Learning Trajectory-Aware Transformer for Video Super-Resolution

Chengxu Liu[1]*, Huan Yang[2], Jianlong Fu[2], Xueming Qian[1]
[1]Xi'an Jiaotong University [2]Microsoft Research Asia
liuchx97@gmail.com, {huayan, jianf}@microsoft.com, qianxm@mail.xjtu.edu.cn

Abstract

*Video super-resolution (VSR) aims to restore a sequence of high-resolution (HR) frames from their low-resolution (LR) counterparts. Although some progress has been made, there are grand challenges to effectively utilize temporal dependency in entire video sequences. Existing approaches usually align and aggregate video frames from limited adjacent frames (e.g., 5 or 7 frames), which prevents these approaches from satisfactory results. In this paper, we take one step further to enable effective spatio-temporal learning in videos. We propose a novel **T**rajectory-aware **T**ransformer for **V**ideo **S**uper-**R**esolution (TTVSR). In particular, we formulate video frames into several pre-aligned trajectories which consist of continuous visual tokens. For a query token, self-attention is only learned on relevant visual tokens along spatio-temporal trajectories. Compared with vanilla vision Transformers, such a design significantly reduces the computational cost and enables Transformers to model long-range features. We further propose a cross-scale feature tokenization module to overcome scale-changing problems that often occur in long-range videos. Experimental results demonstrate the superiority of the proposed TTVSR over state-of-the-art models, by extensive quantitative and qualitative evaluations in four widely-used video super-resolution benchmarks. Both code and pre-trained models can be downloaded at https://github.com/researchmm/TTVSR.*

1. Introduction

Video super-resolution (VSR) aims to recover a high-resolution (HR) video from a low-resolution (LR) counterpart [39]. As a fundamental task in computer vision, VSR is usually adopted to enhance visual quality, which has great value in many practical applications, such as video surveillance [48], high-definition television [10], and satellite imagery [6, 27], etc. From a methodology perspective, unlike image super-resolution that usually learns on spatial dimensions, VSR tasks pay more attention to exploiting temporal information. In Fig. 1, if detailed textures to recover the target frame can be discovered and leveraged at relatively distant frames, video qualities can be greatly enhanced.

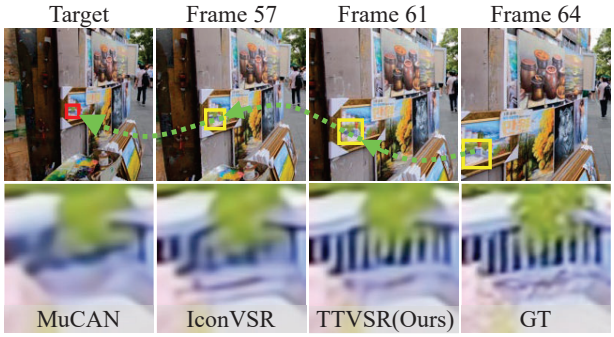

Figure 1. A comparison between TTVSR and other SOTA methods: MuCAN [24] and IconVSR [4]. We introduce finer textures for recovering the target frame from the boxed areas (indicated by yellow) tracked by the trajectory (indicated by green).

To solve this challenge, recent years have witnessed an increasing number of VSR approaches, which can be categorized into two paradigms. The former makes attempts to utilize adjacent frames as inputs (e.g., 5 or 7 frames), and align temporal features in an implicit [18, 23] or explicit manners [34, 39]. One of the classic works is EDVR that adopts deformable convolutions to capture features within a sliding window [39]. However, larger window sizes will dramatically increase computational costs which makes this paradigm infeasible to capture distant frames. The latter investigates temporal utilization by recurrent mechanisms [4, 32, 44]. One of the representative works is IconVSR that uses a hidden state to convey relevant features from entire video frames [4]. Nonetheless, recurrent networks usually lack long-term modeling capability due to vanishing gradient [12], which inevitably leads to unsatisfied results as shown in Fig. 1.

Inspired by the recent progress of Transformer in natural language processing [36], significant progresses have been made in both visual recognition [3, 8] and generation tasks [43, 46]. For example, MuCAN proposes to use attention mechanisms to aggregate inter-frame features [24]

*This work was done while Chengxu Liu was a research intern at Microsoft Research Asia.

for VSR tasks. However, due to the high computational complexity in a video, it only learns from a narrow temporal window, which results in sub-optimal performance as shown in Fig. 1. Therefore, exploring proper ways of utilizing Transformers in videos remains a big challenge.

In this paper, we propose a novel Trajectory-aware Transformer to enable effective video representation learning for Video Super-Resolution (TTVSR). The key insight of TTVSR is to formulate video frames into pre-aligned trajectories of visual tokens, and calculate $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$ in the same trajectory. In particular, we learn to link relevant visual tokens together along temporal dimensions, which forms multiple trajectories to depict object motions in a video (e.g., the green trajectory in Fig. 1). We update token trajectories by a proposed location map that online aggregates pixel motions around a token by average pooling. Once video trajectories have been learned, TTVSR calculates self-attention only on the most relevant visual tokens that are located in the same trajectory. Compared with MuCAN that calculates attention across visual tokens in space and time [24], the proposed TTVSR significantly reduces the computational cost and thus makes long-range video modeling practicable.

To further deal with the scale-changing problem that often occur in long-range videos (e.g., the yellow boxes in Fig. 1), we devise a cross-scale feature tokenization module and enhance feature representations from multiple scales. Our contributions are summarized as follows:

- We propose a novel trajectory-aware Transformer, which is one of the first works to introduce Transformer into video super-resolution tasks. Our method significantly reduces computational costs and enables long-range modeling in videos.
- Extensive experiments demonstrate that the proposed TTVSR can significantly outperform existing SOTA methods in four widely-used VSR benchmarks. In the most challenging REDS4 dataset, TTVSR gains 0.70db and 0.45db PSNR improvements than BasicVSR and IconVSR, respectively.

2. Related Work

2.1. Video Super-Resolution

In VSR tasks, it is crucial to assist frame recovery with other frames in the sequence. Therefore, according to the number of input frames, VSR tasks can be mainly divided into two kinds of paradigms: based on sliding-window structure [1, 2, 15, 19, 20, 24, 34, 39, 41, 45] and based on recurrent structure [4, 9, 11, 13, 14, 16, 32, 44].

Sliding-window structure. The methods based on sliding-window structure use adjacent frames within a sliding window as inputs to recover the HR frame (e.g., 5 or 7 frames). They mainly focus on using 2D or 3D CNN [15, 17, 18, 23], optical flow estimation [1, 20, 33] or deformable convolutions [5, 34, 39] to design advanced alignment modules and fuse detailed textures from adjacent frames. Typically, to fully utilize the complementary information across frames, FSTRN [23] presented a fast spatio-temporal residual network for VSR by adopting 3D convolutions [35]. To better align adjacent frames, VESCPN [1] introduced a spatio-temporal sub-pixel convolution network and first combined the motion compensation and VSR together. EDVR [39] and TDAN [34] used deformable convolutions [5] to align adjacent frames. However, they cannot utilize textures at other moments, especially in relatively distant frames.

Recurrent structure. Rather than aggregating information from adjacent frames, methods based on recurrent structure use a hidden state to convey relevant information in previous frames. FRVSR [32] used the previously SR frame to recover the subsequent frame. Inspired by the back-projection, RBPN [11] treated each frame as a separate source, which is combined in an iterative refinement framework. RSDN [14] divided the input into structure and detail components and proposed the two-steam structure-detail block to learn textures. Representatively, OVSR [44], BasicVSR [4], and IconVSR [4] fused the bidirectional hidden state from the past and future for reconstruction and got significant improvements. They try to fully utilize the information of the whole sequence and synchronously update the hidden state by the weights of reconstruction network. However, due to the vanishing gradient [12], this mechanism makes the updated hidden state loses the long-term modeling capabilities to some extent.

2.2. Vision Transformer

Recently, Transformer [36] has been proposed to improve the long-term modeling capabilities of sequence in various fields [7, 8]. In the field of computer vision [8], Transformer is used as a new attention-based module to model relationships between tokens in many image-based tasks, such as classification [8], inpainting [46], super-resolution [43], generation [47] and so on. Typically, ViT [8] unfolded an image into patches as tokens for attention to capture the long-range relationship in high-level vision. TTSR [43] proposed a texture Transformer in low-level vision to search relevant texture patches from Ref image to LR image.

In VSR tasks, VSR-Transformer [2] and MuCAN [24] tried to use attention mechanisms for aligning different frames with great success. However, due to the heavy computational costs of attention calculation on videos, these methods only aggregate information on the narrow temporal window. Therefore, in this paper, we introduce a trajectory-aware Transformer to improve the long-term modeling capabilities for VSR tasks while keeping the computational cost of attention within an acceptable range.

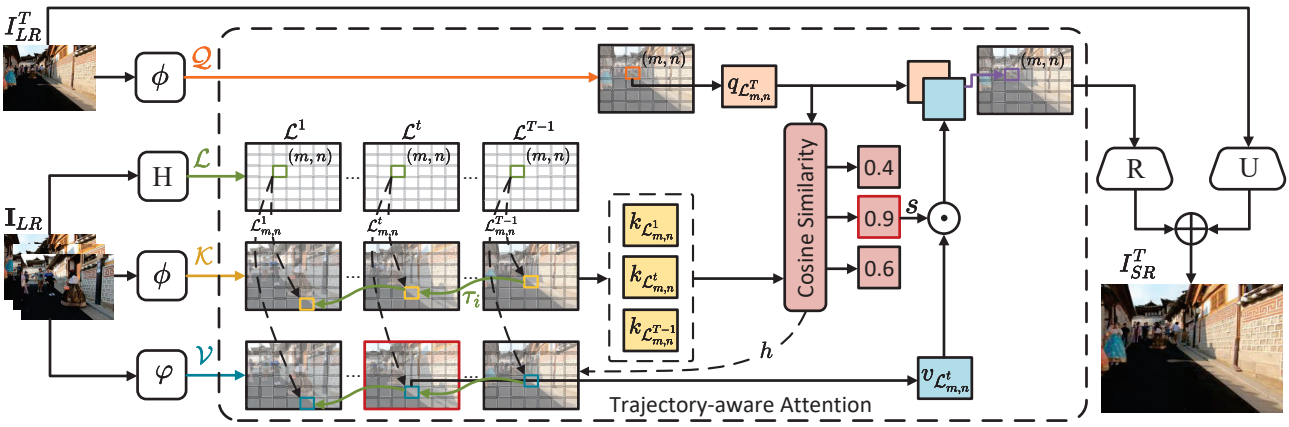

Figure 2. The overview of TTVSR based on location maps. $\mathcal{Q}$, $\mathcal{K}$ and $\mathcal{V}$ are tokens from video frames extracted by embedding networks $\phi(\cdot)$ and $\varphi(\cdot)$, respectively. τ_i indicates a trajectory of $\mathcal{T}$. $\mathcal{L}$ is the set of location map generated by the motion estimation network H. The dotted lines indicates the indexing operation from $\mathcal{K}$ and $\mathcal{V}$ by location maps $\mathcal{L}$ and hard index h. R$(\cdot)$ represents the reconstruction network followed by a pixel-shuffle layer to resize feature maps to the desired size. U$(\cdot)$ represents the bicubic upsampling operation. $\odot$ and $\oplus$ indicate multiplication and element-wise addition, respectively.

3. Our Approach

In this section, we first introduce the proposed **T**rajectory-aware **T**ransformer for **V**ideo **S**uper-**R**esolution (TTVSR) in Sec. 3.1, and then discuss the proposed location map for trajectory generation in Sec. 3.2. Finally, we refocus to our Transformer design based on the location maps and discuss its advantages in Sec. 3.3.

3.1. Trajectory-Aware Transformer

We introduce the formulation of the TTVSR firstly, followed by trajectory-aware attention and cross-scale feature tokenization. More illustrations can be found in Fig. 2.

Formulation. Given a LR sequence, the goal of VSR tasks is to recover a HR version. Specifically, for our task, when restoring the T^{th} frame I^T_{SR}, we denote the current LR frame as I^T_{LR} and other LR frames as $\mathbf{I}_{LR} = \{I^t_{LR}, t \in [1, T-1]\}$.

We use two embedding networks $\phi(\cdot)$ and $\varphi(\cdot)$ to get features from video frames and extract tokens by sliding-windows. The queries $\mathcal{Q}$ and keys $\mathcal{K}$ are extracted by $\phi(\cdot)$ and denoted as $\mathcal{Q} = \phi(I_{LR}) = \{q^T_i, i \in [1, N]\}$ and $\mathcal{K} = \phi(\mathbf{I}_{LR}) = \{k^t_i, i \in [1, N], t \in [1, T-1]\}$, respectively. The values are extracted by $\varphi(\cdot)$ and denoted as $\mathcal{V} = \varphi(\mathbf{I}_{LR}) = \{v^t_i, i \in [1, N], t \in [1, T-1]\}$.

The trajectories $\mathcal{T}$ in our approach can be formulated as a set of trajectory, in which each trajectory τ_i is a sequence of coordinate over time and the end point of trajectory τ_i is associated with the coordinate of token q_i:

$$\begin{aligned} \mathcal{T} &= \{\tau_i,\ i \in [1, N]\}, \\ \tau_i &= \langle \tau^t_i = (x^t_i, y^t_i),\ t \in [1, T] \rangle, \end{aligned} \tag{1}$$

where $x^t_i \in [1, H]$, $y^t_i \in [1, W]$, and (x^t_i, y^t_i) represents the coordinate of trajectory τ_i at time t. H and W represents the height and width of the feature maps, respectively.

From the aspect of trajectories, the inputs of proposed trajectory-aware transformer can be further represented as visual tokens which are aligned by trajectories $\mathcal{T}$:

$$\begin{aligned} \mathcal{T} &= \{\tau_i,\ i \in [1, N]\}, \\ \mathcal{Q} &= \{q_{\tau^T_i},\ i \in [1, N]\}, \\ \mathcal{K} &= \{k_{\tau^t_i},\ i \in [1, N],\ t \in [1, T-1]\}, \\ \mathcal{V} &= \{v_{\tau^t_i},\ i \in [1, N],\ t \in [1, T-1]\}. \end{aligned} \tag{2}$$

The process of recovering the T^{th} HR frame I^T_{SR} can be further expressed as:

$$\begin{aligned} I^T_{SR} &= \mathrm{T}_{traj}(\mathcal{Q}, \mathcal{K}, \mathcal{V}, \mathcal{T}) \\ &= \mathrm{R}(\underset{\tau_i \in \mathcal{T}}{\mathrm{A}_{traj}}(q_{\tau^T_i}, k_{\tau^t_i}, v_{\tau^t_i})) + \mathrm{U}(I^T_{LR}), \end{aligned} \tag{3}$$

where $\mathrm{T}_{traj}(\cdot)$ denotes the trajectory-aware Transformer. $\mathrm{A}_{traj}(\cdot)$ denotes the trajectory-aware attention. R$(\cdot)$ represents the reconstruction network followed by a pixel-shuffle layer to resize feature maps to the desired size. U$(\cdot)$ represents the bicubic upsampling operation.

By introducing trajectories into Transformer, the attention calculation on $\mathcal{K}$ and $\mathcal{V}$ can be significantly reduced because it can avoid the computation on spatial dimension compared with vanilla vision Transformers.

Trajectory-aware attention. Thanks to the powerful long-range model ability, the attention mechanisms in vanilla vision Transformer is used to model dependencies of tokens within an image [3, 8]. However, empowering the attention mechanisms to videos remains a challenge. Thus, we propose a trajectory-aware attention module, which integrates relevant visual tokens located in the same spatio-temporal trajectories with less computational costs.

Different from the traditional attention mechanisms that take a weighted sum of keys in temporal. We use hard attention to select the most relevant token along trajectories, its purpose is to reduce blur introduced by weighted sum. We use soft attention to generate the confidence of relevant patches, it is used to reduce the impact of irrelevant tokens when hard attention gets inaccurate results. We use h_{τ_i} and s_{τ_i} to represent the results of hard and soft attention. The calculation process can be formulated as:

$$h_{\tau_i} = \arg\max_t \langle \frac{q_{\tau_i^T}}{\| q_{\tau_i^T} \|_2^2}, \frac{k_{\tau_i^t}}{\| k_{\tau_i^t} \|_2^2} \rangle, \\ s_{\tau_i} = \max_t \langle \frac{q_{\tau_i^T}}{\| q_{\tau_i^T} \|_2^2}, \frac{k_{\tau_i^t}}{\| k_{\tau_i^t} \|_2^2} \rangle. \tag{4}$$

Based on such formula, the attention calculation in Equ. 3 can be formulated as:

$$\mathrm{A}_{traj}(q_{\tau_i^T}, k_{\tau_i}, v_{\tau_i}) = \mathrm{C}(q_{\tau_i^T}\ ,\ s_{\tau_i} \odot v_{\tau_i^{h_{\tau_i}}}), \tag{5}$$

where the operator $\odot$ denotes multiplication. $\mathrm{C}(\cdot)$ denotes the concatenation operation. We fold all the tokens and output a feature map.

In general, in the proposed trajectory-aware attention, we integrate features from the whole sequence. Such a design allows attention calculation only along its spatio-temporal trajectory, mitigating the computational cost.

Cross-scale feature tokenization. The premise of utilizing multi-scale texture from sequences is that the model can adapt to the multi-scale variations in content that often occur. Therefore, we propose a cross-scale feature tokenization module before trajectory-aware attention to extract tokens from multiple scales. It can uniform multi-scale features into a uniform-length token and allows rich textures from larger scales to be utilized for the recovery of smaller ones in the attention mechanism.

Specifically, we follow three steps to extract tokens. First, the successive unfold and fold operations are used to expand the receptive field of features. Second, features from different scales are shrunk to the same scale by a pooling operation. Third, the features are split by unfolding operation to obtain the output tokens. It is noteworthy that this process can extract features from a larger scale while keeping the same size as output tokens. It is convenient for attention calculation and token integration. More analyses can be found in the supplementary.

3.2. Location Maps for Trajectory Generation

Existing approaches use feature alignment and global optimization to calculate trajectories of video which are time-consuming and less efficient [30, 37, 38]. Especially in our task, trajectories are updated over time, the computation cost will be further exploded. To solve this problem, we propose a location map for trajectory generation in which the location maps are represented as a group of matrices over time. By such a design, the trajectory generation can be expressed as some matrix operations which are both efficient for computing and friendly for model implementation.

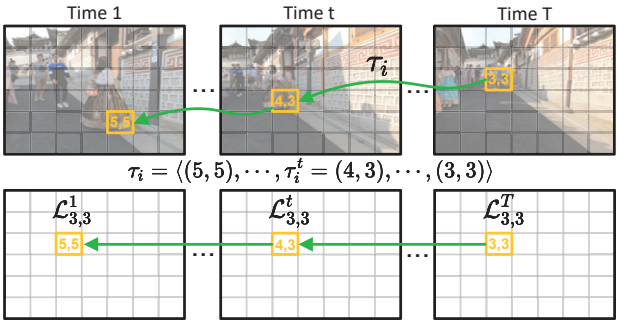

Figure 3. An illustration of the relationship between trajectory τ and the location maps $\mathcal{L}$ at time t.

Since the trajectories are updated over time, our location maps need also to be updated accordingly. In the formulation of it, we fix the time to T for better illustration. The proposed location maps can be formulated as:

$$\mathcal{L}^t = \begin{bmatrix} (x_1, y_1) & \dots & (x_1, y_W) \\ \dots & \dots & \dots \\ (x_H, y_1) & \dots & (x_H, y_W) \end{bmatrix}, t \in [1, T], \tag{6}$$

where $\mathcal{L}^t_{m,n}$ represents the coordinate at time t in a trajectory which is ended at (m, n) at time T. The relationship between the location map $\mathcal{L}^t_{m,n}$ and the trajectory τ_i^t defined in Equ. 1 can be further expressed as:

$$\mathcal{L}^t_{m,n} = \tau_i^t, \text{ where } \tau_i^T = (m, n),\ i \in [1, N], \tag{7}$$

where $m \in [1, H]$ and $n \in [1, W]$. In Fig. 3, we use a simple case to further illustrate the relationship between location maps and trajectories.

Location map updating. As discussed in the formulation part, the location maps will change over time. We denotes the updated location maps as ${}^*\mathcal{L}^t$. When changing from time T to time $T+1$, a new location map ${}^*\mathcal{L}^{T+1}$ at time $T+1$ should be initialized. Based on Equ. 7, the element values of ${}^*\mathcal{L}^{T+1}$ are exactly the coordinates of frame $T+1$ [1].

Then the rest updated location maps $\{{}^*\mathcal{L}^1, \cdots, {}^*\mathcal{L}^T\}$ can be obtained by tracking the location maps $\{\mathcal{L}^1, \cdots, \mathcal{L}^T\}$ from time $T+1$ to time T using backward flow O^{T+1}. Specifically, O^{T+1} can build the connection of trajectories between time T and time $T+1$ and obtain from a lightweight motion estimation network. Due to the correlations in flow are usually float numbers, we get the updated coordinates in location map $\mathcal{L}^t$ by interpolating between its adjacent coordinates:

$${}^*\mathcal{L}^t = \mathrm{S}(\mathcal{L}^t, O^{T+1}), \tag{8}$$

[1] Where the element values of the matrix are equal to the index matrix.

where $\mathrm{S}(\cdot)$ represents the spatial sampling operation by spatial correlation O^{T+1} (i.e., $grid\_sample$ in PyTorch). Thus far, we have all the updated location maps for time $T+1$.

With the careful design of the location maps, the trajectories in our proposed trajectory-aware Transformer can be effectively calculated and maintained through one parallel matrix operation (i.e., the operation $\mathrm{S}(\cdot)$). More analyses can be found in the supplementary.

3.3. TTVSR based on Location Maps

In this section, we recap the formulation of our proposed TTVSR in Sec. 3.1 and show the relation between TTVSR and location maps in a more intrinsical way. More details can be found in Fig. 2. Since the location map $\mathcal{L}^t$ in Equ. 7 is an interchangeable formulation of trajectory τ_i in Equ. 3, the proposed TTVSR can be further expressed as:

$$\begin{aligned} I_{SR}^{T} &= \mathrm{T}_{traj}(\mathcal{Q}, \mathcal{K}, \mathcal{V}, \mathcal{L}) \\ &= \mathrm{R}(\underset{t,m,n}{\mathrm{A}_{traj}}(q_{\mathcal{L}^T_{m,n}}, k_{\mathcal{L}^t_{m,n}}, v_{\mathcal{L}^t_{m,n}})) + \mathrm{U}(I_{LR}^{T}), \end{aligned} \tag{9}$$

where $m \in [1, H]$, $n \in [1, W]$, and $t \in [1, T-1]$.

In this formulation, we transform the coordinate system in our transformer from the one defined by trajectories to a group of aligned matrices (i.e., the location maps). Such a design has two advantages: First, the location maps provide a more efficient way to enable our TTVSR can directly leverage the information from a distant video frame. Second, as the trajectory is a widely used concept in videos, our design can motivate other video tasks to achieve a more efficient and powerful implementation.

3.4. Training Details

For fair comparisons, we follow IconVSR [4] and VSR-Transformer [2] to use the same feature extraction network, reconstruction network, and pre-trained SPyNet [31] for motion estimation. To leverage the information of the whole sequence, we follow previous works [4, 13] to adopt a bidirectional propagation scheme, where the features in different frames can be propagated backward and forward, respectively. To reduce consumption in terms of time and memory, we generate the visual tokens of different scales from different frames. Features from adjacent frames are finer, so we generate tokens of size 1×1. Features from a long distance are coarser, so we select these frames at a certain time interval and generate tokens of size 4×4. Besides, in Sec. 3.1, we use kernels of size 4×4, 6×6, and 8×8 for cross-scale feature tokenization. During training, we use Cosine Annealing scheme [26] and Adam [21] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The learning rates of the motion estimation and other parts are set as 1.25×10^{-5} and 2×10^{-4}, respectively. We set the batch size as 8 and input patch size as 64×64. To keep fair comparison, we augment the training data with random horizontal flips, vertical flips, and 90° rotations. Besides, to enable long-range sequence capability, we use sequences with a length of 50 as inputs. The Charbonnier penalty loss [22] is applied on whole frames between the ground-truth I_{HR} and restored SR frame I_{SR}, which can be defined by $\ell = \sqrt{\|I_{HR} - I_{SR}\|^2 + \varepsilon^2}$. To stabilize the training of TTVSR, we fix the weights of the motion estimation module in the first 5K iterations, and make it trainable later. The total number of iterations is 400K.

4. Experiments

4.1. Datasets and Metrics

We evaluate the proposed TTVSR and compare its performance with other SOTA approaches on two widely-used datasets: **REDS** [29] and **Vimeo-90K** [42]. For **REDS** [29], it is published in the NTIRE19 challenge [29]. It contains a total of 300 video sequences, in which 240 for training, 30 for validation, and 30 for testing. Each sequence contains 100 frames with a resolution of 720×1280. To create training and testing sets, we follow previous works [4, 24, 39] to select four sequences[2] as the testing set which is called **REDS4** [29]. And we select the rest 266 sequences from the training and validation set as the training set. For **Vimeo-90K** [42], it contains 64,612 sequences for training and 7,824 for testing. Each sequence contains seven frames with a resolution of 448×256. For fair comparison, we follow previous works [4] to evaluate TTVSR with $4\times$ downsampling by using two degradations: 1) MATLAB bicubic downsample (BI), and 2) Gaussian filter with a standard deviation of $\sigma = 1.6$ and downsampling (BD). Same with previous works [14, 15, 24, 34], we apply the BI degradation on **REDS4** [29] and BD degradation on **Vimeo-90K-T** [42], **Vid4** [25] and **UDM10** [45]. We keep the same evaluation metrics: 1) Peak signal-to-noise ratio (PSNR) and 2) structural similarity index (SSIM) [40] as previous works [4, 24].

4.2. Comparisons with State-of-the-art Methods

We compare TTVSR with 15 start-of-the-art methods. These methods can be summarized into three categories: single image super-resolution (SISR) [28, 49], sliding window-based [2, 15, 17, 24, 34, 39, 42], and recurrent structure-based [4, 9, 11, 14, 32]. For fair comparisons, we obtain the performance from their original paper or reproduce results by authors officially released models.

Quantitative comparison. We compare TTVSR with other SOTA methods on the most widely-used REDS dataset [29]. As shown in Tab. 1, we categorize these approaches according to the frames used in each inference. Among them, since only one LR frame is used, the performance of SISR methods [28, 49] is very limited. MuCAN [24]

[2] Clips 000,011,015,020 of the REDS training set.

Table 1. Quantitative comparison (PSNR↑ and SSIM↑) on the REDS4 [29] dataset for 4× video super-resolution. The results are tested on RGB channels. Red indicates the best and blue indicates the second best performance (best view in color). #Frame indicates the number of input frames required to perform an inference, and "r" indicates to adopt the recurrent structure.

Method	#Frame	Clip_000	Clip_011	Clip_015	Clip_020	Average
Bicubic	1	24.55/0.6489	26.06/0.7261	28.52/0.8034	25.41/0.7386	26.14/0.7292
RCAN [49]	1	26.17/0.7371	29.34/0.8255	31.85/0.8881	27.74/0.8293	28.78/0.8200
CSNLN [28]	1	26.17/0.7379	29.46/0.8260	32.00/0.8890	27.69/0.8253	28.83/0.8196
TOFlow [42]	7	26.52/0.7540	27.80/0.7858	30.67/0.8609	26.92/0.7953	27.98/0.7990
DUF [17]	7	27.30/0.7937	28.38/0.8056	31.55/0.8846	27.30/0.8164	28.63/0.8251
EDVR [39]	7	28.01/0.8250	32.17/0.8864	34.06/0.9206	30.09/0.8881	31.09/0.8800
MuCAN [24]	5	27.99/0.8219	31.84/0.8801	33.90/0.9170	29.78/0.8811	30.88/0.8750
VSR-T [2]	5	28.06/0.8267	32.28/0.8883	34.15/0.9199	30.26/0.8912	31.19/0.8815
BasicVSR [4]	r	28.39/0.8429	32.46/0.8975	34.22/0.9237	30.60/0.8996	31.42/0.8909
IconVSR [4]	r	28.55/0.8478	32.89/0.9024	34.54/0.9270	30.80/0.9033	31.67/0.8948
TTVSR	r	28.82/0.8566	33.47/0.9100	35.01/0.9325	31.17/0.9094	32.12/0.9021

Table 2. Quantitative comparison (PSNR↑ and SSIM↑) on Vid4 [25], UDM10 [45] and Vimeo-90K-T [42] dataset for 4× video super-resolution. All the results are calculated on Y-channel. Red indicates the best and blue indicates the second best performance (best view in color).

Method	Vid4 [25]	UDM10 [45]	Vimeo-90K-T [42]
Bicubic	21.80/0.5246	28.47/0.8253	31.30/0.8687
TOFlow [42]	25.85/0.7659	36.26/0.9438	34.62/0.9212
FRVSR [32]	26.69/0.8103	37.09/0.9522	35.64/0.9319
DUF [17]	27.38/0.8329	38.48/0.9605	36.87/0.9447
RBPN [11]	27.17/0.8205	38.66/0.9596	37.20/0.9458
RLSP [9]	27.48/0.8388	38.48/0.9606	36.49/0.9403
EDVR [39]	27.85/0.8503	39.89/0.9686	37.81/0.9523
TDAN [34]	26.86/0.8140	38.19/0.9586	36.31/0.9376
TGA [15]	27.59/0.8419	39.05/0.9634	37.59/0.9516
RSDN [14]	27.92/0.8505	39.35/0.9653	37.23/0.9471
BasicVSR [4]	27.96/0.8553	39.96/0.9694	37.53/0.9498
IconVSR [4]	28.04/0.8570	40.03/0.9694	37.84/0.9524
TTVSR	28.40/0.8643	40.41/0.9712	37.92/0.9526

and VSR-T [2] use attention mechanisms in sliding window, which has a significant improvement over the SISR methods. However, they do not fully utilize the information of the sequence. BasicVSR [4] and IconVSR [4] try to model the whole sequence through hidden states. Nonetheless, the well-known vanishing gradient issue limits their capabilities of long-term modeling, thus the information at a distance will be lost. Different from them, our TTVSR tries to link the relevant visual token together along the same trajectory in an efficient way. TTVSR also uses the whole sequence information to recover the lost textures. Due to such merits, TTVSR achieves a result of 32.12dB PSNR and significantly outperforms IconVSR [4] by **0.45dB** on the REDS4 [29]. This large margin demonstrates the power of TTVSR in long-range modeling.

To further verify the generalization capabilities of TTVSR, we train TTVSR on Vimeo-90K dataset [42], and evaluate the results on Vid4 [25], UDM10 [45], and Vimeo-90K-T datasets [42], respectively. As shown in Tab. 2,

Table 3. Comparison of params, FLOPs and numbers. FLOPs is computed on one LR frame with the size of 180 × 320 and ×4 upsampling on the REDS4 [29] dataset.

Method	#Params(M)	FLOPs(T)	PSNR/SSIM
DUF [17]	5.8	2.34	28.63/0.8251
RBPN [11]	12.2	8.51	30.09/0.8590
EDVR [39]	20.6	2.95	31.09/0.8800
MuCAN [24]	13.6	>1.07	30.88/0.8750
BasicVSR [4]	6.3	0.33	31.42/0.8909
IconVSR [4]	8.7	0.51	31.67/0.8948
TTVSR	6.8	0.61	32.12/0.9021

on the Vid4 [25], UDM10 [45], and Vimeo-90K-T [42] test sets, TTVSR achieves the results of 28.40dB, 40.41dB, and 37.92dB in PSNR respectively, which is superior to other SOTA methods. Specifically, on the Vid4 [25] and UDM10 [45] datasets, TTVSR outperforms IconVSR [4] by **0.36dB** and **0.38dB** respectively. At the same time, we notice that compared with the evaluation on Vimeo-90K-T [42] dataset with only seven frames in each testing sequence, TTVSR has a better improvement on other datasets which have at least 30 frames per video. The results verify that TTVSR has strong generalization capabilities and is good at modeling the information in long-range sequences.

Qualitative comparison. To further compare visual qualities of different approaches, we show visual results generated by TTVSR and other SOTA methods on four different test sets in Fig. 4. For fair comparisons, we either directly take the original SR images of the author-released or use author-released models to get results. It can be observed that TTVSR has a great improvement in visual quality, especially for areas with detailed textures. For example, in the fourth row in Fig. 4, TTVSR can recover more striped details from the stonework in the oil painting. The results verify that TTVSR can utilize textures from relevant tokens to produce finer results. More visual results can be found in the supplementary materials.

Model sizes and computational costs. In real applica-

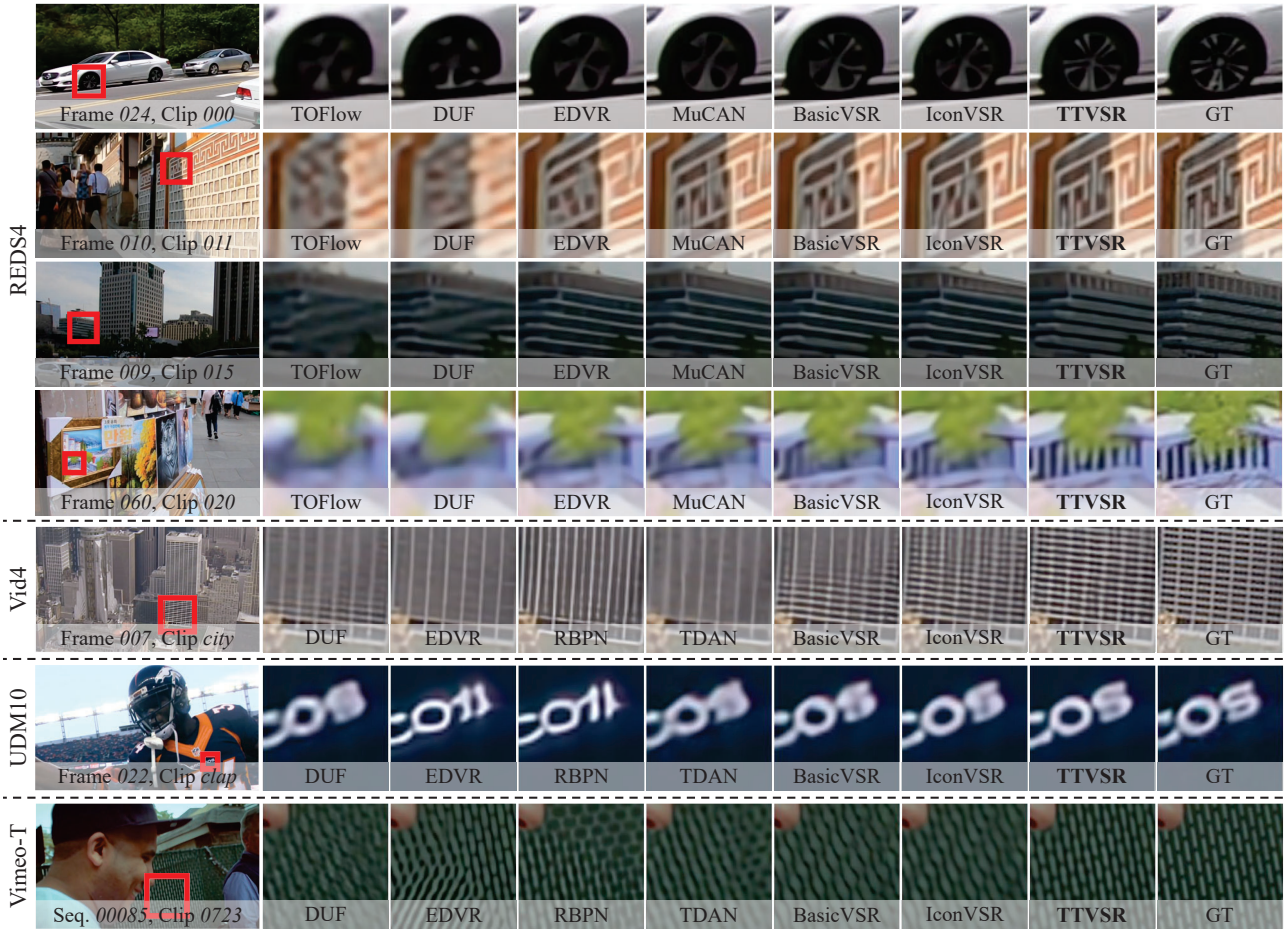

Figure 4. Visual results on REDS4 [29], Vid4 [25], UDM10 [45] and Vimeo-90K-T [42] for 4× scaling factor. The frame number is shown at the bottom of each case. Zoom in to see better visualization.

tions, model sizes and computational costs are usually important. To avoid the gap between different hardware devices, we use two hardware-independent metrics, including the number of parameters (#Params) and FLOPs. As shown in Tab. 3, the FLOPs are computed with the input of LR size 180×320 and $\times 4$ upsampling settings. Compared with IconVSR [4], TTVSR achieves higher performance while keeping comparable #Params and FLOPs. Besides, it should be emphasized that our method is much lighter than MuCAN [24] which is the SOTA attention-based method. Such superior performances mainly benefit from the use of trajectories in attention calculation which significantly reduces computational costs.

4.3. Ablation Study

In this section, we conduct the ablation study on the proposed trajectory-aware attention and study the influence of frames number used in this module. In addition, we further analyze the effect of the cross-scale feature tokenization.

Trajectory-aware attention. Trajectory generation (TG) is a prerequisite for trajectory-aware attention (TA), so we study them together in this part. We directly use convolution layers to integrate the aligned previous tokens and current token as our "Base" model. We denote the model that aggregates the most relevant tokens on the trajectory as our "Base+TG" model. We denote the model that adds trajectory-aware attention progressively as our "Base+TG+TA" model. The results are shown in Tab. 4. With the addition of TG, PSNR can be improved from 30.46 to 31.91, which verifies that the trajectory can link relevant visual tokens together precisely. When TA is involved, we integrate tokens from trajectories, and the performance is improved to 31.99. This demonstrates the superiority of TA

Table 4. Ablation study results of trajectory-aware attention module on the REDS4 [29] dataset. TG: trajectory generation. TA: trajectory-aware attention.

Method	TG	TA	PSNR/SSIM
Base			30.46/0.8661
Base+TG	✓		31.91/0.8985
Base+TG+TA	✓	✓	**31.99/0.9007**

Table 5. Ablation study results of the frame number used on the REDS4 [29] dataset.

#Frame	5	10	20	33	45
PSNR	31.89	31.93	31.97	31.99	32.01
SSIM	0.8984	0.8994	0.9005	0.9007	0.9004

Base Base+TG Base+TG+TA GT

Figure 5. Ablation study on the trajectory generation (TG) and trajectory-aware attention (TA) on the REDS4 [29] dataset.

for modeling long-range information. We further explore the visual differences as shown in Fig. 5. TG can capture the relevant tokens, while TA integrates tokens into the current frame to produce clearer textures.

Influence of frame number during inference. To explore the influence of the frame number used during inference on the ability of modeling long-range sequences. As shown in Tab. 5, we use different temporal intervals to sample frames from the entire sequence (100 frames). The performance is positively correlated with the number of sampled frames. It demonstrates the effectiveness of the trajectory-aware attention module for long-range modeling. However, the performance gain gradually decreases when the frame number is more than 45. It indicates that choosing three as the temporal interval (i.e., 33 frames) is sufficient to model the entire sequences. Using smaller intervals may not provide more information since the adjacent frames are too similar.

Cross-scale feature tokenization. To alleviate the scale-changing problem in sequences, we discuss the impact of token size in the cross-scale feature tokenization (CFT). As shown in Tab. 6, the first three rows of results show that CFT can extract richer textures as the token scale increases. The performance can improve PSNR from 31.99 to 32.12, indicating that CFT can adapt to scale changes in sequences. In addition, according to the visualizations, as shown in Fig. 6, cross-scale feature tokenization can introduce finer textures from a larger scale, avoiding the loss of textures caused by scale-changing in long-range sequences. It is also observed that using the larger scale (e.g., 12) leads to undesirable results. This is because oversized tokens are not conducive to textures learning. In our model, we choose 4, 6, and 8 scales as the token size in CFT.

5. Limitations

In this section, we visualize the failure cases of TTVSR in Fig. 7. The motion trajectories are inaccurate when rotation occurs and useful information cannot be transferred through it, thus limiting the performance of our method. However, due to the high difficulty of modeling rotation, other SOTA methods also fail to obtain better performance. It is notable that TTVSR still achieves greater gains than other methods through its powerful long-range modeling ability. More analyses can be found in the supplementary.

Table 6. Ablation study results of cross-scale feature tokenization (CFT) module on the REDS4 [29] dataset, "S2" and "S3" represent extracting features from two and three scales, respectively. TTVSR can be interpreted as "Base+TG+TA+CFT(S3)".

Method	Token sizes in CFT	PSNR/SSIM
Base+TG+TA	4	31.99/0.9007
Base+TG+TA+CFT(S2)	4, 6	32.08/0.9011
Base+TG+TA+CFT(S3)	4, 6, 8	**32.12/0.9021**
Base+TG+TA+CFT(S3.1)	6, 9, 12	31.95/0.9004
Base+TG+TA+CFT(S3.2)	8, 12, 16	31.91/0.8991

w/o CFT w/ CFT GT

Figure 6. Example of without and with the cross-scale feature tokenization (CFT) on the REDS4 [29] dataset. CFT can transfer the clearer textures from larger scales to restore the detailed textures.

Figure 7. A failure case when rotation occurs.

6. Conclusion

In this paper, we study video super-resolution by leveraging long-range frame dependencies. In particular, we propose a novel trajectory-aware Transformer (TTVSR), which is one of the first works to introduce Transformer architectures in video super-resolution tasks. Specifically, we formulate video frames into pre-aligned trajectories of visual tokens, and calculate attention along trajectories. To implement such formulations, we propose a novel location map to record trajectories, and the location map can online update efficiently by design. TTVSR significantly mitigates computational costs and enables Transformers to model long-range information in videos in an effective way. Experimental results show clear visual margins between the proposed TTVSR and existing SOTA models. In the future, we will focus on 1) evaluating our method in more low-level vision tasks, and 2) extending the trajectory-aware Transformer to high-level vision tasks by more explorations.

Acknowledgement. This work was supported by the NSFC under grants No.61772407. We would like to also thank Tiankai Hang for his help with the paper discussion.

References

[1] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *CVPR*, pages 4778–4787, 2017. 2

[2] Jiezhang Cao, Yawei Li, Kai Zhang, and Luc Van Gool. Video super-resolution transformer. *arXiv preprint arXiv:2106.06847*, 2021. 2, 5, 6

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 1, 3

[4] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. BasicVSR: The search for essential components in video super-resolution and beyond. In *CVPR*, pages 4947–4956, 2021. 1, 2, 5, 6, 7

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 2

[6] Michel Deudon, Alfredo Kalaitzis, Israel Goytom, Md Rifat Arefin, Zhichao Lin, Kris Sankaran, Vincent Michalski, Samira E Kahou, Julien Cornebise, and Yoshua Bengio. HighRes-net: Recursive fusion for multi-frame super-resolution of satellite imagery. *arXiv preprint arXiv:2002.06460*, 2020. 1

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 3

[9] Dario Fuoli, Shuhang Gu, and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation. In *ICCVW*, pages 3476–3485, 2019. 2, 5, 6

[10] Tomio Goto, Takafumi Fukuoka, Fumiya Nagashima, Satoshi Hirano, and Masaru Sakurai. Super-resolution system for 4K-HDTV. In *ICPR*, pages 4453–4458, 2014. 1

[11] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *CVPR*, pages 3897–3906, 2019. 2, 5, 6

[12] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998. 1, 2

[13] Yan Huang, Wei Wang, and Liang Wang. Video super-resolution via bidirectional recurrent convolutional networks. *IEEE TPAMI*, 40(4):1015–1028, 2017. 2, 5

[14] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *ECCV*, pages 645–660, 2020. 2, 5, 6

[15] Takashi Isobe, Songjiang Li, Xu Jia, Shanxin Yuan, Gregory Slabaugh, Chunjing Xu, Ya-Li Li, Shengjin Wang, and Qi Tian. Video super-resolution with temporal group attention. In *CVPR*, pages 8008–8017, 2020. 2, 5, 6

[16] Takashi Isobe, Fang Zhu, Xu Jia, and Shengjin Wang. Revisiting temporal modeling for video super-resolution. *arXiv preprint arXiv:2008.05765*, 2020. 2

[17] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, pages 3224–3232, 2018. 2, 5, 6

[18] Soo Ye Kim, Jeongyeon Lim, Taeyoung Na, and Munchurl Kim. 3DSRNet: Video super-resolution using 3D convolutional neural networks. *arXiv preprint arXiv:1812.09079*, 2018. 1, 2

[19] Soo Ye Kim, Jeongyeon Lim, Taeyoung Na, and Munchurl Kim. Video super-resolution based on 3D-CNNS with consideration of scene change. In *ICIP*, pages 2831–2835, 2019. 2

[20] Tae Hyun Kim, Mehdi SM Sajjadi, Michael Hirsch, and Bernhard Scholkopf. Spatio-temporal transformer network for video restoration. In *ECCV*, pages 106–122, 2018. 2

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[22] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, pages 624–632, 2017. 5

[23] Sheng Li, Fengxiang He, Bo Du, Lefei Zhang, Yonghao Xu, and Dacheng Tao. Fast spatio-temporal residual network for video super-resolution. In *CVPR*, pages 10522–10531, 2019. 1, 2

[24] Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. MuCAN: Multi-correspondence aggregation network for video super-resolution. In *ECCV*, pages 335–351, 2020. 1, 2, 5, 6, 7

[25] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *IEEE TPAMI*, 36(2):346–360, 2013. 5, 6, 7

[26] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[27] Yimin Luo, Liguo Zhou, Shu Wang, and Zhongyuan Wang. Video satellite imagery super resolution via convolutional neural networks. *IEEE TGRS Letters*, 14(12):2398–2402, 2017. 1

[28] Yiqun Mei, Yuchen Fan, Yuqian Zhou, Lichao Huang, Thomas S Huang, and Honghui Shi. Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *CVPR*, pages 5690–5699, 2020. 5, 6

[29] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPRW*, pages 0–0, 2019. 5, 6, 7, 8

[30] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi,

and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *NeurIPS*, 34, 2021. 4

[31] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, pages 4161–4170, 2017. 5

[32] Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *CVPR*, pages 6626–6634, 2018. 1, 2, 5, 6

[33] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *ICCV*, pages 4472–4480, 2017. 2

[34] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. TDAN: Temporally-deformable alignment network for video super-resolution. In *CVPR*, pages 3360–3369, 2020. 1, 2, 5, 6

[35] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, pages 4489–4497, 2015. 2

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 1, 2

[37] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013. 4

[38] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013. 4

[39] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. EDVR: Video restoration with enhanced deformable convolutional networks. In *CVPRW*, 2019. 1, 2, 5, 6

[40] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 5

[41] Gang Xu, Jun Xu, Zhen Li, Liang Wang, Xing Sun, and Ming-Ming Cheng. Temporal modulation network for controllable space-time video super-resolution. In *CVPR*, pages 6388–6397, 2021. 2

[42] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 127(8):1106–1125, 2019. 5, 6, 7

[43] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *CVPR*, pages 5791–5800, 2020. 1, 2

[44] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, Tao Lu, Xin Tian, and Jiayi Ma. Omniscient video super-resolution. *arXiv preprint arXiv:2103.15683*, 2021. 1, 2

[45] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *ICCV*, pages 3106–3115, 2019. 2, 5, 6, 7

[46] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *ECCV*, pages 528–543, 2020. 1, 2

[47] Yanhong Zeng, Huan Yang, Hongyang Chao, Jianbo Wang, and Jianlong Fu. Improving visual quality of image synthesis by a token-based generator with transformers. *NeurIPS*, 34, 2021. 2

[48] Liangpei Zhang, Hongyan Zhang, Huanfeng Shen, and Pingxiang Li. A super-resolution reconstruction algorithm for surveillance images. *Signal Processing*, 90(3):848–859, 2010. 1

[49] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pages 286–301, 2018. 5, 6