Event-driven Video Restoration with Spiking-Convolutional Architecture

Chengzhi Cao, Xueyang Fu*, Yurui Zhu, Zhijing Sun, Zheng-jun Zha

***Abstract*—With high temporal resolution, high dynamic range, and low latency, event cameras have made great progress in numerous low-level vision tasks. To help restore low-quality video sequences, most existing event-based methods usually employ convolutional neural networks (CNNs) to extract sparse event features without considering the spatial sparse distribution or the temporal relation in neighboring events. It brings about insufficient use of spatial and temporal information from events. To address this problem, we propose a new spiking-convolutional network (SC-Net) architecture to facilitate event-driven video restoration. Specifically, to properly extract the rich temporal information contained in the event data, we utilize a spiking neural network (SNN) to suit the sparse characteristics of events and capture temporal correlation in neighboring regions; to make full use of spatial consistency between events and frames, we adopt CNNs to transform sparse events as an extra brightness prior to being aware of detailed textures in video sequences. In this way, both the temporal correlation in neighboring events and the mutual spatial information between the two types of features are fully explored and exploited to accurately restore detailed textures and sharp edges. The effectiveness of the proposed network is validated in three representative video restoration tasks: de-blurring, super-resolution, and de-raining. Extensive experiments on synthetic and real-world benchmarks have illuminated that our method performs better than existing competing methods.**

***Index Terms*—Video restoration, event camera, spiking neural networks, convolutional neural networks.**

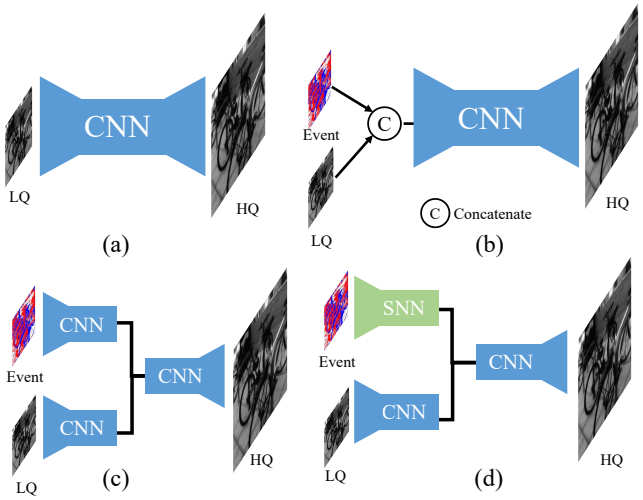

Fig. 1: The categorization of existing video restoration methods. (a) is CNN-based network fed by low-quality (LQ) frames only; (b) directly concatenates the events and frames along channel dimension before feeding them into the CNN-based network; (c) extracts the feature of events and frames by CNN and fuse them; (d) is the proposed hybrid network, including spiking neural network (SNN) and CNN to encode events and frames, respectively.

I. Introduction

VIDEO is a crucial data source for computer vision. Compared with the single image, the video contains abundant spatial and temporal information, so numerous researchers focus on the application of videos in daily life [1]. Videos recorded in real-world scenes, however, always suffer from various degradation factors, such as low-resolution, adverse weather and blurring [2], [3]. These degradations often result in detailed information loss, which seriously influence the visual experience. To address this problem, video restoration aims to obtain a visually pleasing high-quality frame sequence from low-quality measurements. It plays a fundamental part in lots of computer vision tasks, such as tracking and video generation [4]–[6]. In recent years, with the widely use of convolutional neural network, researchers have achieved promising performance in numerous restoration tasks, such as video de-blurring [7]–[9] and video super-resolution [10]–[12]. Although they are valid in certain scenarios, they may fall short when suffering from serious motion blur in high-speed or high-dynamic scenes in complex environments. Due to the considerable loss of visual information, restoring the scene details is hardly possible [13].

Recently, a novel bio-inspired sensor called an event camera has provided a new perspective for video restoration. Unlike the imaging process of traditional cameras, where frames are captured at a fixed rate, the event camera asynchronously records per-pixel brightness changes and generates a stream of events to encode the location, time, and polarity of brightness changes [14]–[16]. With low latency, high dynamic range, and high temporal resolution, the event camera can precisely capture the pixel variants in a very short time, offering a more detailed transition between frames. Due to its numerous advantages, events have been widely used in visual processing and analysis, including video restoration. However, they still have some limitations [17], [18]. First, the current video restoration methods [19], [20] use events as extra information without taking the association between neighboring events into

This work was supported by the National Key R&D Program of China under Grant 2020AAA0105702, the National Natural Science Foundation of China (NSFC) under Grants 61901433, 61906151, U19B2038 and 61620106009, the University Synergy Innovation Program of Anhui Province under Grant GXXT-2019-025, the USTC Research Funds of the Double First-Class Initiative under Grant YD2100002003. (*Corresponding author: Xiangyong Cao*)

The authors are with School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China. (*Corresponding author: Xueyang Fu, xyfu@ustc.edu.cn.)

account, so the high temporal resolution of events cannot be fully utilized. Second, these networks ignore the rich spatial and temporal correlation between events and frames, simply concatenating them or their feature maps (as shown in Fig. 1 (b) and (c)) as the input of CNNs. However, the deployment of events is critical for video restoration since events have recorded scene illumination changes asynchronously, which can present the movement of objects in the field of view. The above issues impede the further development of event-driven video restoration research.

To tackle the limitations mentioned above, we develop a spiking-convolutional network architecture to tackle with event-driven video restoration, where the spatial consistency between events and frames is extensively leveraged to recover spatial textures, and the temporal correlation in event sequences is explored to constantly capture their long-term dependencies. Specifically, to make use of the high temporal correlation in events, the spiking neural temporal memory module is proposed to continually restore temporal information among various sequences of events. This module is built by adopting a spiking neural network (SNN) to suit the distribution of events and compute the long-term dependency of neighboring events to capture temporal correlation. Then, we build a frame-event spatial aggregation module based on CNNs to fuse the two types of features by computing the non-local features between frames and events to capture spatial consistency. As shown in Fig. 1 (d), the hybrid network deploys the SNN to process events considering their sparse nature, while frames are fed into the CNNs to extract spatial features. Compared with (a), (b) and (c), our spiking-convolutional network architecture aims to continuously preserve the spatial and temporal characteristics from event sequences. Furthermore, the proposed network architecture can cope with different low-level vision tasks, including video de-blurring, video super-resolution and video de-raining.

The main contributions of our work are as follows:

- We propose a unified spiking-convolutional network architecture that is able to fully explore and exploit the spatiotemporal correlations between frames and events to reconstruct high-quality frames.
- To utilize high temporal resolution from events effectively, we propose a spiking neural temporal memory module by capturing long-term relations of event sequences in each timestamp. It transforms the brightness changes as an extra prior to aware detailed textures.
- To make use of the spatial consistency, we propose a frame-event spatial aggregation module, which extracts the spatial correlation between frames and events to exploit the complementary information from them.
- Extensive experiments have demonstrated the effectiveness of our approach on a range of restoration tasks, including video de-blurring, super-resolution and de-raining. The numerical results confirm the superiority of our approach.

A preliminary version of this paper appears in [21]. The new contributions are mainly from four aspects: 1) we investigate the performance of spiking neural network (SNN) to deal with events and deploy SNN to cope with the sparse characteristics of events, extracting spatial and temporal information from events properly; 2) we redesign our network and find that the proposed hybrid architecture obtains better performance than our previous method STRA [21], e.g., it obtains 0.46 dB PSNR gains in GoPro dataset, with almost the same parameter numbers; 3) our previous work [21] only handles video de-blurring. In this work, we extend our proposed approach into video super-resolution and de-raining, and compare our method with previous work [21] to illuminate the superior performance of our model; 4) we add considerable analysis to the original version, such as the effect of the SNN, visualization of the SNN, and network parameters.

The structure of this paper is as follows. Section II provides an overview of related research on event-driven video restoration. Section III presents the spiking-convolutional network architecture that we propose. The implementation details and experimental results for three video restoration tasks are discussed in Section IV. Section V presents some limitation of our method, and Section VI offers concluding remarks on this work.

II. Related Work

A. Event-driven Video Restoration

Event cameras record intensity changes of the scene at microsecond level with slight power consumption [22], and have potential applications in video restoration, including deblurring [23] [17], super-resolution [24] [25], reconstruction [26] and so on. Event-based video restoration methods can be categorized into two groups, including model-driven and data-driven algorithms. Model-based algorithms rely on the physical event generation principle to reconstruct high-quality frames. However, due to the imperfections of physical implementation including intrinsic noise and limited bandwidth, real events in temporal and spatial domains inevitably degrades performance. Shang $et\ al.$ [27] formulates event-based motion blurring with a double integral model. Yet, the noisy hard sampling mechanism of event cameras often introduces strong accmulated noise and loss of scene details/contrast. Jiang $et\ al.$ [17] proposed a sequential formulation of event-based motion deblurring, then unfolded its optimization steps as an end-to-end deep architecture. However, events modules in these methods are too trivial to incorporate into other existing networks, making it infeasible to benefit from state-of-the-art video deblurring methods. Data driven algorithms [28] [24] address the above limitations by utilizing neural networks and directly learn the relation of a low-quality image and a sequence of sharp clear images with the aid of events. For training purpose, a synthesized dataset composed of labeled events and blurry images is commonly simulated from sharp clear video sequences. Wang $et\ al.$ [19] built a synthetic dataset based on GoPro [29] to connect events, LR blurry images and the HR sharp clear images, and proposed an sparse learning network to recover the high-quality images from event cameras.Comparing to the above-mentioned methods, our proposed approach differs in the following aspects: (1) we consider the long-range dependency of different events to

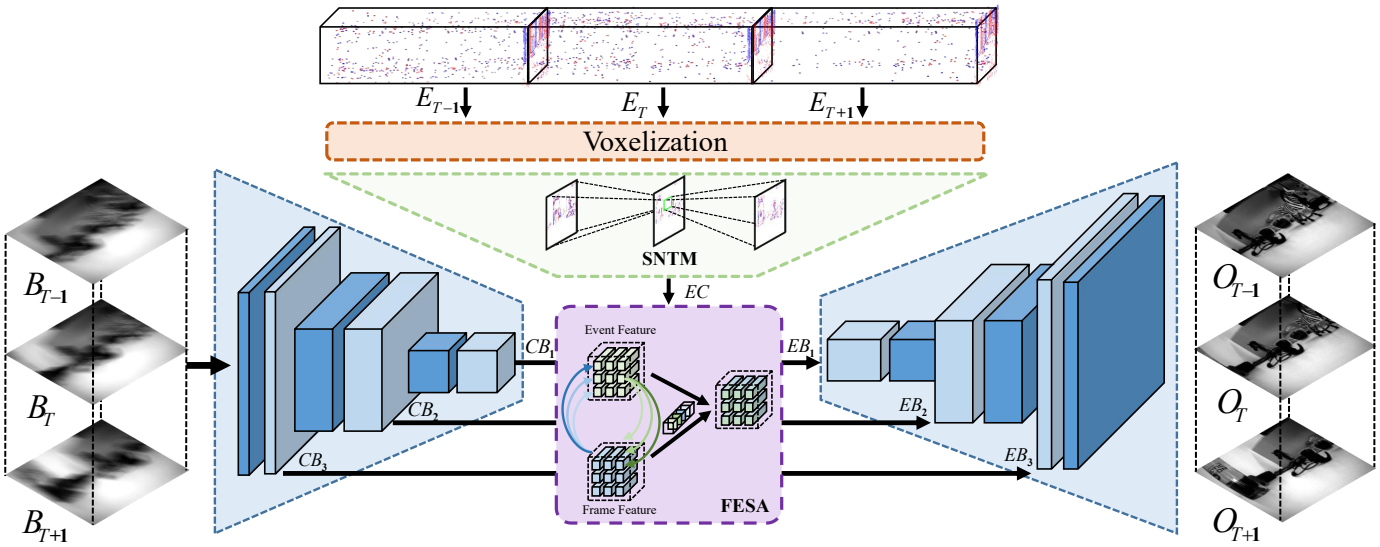

Fig. 2: The overall architecture of our proposed method. It contains two specifically designed modules: the spiking neural temporal memory (SNTM) module and the frame-event spatial aggregation module (FESA). Given the triplet blurring consecutive frames $\{B_{T-1}, B_T, B_{T+1}\}$ and their corresponding event sequences $\{E_{T-1}, E_T, E_{T+1}\}$, SNTM can fuse them by computing the spatial consistency between them; FESA can capture long-term correlations of neighboring event sequences continuously in temporal order.

restore temporal event correlation, and (2) calculate spatial consistency between events and frames with improved non-local operation to capture degraded contexts. (3) Our unified framework can resolve several restorations, such as deblurring, super-resolution and deraining.

B. Spiking Neural Networks

Spiking neural network (SNN) is a bio-inspired network that can naturally suit the asynchronous and sparse characteristics of event streams by using biomimetic spike neurons as computing units [30] [31]. For the reproduction of this asynchronous and spike-based mechanism, multiple models of spiking neurons have been presented at various levels of abstraction. Biophysical formulations lead to accurate representations of neural dynamics, however, their complexity limits their use in large-scale networks [32] [33]. Because it has a specific event-triggered computation characteristic that can respond to the events in a nearly latency-free way, it is naturally fit for processing events and can preserve the spatial and temporal information of events by utilizing a discretized input representation. Researchers in [34] proposed a temporal-wise attention SNN by implementing an attention mechanism in temporal-wise input to filter out useless frames for event stream classification. Authors of [35] presented a methodology for optical flow estimation using convolutional SNNs based on Spike-Time-Dependent-Plasticity learning. The main limitation of these works is that they employ shallow SNN architectures, because deep SNNs suffer in terms of performance. In view of these, a hybrid approach is expected to obtain the advantages of SNN and CNN.

III. Methods

A. Event Representation

Event-driven video restoration intends to recover high-quality frames $\{I_T\}$ from low-quality frames $\{B_T\}$ in one time step T and its corresponding event sequence E_T:

$$E_T \triangleq \{(x_n, y_n, p_n, t_n)\}_{n \in T}, \tag{1}$$

where (x_n, y_n) and t_n denote the spatial and temporal location of the n-th event, respectively. $p_n \in \{+1, -1\}$ is the polarity (increase or decrease). The polarity is computed by:

$$p_n = \Phi(log(\frac{\mathrm{L}_{xy}(n)}{\mathrm{L}_{xy}(n-1)}), c), \tag{2}$$

where c is the intensity threshold deciding whether an event can be recorded or not, $\mathrm{L}_{xy}(n)$ and $\mathrm{L}_{xy}(n-1)$ represent the instantaneous brightness intensity at location (x, y) at time n and $n-1$, respectively. $\Phi(\cdot, \cdot)$ is a piecewise function:

$$\Phi(x, c) = \begin{cases} -1, x \leqslant -c, \\ +1, x \geqslant +c, \end{cases} \tag{3}$$

many deep learning-based methods are proposed to solve this problem by learning an mapping function, $i.e.$,

$$I = f\left(B, E\right), \tag{4}$$

where $f(\cdot, \cdot)$ is the representation from low-quality frame B and events E to the corresponding high-quality frame I.

B. Network Structure

The overview of our proposed spiking-convolutional network architecture is shown in Fig. 2. Firstly, our network feeds three low-quality frames into an encoder to extract frame features. Then, to utilize the temporal information from events,

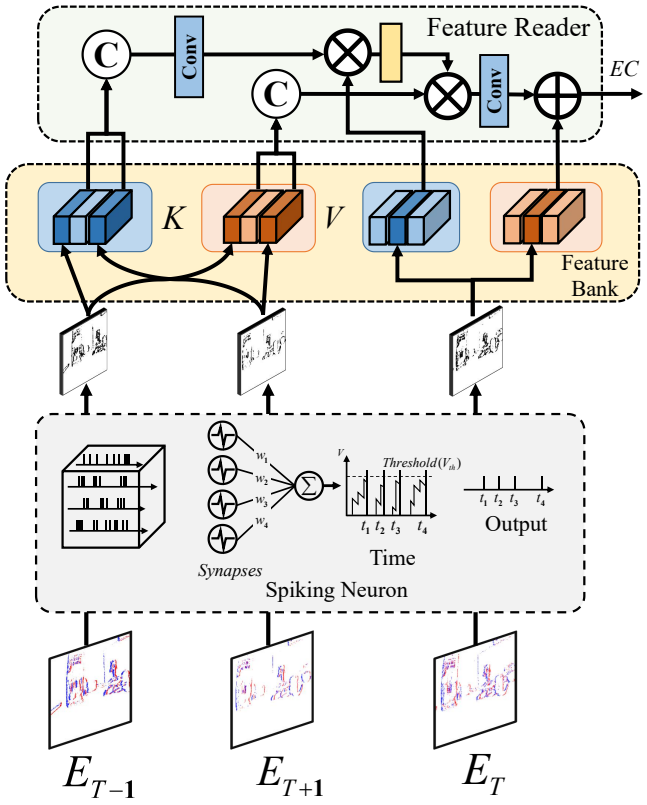

Fig. 3: The structure of spiking neural temporal memory module (SNTM). The input event sequences are fed into spiking neural network to extract features, and then the "Feature Reader" calculate the correlation of neighboring events.

the spiking neural temporal memory module (SNTM) will continuously capture temporal information between different sequences of events. Moreover, the frame-event spatial aggregation module (FESA) is proposed to fuse two types of features by computing non-local correlations.

C. Spiking Neural Temporal Memory Module

The structure of spiking neural temporal memory module is shown in Fig. 3. For the event camera, the polarity of event streams denotes whether the brightness of a given pixel is increasing (ON) or decreasing (OFF). In biological brain systems, a neuron that accepts stimuli and fires a spike can be modeled using the leaky integrate-and-fire (LIF) model. It can be calculated as follows:

$$\tau_m \frac{dV^l}{dt} = -V^l + C(t), \tag{5}$$

where V^l denotes the post-neuronal membrane potential and τ_m presents the membrane potential decay time constant. The input current $C(t)$ can be calculated as the weight sum of pre-spikings at each timestamp:

$$C(t) = \sum_{i=1}^{n^l} (w_i \sum_k \theta_i(t - t_k)), \tag{6}$$

where n_l denotes the number of pre-synaptic weights, w_i is the synaptic weight connecting i-th pre-neuron to post-neuron.

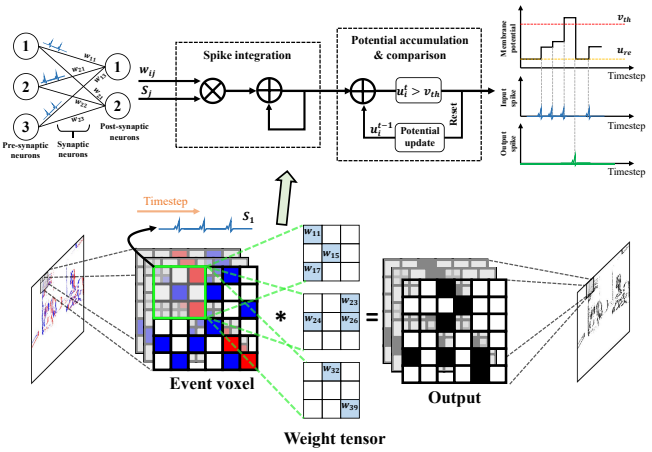

Fig. 4: Detailed spiking neural network operations.

$\theta_i(t - t_k)$ is a spiking event from i-th pre-neuron at time t_k, which can be formulated as follows:

$$\theta_i(t - t_k) = \begin{cases} 1, & t = t_k, \\ 0, & otherwise, \end{cases} \tag{7}$$

where t_k is the time instant that k-th spiking occurred. For training and inference, the iterative representation of LIF model can be described as:

$$D[t] = f(V[t-1], I[t]), \tag{8}$$
$$S[t] = \Theta(D[t] - V_{th}), \tag{9}$$
$$V[t] = D[t](1 - S[t]) + V_{reset} S[t], \tag{10}$$

where $I[t]$ is the input current at time-step t, $D[t]$ and $V[t]$ denote the membrane potential after neuronal dynamics and after the trigger of a spike, respectively. $\Theta(x)$ is defined by $\Theta(x) = 1$ for $x \geq 0$ and $\Theta(x) = 0$ for $x < 0$. V_{th} is the firing threshold, and $S[t]$ is the output spike, which equals 1 if there is a spike and 0 otherwise. V_{reset} is the reset potential. The function $f(\cdot)$ describes the neuronal dynamics and takes different forms for different spiking neuron models:

$$H[t] = V[t-1] + \frac{1}{\tau}(I[t] - (V[t-1] - V_{reset})). \tag{11}$$

where τ represents the membrane time constant. Detailed spiking neural network operations are shown in Fig. 4.

For training method in SNN, the total loss is evaluated after the forward propagation of all consecutive events and frames through the network, including the spiking neural network. Due to this discontinuous and non-differentiable neuron model, standard backpropagation algorithms cannot be applied to SNNs in their native form. We adopt the approximate gradient method [36] for back-propagating errors through SNN layers. The approximate IF gradient is computed as $\frac{1}{V_{th}}$, where the threshold value accounts for the change of the spiking output with respect to the input. In the forward phase, neurons in the SNN layers accumulate the weighted sum of the spike inputs in membrane potential. If the membrane potential exceeds a threshold, a neuron emits a spike as its output and resets. The final SNN layer neurons just integrate the weighted sum of spike inputs in the output accumulator, while not

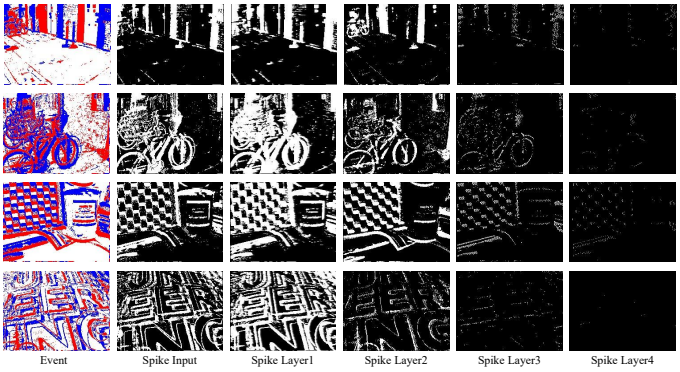

Fig. 5: The visualization of degradations in SNN at deeper layers. From left to right, the layer of SNN get deeper, but the number of spikes vanishes drastically.

producing any spikes as the output. At the last time-step, the integrated outputs of SNN layers propagate to the CNN layers. After the forward pass, the final loss is evaluated, followed by backpropagation of gradients through the ANN layers using standard back-propagation.

Compared with traditional convolutional layers, spike neuron can certainly maintain the spatial and temporal dynamics. However, most of recent works have proved that the number of spikes drastically vanishes at deeper layers, leading to serious degradation in performance [37], as shown in Fig. 5. So in our spiking neural extraction, two LIF neurons are employed to extract event features.

The proposed Spiking Neural Temporal Memory module has a feature bank and a feature reader. For neighboring event sequence in $T-1$ and $T+1$, one common-used and two special-used spiking neural layers (SNL) are deployed to obtain the key and the value. Their own spiking neural layers are deployed on the common feature map F_m so that all keys and values of each frame will be stored by temporal order in the feature bank. The equation can be shown by:

$$K_{T-1}, V_{T-1} = SNL_{T-1}(F_m), \tag{12}$$

$$K_{T+1}, V_{T+1} = SNL_{T+1}(F_m). \tag{13}$$

In the feature reader, the keys and values of neighboring events are concatenated respectively, and the similarities between query and key to present temporal correlation with current events are calculated. In this way, the temporal correlation between current sequence and its adjacent sequences will be completely utilized to capture long-term dependencies.

D. Frame-Event Spatial Aggregation Module

Motivated by the asymmetric integration of intra-scale characteristics from different scales [38], three independent feature maps are resized to the same scale and concatenate them for fusion with event features. It can be represented as:

$$CB = Conv(Concat(CB_1^{\downarrow}, CB_2^{\downarrow}, CB_3)), \tag{14}$$

in which CB_1, CB_2 and CB_3 denote three independent feature maps, and CB is the output of feature fusion. To maintain the same scale as the frame feature, we also utilize a

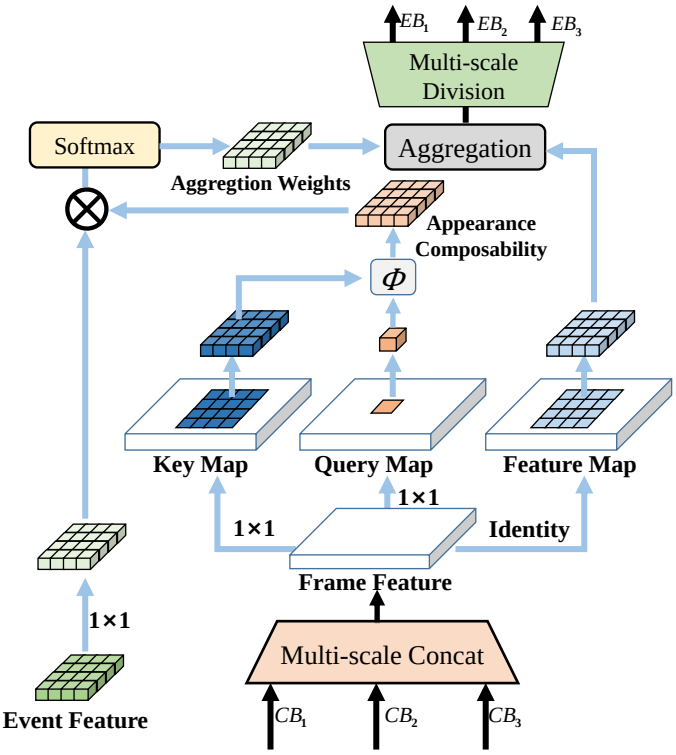

Fig. 6: The structure of frame-event spatial aggregation module (FESA). The three maps are transformed in the same scale, and then the spatial correlation between events is calculated.

downsampling layer for events. Then, the non-local attention operation is deployed to fuse spatial features from frames and events [39], [40] by taking into account the spatial correlation between all regions in each frame and their corresponding events.

$$Feat_i = \frac{1}{C(E)} \sum_{\forall j} f(E_i, F_j) g(F_j), \tag{15}$$

where E and F denote the event feature maps and low-quality frame feature maps, respectively. i and j represent the index of positions. The function $g(\cdot)$ computes the representation of feature maps at the position j. We set $C(x) = N$ to normalize the final result, in which N is the number of pixels in the feature map. The function $f(\cdot, \cdot)$ can be calculated as:

$$f(E_i, F_j) = \theta(E_i)^T \varphi(F_j). \tag{16}$$

Finally, we deploy multi-scale division by upsampling the output to three maps (EB_1,EB_2, EB_3) to connect with U-net decoder feature maps. Fig. 6 shows the structure of FESA. Unlike traditional non-local network, which only computes the weighted sum of all features in a single feature map, our FESA combines three frame feature maps in different scales and records correlation with corresponding event sequences.

E. Loss Function

In our work, the mean squared error (MSE) is applied to train our network in an end-to-end fashion:

$$\mathcal{L}_{MSE} = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}(\hat{I}_{x,y} - I_{x,y})^2, \tag{17}$$

where $I_{x,y}$ and $\hat{I}_{x,y}$ correspond to the values of ground-truth and the output of our network at location (x, y), respectively.

IV. Experiment

A. Dataset

a) GoPro [29]: It contains 2,103 pairs of blurry and sharp frames without corresponding events in training process, and 1,111 pairs in testing process. Following [19], we increase the frame rate by interpolating 7 images between consecutive frames and then generate both events and blurry images based on the interpolated high frame-rate sequences. V2E [41] simulator is utilized to generate the corresponding event sequences. To increase the noise diversity, we set different contract thresholds for pixel-level from Gaussian distribution $N(0.18, 0.03)$ [23]. According to [28], the images are downsampled to a small size to match the real event cameras. Some details are provided in the Table I, where the number of diverse scene sequences $\#seq$, image size, and average number of events within the exposure time of the corresponding blurry image $\#AVG$ are presented.

b) HQF [18]: It consists of sharp ground-truth frames and corresponding events that were simultaneously captured from DAVIS240C event camera [14] that are well-exposed with little motion blur. Following [28], we synthesize motion blurs in the same way as the GoPro dataset [29]. Moreover, because HQF is captured in the real world, we also regard it as the real-world testing dataset in video SR task to demonstrate the robustness of our method in the real-world scene.

c) NFS-SR: Following [25], we rely on synthetic datasets to train our network because there is no public dataset that contains low-resolution (LR) frames, ground-truth high-resolution (HR) frames and corresponding event streams. Specially, we employ Need for Speed (NFS) [42] as sources, and all frames are downsampled to 128×128 as LR frames. The corresponding event streams are generated from V2E [41], and the HR frames are produced in different scale according to upscale factors (×2 and ×4). Finally, there are 3828 tuples produced from 132 video sequences. To evaluate our method, the testing dataset, which is the same as [25], contains 841 intensity images and the related simulated event streams between two successive frames from 19 videos.

d) NTURain: It is proposed by Chen *et al*. [43]. The videos are captured by the camera with diverse motions for each clip, e.g., panning slowly with unstable movements, or mounting on a fast moving vehicle. The rain streaks are synthesized with adjustable parameters such as raindrop size, opacity, scene depth, wind direction, and camera shutter speed. It has 24 paired rainy sequences with the clear versions for training and 8 pairs for testing. The corresponding events are also generated by V2E [41].

TABLE I: The details of GoPro and HQF datasets. $\#seq$, $size$ and $\#AVG$ represent the number of diverse scene sequences, image size and average number of events within the exposure time of the corresponding image, respectively.

Dataset	$\#seq.$	$size$	$\#AVG$
GoPro (train)	240	180×320	164827
GoPro (test)	30	180×320	161834
HQF (train)	9	180×240	65227
HQF (test)	5	180×240	71895

B. Implementation Details

We randomly cropped frames in 256×256 size and horizontally flipped them with a probability of 0.5 to augment the training set. ADAM optimizer [44] with parameter $\beta_1 = 0.9, \beta_2 = 0.999$, and train our network in a batch size of 8. The initial learning rate is set as 10^{-4}, and it decreases by 25% every 50 epochs, and the maximum training epoch is 200 . The framework is implemented in Pytorch and our experiments are deployed on a single NVIDIA RTX 2080Ti GPU.

C. Evaluated Metrics

Image quality refers to visual attributes of images and focuses on the perceptual assessments of viewers. We choose Peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) and perceptual image patch similarity (LPIPS) as our metrics.

a) PSNR: It is defined via the maximum pixel value and the mean squared error between images. Given the ground truth image I with N pixels and the reconstruction $\hat{I}$, the PSNR value can be calculated as:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{L^2}{\frac{1}{N}\sum_{i=1}^{N}(I(i) - \hat{I}(i))^2}\right), \tag{18}$$

where L equals to 255 in general cases using 8-bit representations. Since the PSNR is only related to the pixel-level MSE, only caring about the differences between corresponding pixels instead of visual perception, it often leads to poor performance in representing the reconstruction quality in real scenes, where we're usually more concerned with human perceptions.

b) SSIM: It is proposed for measuring the structural similarity between images, based on independent comparisons in terms of luminance, contrast, and structures. For an image I with N pixels, the luminance μ_I and contrast σ_I are estimated as the mean and standard deviation of the image intensity, respectively. And the comparison functions on luminance, contrast and structure, denoted as $\mathcal{C}_l(I, \hat{I})$, $\mathcal{C}_c(I, \hat{I})$ and $\mathcal{C}_s(I, \hat{I})$, can be calculated as:

$$\mathcal{C}_l(I, \hat{I}) = \frac{2\mu_I\mu_{\hat{I}} + C_1}{\mu_I^2 + \mu_{\hat{I}}^2 + C_1}, \tag{19}$$

$$\mathcal{C}_c(I, \hat{I}) = \frac{2\sigma_I\sigma_{\hat{I}} + C_2}{\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2}, \tag{20}$$

$$\mathcal{C}_s(I, \hat{I}) = \frac{\sigma_{I\hat{I}} + C_3}{\sigma_I\sigma_{\hat{I}} + C_3}, \tag{21}$$

$$\text{SSIM}(I, \hat{I}) = [\mathcal{C}_l(I, \hat{I})^\alpha][\mathcal{C}_c(I, \hat{I})]^\beta[\mathcal{C}_s(I, \hat{I})]^\gamma, \tag{22}$$

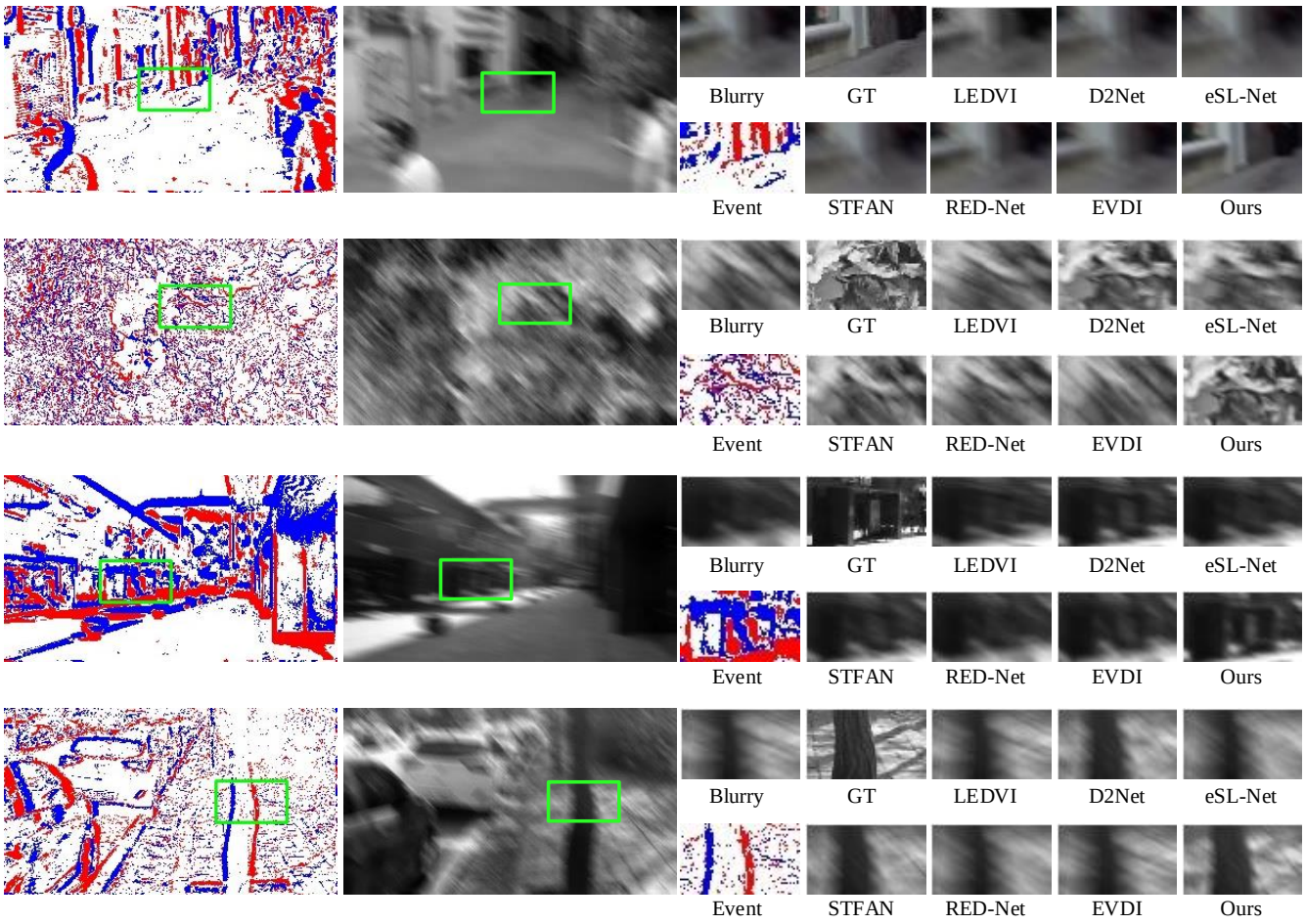

Fig. 7: Qualitative comparison of event-driven video de-blurring in GoPro dataset. ”GT” is the abbreviation of ”ground-truth”, which means the high-quality images without any blur. Our network generates much sharper frame with clearer edges than other methods. eSL-Net [19] and EVDI [45] do not perform well without consideration of temporal correlation in events. Others contain some blurry residual.

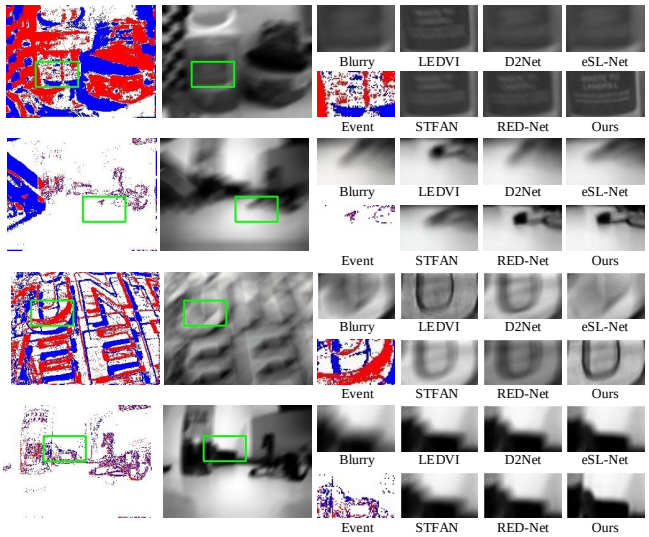

Fig. 8: Qualitative comparison of event-driven video de-blurring in HQF dataset, where the third row is the outdoor image, and others are indoor images. The first column is the corresponding events to the blurry image in the second column.

where α, β, γ are control parameters for adjusting the relative importance.

c) LPIPS: Zhang *et al.* [46] collect a large-scale perceptual similarity dataset, evaluate the perceptual image patch similarity (LPIPS) according to the difference in deep features by trained deep networks, and show that the deep features learned by CNNs model perceptual similarity much better than measures without CNNs.

D. Video De-blurring

In this section, we compare the performance of our proposed framework with several state-of-the-art video de-blurring methods, including eSL-Net [19], STRA [21], STFAN [47], RED-Net [28], MemDeblur [48], LEDVI [23], GShift-Net [49], DLEFNet [50], and EVDI [45].

Table II has shown the performance in GoPro and HQF dataset. Note that when the input are videos and events (“V+E”), we do not change the original structure of video-based network, and use a CNN backbone to extract features of events, then employ an attention module to feed them into the main stream. It is obvious that when we feed both event streams and frames into these networks, most of them achieve

Fig. 9: Qualitative comparison of event-driven video SR×4 in HQF dataset. The first column shows the input Low-resolution frames, and other columns show the corresponding events and SR results by different methods. Zoom in for a better view.

TABLE II: Average PSNR, SSIM and LPIPS results on the GoPro and HQF dataset. The best performance is highlighted in **bold** and the second-best performance is underlined, respectively. "V" and "E" represent the image sequences and event sequences, respectively.

Dataset	Input	Methods	LEDVI	GShift-Net	eSL-Net	STFAN	RED-Net	MemDeblur	EVDI	DLEFNet	STRA	**Ours**
GoPro	V	PSNR	20.46	27.48	21.30	26.62	27.16	27.01	24.58	27.30	26.82	26.99
	V	SSIM	0.825	0.860	0.733	0.810	0.814	0.852	0.820	0.872	0.792	0.856
	V	LPIPS	0.206	0.081	0.187	0.101	0.089	0.097	0.128	0.079	0.099	0.087
GoPro	V+E	PSNR	22.86	28.57	22.59	28.07	28.98	27.31	26.40	28.95	29.73	**30.19**
	V+E	SSIM	0.733	0.889	0.750	0.836	0.849	0.860	0.829	0.893	0.927	**0.939**
	V+E	LPIPS	0.156	0.075	0.161	0.086	0.077	0.093	0.104	0.074	0.071	**0.063**
HQF	V	PSNR	21.37	26.57	24.99	23.19	24.63	22.69	24.41	26.98	26.33	26.69
	V	SSIM	0.789	0.873	0.841	0.812	0.831	0.807	0.818	0.879	0.866	0.865
	V	LPIPS	0.185	0.101	0.132	0.150	0.127	0.159	0.131	0.103	0.105	0.109
HQF	V+E	PSNR	22.22	26.86	25.42	24.17	25.72	23.10	25.27	26.91	27.54	**27.85**
	V+E	SSIM	0.687	0.901	0.754	0.711	0.763	0.813	0.841	0.907	0.834	**0.937**
	V+E	LPIPS	0.168	0.098	0.116	0.126	0.112	0.151	0.118	0.093	0.091	**0.087**

better PSNR, SSIM and LPIPS values than feeding videos only. For example, with the guidance of events, RED-Net [28] and STFAN [47] achieves 1.82 dB and 1.45 dB gains over video conditions without events in GoPro dataset, respectively. The gains on HQF dataset become even larger. Moreover, the experiments have shown the superior performance of proposed framework in all metrics than other competing methods, with 2.88 dB improvement than MemDeblur [48] in terms of PSNR. It is because that our method can effectively make full use of the high-temporal information from events to capture the temporal relation between adjacent events as an extra prior to reconstruct sharp edges.

The performance gains over other state-of-the-art methods are consistent with the visual results. Fig. 7 and Fig. 8 present de-blurring results of our method and other comparing methods in HQF and GoPro dataset, respectively. Note that "GT" is the abbreviation of "ground-truth", which means the high-quality images without any blur. Visual quality comparisons show that high-quality video de-blurring with more structural details can be achieved with the right combination of frames and events. For example, in the third row of Fig. 8, the the blurry edges of letter "U" is removed better by our proposed method. Other networks such as LEDVI [23] and eSL-Net [19], however, fail to recover clear edges. Moreover, in the fourth row of Fig. 7, it is observed that in the course of blurring process, various undesired artifacts are produced in the tree's edges. All the compared methods can not tackle this case. But the proposed framework can produce better visual results, more similar to the ground truth. Although the low-quality frames suffer from significant blur, events can also record the clear moving pattern of the scene, which can guide motion de-blurring.

E. Video Super-Resolution

Our method is compared with several state-of-the-art methods, including BSRDM [51], STRA [21], eSL-Net [19], EvInt [25], EVSR [24], LBNet [52], MRVSR [53], SLS [54] and TTVSR [55].

1) Synthetic Video SR: Tthe PSNR, SSIM and LPIPS results of SR×2 and SR×4 are presented in Table III. From

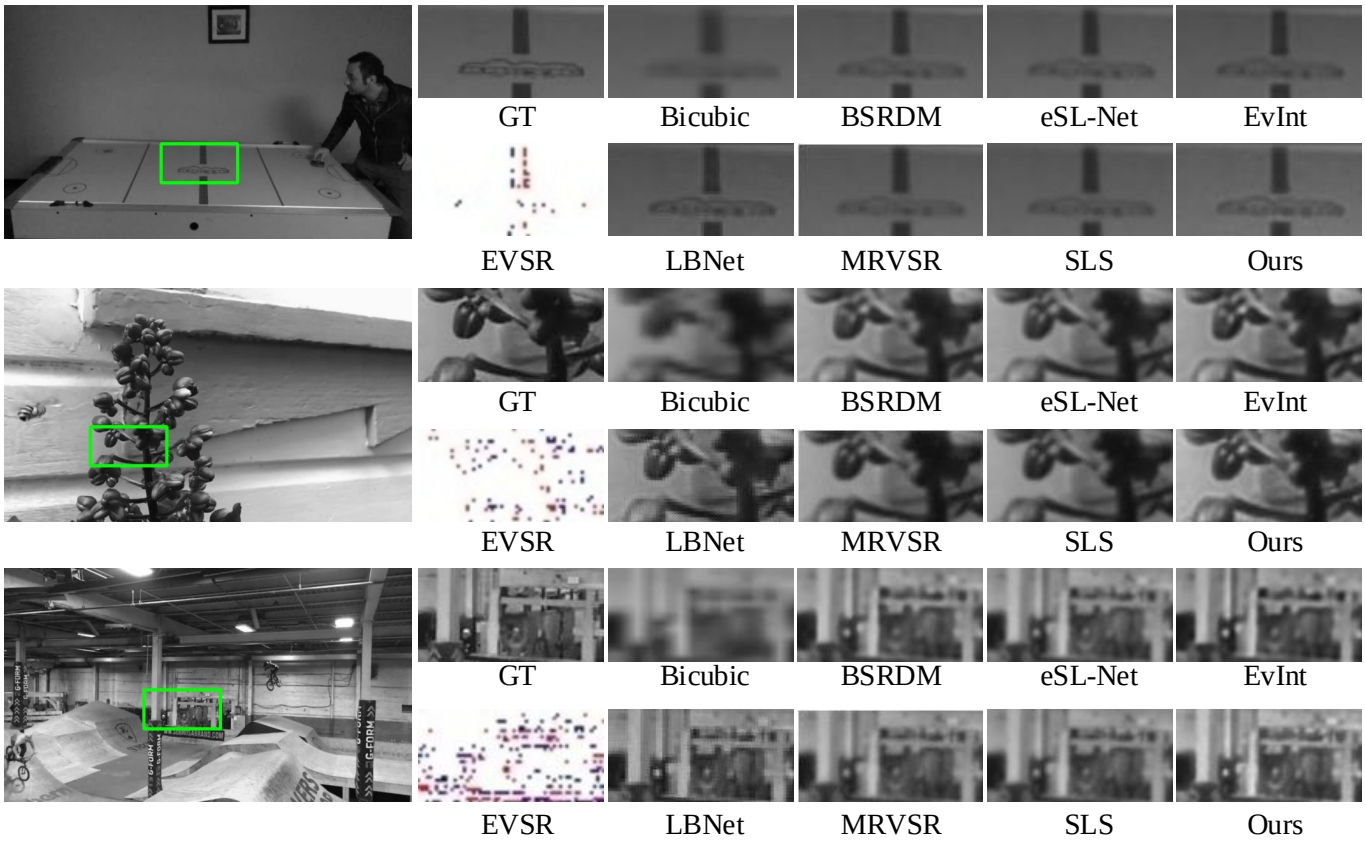

Fig. 10: Qualitative comparison of event-driven video SR×4 in NFS dataset. Zoom in for a better view. "GT" is the abbreviation of "ground-truth". The sharpness of the edges on the objects and textures restored by our method is the best.

TABLE III: Quantitative results (PSNR, SSIM and LPIPS) of SR ×2 and SR ×4 results on the synthetic test dataset. The best performance is highlighted in **bold** and the second-best performance is underlined, respectively. For the input, "V" and "E" represent that the input are the image sequences and event sequences, respectively.

Scale	Input	Methods	BSRDM	eSL-Net	EvInt	EVSR	LBNet	MRVSR	SLS	TTVSR	STRA	**Ours**	Ideal value
SRx2	V	PSNR	34.33	32.84	28.09	28.84	29.43	34.67	33.44	32.65	34.47	34.57	$+\infty$
	V	SSIM	0.926	0.941	0.859	0.860	0.883	0.925	0.919	0.908	0.948	0.949	1.00
	V	LPIPS	0.047	0.043	0.086	0.080	0.075	0.046	0.051	0.055	0.038	0.042	0
SRx2	V+E	PSNR	35.03	34.63	30.07	30.10	32.71	35.92	35.29	34.86	35.55	**36.01**	$+\infty$
	V+E	SSIM	0.954	0.926	0.912	0.914	0.943	0.955	0.952	0.950	**0.960**	0.957	1.00
	V+E	LPIPS	0.038	0.046	0.073	0.051	0.049	0.035	0.037	0.036	0.033	**0.030**	0
SRx4	V	PSNR	30.42	29.71	21.84	26.07	29.64	29.99	30.59	23.45	30.85	30.67	$+\infty$
	V	SSIM	0.919	0.912	0.851	0.816	0.915	0.914	0.913	0.887	0.920	0.921	1.00
	V	LPIPS	0.061	0.060	0.143	0.108	0.071	0.055	0.064	0.142	0.057	0.053	0
SRx4	V+E	PSNR	31.01	30.57	25.43	29.46	31.16	30.85	31.20	26.76	31.12	**31.56**	$+\infty$
	V+E	SSIM	0.925	0.920	0.847	0.898	0.919	0.921	0.922	0.879	0.924	**0.926**	1.00
	V+E	LPIPS	0.061	0.051	0.116	0.060	0.062	0.058	0.058	0.091	0.054	**0.044**	0

it, we have some observations. First, most of video-based SR networks can obtain obvious improvements with the help of events. Take EVSR [24] as an example, it achieves 1.26 dB and 0.86 dB improvements in terms of PSNR in SR×2 and SR×4, respectively. Other competing methods also obtain similar benefits from events. This is because events can provide extra prior about high frequency details in scenes, which contribute to the final SR results. Secondly, as expected, our method get promising results in both SR×2 and SR×4 tasks in "V+E" condition. Specifically, it achieves 36.01 dB in PSNR values in SR×2, getting the highest performance among all methods. The average gains of ours over STRA are 0.46 dB in PSNR and 0.03 in LPIPS. When compared with the other methods, our method is far ahead.

Our method not only outperforms all the above comparative methods in quantitative evaluations, but also produces visually pleasing results among them. In Fig. 10, we show some visual comparisons on scales ×4 in NFS dataset. We can see that most of state-of-the-art methods lost many texture details in the LR frame, even though MRVSR can recover partial regions. In the third row of Fig. 10, it is obvious that event has captured the clear edges of objects in the scene, which is beneficial to recovering detailed textures from LR. But most compared networks can not fully utilize events to recover the

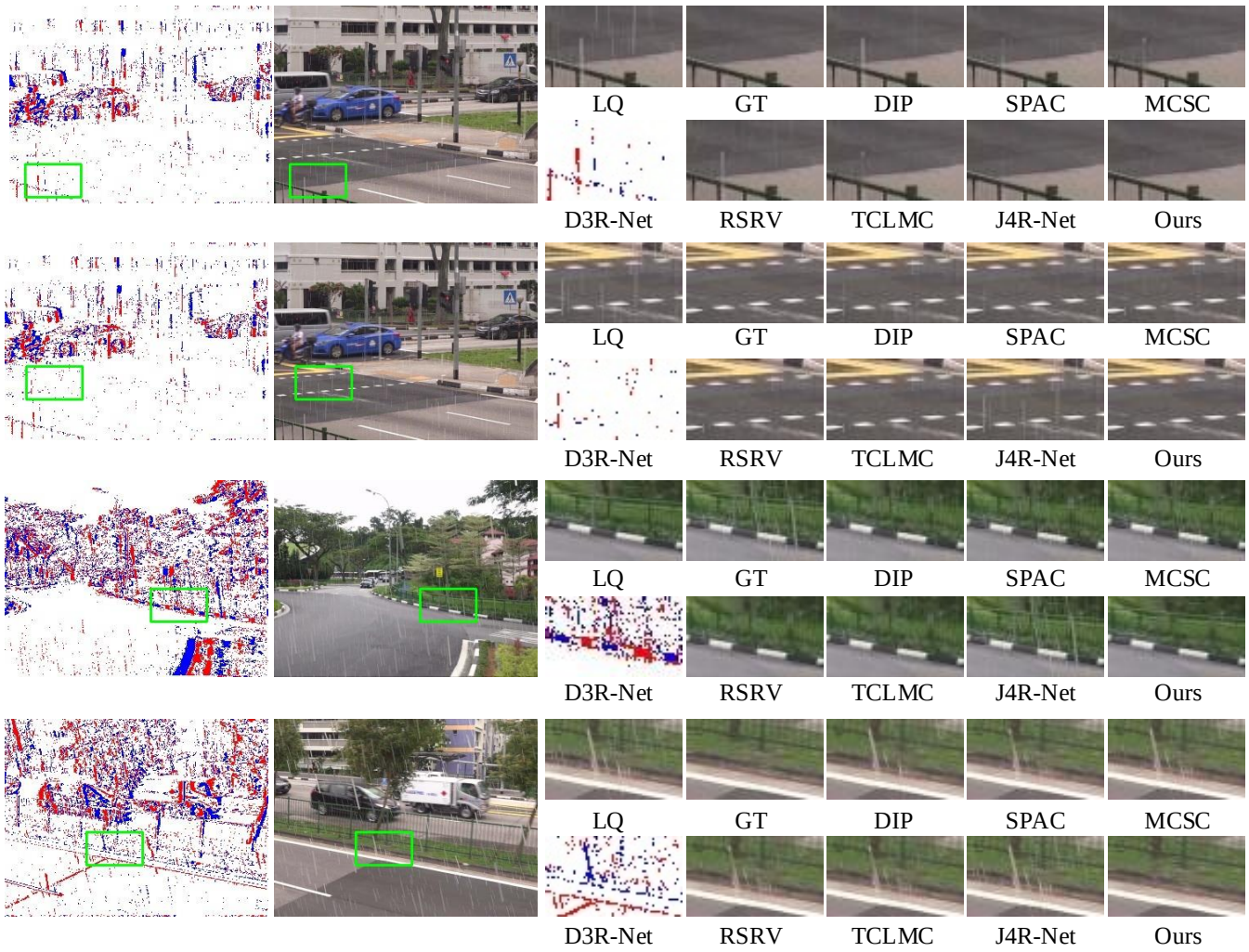

Fig. 11: Visual results of event-driven video de-raining in NTURain dataset. The first column shows the events, while the second column shows the input rainy frames. The remaining rain streaks and artifacts are denoted with green box. Zoom in for a better view.

lost information. On the contrary, with the proper usage of spatial and temporal correlation between events and frames, our method can obtain promising visual results with clearer structures.

2) Real-World Video SR: To further illuminate the performance of our proposed method, we super-resolve the real-world frames with corresponding events in HQF dataset. The results are shown in Fig. 9. For the letters "Theo" the first row, both BSRDM and eSL-Net contain some unwanted and blurred artifacts, but our method generates sharper and clearer textures. Moreover, in the last row in Fig. 9, all of compared methods can not reconstruct the face of the "person" in the poster, suffering from serious texture degradation. But our network can still produce better qualitative results. These comparisons have validated the superior performance of spatio-temporal relation-aware mechanism, allowing our method to maintain robustness in different unknown degradation conditions.

F. Video De-raining

We compare our method with DIP [56], STRA [21], SPAC [57], MCSC [58], D3R-Net [59], RSRV [60], TCLMC [61], J4R-Net [62], and VRGNet [63].

For video de-raining task, Table IV has shown that our proposed method significantly outperforms the state-of-the-art methods by consistently achieving better performance in all metrics. Compared with the second best algorithm STRA, our method obtains a performance gain of 0.28 dB in PSNR and 0.01 in SSIM. In addition, our method gets considerable gains over other methods, $i.e.$, 1.80 dB over MCSC [58], 2.43 dB over D3R-Net [59] and 2.74 dB over J4R-Net [62]. Clearly, when adding events into video-based networks, all of them obtain varying degrees of gains, which illuminates the benefit of guidance from events for video de-raining.

Moreover, some visual comparisons on NTURain dataset are shown in Fig. 11. We have several observations: **(1)** the event camera can not only record the movement of objects in the field of view, but also detect the moving streaks in the rainy day. Take the last row as an example, the events have

TABLE IV: Quantitative results (PSNR, SSIM and LPIPS) of results on the NTURain dataset. The best performance is highlighted in **bold** and the second-best performance is underlined, respectively. For the input, "V" and "E" represent that the input are the image sequences and event sequences, respectively.

Input	Methods	DIP	SPAC	MCSC	D3R-Net	RSRV	TCLMC	J4R-Net	VRGNet	STRA	**Ours**	Ideal value
V	PSNR	32.28	28.31	33.30	30.61	28.45	30.49	32.20	33.47	32.30	33.82	$+\infty$
V	SSIM	0.944	0.908	0.956	0.898	0.894	0.913	0.949	0.958	0.946	0.961	1.00
V	LPIPS	0.038	0.066	0.037	0.063	0.082	0.058	0.053	0.046	0.047	0.031	0
V+E	PSNR	34.92	30.34	33.49	32.86	30.40	31.45	32.55	33.83	35.01	**35.29**	$+\infty$
V+E	SSIM	0.967	0.913	0.959	0.950	0.919	0.935	0.949	0.958	0.967	**0.968**	1.00
V+E	LPIPS	0.016	0.066	0.014	0.018	0.041	0.058	0.021	0.022	0.015	**0.013**	0

TABLE V: Quantitative analysis of SNTM and FESA on GoPro and HQF dataset.

SNN	SNTM	FESA	GoPro	HQF
0	✗	✗	27.81	25.03
0	✓	✗	29.04	25.94
0	✗	✓	28.50	25.38
0	✓	✓	29.73	27.54
1	✓	✓	29.81	27.76
2	✓	✓	**30.19**	**27.85**
3	✓	✓	28.31	26.57
4	✓	✓	28.02	26.23
5	✓	✓	27.89	26.04

shown the sharp edges of street as well as the tiny rain streaks in a small region. It means that the high-temporal resolution of events is beneficial to aware the rain streaks; **(2)** although the rain streaks can be detected and removed by TCLMC and DIP, they can not recover the missing details properly; **(3)** there are more or less noticeable streaks generated by SPAC and MCSC, while our network can recover global texture information. In short, the rain-free frames recovered by ours achieves better performance qualitatively and quantitatively.

G. Ablation Study

a) Benefit of Events: The effect of events for video-based restoration has been shown in Table II, Table III and Table IV. We test all compared methods in both with-event and without-event conditions. As we can see, the event-driven methods ("V+E") have better performance than video-based methods, which means that events can actually bring noticeable gains in different restoration tasks. Moreover, the proposed framework can also achieve superior performance than all other networks with the guidance of events.

b) FESA and SNTM: We use the GoPro dataset to validate the significance of the Spiking Neural Temporal Memory Module (SNTM), and there is a significant quantitative performance gap in the first two rows of Table V. It demonstrates that due to the aid of temporal correlation between neighboring events, the de-blurring performance is effectively improved with gains of 0.91 dB in HQF dataset and 1.23 dB in GoPro dataset in terms of PSNR.

The same procedure is used to examine the impact of the Frame-Event Spatial Aggregation Module (FESA), and the results are displayed in the first and third rows of Table V. Clearly, the usage of FESA leads to better performance, with gains up to 0.69 dB and 0.35 dB on GoPro and HQF dataset, respectively. Most importantly, we outperform inserting just one block in terms of de-blurring performance when both SNTM and FESA are included in our network.

TABLE VI: Comparison of different backbones on parameter numbers, FLOPS and Throughput. "*" means that SNN is utilized to replace the TEM (Temporal Memory Block) in STRA.

Methods	# Params	FLOPS	Throughput (images/s)	PSNR
STRA	2.36M	1.10G	22.54	29.73
STRA*	2.01M	0.87G	23.18	27.12

TABLE VII: PSNR values on different parameter settings.

Channel # \ Res-unit #	$R=4$	$R=6$	$R=8$	$R=10$	$R=12$
$C=16$	27.99	28.77	28.98	29.12	29.36
$C=24$	28.10	29.73	29.21	29.47	29.68
$C=32$	28.31	29.57	30.19	30.21	30.24
$C=48$	28.54	29.64	30.19	30.23	30.25
$C=64$	28.67	29.72	30.22	30.24	30.25

c) Effect of SNN: We also evaluate the effect of SNN, and the results are shown in the last five rows in Table V. When adding two layers of SNN, they bring 0.46 dB and 0.31 dB PSNR values in GoPro and HQF dataset, getting the best performance among other settings. Interestingly, if there are three SNN layers in our network, our method gets obvious degradation of performance, with dropping by 1.88 dB in terms of PSNR. It also illuminates the degradation of SNN in deeper layers, as shown in Fig. 5.

We also show FLOP changes by using SNN to replace the TEM (Temporal Memory Block) in STRA, and make comparison with the original STRA. The results are shown in Table VI. By using SNN to replace Temporal Memory Block, our method obtains comparable FLOPS and running time but the performance drops largely, which demonstrates the necessity of Temporal Memory Block. Due to the limitation of shallow SNN architectures, our proposed spiking neural temporal memory module is a hybrid approach for constructing deep network architectures to explore the temporal correlation and capture their long-term dependencies, obtaining the benefits of both SNNs and CNNs.

d) Effect of the ResNet Depth: We also conduct analysis about the influence of hyper-parameters, including the number of channels C and res-units in our network R. Note that we set $C \in (16, 24, 32, 48, 64)$ and $R \in (4, 6, 8, 10, 12)$

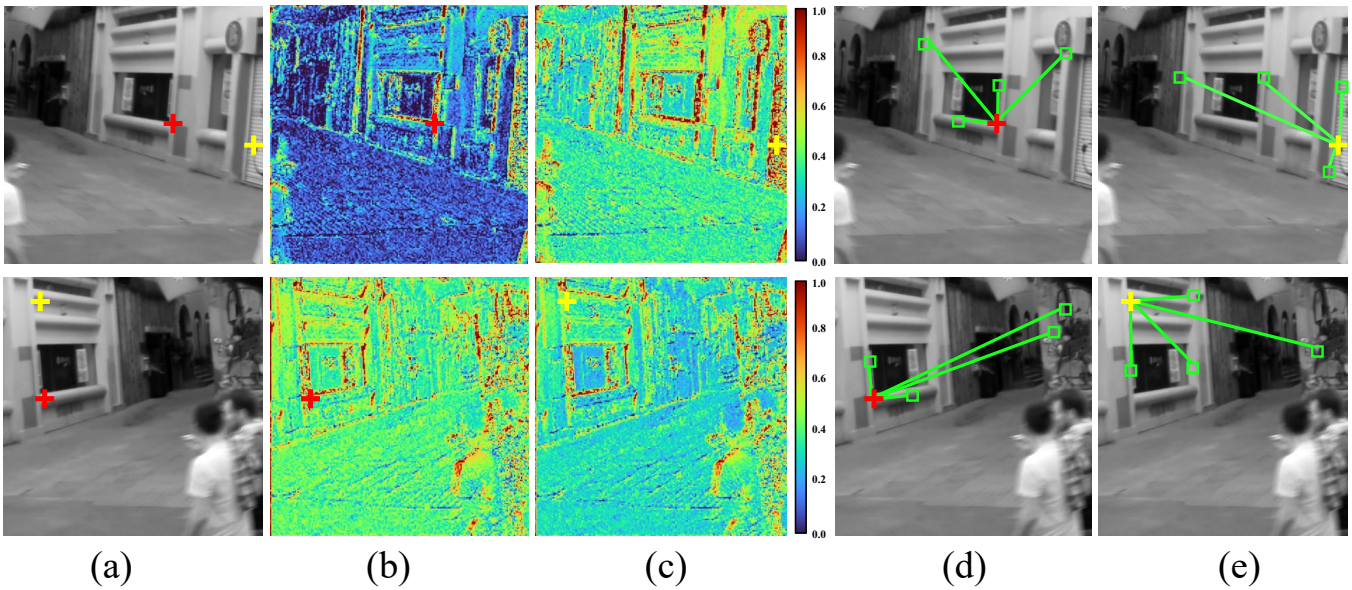

Fig. 12: Visualizations of similarity matrixes in the FESA. Heat maps are used to display the similarity matrices of the query points in (b) and (c). The regions of query points are identified by red and yellow cross. We give some highly linked neighbors in (d) and (e) as examples (marked by green boxes).

TABLE VIII: Ablation study about loss functions on deblurring in GoPro dataset.

Loss function	$\mathcal{L}_1$	$\mathcal{L}_{MSE}$	$\mathcal{L}_{SSIM}$
PSNR↑	29.97	**30.19**	29.85
SSIM↑	0.930	**0.939**	0.927
LPIPS↓	0.077	**0.063**	0.083

in comparison. The results are shown in Table VII. Clearly, our network yields better results as the number of channels increases. Specifically, When $C = 64$ and $R = 12$, our method gets 30.25 dB PSNR values, achieving PSNR gains 0.06 dB than our original settings, but it requires lots of memory assumption and this model has to be trained for more time to obtain good performance. To achieve a good trade-off between efficiency and performance, we set $C = 32$ and $R = 8$ as the default setting.

e) Parameters and running time analysis: To compare the running time, we evaluate different methods by processing 200 frames on a RTX 2080Ti GPU, and record the average time. Results are shown in Fig. 13. With reasonable storage consumption, our method has a comparable running time and provides promising video restoration performance. Note that SNN can improve energy efficiency over time because it is only ever active when it receives or emits spikes, while CNNs, on the other hand, operate with all units active regardless of real-valued input or output values. Additionally, the mathematical dot-product operation is lowered by the fact that an SNN's inputs can only be either 1 or 0. So we deploy SNN with adding less than 0.01M parameters. It is because SNN utilizes the intrinsic sparsity of spatio-temporal events and provides asynchronous computations. Its mechanism naturally encapsulates the asynchronous processing capability, leading to energy-efficient computation.

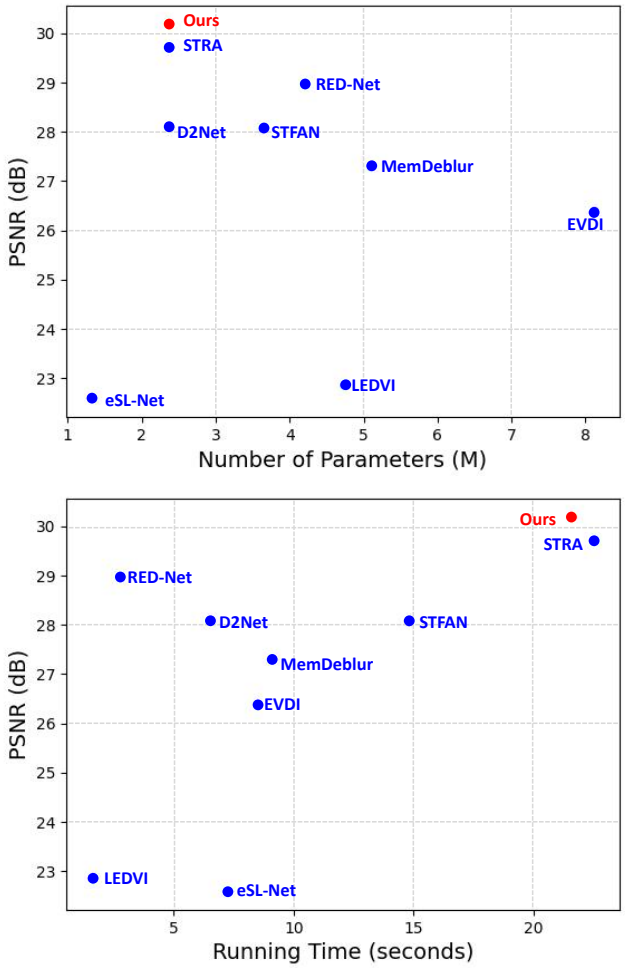

Fig. 13: Comparison of running time and parameter numbers.

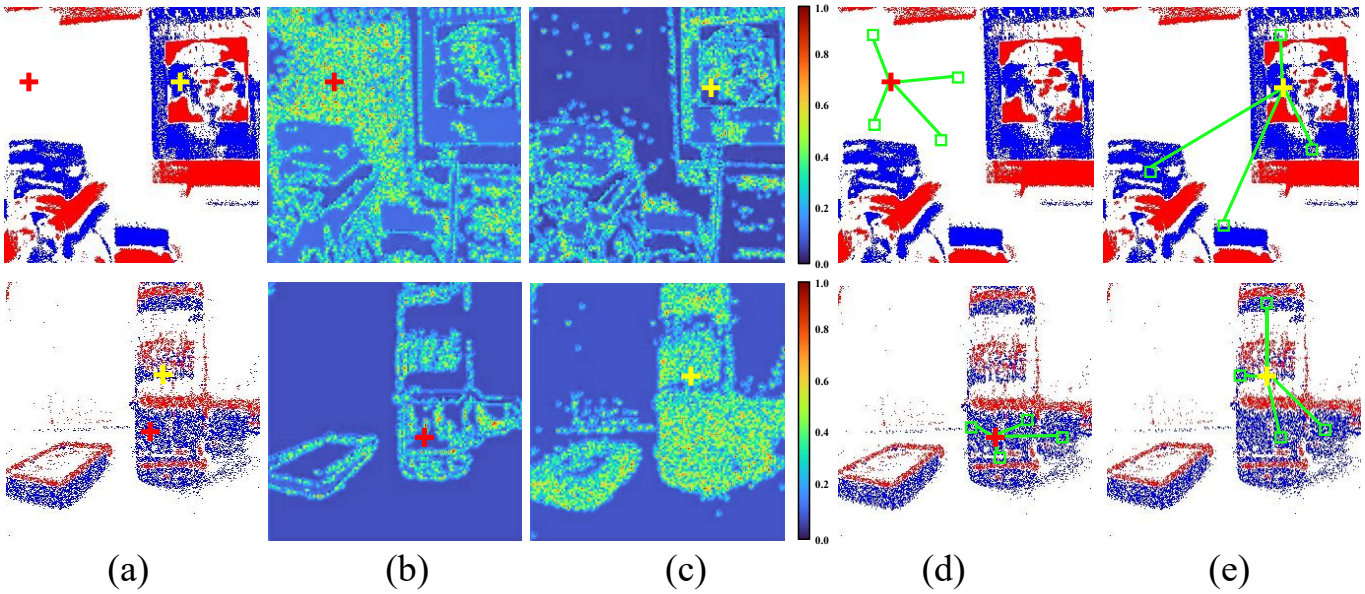

Fig. 14: Visualizations of similarity matrixes in the SNTM. Heat maps are used to display the similarity matrices of the query points in (b) and (c). The regions of query points are identified by red and yellow cross. We give some highly linked neighbors in (d) and (e) as examples (marked by green boxes).

f) Effects of different Loss functions: We conduct ablation study on different loss functions, including $\mathcal{L}_{MSE}$, $\mathcal{L}_1$ and $\mathcal{L}_{SSIM}$. Note that [64] proposes $\mathcal{L}_{SSIM}$ and shows that it has some benefits for low-level computer vision tasks. The results are shown in Table VIII. Obviously, using $\mathcal{L}_{MSE}$ can obtain the best performance. $\mathcal{L}_{MSE}$ is chosen as loss function to train our network to get better results.

H. Visual Analysis

To demonstrate the operation of spatial consistency between frames and events, various attention maps are shown in Fig. 12. Note that The results are normalized and presented in the form of heat maps, and the lighter color indicates higher importance. The location of two query patches is labeled by red and yellow boxes, and some regions with the highest weights are labeled by green boxes in (b) and (c). Take the first row as an example, the lighter color in (b) always appears in the edge of objects, while in (c), it appears in the surface of walls. Clearly, the heat maps of red cross and yellow cross have great difference. The similar brightness actually have more influence on the query patches, which illuminates the effectiveness of our frame-event spatial aggregation module to present long-term spatial correlations in particular patches.

For spiking neural temporal memory module, we present some similarity matrixes with neighbor events in the same way. some highly correlated neighbors are also labeled by green boxes. In the first row of Fig. 14, the patches of red cross has no events, and its higher related regions also appears in marginal areas. In contrast, the higher related regions of the patches in yellow cross always have more events. For event patches, the regions with frequent brightness changes are more significant, which demonstrates how the temporal relationship between adjacent event sequences was constructed.

V. Limitations

While our method achieves the state-of-the-art performance, it still fails on some special scenarios. One limitation of the proposed method is that the network need be retrained if we aim to further increase the frame rate. We can solve it by applying an additional interpolation network recursively between pairs of restored sharp frames. Further research will be devoted to arbitrary frame-rate video reconstruction. For computational requirements, an approach of interest would be to perform a distributed edge-cloud implementation where the SNN blocks and CNN blocks are administered on the edge device and the cloud, respectively. This would lead to high energy benefits on edge devices, which are limited by resource constraints while not compromising on algorithmic performance.

VI. Conclusion

In this paper, we propose a novel Spiking-Convolutional Architecture for event-driven video restoration. By bringing in advantages of SNN and CNN to suit the characteristics of events and frames, the proposed network is capable of restoring videos suffering from various degradations. To model the long-term dependencies in events, the spiking neural temporal memory module continuously maintains the temporal correlation between adjacent event sequences to obtain better performance. Moreover, the frame-event spatial aggregation module calculates spatial consistency by enhancing non-local operations to utilize spatial contexts from them. Consequently, the framework learns to reconstruct high-quality regions depending on adequate temporal and spatial information provided by events. We evaluate the proposed method on numerous synthetic and real-world datasets, which clearly shows that our method outperforms the state-of-the-art methods for several video restoration tasks.

REFERENCES

[1] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren, "Spatio-temporal filter adaptive network for video deblurring," pp. 2482–2491, 2019.

[2] Haibin Duan and Xiaohua Wang, "Echo state networks with orthogonal pigeon-inspired optimization for image restoration," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 11, pp. 2413–2425, 2015.

[3] Saeed Anwar, Nick Barnes, and Lars Petersson, "Attention-based real image restoration," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2021.

[4] Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza, "Time lens: Event-based video frame interpolation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16155–16164.

[5] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu, "Tdan: Temporally-deformable alignment network for video super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3360–3369.

[6] Zihao W Wang, Weixin Jiang, Kuan He, Boxin Shi, Aggelos Katsaggelos, and Oliver Cossairt, "Event-driven video frame synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[7] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng, "Efficient spatio-temporal recurrent neural network for video deblurring," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer, 2020, pp. 191–207.

[8] Songnan Lin, Jiawei Zhang, Jinshan Pan, Zhe Jiang, Dongqing Zou, Yongtian Wang, Jing Chen, and Jimmy Ren, "Learning event-driven video deblurring and interpolation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*. Springer, 2020, pp. 695–710.

[9] Jinshan Pan, Haoran Bai, and Jinhui Tang, "Cascaded deep video deblurring using temporal sharpness prior," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3043–3051.

[10] Xuecai Hu, Zhang Zhang, Caifeng Shan, Zilei Wang, Liang Wang, and Tieniu Tan, "Meta-usr: A unified super-resolution network for multiple degradation parameters," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 9, pp. 4151–4165, 2020.

[11] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian, "Video super-resolution with recurrent structure-detail network," pp. 645–660, 2020.

[12] Lei Zhang, Jiangtao Nie, Wei Wei, Yong Li, and Yanning Zhang, "Deep blind hyperspectral image super-resolution," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2388–2400, 2020.

[13] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4947–4956.

[14] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck, "A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[15] Gregory Cohen, Saeed Afshar, Garrick Orchard, Jonathan Tapson, Ryad Benosman, and Andre van Schaik, "Spatial and temporal downsampling in event-based visual classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 5030–5044, 2018.

[16] Yi Zhou, Guillermo Gallego, Xiuyuan Lu, Siqi Liu, and Shaojie Shen, "Event-based motion segmentation with spatio-temporal graph cuts," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2021.

[17] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu, "Learning event-based motion deblurring," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3320–3329.

[18] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony, "Reducing the sim-to-real gap for event cameras," pp. 534–549, 2020.

[19] Bishan Wang, Jingwei He, Lei Yu, Gui-Song Xia, and Wen Yang, "Event enhanced high-quality image recovery," pp. 155–171, 2020.

[20] Wei Shang, Dongwei Ren, Dongqing Zou, Jimmy S Ren, Ping Luo, and Wangmeng Zuo, "Bringing events into video deblurring with non-consecutively blurry frames," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4531–4540.

[21] Chengzhi Cao, Xueyang Fu, Yurui Zhu, Gege Shi, and Zheng-Jun Zha, "Event-driven video deblurring via spatio-temporal relation-aware network," *IJCAI*, 2022.

[22] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza, "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2022.

[23] Songnan Lin, Jiawei Zhang, Jinshan Pan, Zhe Jiang, Dongqing Zou, Yongtian Wang, Jing Chen, and Jimmy Ren, "Learning event-driven video deblurring and interpolation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*. Springer, 2020, pp. 695–710.

[24] Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao, "Turning frequency to resolution: Video super-resolution via event cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7772–7781.

[25] Jin Han, Yixin Yang, Chu Zhou, Chao Xu, and Boxin Shi, "Evintsr-net: Event guided multiple latent frames reconstruction and super-resolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4882–4891.

[26] Yunhao Zou, Yinqiang Zheng, Tsuyoshi Takatani, and Ying Fu, "Learning to Reconstruct High Speed and High Dynamic Range Videos from Events," in *CVPR*, 2021.

[27] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai, "Bringing a blurry frame alive at high frame-rate with an event camera," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6820–6829.

[28] Fang Xu, Lei Yu, Bishan Wang, Wen Yang, Gui-Song Xia, Xu Jia, Zhendong Qiao, and Jianzhuang Liu, "Motion deblurring with real events," pp. 2583–2592, 2021.

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[30] Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian, "Deep Residual Learning in Spiking Neural Networks," in *Advances in Neural Information Processing Systems*, 2021.

[31] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothee Masquelier, and Anthony S. Maida, "Deep Learning in Spiking Neural Networks," *Neural Networks*, 2019.

[32] Isha Garg, Sayeed Shafayet Chowdhury, and Kaushik Roy, "DCT-SNN: Using DCT to Distribute Spatial Information over Time for Low-Latency Spiking Neural Networks," in *ICCV*, 2021.

[33] Chankyu Lee, Adarsh Kumar Kosta, Alex Zihao Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy, "Spike-FlowNet: Event-Based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks," in *ECCV*. 2020.

[34] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li, "Temporal-Wise Attention Spiking Neural Networks for Event Streams Classification," in *ICCV*, 2021.

[35] Federico Paredes-Vallés, Kirk Y. W. Scheper, and Guido C. H. E. de Croon, "Unsupervised Learning of a Hierarchical Spiking Neural Network for Optical Flow Estimation: From Events to Global Motion Perception," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[36] Chankyu Lee, Syed Shakib Sarwar, Priyadarshini Panda, Gopalakrishnan Srinivasan, and Kaushik Roy, "Enabling spike-based backpropagation for training deep neural network architectures," *Frontiers in neuroscience*, p. 119, 2020.

[37] Siqi Li, Yutong Feng, Yipeng Li, Yu Jiang, Changqing Zou, and Yue Gao, "Event stream super-resolution via spatiotemporal constraint learning," pp. 4480–4489, 2021.

[38] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko, "Rethinking coarse-to-fine approach in single image deblurring," pp. 4641–4650, 2021.

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He, "Non-local neural networks," pp. 7794–7803, 2018.

[40] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang, "Selective kernel networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 510–519.

[41] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck, "v2e: From video frames to realistic dvs events," pp. 1312–1321, 2021.

[42] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey, "Need for speed: A benchmark for higher frame rate object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1125–1134.

[43] Jie Chen, Cheen-Hau Tan, Junhui Hou, Lap-Pui Chau, and He Li, "Robust video content alignment and compensation for rain removal in a cnn framework," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6286–6295.

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

[45] Xiang Zhang and Lei Yu, "Unifying motion deblurring and frame interpolation with events," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17765–17774.

[46] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[47] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren, "Spatio-temporal filter adaptive network for video deblurring," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2482–2491.

[48] Bo Ji and Angela Yao, "Multi-scale memory-based video deblurring," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1919–1928.

[49] Dasong Li, Xiaoyu Shi, Yi Zhang, Ka Chun Cheung, Simon See, Xiaogang Wang, Hongwei Qin, and Hongsheng Li, "A simple baseline for video restoration with grouped spatial-temporal shift," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9822–9832.

[50] Dan Yang and Mehmet Yamac, "Deformable convolutions and lstm-based flexible event frame fusion network for motion deblurring," *arXiv preprint arXiv:2306.00834*, 2023.

[51] Zongsheng Yue, Qian Zhao, Jianwen Xie, Lei Zhang, Deyu Meng, and Kwan-Yee K Wong, "Blind image super-resolution with elaborate degradation modeling on noise and kernel," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2128–2138.

[52] Guangwei Gao, Zhengxue Wang, Juncheng Li, Wenjie Li, Yi Yu, and Tieyong Zeng, "Lightweight bimodal network for single-image super-resolution via symmetric cnn and recursive transformer," *arXiv preprint arXiv:2204.13286*, 2022.

[53] Benjamin Naoto Chiche, Arnaud Woiselle, Joana Frontera-Pons, and Jean-Luc Starck, "Stable long-term recurrent video super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 837–846.

[54] Junghun Oh, Heewon Kim, Seungjun Nah, Cheeun Hong, Jonghyun Choi, and Kyoung Mu Lee, "Attentive fine-grained structured sparsity for image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17673–17682.

[55] Chengxu Liu, Huan Yang, Jianlong Fu, and Xueming Qian, "Learning trajectory-aware transformer for video super-resolution," in *CVPR*, 2022.

[56] Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, Liang-Jian Deng, and Yao Wang, "A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2017, pp. 4057–4066.

[57] Jie Chen, Cheen-Hau Tan, Junhui Hou, Lap-Pui Chau, and He Li, "Robust video content alignment and compensation for rain removal in a cnn framework," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6286–6295.

[58] Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng, "Video rain streak removal by multiscale convolutional sparse coding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6644–6653.

[59] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo, "D3r-net: Dynamic routing residue recurrent network for video rain removal," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 699–712, 2018.

[60] Wei Wei, Lixuan Yi, Qi Xie, Qian Zhao, Deyu Meng, and Zongben Xu, "Should we encode rain streaks in video as deterministic or stochastic?," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2516–2525.

[61] Jin-Hwan Kim, Jae-Young Sim, and Chang-Su Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2658–2670, 2015.

[62] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo, "Erase or fill? deep joint recurrent rain removal and reconstruction in videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3233–3242.

[63] Hong Wang, Zongsheng Yue, Qi Xie, Qian Zhao, Yefeng Zheng, and Deyu Meng, "From rain generation to rain removal," in *CVPR*, 2021.

[64] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng, "Progressive image deraining networks: A better and simpler baseline," pp. 3937–3946, 2019.

Chengzhi Cao is currently pursuing the M.S. degree with the National Engineering Laboratory for Brain-inspired Intelligence Technology and Application (NEL-BITA), University of Science and Technology of China, Hefei, China. His research interests include bio-inspired computer vision and image processing.

Xueyang Fu received the Ph.D. degree in signal and information processing from Xiamen University in 2018. He was a Visiting Scholar with Columbia University, sponsored by the China Scholarship Council, from 2016 to 2017. He is currently an Associate Researcher with the Department of Automation, University of Science and Technology of China. His research interests include machine learning and image processing.

Yurui Zhu received the BEng degree from the University of Science and Technology of China. He is currently working toward the PhD degree with the National Engineering Laboratory for Brain-inspired Intelligence Technology and Application (NEL-BITA), University of Science and Technology of China, Hefei, China. His research interests focus on computer vision, especially low-level vision tasks.

Zhijing Sun received the B.S. degree in nuclear engineering and technology from Shanghai Jiao Tong University in 2022. He is currently working toward the M.S. degree in Automation with the Department of Automation, University of Science and Technology of China. His research interests include low-level vision, event vision and image processing.

Zheng-Jun Zha received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2004 and 2009, respectively.

He is currently a Full Professor with the School of Information Science and Technology and the Executive Director of the National Engineering Laboratory for Brain-Inspired Intelligence Technology and Application, University of Science and Technology of China. His research interests include multimedia analysis, retrieval and applications, computer vision, as well as pattern recognition. He has authored or coauthored over 100 papers in these areas with a series of publications on top journals and conferences.

Dr. Zha was the recipient of multiple paper awards from prestigious multimedia conferences, including the Best Paper Award and Best Student Paper Award in ACM Multimedia. He serves as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.