longer sequences, it shows good potential in temporal modelling and further

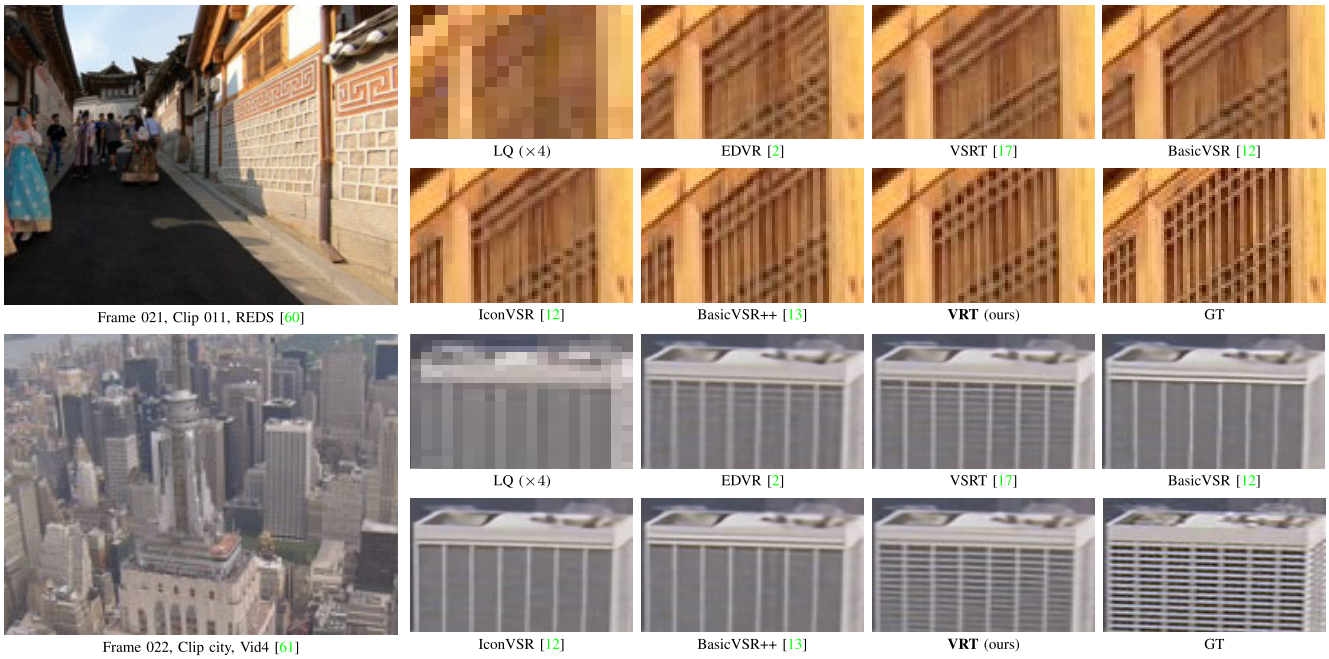


Fig. 5. Visual comparison of **video super-resolution** (×4) methods.

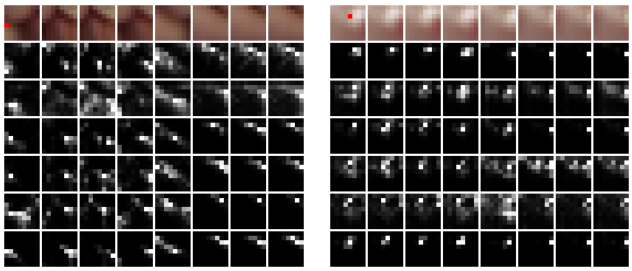
increases the PSNR by 0.52dB. As indicated in [17], recurrent models often suffer from significant performance drops on short sequences. In contrast, VRT performs well on both short and long sequences. We note that VRT is slightly lower than the 32-frame model BasicVSR++. This is expected since VRT is only trained on 16 frames. When we reduce the model size of VRT to be similar to BasicVSR++ and train it with 30 or 14 frames, we find that it outperforms BasicVSR++ on Vimeo-90K-T and Vid4, but not on REDS4. We guess it is due to the difference of the characteristics of different testing sets.

We also provide comparison on parameter number and runtime in Table I. As a parallel model, VRT needs to restore all frames at the same time, which leads to relatively larger model size and longer runtime per frame compared with recurrent models. However, VRT has the potential for distributed deployment, which is hard for recurrent models that restore a video clip recursively by design.

2) Qualitative Results: Visual results of different methods are shown in Fig. 5. As one can see, in accordance with its significant quantitative improvements, VRT can generate visually pleasing images with sharp edges and fine details, such as horizontal strip patterns of buildings. By contrast, its competitors suffer from either distorted textures or lost details.

3) Performance in Different Motion Conditions: Following [23], we compare different methods on Vimeo90K [31] (×4, BI degradation) with fast/ medium/ slow motions. As shown in Table II, VRT leads to larger improvement on fast motion videos than on slow ones when compared with existing methods.

4) Performance on Different Testing Frame Numbers: To compare the characteristics of recurrent models and the proposed parallel model VRT, we compare VRT with a representative recurrent model BasicVSR++ [13] using different testing frame numbers. As shown in Table III, from 100 frames to 5 frames, the PSNR of BasicVSR++ drops significantly, while the VRT is less sensitive to the testing frame numbers and outperforms BasicVSR++ by large margins on few-frame cases.



(a) (b)

Fig. 6. Visualization of attention maps. The first row shows the original image patches at the same position from different frames, while the rest rows are the attention weight visualizations of six different attention heads. The query pixel is marked by a red point in the first frame.

5) Robustness to Noise: In addition, to compare the noise robustness of parallel models and recurrent models, we hack the LQ input video by manually setting all pixels of the 50-th frame as zero in testing. As shown in Fig. 7, VRT suffers from less performance drop and has less adverse impact on neighbouring frames than BasicVSR, indicating that VRT is more robust to noise.

6) Attention Visualization: To show exploit what the attention mechanism has learned, we plot the attention maps between a pixel (marked as red points) from the first frame and the rest pixels in the same attention window. As shown in Fig. 6, when the red point moves towards the top-right direction from the first frame to the last frame, it moves most attention to the top-right direction as well. Similar observations can be concluded from other examples. This shows the temporal dependency modelling ability of our model across different frames.

TABLE II

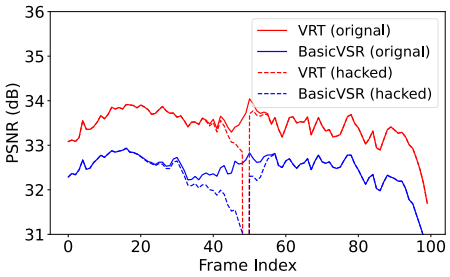
VIDEO SR (×4, BI DEGRADATION) RESULTS ON VIMEO-FAST/ MEDIUM/ SLOW SUBSETS

Subset	EDVR [2]	BasicVSR [12]	BasicVSR++ [13]	VRT (TRSA)	VRT (PA)	VRT (ours, TRSA+PA)
Fast	40.77	40.34	40.98	40.78	41.29	41.44
Medium	37.81	37.35	37.99	37.95	38.22	38.42
Slow	34.52	34.11	34.57	34.90	34.65	34.98

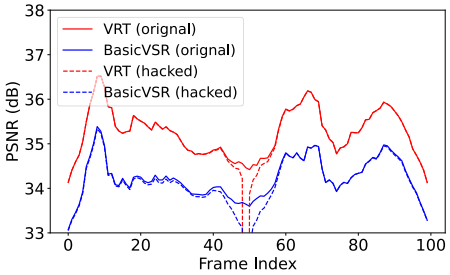
TABLE III

ABLATION STUDY ON TESTING FRAME NUMBERS

# Frame	5	10	20	40	60	100
BasicVSR++ [13]	30.73	31.12	31.43	31.81	32.21	32.39
VRT (ours)	31.69	31.88	32.01	32.10	32.15	32.19



(a) Clip 011, REDS [60]



(b) Clip 015, REDS [60]

Fig. 7. Comparison of per-frame PSNR drop when pixels of the 50-th frame of the LQ input video is hacked to be all zeros during testing.

C. Video Deblurring

1) Quantitative Results: We conducts experiments on three different datasets for fair comparison with existing methods [2], [4], [5], [6], [16], [22], [24], [36], [62], [64]. Table IV shows the results on the DVD [4] dataset. It is clear that VRT achieves the best performance, outperforming the second best method ARVo by a remarkable improvement of 1.47dB and 0.0299 in terms of PSNR and SSIM. PVDNet proposes motion estimation learning to better aggregate information from multiple frames, but it is inferior to the proposed VRT, which uses reciprocal attention for alignment. Related to the attention mechanism, GSTA designs a gated spatio-temporal attention mechanism, while ARVo calculates the correlation between pixel pairs for correspondence learning. However, both of them are based on CNN, achieving significantly worse performance compared with the transformer-based VRT. We also compare VRT on the GoPro [62] (Table V) and REDS [60] (Table VI) datasets. VRT shows its superiority over other methods with significant PSNR gains of 2.15dB and 1.99dB. The total number of parameters of VRT is 18.3M and the runtime is 2.2s per frame on 1280 × 720 blurred videos. In contrast, the model sizes of EDVR and PVDNet are 23.6M and 23.5M, respectively, and their runtimes are 0.4s and 1.1s, respectively. Notably, during evaluation, we do not use any pre-processing techniques such as sequence truncation and image alignment [16], [24].

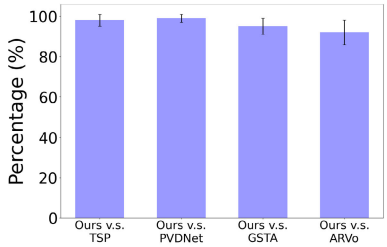


Fig. 8. User study of video deblurring on the DVD [4] dataset.

2) Qualitative Results: Fig. 9 shows the visual comparison of different methods. VRT is effective in removing motion blurs and restoring faithful details, such as the pole in the first example and characters in the second one. In comparison, other approaches fail to remove blurs completely and do not produce sharp edges.

In addition, we conduct a user study with 20 users on video deblurring. Each user is given multiple pairs of deblurred videos from DVD [4], where one is our result. As shown in Fig. 8, over 90% of the users vote that VRT has better visual quality than existing methods.

D. Video Denoising

We also conduct experiments on video denoising to show the effectiveness of VRT. Following [28], [29], we train one non-blind model for noise level $\sigma \in [0, 50]$ on the DAVIS [63] dataset and test it on different noise levels. Table VII shows the superiority of VRT on two benchmark datasets over existing methods [26], [28], [29], [67]. Even though PaCNet [26] trains different models separately for different noise levels, VRT still improves the PSNR by 0.82∼2.16dB. As for the efficiency, the VRT has a runtime of 2.2s, while the runtimes of DVDnet and PaCNet are 8s and 35.4s, respectively.

E. Video Frame Interpolation

To show the generalizability of our framework, we conduct experiments on video frame interpolation. Following [68], [69], we train the model on Vimeo-90K [31] for single frame interpolation and test it on quintuples generated from Vimeo-90K-T [31], UCF101 [70] and DAVIS [63]. As shown in Table VIII, VRT achieves best or competitive performance on all datasets compared with it competitors, including those using depth maps or optical flows. As for the model size, VRT only has 9.9M parameters, which is much smaller than the recent best model FLAVR (42.4M).

TABLE IV

QUANTITATIVE COMPARISON (AVERAGE RGB CHANNEL PSNR/SSIM) WITH STATE-OF-THE-ART METHODS FOR **VIDEO DEBLURRING** ON **DVD** [4]. FOLLOWING [6], [24], ALL RESTORED FRAMES INSTEAD OF RANDOMLY SELECTED 30 FRAMES FROM EACH TEST SET [4] ARE USED IN EVALUATION. BEST AND SECOND BEST RESULTS ARE IN RED AND BLUE COLORS, RESPECTIVELY

Method	DeepDeblur [62]	SRN [64]	DBN [4]	STFAN [5]	SFE [22]	EDVR [2]	TSP [24]	PVDNet [16]	GSTA [36]	ARVo [6]	**VRT** (ours)
PSNR	29.85	30.53	30.01	31.24	31.71	31.82	32.13	32.31	32.53	32.80	34.27 (+1.47)
SSIM	0.8800	0.8940	0.8877	0.9340	0.9160	0.9160	0.9268	0.9260	0.9468	0.9352	0.9651 (+0.03)

TABLE V

QUANTITATIVE COMPARISON (AVERAGE RGB CHANNEL PSNR/SSIM) WITH STATE-OF-THE-ART METHODS FOR **VIDEO DEBLURRING** ON **GOPRO** [62]. BEST AND SECOND BEST RESULTS ARE IN RED AND BLUE COLORS, RESPECTIVELY

Method	DeepDeblur [62]	SRN [64]	SAPHN [65]	MPRNet [66]	SFE [22]	ESTRNN [15]	EDVR [2]	TSP [24]	PVDNet [16]	GSTA [36]	**VRT** (ours)
PSNR	29.23	30.26	31.85	32.66	31.01	31.07	31.54	31.67	31.98	32.10	34.81 (+2.15)
SSIM	0.9162	0.9342	0.9480	0.9590	0.9130	0.9023	0.9260	0.9279	0.9280	0.9600	0.9724 (+0.01)

Fig. 9. Visual comparison of **video deblurring** methods.

TABLE VI

QUANTITATIVE COMPARISON (AVERAGE RGB CHANNEL PSNR/SSIM) WITH STATE-OF-THE-ART METHODS FOR **VIDEO DEBLURRING** ON **REDS** [60]. BEST AND SECOND BEST RESULTS ARE IN RED AND BLUE COLORS, RESPECTIVELY

Method	DeepDeblur [62]	SRN [64]	DBN [4]	EDVR [2]	**VRT** (ours)
PSNR	26.16	26.98	26.55	34.80	36.79 (+1.99)
SSIM	0.8249	0.8141	0.8066	0.9487	0.9648 (+0.02)

F. Space-Time Video Super-Resolution

With the pretrained models on video SR (VSR) and video frame interpolation (VFI), we directly test VRT on space-time video super-resolution by cascading VRT models in two ways: VFI followed by VSR, or VSR followed by VFI. As shown in Table IX, compared with existing methods, VRT provides a strong baseline for space-time video super-resolution, even though it serves as a two-stage model and is not specifically trained for this task. In particular, it improves the PSNR by 1.03dB on the Vid4 dataset.

TABLE VII

QUANTITATIVE COMPARISON (AVERAGE RGB CHANNEL PSNR) WITH STATE-OF-THE-ART METHODS FOR **VIDEO DENOISING** ON **DAVIS** [63] AND **SET8** [28]. σ IS THE ADDITIVE WHITE GAUSSIAN NOISE LEVEL. BEST AND SECOND BEST RESULTS ARE IN RED AND BLUE COLORS, RESPECTIVELY

Dataset	σ	VLNB [67]	DVDnet [28]	FastDVDnet [29]	PaCNet [26]	**VRT** (ours)
DAVIS	10	38.85	38.13	38.71	39.97	40.82 (+0.85)
	20	35.68	35.70	35.77	36.82	38.15 (+1.33)
	30	33.73	34.08	34.04	34.79	36.52 (+1.73)
	40	32.32	32.86	32.82	33.34	35.32 (+1.98)
	50	31.13	31.85	31.86	32.20	34.36 (+2.16)
Set8	10	37.26	36.08	36.44	37.06	37.88 (+0.82)
	20	33.72	33.49	33.43	33.94	35.02 (+1.08)
	30	31.74	31.79	31.68	32.05	33.35 (+1.30)
	40	30.39	30.55	30.46	30.70	32.15 (+1.45)
	50	29.24	29.56	29.53	29.66	31.22 (+1.56)

G. Ablation Study

For ablation study, we set up a small version of VRT as the baseline model by halving the layer and channel numbers. All models are trained on Vimeo-90K [31] for bicubic video SR ($\times$4) and tested it on Vid4 [61].

TABLE VIII

QUANTITATIVE COMPARISON (AVERAGE RGB CHANNEL PSNR) WITH STATE-OF-THE-ART METHODS FOR **VIDEO FRAME INTERPOLATION** (SINGLE FRAME INTERPOLATION, ×2) ON **VIMEO-90K-T** [31], **UCF101** [70] AND **DAVIS** [63]. R, D AND F MEANS THAT THE MODEL USES RGB IMAGES, DEPTH MAPS OR OPTICAL FLOWS. BEST AND SECOND BEST RESULTS ARE IN RED AND BLUE COLORS, RESPECTIVELY

Method	Inputs	Vimeo-90K-T [31]	UCF101 [70]	DAVIS [63]
DAIN [71]	R+D+F	33.35/0.945	31.64/0.957	26.12/0.870
QVI [68]	R+F	35.15/0.971	32.89/0.970	27.17/0.874
CAIN [72]	R	33.93/0.964	32.28/0.965	26.46/0.856
SuperSloMo [73]	R	32.90/0.957	32.33/0.960	25.65/0.857
AdaCoF [74]	R	35.40/0.971	32.71/0.969	26.49/0.866
FLAVR [69]	R	36.25/0.975	33.31/0.971	27.43/0.874
VRT (ours)	R	36.53/0.977	33.30/0.970	27.88/0.889

TABLE IX

QUANTITATIVE COMPARISON (AVERAGE Y CHANNEL PSNR) WITH STATE-OF-THE-ART METHODS FOR **SPACE-TIME VIDEO SUPER-RESOLUTION** (TIME: ×2, SPACE: ×4) ON **VID4** [61] AND **VIMEO-90K-T** [31]. [71] IS THE FRAME INTERPOLATION METHOD DAIN. NOTE THAT THE PROPOSED VRT IS NOT TRAINED ON THIS TASK. WE DIRECTLY TEST IT BY CASCADING PRE-TRAINED VIDEO SUPER-RESOLUTION (VSR) AND VIDEO FRAME INTERPOLATION (VFI) MODELS. BEST AND SECOND BEST RESULTS ARE IN RED AND BLUE COLORS, RESPECTIVELY

VFI+VSR Methods	Vid4 [61]	Vimeo-Fast [31]	Vimeo-Medium [31]	Vimeo-Slow [31]
[71]+Bicubic	23.55/0.6268	32.41/0.8910	30.67/0.8636	29.06/0.8289
[71]+RCAN [19]	25.03/0.7261	35.27/0.9242	33.82/0.9146	32.26/0.8974
[71]+RBPN [10]	25.96/0.7784	35.55/0.9300	34.45/0.9262	32.92/0.9097
[71]+EDVR [2]	26.12/0.7836	35.81/0.9323	34.66/0.9281	33.11/0.9119
ZSM [23]	26.31/0.7976	36.81/0.9415	35.41/0.9361	33.36/0.9138
CFT [51]	26.02/0.7803	36.41/0.9293	32.60/0.8841	31.49/0.8944
STARnet [75]	26.06/0.8046	36.19/0.9368	34.86/0.9356	33.10/0.9164
TMNet [76]	26.43/0.8016	37.04/0.9435	35.60/0.9380	33.51/0.9159
RSTT [77]	26.43/0.7994	36.80/0.9403	35.66/0.9381	33.50/0.9147
VRT (VFI+VSR)	26.59/0.8014	36.56/0.9372	35.28/0.9343	33.75/0.9204
VRT (VSR+VFI)	27.46/0.8392	36.98/0.9439	36.01/0.9434	34.01/0.9236

TABLE X

ABLATION STUDY ON MULTI-SCALE ARCHITECTURE AND PARALLEL WARPING. GIVEN AN INPUT OF SPATIAL SIZE 64 × 64, THE CORRESPONDING FEATURE SIZES OF EACH SCALE ARE SHOWN IN BRACKETS. WHEN SOME SCALES ARE REMOVED, WE ADD MORE LAYERS TO THE REST SCALES TO KEEP SIMILAR MODEL SIZE

1 (64 × 64)	2 (32 × 32)	3 (16 × 16)	4 (8 × 8)	Parallel warping	PSNR
✓				✓	27.13
✓	✓			✓	27.20
✓	✓	✓		✓	27.25
✓	✓	✓	✓		27.11
✓	✓	✓	✓	✓	27.28

1) Impact of Multi-Scale Architecture & Parallel Warping: Table X shows the ablation study on the multi-scale architecture and parallel warping. When the number of model scales is reduced, the performance drops gradually, even though the computation burden becomes heavier. This is expected because multi-scale processing can help the model utilize information from a larger area and deal with large motions between frames. Besides, parallel warping also helps, bringing an improvement of 0.17dB.

2) Impact of Temporal Reciprocal Self Attention & the Comparison With Parallel Warping: To test the effectiveness of reciprocal and self attention in TRSA, we conduct ablation study in Table XI. When we replace reciprocal attention with self attention (*i.e.*, two self attentions) or only use one self attention, the performance drops by 0.11∼0.17dB. One possible reason is that the model may be more focused on the reference frame rather than on the supporting frame during the computation of attention maps. In contrast, using the reciprocal attention can help the model to explicitly attend to the supporting frame and benefit from feature fusion. In addition, we can find that only using reciprocal attention is not enough. This is because reciprocal attention cannot preserve information of reference frames. Furthermore, we compare the performance of TRSA and parallel warping under different motion conditions. As shown in Table II, both of these two modules are effective and important for the model, but they show different characteristics. TRSA performs better on slow-motion videos, while parallel warping is better at fast-motion videos. We guess it is because TRSA allows for direct interactions within the attention window for small motions, but is less efficient than parallel warping in dealing with large motions outside of the attention window.

TABLE XI

ABLATION STUDY ON TEMPORAL RECIPROCAL SELF ATTENTION

Attention 1	Self Attn.	-	Reciprocal Attn.	Reciprocal Attn.
Attention 2	Self Attn.	Self Attn.	-	Self Attn.
PSNR	27.17	27.11	26.92	27.28

TABLE XII

ABLATION STUDY ON ATTENTION WINDOW SIZE (FRAME × HEIGHT × WIDTH)

Window Size	1 × 8 × 8	2 × 8 × 8	4 × 8 × 8	8 × 8 × 8
PSNR	27.10	27.13	27.18	27.28

3) Impact of Attention Window Size: We conduct ablation study in Table XII to investigate the impact of attention window size in the last few TRSAs of each scale. When the temporal window size increases from 1 to 2, the performance only improves slightly, possibly due to the fact that previous TRSA layers can already make good use of neighboring two-frame information. When the size is increased to 8, we can see an obvious improvement of 0.18dB. As a result, we use the window size of 8 × 8 × 8 for those layers.

V. CONCLUSION

In this paper, we proposed the Video Restoration Transformer (VRT) for video restoration. Based on a multi-scale framework, it jointly extracts, aligns, and fuses information from different frames at multiple resolutions by two kinds of modules: multiple temporal reciprocal self attention (TRSA) and parallel warping. More specifically, TRSA is composed of reciprocal and self attention. Reciprocal attention allows joint implicit flow estimation and feature warping, while self attention is responsible for feature extraction. Parallel warping is also used to further enhance feature alignment and fusion. Extensive experiments on various benchmark datasets show that VRT brings significant performance gains (up to 2.16dB) for various video restoration tasks, including video super-resolution, video deblurring, video denoising, video frame interpolation and space-time video super-resolution.

ACKNOWLEDGMENT

The authors would like to thank Dr. Gurkirt Singh for insightful discussions.

REFERENCES

[1] J. Caballero et al., "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 4778–4787.

[2] X. Wang, K. C. K. Chan, K. Yu, C. Dong, and C. C. Loy, "EDVR: Video restoration with enhanced deformable convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1954–1963.

[3] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, "TDAN: Temporally-deformable alignment network for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3360–3369.

[4] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1279–1288.

[5] S. Zhou, J. Zhang, J. Pan, W. Zuo, H. Xie, and J. Ren, "Spatio-temporal filter adaptive network for video deblurring," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2482–2491.

[6] D. Li et al., "ARVo: Learning all-range volumetric correspondence for video deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7721–7731.

[7] Y. Huang, W. Wang, and L. Wang, "Bidirectional recurrent convolutional networks for multi-frame super-resolution," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 235–243.

[8] M. S. M. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-recurrent video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6626–6634.

[9] D. Fuoli, S. Gu, and R. Timofte, "Efficient video super-resolution through recurrent latent space propagation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3476–3485.

[10] M. Haris, G. Shakhnarovich, and N. Ukita, "Recurrent back-projection network for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3897–3906.

[11] T. Isobe, X. Jia, S. Gu, S. Li, S. Wang, and Q. Tian, "Video super-resolution with recurrent structure-detail network," in *Proc. Eur. Conf. Comput.*, Aug. 2020, pp. 645–660.

[12] K. C. K. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, "BasicVSR: The search for essential components in video super-resolution and beyond," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4947–4956.

[13] K. C. K. Chan, S. Zhou, X. Xu, and C. Change Loy, "BasicVSR++: Improving video super-resolution with enhanced propagation and alignment," 2021, *arXiv:2104.13371*.

[14] J. Lin, Y. Huang, and L. Wang, "FDAN: Flow-guided deformable alignment network for video super-resolution," 2021, *arXiv:2105.05640*.

[15] Z. Zhong, Y. Gao, Y. Zheng, and B. Zheng, "Efficient spatio-temporal recurrent neural network for video deblurring," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 191–207.

[16] H. Son, J. Lee, J. Lee, S. Cho, and S. Lee, "Recurrent video deblurring with blur-invariant motion estimation and pixel volumes," *ACM Trans. Graph.*, vol. 40, no. 5, pp. 1–18, Oct. 2021.

[17] J. Cao, Y. Li, K. Zhang, and L. Van Gool, "Video super-resolution transformer," 2021, *arXiv:2106.06847*.

[18] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[19] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 286–301.

[20] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image restoration using Swin transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1833–1844.

[21] P. Yi, Z. Wang, K. Jiang, J. Jiang, and J. Ma, "Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3106–3115.

[22] X. Xiang, H. Wei, and J. Pan, "Deep video deblurring using sharpness features from exemplars," *IEEE Trans. Image Process.*, vol. 29, pp. 8976–8987, 2020.

[23] X. Xiang, Y. Tian, Y. Zhang, Y. Fu, J. P. Allebach, and C. Xu, "Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3367–3376.

[24] J. Pan, H. Bai, and J. Tang, "Cascaded deep video deblurring using temporal sharpness prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3043–3051.

[25] P. Yi et al., "Omniscient video super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4429–4438.

[26] G. Vaksman, M. Elad, and P. Milanfar, "Patch craft: Video denoising by deep modeling and patch matching," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2137–2146.

[27] R. Yang et al., "NTIRE 2021 challenge on quality enhancement of compressed video: Methods and results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 647–666.

[28] M. Tassano, J. Delon, and T. Veit, "DVDNET: A fast network for deep video denoising," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1805–1809.

[29] M. Tassano, J. Delon, and T. Veit, "FastDVDnet: Towards real-time deep video denoising without flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1354–1363.

[30] S. Li, F. He, B. Du, L. Zhang, Y. Xu, and D. Tao, "Fast spatio-temporal residual network for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10514–10523.

[31] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, Aug. 2019.

[32] D. Liu et al., "Robust video super-resolution with learned temporal dynamics," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2526–2534.

[33] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4161–4170.

[34] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, "Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3224–3232.

[35] J. Dai et al., "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[36] M. Suin and A. N. Rajagopalan, "Gated spatio-temporal attention-guided video deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7798–7807.

[37] A. Vaswani et al., "Attention is all you need," 2017, *arXiv:1706.03762*.

[38] N. Shazeer, "GLU variants improve transformer," 2020, *arXiv:2002.05202*.

[39] X. Li, Y. Hou, P. Wang, Z. Gao, M. Xu, and W. Li, "Trear: Transformer-based RGB-D egocentric action recognition," *IEEE Trans. Cognit. Develop. Syst.*, vol. 14, no. 1, pp. 246–252, Mar. 2022.

[40] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," 2021, *arXiv:2103.14030*.

[41] L. Liu et al., "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020.

[42] H. Chen et al., "Pre-trained image processing transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12299–12310.

[43] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, "Uformer: A general U-shaped transformer for image restoration," 2021, *arXiv:2106.03106*.

[44] A. Buades, B. Coll, and J.-M. Morel, "Nonlocal image and movie denoising," *Int. J. Comput. Vis.*, vol. 76, no. 2, pp. 123–139, Feb. 2008.

[45] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 3952–3966, Sep. 2012.

[46] C. Liu, H. Yang, J. Fu, and X. Qian, "Learning trajectory-aware transformer for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5687–5696.

[47] M. Song, Y. Zhang, and T. O. Aydın, "TempFormer: Temporally consistent transformer for video denoising," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 481–496.

[48] M. Liu, S. Jin, C. Yao, C. Lin, and Y. Zhao, "Temporal consistency learning of inter-frames for video super-resolution," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 4, pp. 1507–1520, Apr. 2023.

[49] D. Li et al., "A simple baseline for video restoration with grouped spatial–temporal shift," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 9822–9832.

[50] K. Zhou, W. Li, X. Han, and J. Lu, "Exploring motion ambiguity and alignment for high-quality video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22169–22179.

[51] W. Zhang, M. Zhou, C. Ji, X. Sui, and J. Bai, "Cross-frame transformer-based spatio-temporal video super-resolution," *IEEE Trans. Broadcast.*, vol. 68, no. 2, pp. 359–369, Jun. 2022.

[52] Z. Qiu, H. Yang, J. Fu, and D. Fu, "Learning spatiotemporal frequency-transformer for compressed video super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 257–273.

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

[54] W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1874–1883.

[55] A. Dosovitskiy et al., "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[56] Z. Liu et al., "Video Swin transformer," 2021, *arXiv:2106.13230*.

[57] S. Niklaus. (2018). *A Reimplementation of SPyNet Using PyTorch*. [Online]. Available: https://github.com/sniklaus/pytorch-spynet

[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[59] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.

[60] S. Nah et al., "NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1996–2005.

[61] C. Liu and D. Sun, "On Bayesian adaptive video super resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 346–360, Feb. 2014.

[62] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3883–3891.

[63] A. Khoreva, A. Rohrbach, and B. Schiele, "Video object segmentation with language referring expressions," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 123–141.

[64] X. Tao, H. Gao, X. Shen, and J. Wang, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE Conf. CVPR*, Jun. 2018, pp. 8174–8182.

[65] M. Suin, K. Purohit, and A. N. Rajagopalan, "Spatially-attentive patch-hierarchical network for adaptive motion deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3606–3615.

[66] S. W. Zamir et al., "Multi-stage progressive image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14821–14831.

[67] P. Arias and J.-M. Morel, "Video denoising via empirical Bayesian estimation of space-time patches," *J. Math. Imag. Vis.*, vol. 60, no. 1, pp. 70–93, Jan. 2018.

[68] X. Xu, L. Siyao, W. Sun, Q. Yin, and M.-H. Yang, "Quadratic video interpolation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1647–1656.

[69] T. Kalluri, D. Pathak, M. Chandraker, and D. Tran, "FLAVR: Flow-agnostic video representations for fast frame interpolation," 2020, *arXiv:2012.08512*.

[70] K. Soomro, A. Roshan Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.

[71] W. Bao, W. Lai, C. Ma, X. Zhang, Z. Gao, and M. Yang, "Depth-aware video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3703–3712.

[72] M. Choi, H. Kim, B. Han, N. Xu, and K. L. Lee, "Channel attention is all you need for video frame interpolation," in *Proc. 34th AAAI Conf. Artif. Intell. (AAAI)*, vol. 34, no. 7, 2020, pp. 10663–10671.

[73] H. Jiang, D. Sun, V. Jampani, M. Yang, E. Learned-Miller, and J. Kautz, "Super SloMo: High quality estimation of multiple intermediate frames for video interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9000–9008.

[74] H. Lee, T. Kim, T.-Y. Chung, D. Pak, Y. Ban, and S. Lee, "AdaCoF: Adaptive collaboration of flows for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5316–5325.

[75] M. Haris, G. Shakhnarovich, and N. Ukita, "Space-Time-Aware multi-resolution video enhancement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2856–2865.

[76] G. Xu, J. Xu, Z. Li, L. Wang, X. Sun, and M.-M. Cheng, "Temporal modulation network for controllable space-time video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6384–6393.

[77] Z. Geng, L. Liang, T. Ding, and I. Zharkov, "RSTT: Real-time spatial temporal transformer for space-time video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17420–17430.