# Event-Based Frame Interpolation with Ad-hoc Deblurring

Lei Sun[1,2] Christos Sakaridis[2] Jingyun Liang[2] Peng Sun[1] Jiezhang Cao[2] Kai Zhang[2]
Qi Jiang[1] Kaiwei Wang[1] Luc Van Gool[2,3]

[1]Zhejiang University [2]ETH Zürich [3]KU Leuven

## Abstract

*The performance of video frame interpolation is inherently correlated with the ability to handle motion in the input scene. Even though previous works recognize the utility of asynchronous event information for this task, they ignore the fact that motion may or may not result in blur in the input video to be interpolated, depending on the length of the exposure time of the frames and the speed of the motion, and assume either that the input video is sharp, restricting themselves to frame interpolation, or that it is blurry, including an explicit, separate deblurring stage before interpolation in their pipeline. We instead propose a general method for event-based frame interpolation that performs deblurring ad-hoc and thus works both on sharp and blurry input videos. Our model consists in a bidirectional recurrent network that naturally incorporates the temporal dimension of interpolation and fuses information from the input frames and the events adaptively based on their temporal proximity. In addition, we introduce a novel real-world high-resolution dataset with events and color videos named HighREV, which provides a challenging evaluation setting for the examined task. Extensive experiments on the standard GoPro benchmark and on our dataset show that our network consistently outperforms previous state-of-the-art methods on frame interpolation, single image deblurring and the joint task of interpolation and deblurring. Our code and dataset are available at https://github.com/AHupuJR/REFID.*

## 1. Introduction

Video frame interpolation (VFI) methods synthesize intermediate frames between consecutive input frames, increasing the frame rate of the input video, with wide applications in super-slow generation [11, 13, 20], video editing [27, 45], virtual reality [1], and video compression [40]. With the absence of inter-frame information, frame-based methods explicitly or implicitly utilize motion models such as linear motion [13] or quadratic motion [41]. However, the non-linearity of motion in real-world videos makes it hard to accurately capture inter-frame motion with these simple models.

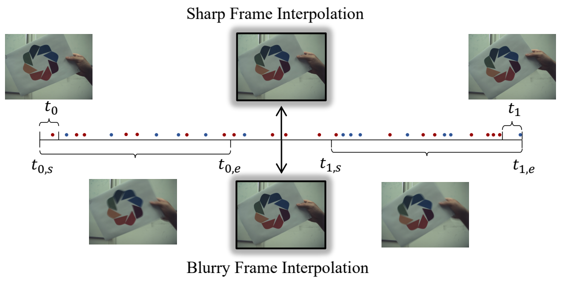

Figure 1. Our unified framework for event-based sharp and blurry frame interpolation. Red/blue dots: negative/positive events; Curly braces: exposure time range.

Recent works introduce event cameras in VFI as a proxy to estimate the inter-frame motion between consecutive frames. Event cameras [7] are bio-inspired asynchronous sensors that report per-pixel intensity changes, *i.e.*, *events*, instead of synchronous full intensity images. The events are recorded at high temporal resolution (in the order of $\mu$s) and high dynamic range (over 140 dB) within and between frames, providing valid compressed motion information. Previous works [9, 36, 37] show the potential of event cameras in VFI, comparing favorably to frame-only methods, especially in high-speed non-linear motion scenarios, by using spatially aligned events and RGB frames. These event-based VFI methods make the crucial assumption that the input images are sharp. However, this assumption is violated in real-world scenes because of the ubiquitous motion blur. In particular, because of the finite exposure time of frames in real-world videos, especially of those captured with event cameras that output both image frames and an event stream (*i.e.*, Dynamic and Activate VIsion Sensor (DAVIS) [3])—which have a rather long exposure time and low frame rate, motion blur is inevitable for high-speed scenes. In such a scenario, where the reference frames for VFI are degraded by motion blur, the performance of frame interpolation also degrades.

As events encode motion information within and between frames, several studies [4, 18, 22] are carried out on event-based deblurring in conjunction with VFI. However, these works approach the problem via cascaded deblurring and interpolation pipelines and the performance of VFI is limited by the image deblurring performance.

Thus, the desideratum in event-based VFI is robust performance on both sharp image interpolation and blurry image interpolation. Frame-based methods [12, 17, 17, 21, 30, 46] usually treat these two aspects as separate tasks. Different from frames, events are not subject to motion blur. No matter whether the frame is sharp or blurry, the corresponding events are the same. Based on this observation, we propose to unify the two aforementioned tasks into one problem: *given two input images and a corresponding event stream, restore the latent sharp images at arbitrary times between the input images.* The input images could be either blurry or sharp, as Fig. 1 shows. To solve this problem, we first revisit the physical model of event-based deblurring and frame interpolation. Based on this model, we propose a novel recurrent network, which can perform both event-based sharp VFI and event-based blurry VFI. The network consists of two branches, an image branch and an event branch. The recurrent structure pertains to the event branch, in order to enable the propagation of information from events across time in both directions. Features from the image branch are fused into the recurrent event branch at multiple levels using a novel attention-based module for event-image fusion, which is based on the squeeze-and-excitation operation [10].

To test our method on a real-world setting and motivated by the lack of event-based datasets recorded with high-quality event cameras, we record a dataset, HighREV, with high-resolution chromatic image sequences and corresponding events. From the sharp image sequences, we synthesize blurry images by averaging several consecutive frames [19]. To our knowledge, HighREV has the highest event resolution among all publicly available event datasets.

In summary, we make the following contributions:

- We propose a framework for solving general event-based frame interpolation and event-based single image deblurring, which builds on the underlying physical model of high-frame-rate video frame formation and event generation.
- We introduce a novel network for solving the above tasks, which is based on a bi-directional recurrent architecture, includes an event-guided channel-level attention fusion module that adaptively attends to features from the two input frames according to the temporal proximity with features from the event branch, and achieves state-of-the-art results on both synthetic and real-world datasets.
- We present a new real-world high-resolution dataset with events and RGB videos, which enables real-world evaluation of event-based interpolation and deblurring.

## 2. Related Work

**Event-based frame interpolation.** Because event cameras report the per-pixel intensity changes, they provide useful spatio-temporal information for frame interpolation. Tulyakov *et al.* [37] propose Time Lens, which combines a warping-based method and a synthesis-based method with a late-fusion module. Time Lens++ [36] further improves the efficiency and performance via computing motion splines and multi-scale fusion separately. TimeReplayer [9] utilizes a cycle-consistency loss as supervision signal, making a model trained on low-frame-rate videos also able to predict high-speed videos. All the methods above assume that the key frame is sharp, but in high-speed or low-illumination scenarios, the key frame inevitably gets blurred because of the high-speed motion within the exposure time, where these methods failed (Tab. 1). Hence, the exposure time should be taken into consideration in real-world scenes.

**Event-based deblurring.** Due to the high temporal resolution, event cameras provide motion information within the exposure time, which is a natural motion cue for image deblurring. Thus, several works have focused on event-based image deblurring. Jiang *et al.* [14] used convolutional models and mined the motion information and edge information to assist deblurring. Sun *et al.* [34] proposed a multi-head attention mechanism for fusing information from both modalities, and designed an event representation specifically for the event-based image deblurring task. Kim *et al.* [15] further extended the task to images with unknown exposure time by activating the events that are most related to the blurry image. These methods only explore the single image deblurring setting, where the timestamp of the deblurred image is in the middle of the exposure time. However, the events encode motion information for the entire exposure time, and latent sharp images at arbitrary points within the exposure time can be estimated in theory.

**Joint frame interpolation and enhancement.** Pan *et al.* [22] formulate the Event Double Integral (EDI) deblurring model, which is derived from the definition of image blur and the measurement mechanism of event cameras, and perform both image deblurring and frame interpolation by accumulating events and applying the intensity changes within the exposure time and from the key frame to the synthesized frames, respectively. This seminal work optimizes the model by minimizing an energy function but is limited by practical issues in the measurement mechanism of event cameras, e.g. accumulated noise, dynamic contrast thresholds and missing events. Based on EDI, a differentiable model and a residual learning denoising model to improve the result is introduced in [39]. Recent works [23, 44] identify the relationship between the events and the latent sharp image, and apply it to self-supervised event-based image reconstruction and image deblurring. However, the above works on joint frame interpolation and deblurring predict

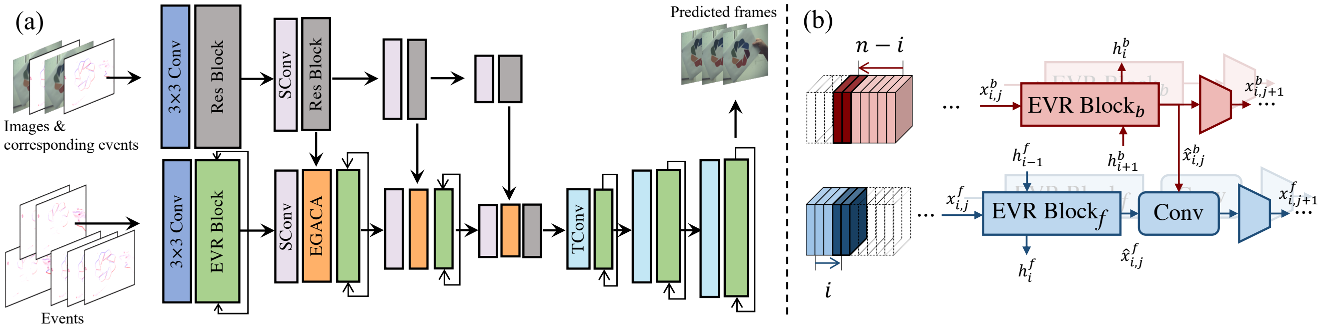

Figure 2. (a): **The architecture of our Recurrent Event-based Frame Interpolation with ad-hoc Deblurring (REFID) network.** The input of the image branch consists of two key frames and their corresponding events, and the event branch consumes sub-voxels of events recurrently. "EGACA": event-guided adaptive channel attention, "SConv": strided convolution, "TConv": transposed convolution. (b): **The proposed bidirectional event recurrent (EVR) blocks.** In each recurrent step, the events from the forward and backward direction are fed to the network. For notations, cf. (8).

the latent frames with a two-stage deblurring+interpolation approach, which limits the performance of VFI.

## 3. Method

We first revisit the physical model of event-based frame interpolation and deblurring in Sec. 3.1. Based on this model, we argue that the events within the exposure time should not be ignored in event-based frame interpolation, and present our model architecture abstracted from the physical model in Sec. 3.2. To perform the bidirectional recurrent propagation, we demonstrate the data preparation in Sec. 3.3. In Sec. 3.4 and Sec 3.5, we introduce the proposed bidirectional Event Recurrent Block and Event-Guided Adaptive Channel Attention in detail.

### 3.1. Problem Formulation

Once the change in intensity $\mathcal{I}$ at a pixel between the current moment and the moment of the last generated event at that pixel surpasses the contrast threshold $c$, an event camera emits the $i$-th event $e_i$, represented as a tuple $e_i = (x_i, y_i, t_i, p_i)$, where $x_i$ and $y_i$ represent the pixel coordinates of the event, $t_i$ represents its timestamp, and $p_i$ is the polarity of the event. More formally, this can be written as

$$p_i = \begin{cases} +1, \text{if} \log\left(\frac{\mathcal{I}_t(x_i,y_i)}{\mathcal{I}_{t-\Delta t}(x_i,y_i)}\right) > c, \\ -1, \text{if} \log\left(\frac{\mathcal{I}_t(x_i,y_i)}{\mathcal{I}_{t-\Delta t}(x_i,y_i)}\right) < -c. \end{cases} \tag{1}$$

Ideally, given two consecutive images, referred to as the left frame $I_0$ and the right frame $I_1$, and the corresponding event stream in the time range between the timestamps of the two images $[t_0, t_1]$, we can get any latent image $\hat{I}_\tau$ with timestamp $\tau$ in $[t_0, t_1]$ via

$$\begin{aligned} \hat{I}_\tau &= I_0 \exp(c \int_{t_0}^{t} p(s)ds), \\ \hat{I}_\tau &= I_1 \exp(c \int_{t_1}^{t} p(s)ds), \end{aligned} \tag{2}$$

where $p(s)$ is the polarity component of the event stream.

Previous event-based methods [9, 36, 37] solve event-based frame interpolation based on (2). However, in the real-world setting, because of the finite exposure times of the two frames, the timestamps $t_0$ and $t_1$ should be replaced by time ranges, and the images $I_0$ and $I_1$ may be either sharp (small motion in the exposure time) or blurry (large motion in the exposure time). Thus, the events within the exposure time of the frames, $E$, should also be utilized for removing potential blur from the frames:

$$\text{Deblur}(I, E) = \frac{B \times T}{\int_{t_s}^{t_e} \exp\left(c \int_{\frac{t_s+t_e}{2}}^{t} p(s)ds\right) dt}, \tag{3}$$

where $B$, $T$, $t_s$ and $t_e$ are the blurry frame, length of exposure time, start and end of exposure time, respectively. Previous studies [4, 14, 18, 22] combine the above deblur equation with the frame interpolation equation (2) (denoted as Interpo) to synthesize the target frame:

$$\begin{aligned} \hat{I}_{\tau,0} &= \text{Deblur}(I_0, E_0)\text{Interpo}(E_{t_{0,s}\to\tau}), \\ \hat{I}_{\tau,1} &= \text{Deblur}(I_1, E_1)\text{Interpo}(E_{\tau\leftarrow t_{1,e}}), \end{aligned} \tag{4}$$

where $E_{t_{0,s}\to\tau}$ and $E_{\tau\leftarrow t_{1,e}}$ indicate the intensity changes—recorded as events—from the start of the exposure time of the left frame, $t_{0,s}$, and the end of the exposure time of the right frame, $t_{1,e}$, to the target timestamp $\tau$.

However, the physical model (4) is prone to sensor noise and to the varying contrast threshold of an event camera, which is a inherent drawback of such cameras.

Based on (4), [4, 18, 22, 44] design deep neural networks with a cascaded *first-deblur-then-interpolate* pipeline to perform blurry frame interpolation. In these two-stage methods, the performance of frame interpolation (second stage) is limited by the performance of image deblurring (first stage). Moreover, these methods are only evaluated on blurry frame interpolation.

Given the left and right frame, we design a unified framework to perform event-based frame interpolation both for sharp and blurry inputs with a one-stage model, which applies deblurring ad-hoc.

### 3.2. General Architecture

The physical model of (4) indicates that the latent sharp frame at time $\tau$ can be derived from the two consecutive frames and the corresponding events as

$$\begin{aligned} \hat{I}_{\tau,0} &= \mathbf{F}(\mathbf{G}(I_0, E_0), E_{t_{0,s}\to\tau}), \\ \hat{I}_{\tau,1} &= \mathbf{F}(\mathbf{G}(I_1, E_1), E_{\tau\leftarrow t_{1,e}}), \end{aligned} \tag{5}$$

where $\mathbf{G}$ and $\mathbf{F}$ are learned parametric mappings. Contrary to the formulation of (4), $\mathbf{G}$ does not accomplish solely image deblurring, but rather extracts features of both absolute intensities (image) and relative intensity changes (events) within the exposure time. We use cascaded residual blocks to model this mapping. For each latent frame, previous methods collect the events in both time ranges and convert them to an event representation [9,37], which may incur inconsistencies in the result [36]. To mitigate this, we use a recurrent network to naturally model temporal information. Thus, we abstract the physical model (4) to:

$$\begin{aligned} \hat{I}_{\tau,0} &= \mathbf{EVR}_f(\mathbf{G}(I_0, E_0, I_1, E_1), E_\tau, E_{t_{0,s}\to\tau}), \\ \hat{I}_{\tau,1} &= \mathbf{EVR}_b(\mathbf{G}(I_0, E_0, I_1, E_1), E_\tau, E_{\tau\leftarrow t_{1,e}}), \end{aligned} \tag{6}$$

where $\mathbf{EVR}_f$ and $\mathbf{EVR}_b$ denote forward and backward event recurrent (EVR) blocks, respectively. (6) summarizes the architecture of our proposed method, named **R**ecurrent **E**vent-based **F**rame **I**nterpolation with ad-hoc **D**eblurring (REFID). $E_\tau$ refers to the events in a small time range centered around $\tau$. The recurrent blocks accept as input not only current events, but also previous event information through their hidden states.

Because of the sensor noise and the varying contrast threshold of the event camera sensor, $\hat{I}_{\tau,0}$ ($\hat{I}_{\tau,1}$) approximates the latent sharp image more accurately when the corresponding timestamp of the latter, $\tau$, is closer to $t_0$ ($t_1$). To fuse $\hat{I}_{\tau,0}$ and $\hat{I}_{\tau,1}$ implicitly, we further propose a new Event-Guided Adaptive Channel Attention (EGACA) module to mine and fuse the features from the image branch of REFID with adaptive weights determined by the current events:

$$\hat{I}_\tau = \mathrm{Fuse}(\hat{I}_{\tau,0}, \hat{I}_{\tau,1}) \tag{7}$$

The overall network architecture of REFID is shown in Fig. 2 (a). The image branch extracts features from the two input images and the corresponding events and is connected to the event branch at multiple scales. Overall, REFID has a U-Net [28] structure. A bidirectional recurrent encoder with EVR blocks extracts features from current events and models temporal relationships with previous and future events. In each block of the encoder, the features from the image branch are fused with those from the event branch adaptively with our novel EGACA module, which we detail in Sec. 3.5.

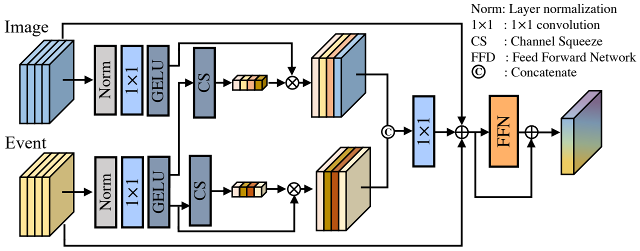

Figure 3. **The Event-Guided Adaptive Channel Attention module.** The channel weights for the image branch are extracted from the event branch.

### 3.3. Data Preparation

To feed the asynchronous events to our network, we first need to convert them to a proper representation. According to (6), the latent image can be derived in both temporal directions. Thus, apart from the forward event stream, we reverse the event stream both in time and polarity to get a backward event stream. Then, event streams from the two directions are converted to two voxel grids [25, 26] $V \in \mathbb{R}^{(n+2)\times H\times W}$, where $n$ is the number of interpolated frames. The channel dimension of the voxel grids holds discrete temporal information. In each recurrent iteration, $V_{\text{sub}} \in \mathbb{R}^{2\times H\times W}$ from both directions are fed to the event branch, which encodes the event information for the latent frame. We also convert events in the exposure time of the two images to voxel grids and concatenate them with corresponding images to form the input of the image branch.

### 3.4. Bidirectional Event Recurrent Block

In previous event-based works [9, 36, 37], for each latent sharp image, the events from both left and right images to the target image are accumulated and converted to an event representation. However, compared to the temporal resolution of events, the length of the exposure time of frames is large and not negligible, so simple accumulation from a single timestamp in the above works loses information and is not reasonable. Moreover, inference for different latent frames is segregated, which leads to inconsistencies in the results [36]. To deal with these problems, we propose a recurrent architecture that models the temporal information both within the exposure time of each frame and between exposure times of different frames. By adopting recurrent blocks, frame interpolation is independent from the exposure time of key frames and it can also be performed inside the exposure time. Features propagated through hidden states of the network also guarantee consistency across the

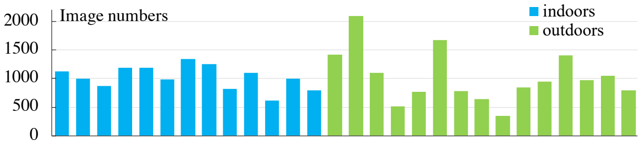

Figure 4. **The distribution of number of ground-truth images per sequence of HighREV dataset.** The x-axis denotes the sequences.

predicted frames. Based on (5), we design a bidirectional Event Recurrent (EVR) block to iteratively extract features from the event branch. As Fig. 2 (b) shows, for each direction, the input sub-voxel $(i-1, i)$ only consists of two voxels of the input voxel. In the next recurrent iteration, the selected sub-voxel moves forward to the next time $(i, i+1)$. For a given recurrent iteration $i$, the forward EVR block cycles for $i$ times and the backward EVR block cycles for $n-i$ times, where $n$ is the index of the latent sharp image at hand:

$$\begin{aligned}
&\hat{x}^b_{i,j+1}, h^b_i = \mathbf{EVR}_b(x^b_{i,j}, h^b_{i+1}),\\
&\hat{x}^f_{i,j+1}, h^f_i = \mathbf{EVR}_f(x^f_{i,j}, h^f_{i-1}),\\
&x^b_{i,j+1} = \mathrm{Down}(\hat{x}^b_{i,j+1}),\\
&x^f_{i,j+1} = \mathrm{Down}(\mathrm{Conv}(\mathrm{Concat}(\hat{x}^b_{i,j+1}, \hat{x}^f_{i,j+1}))),
\end{aligned} \tag{8}$$

where $i$ and $j$ are the indices of sub-voxel and scale, respectively. $x$, $h$, $f$ and $b$ denote feature flow, hidden state, forward and backward, respectively. We select ResNet as the architecture for the EVR block instead of ConvLSTM [31] or ConvGRU [32], because the time range of events between consecutive frames is rather short (cf. Tab 4). In each EVR block, the features from the two directions are fused through convolution and downsampled to half of the original size. The bidirectional EVR blocks introduce the information flow from both directions, which models $E_{t_{0,s} \to \tau}$ and $E_{\tau \leftarrow t_{1,e}}$ in (5), helping reduce artifacts by using the information from the end of the time interval (cf. Fig. 5).

### 3.5. Event-Guided Adaptive Channel Attention

In event-based frame interpolation, fusion happens both between the two input frames and between frames and events. Because of the inherent noise of event cameras, the longer the time range between the key frames is, the more the noise in the event accumulation increases. Ideally, the key frame that is closer to the latent frame should contribute more to the prediction of the latter. In other words, the weights of two key frames should be decided by *time*.

In our REFID network, the two key frames and the corresponding events are concatenated along the channel dimension to provide the input of the image branch. We design the novel Event-Guided Adaptive Channel Attention (EGACA) module to fuse the two key frames and events at the current input sub-voxel in the recurrent structure. The current input sub-voxel contains events in a small range around the timestamp of the latent frame and the fusion weights for the two key frames and the events are determined by the current input sub-voxel, which indicates the time.

Fig. 3 shows the detailed architecture of the proposed EGACA. We simplify the multi-head channel attention of EFNet [34] to channel attention from SENet [10]. Two Channel Squeeze (CS) blocks extract channel weights from the current events, and two weights multiply event features and image features for self-attention and event-guided attention to image features, respectively. Then, feature maps from the two branches are fused by a feed-forward network. In each recurrent iteration, the channel weights from the current events are different, which helps to mine different features from the two images along the channel dimension.

## 4. HighREV Dataset

For event-based low-level tasks, such as event-based image deblurring and event-based frame interpolation, most works evaluate their models on datasets originally designed for image-only methods and having only synthetic events. This is because (1) event cameras are not easy to acquire yet, (2) most event cameras are of low resolution and monochrome [14, 15, 34], and (3) high-resolution chromatic datasets [37, 38] are not publicly available. To fill this gap, we record a high-quality chromatic event-image dataset for training, fine-tuning and evaluating event-based methods for frame interpolation and deblurring.

As Fig. 4 shows, our HighREV dataset consists of 30 sequences with a total of 28589 sharp images and corresponding events. We use 19934 images for training/fine-tuning and 8655 images for evaluation. The size of each RGB image is 1632 × 1224. The events and images are spatially aligned in the sensor. Each event has only one channel (intensity), with pixel coordinates, timestamp and polarity.

The HighREV dataset can be used for event-based sharp frame interpolation. To evaluate event-based blurry frame interpolation, we synthesize blurry images by averaging 11 consecutive original sharp frames. For blurry frame interpolation, we skip 1 or 3 sharp frames (denoted as *11+1* or *11+3* in Tab. 1). To the best of our knowledge, among all event-image datasets, our dataset has the highest resolution.

## 5. Experiments

### 5.1. Tasks and Datasets

We use the popular GoPro dataset [19] for training and evaluation. GoPro provides blurry images, paired sharp images, and sharp image sequences used to synthesize blurry frames. The images have a size of $1280 \times 720$. We leverage the event camera simulator ESIM [24] to generate simulated event data with threshold $c$ following a Gaussian distribution $N(\mu = 0.2, \sigma = 0.03)$. For different tasks, the datasets are as follows:

Table 1. **Comparison of blurry frame interpolation methods on GoPro [19] and HighREV.** "Frames" and "Events" indicate if a method uses frames and events for interpolation. "11+1" (resp. "11+3") indicates that the blurry image is synthesized with 11 sharp frames and 1 (resp. 3) frame(s) is skipped for frame interpolation. The number of network parameters (#Param) is also provided.

| Method | Frames | Events | PSNR ↑ | SSIM ↑ | PSNR ↑ | SSIM ↑ | #Param |
|---|---|---|---|---|---|---|---|
| **GoPro** [19] | | | **11+1** | | **11+3** | | |
| RIFE [12] | ✔ | ✘ | 28.69 | 0.856 | 26.91 | 0.798 | 9.8M |
| EDI [22] | ✔ | ✔ | 18.72 | 0.506 | 18.49 | 0.486 | 0.5M |
| Time Lens [37] | ✔ | ✔ | 21.56 | 0.581 | 21.47 | 0.587 | 72.9M |
| EVDI [44] | ✔ | ✔ | 29.17 | 0.880 | 28.77 | 0.873 | 0.4M |
| EFNet+IFRNet [17, 34] | ✔ | ✔ | 33.05 | 0.955 | 32.89 | 0.950 | 28.2M |
| E2VID+ [26] | ✔ | ✔ | 33.82 | 0.961 | 33.39 | 0.954 | 15.3M |
| **REFID (Ours)** | ✔ | ✔ | **35.90** | **0.973** | **35.47** | **0.971** | 15.9M |
| **HighREV** | | | **11+1** | | **11+3** | | |
| RIFE [12] | ✔ | ✘ | 32.79 | 0.904 | 31.24 | 0.890 | 9.8M |
| EDI [22] | ✔ | ✔ | 24.48 | 0.735 | 23.53 | 0.715 | 0.5M |
| EFNet+IFRNet [17, 34] | ✔ | ✔ | 35.97 | 0.959 | 35.42 | 0.966 | 28.2M |
| E2VID+ [26] | ✔ | ✔ | 36.36 | 0.970 | 35.77 | 0.968 | 15.3M |
| **REFID (Ours)** | ✔ | ✔ | **37.65** | **0.975** | **36.91** | **0.973** | 15.9M |

**Blurry frame interpolation.** We synthesize blurry frames by averaging 11 sharp high-FPS frames in GoPro and HighREV. Between each blurry frame, we skip 1 or 3 frames for the evaluation of blurry frame interpolation (denoted as "11+1" or "11+3" in Tab. 1).

**Sharp frame interpolation.** The high-frame-rate sharp images of GoPro and HighREV are leveraged by skipping 7 or 15 frames and keeping the next one.

**Image deblurring.** We use GoPro with synthesized blurry images (averaged from 7 or 11 sharp frames). For a real-world test, we also fine-tune and evaluate methods on REBlur [34]. We only use a single image and its corresponding events in the event branch as input, for a fair comparison.

For blurry frame interpolation and sharp frame interpolation, we train all the models on each training set and evaluate on the respective test set.

### 5.2. Implementation Details

Different from warping-based methods [36, 37], REFID is an end-to-end network. All its components are optimized from scratch in a single training round, without any pre-trained modules, which makes it train easier. We crop the input images and event voxels to $256 \times 256$ for training and use horizontal and vertical flips, random noise and hot pixels in event voxels [33]. Adam [16] with an initial learning rate of $2 \times 10^{-4}$ and a cosine learning rate annealing strategy with $2 \times 10^{-4}$ as minimum learning rate are adopted for optimization. We train the model on GoPro with a batch size of 1 for 200k iterations on 4 NVIDIA Titan RTX GPUs. For experiments on HighREV, we fine-tune the model trained on GoPro with an initial learning rate of $1 \times 10^{-4}$ for 10k iterations. For image deblurring on REBlur, fine-tuning takes 600 iterations with an initial learning rate of $2 \times 10^{-5}$.

### 5.3. Blurry Frame Interpolation

We compare our method with state-of-the-art image-only and event-based methods. Since most event-based methods do not have public implementations, we use "E2VID+" by adding an extra encoder for images and introduce images as extra inputs for the event-based image reconstruction method E2VID [25]. As a two-stage method, we use EFNet+IFRNet by combining a state-of-the-art event-based image deblurring method [34] with an image-only frame interpolation method [17]. For a fair comparison, IFRNet is also fed with event voxels from two directions as inputs. For Time Lens [37], because the training code is not available, we use the public model and pre-trained weights.

Quantitative results are reported in Tab. 1. Although our method can also interpolate latent frames in the exposure time, the results are reported on the interpolated frame between the two exposure times. REFID achieves 2.08 dB/0.012 and 1.29 dB/0.005 improvement in PSNR and SSIM on the "11+1" setting on GoPro and HighREV, respectively. For the "11+3" setting, the improvements over the second-best method amount to 2.08 dB/0.017 and 1.14 dB/0.005, showing that our principled bidirectional architecture with event-guided fusion leverages events more effectively. The state-of-the-art event-based method Time Lens exhibits a large performance degradation on blurry frame interpolation because of the assumption of sharp key frames and neglecting the intensity changes within the exposure time. Fig. 5 shows qualitative results. Fig. 5 (a) depicts the results for the left, right and interpolated frame on HighREV. EDI [22] is vulnerable to noise and inaccurate events. E2VID+ exhibits artifacts because its unidirectional architecture does not leverage future events. REFID achieves sharp and faithful results both on textured regions and edges thanks to the bidirectional architecture and its event-guided attention fusion. Fig.5 (b) shows the results of frame interpolation within the exposure time.

### 5.4. Sharp Frame Interpolation

We report sharp frame interpolation results in Tab. 2. Our method achieves state-of-the-art performance in the 7- and

Table 2. **Comparison of sharp frame interpolation methods on GoPro [19] and HighREV.** Read as Tab. 1.

| Method | Frames | Events | PSNR ↑ | SSIM ↑ | PSNR ↑ | SSIM ↑ | #Param |
|---|---|---|---|---|---|---|---|
| **GoPro (interpolation)** [19] | | | **7 frames skip** | | **15 frames skip** | | |
| DAIN [2] | ✔ | ✘ | 28.81 | 0.876 | 24.39 | 0.736 | 24.0M |
| SuperSloMo [13] | ✔ | ✘ | 28.98 | 0.875 | 24.38 | 0.747 | 19.8M |
| IFRNet [17] | ✔ | ✘ | 29.84 | 0.920 | - | - | 19.7M |
| EDI [22] | ✔ | ✔ | 18.79 | 0.670 | 17.45 | 0.603 | 0.5M |
| TimeReplayer [9] | ✔ | ✔ | 34.02 | 0.960 | - | - | - |
| Time Lens [37] | ✔ | ✔ | 34.81 | 0.959 | 33.21 | 0.942 | - |
| REFID (Ours) | ✔ | ✔ | **36.80** | **0.980** | **35.635** | **0.974** | 15.9M |
| **HighREV (interpolation)** | | | **7 frames skip** | | **15 frames skip** | | |
| EDI [22] | ✔ | ✔ | 22.32 | 0.716 | 18.65 | 0.654 | 0.5M |
| RIFE [12] | ✔ | ✘ | 32.28 | 0.904 | 28.22 | 0.864 | 9.8M |
| Time Lens [37] | ✔ | ✔ | 32.81 | 0.901 | 27.06 | 0.810 | - |
| REFID (Ours) | ✔ | ✔ | **38.38** | **0.977** | **37.58** | **0.975** | 15.9M |

Table 3. **Comparison of single image motion deblurring methods on GoPro [19] and REBlur [34].** HINet+: event-enhanced versions of HINet [6].

| Method | Events | PSNR ↑ | SSIM ↑ | #Param |
|---|---|---|---|---|
| **GoPro** [19] | | | | |
| D$^2$Nets$^\dagger$ [29] | ✔ | 31.60 | 0.940 | - |
| LEMD$^\dagger$ [14] | ✔ | 31.79 | 0.949 | - |
| MPRNet [43] | ✘ | 32.66 | 0.959 | 20.0M |
| Restormer [42] | ✘ | 32.92 | 0.961 | 26.1M |
| ERDNet [4] | ✔ | 32.99 | 0.935 | - |
| NAFNet [5] | ✘ | 33.69 | 0.967 | - |
| EFNet [34] | ✔ | 35.46 | 0.972 | 8.5M |
| **REFID (Ours)** | ✔ | **35.91** | **0.973** | 15.9M |
| **REBlur [34]** | | | | |
| SRN [35] | ✘ | 35.10 | 0.961 | 10.3M |
| NAFNet [5] | ✘ | 35.48 | 0.962 | 67.9M |
| Restormer [42] | ✘ | 35.50 | 0.959 | 26.1M |
| EDI [22] | ✔ | 36.52 | 0.964 | 0.5M |
| HINet+ [6] | ✔ | 37.68 | 0.973 | 88.9M |
| EFNet [34] | ✔ | 38.12 | **0.975** | 8.5M |
| **REFID (Ours)** | ✔ | **38.34** | **0.975** | 15.9M |

Table 4. **Ablation study of different architectural components of our method** on the GoPro [19] dataset using the "11+1" setting.

| Multi-scale connection | Fusion | Recurrent | PSNR | SSIM |
|---|---|---|---|---|
| ✘ | add | ✘ | 33.24 | 0.950 |
| ✔ | add | ✘ | 33.61 | 0.952 |
| ✔ | add | ConvLSTM | 34.39 | 0.962 |
| ✔ | add | ConvGRU | 34.54 | 0.962 |
| ✔ | add | EVR unidir. | 35.36 | 0.968 |
| ✔ | add | EVR bidir. | 35.81 | 0.971 |
| ✔ | EGACA | EVR bidir. | **36.12** | **0.974** |

15-skip setting on both examined datasets, improving upon competing methods substantially. Fig. 6 shows qualitative results on HighREV. RIFE shows artifacts because of the ambiguity of motion in the time between the two images. Our method exhibits stable performance both on indoor and outdoor scenes.

### 5.5. Single Image Deblurring

As a by-product, REFID can also perform single image motion deblurring, and Tab. 3 reports quantitative comparisons on this task. Compared with the state-of-the-art EFNet [34], our method pushes the performance further to 35.91 dB in PSNR on GoPro. The 0.22 dB improvement in PSNR over EFNet on REBlur also evidences the robustness of REFID on real-world blurry scenes.

### 5.6. Ablation Study

Ablation studies are conducted on GoPro with the "11+1" setting to analyze the effectiveness of the proposed model architecture and its components (Tab. 4). First, the proposed recurrent architecture improves PSNR by 1.75 dB compared to the non-recurrent architecture, proving the effectiveness of temporal modeling of events. Furthermore, the proposed bidirectional EVR block yields an improvement of 0.45 dB in PSNR compared to its unidirectional counterpart, showcasing the informativeness of future events and the merit of our physically-based model design. Compared to ConvLSTM [31] and ConvGRU [32], which model longer time dependencies and are used in video recognition, our EVR block using a simple ResNet [8] yields 0.84 dB improvement in PSNR. Moreover, the proposed EGACA contributes an improvement of 0.31 dB, evidencing the benefit of mining and fusing image features with adaptive weights from current events. The multi-scale connection between the image branch and the event branch also brings a 0.37 dB gain in PSNR. Finally, all our contributions together yield a substantial improvement of 2.88 dB in PSNR and 0.024 in SSIM over the baseline.

## 6. Conclusion

In this paper, we have considered the tasks of event-based sharp frame interpolation and blurry frame interpolation jointly, as motion blur may or may not occur in input videos depending on the speed of the motion and the length of the exposure time. To solve these tasks with a single method, we have proposed REFID, a novel bidirectional recurrent neural network which performs fusion of the reference video frames and the corresponding event stream. The recurrent structure of REFID allows the effective propaga-

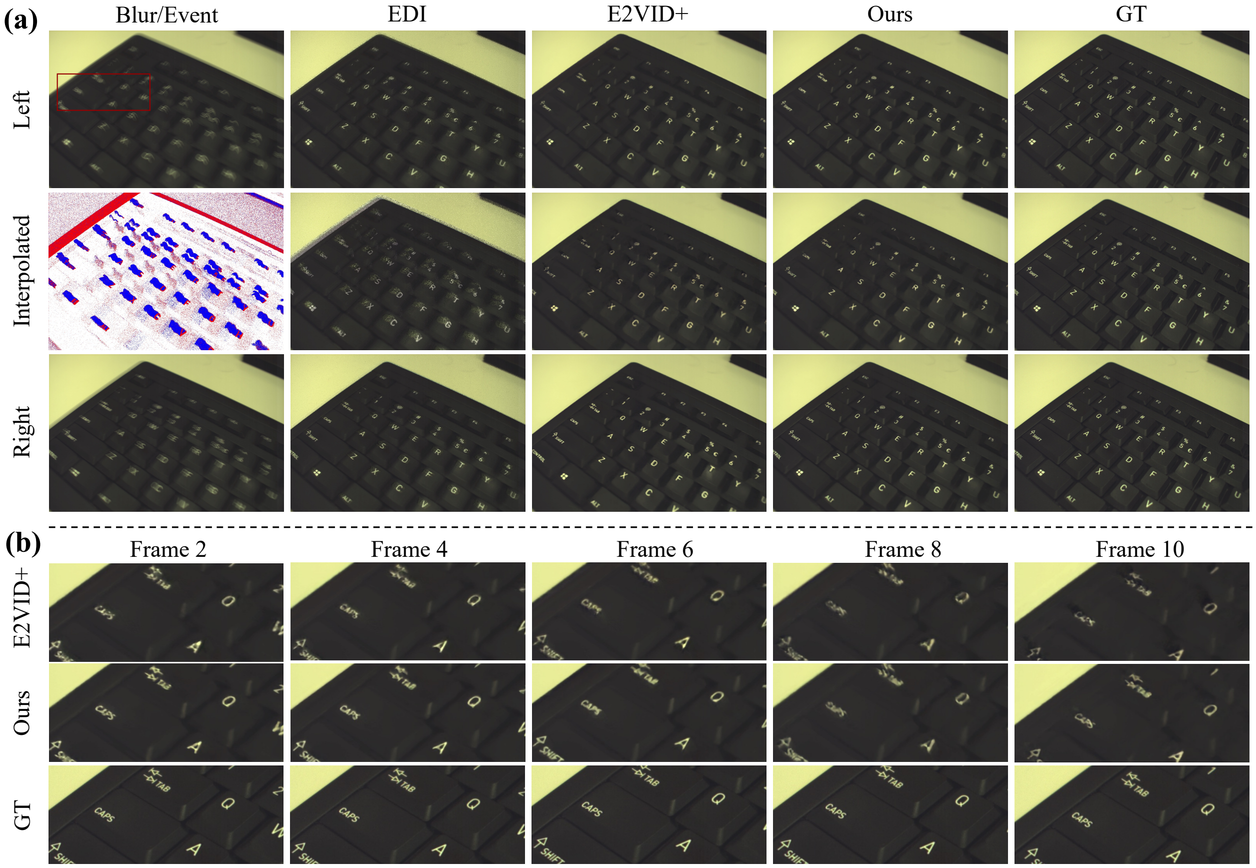

Figure 5. **(a): Visual comparison on HighREV of the restored left, right and interpolated frame.** E2VID+: image-enhanced version of E2VID [25]. Compared to other event-based methods, our method achieves the most faithful results. **(b): The interpolated frames in the exposure time of the left blurry image.** Best viewed on a screen and zoomed in.

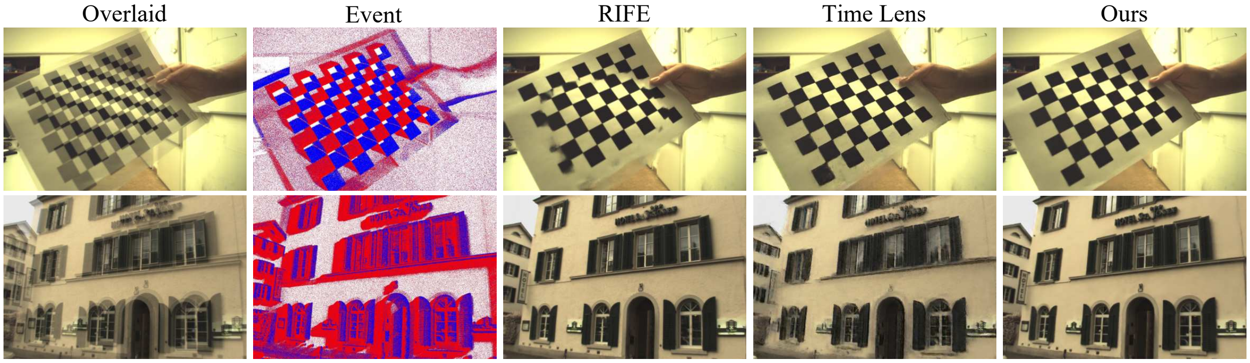

Figure 6. **Qualitative results for sharp frame interpolation on HighREV.** RIFE [12] suffers from motion ambiguity because of the lack of event information. Time Lens [37] is vulnerable to noise. Our REFID shows superior performance both on indoor and outdoor scenes.

tion of event-based information across time, which is crucial for accurate interpolation. Moreover, we have introduced EGACA, a new adaptive event-image fusion module based on channel attention. In order to provide a more realistic experimental setting for the examined low-level event-based tasks, we have presented HighREV, a new event-RGB dataset with the highest spatial event resolution among related sets. We have thoroughly evaluated our network on standard event-based sharp frame interpolation, event-based blurry frame interpolation, and single-image deblurring and shown that it consistently outperforms existing state-of-the-art methods on GoPro and HighREV.

**Acknowledgments.** This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No.12174341, the National Key R&D Program of China under Grant No.2022YFF0705500 and No.2022YFB3206000, the China Scholarship Council, and AlpsenTek GmbH.

## References

[1] Robert Anderson, David Gallup, Jonathan T Barron, Janne Kontkanen, Noah Snavely, Carlos Hernández, Sameer Agarwal, and Steven M Seitz. Jump: virtual reality video. *ACM Transactions on Graphics (TOG)*, 35(6):1–13, 2016. 1

[2] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3703–3712, 2019. 7

[3] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240 × 180 130 dB 3 $\mu$s latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 2014. 1

[4] Haoyu Chen, Minggui Teng, Boxin Shi, Yizhou Wang, and Tiejun Huang. Learning to deblur and generate high frame rate video with an event camera. *arXiv preprint arXiv:2003.00847*, 2020. 1, 3, 7

[5] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *arXiv preprint arXiv:2204.04676*, 2022. 7

[6] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. HINet: Half instance normalization network for image restoration. In *CVPRW*, 2021. 7

[7] 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*, 2022. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[9] Weihua He, Kaichao You, Zhendong Qiao, Xu Jia, Ziyang Zhang, Wenhui Wang, Huchuan Lu, Yaoyuan Wang, and Jianxing Liao. Timereplayer: Unlocking the potential of event cameras for video interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17804–17813, 2022. 1, 2, 3, 4, 7

[10] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2, 5

[11] Ping Hu, Simon Niklaus, Stan Sclaroff, and Kate Saenko. Many-to-many splatting for efficient video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3553–3562, 2022. 1

[12] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. RIFE: Real-time intermediate flow estimation for video frame interpolation. *arXiv preprint arXiv:2011.06294*, 2020. 2, 6, 7, 8

[13] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9000–9008, 2018. 1, 7

[14] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *CVPR*, 2020. 2, 3, 5, 7

[15] Taewoo Kim, Jungmin Lee, Lin Wang, and Kuk-Jin Yoon. Event-guided deblurring of unknown exposure time videos. *arXiv preprint arXiv:2112.06988*, 2021. 2, 5

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[17] Lingtong Kong, Boyuan Jiang, Donghao Luo, Wenqing Chu, Xiaoming Huang, Ying Tai, Chengjie Wang, and Jie Yang. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 6, 7

[18] 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 *ECCV*, 2020. 1, 3

[19] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 2, 5, 6, 7

[20] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020. 1

[21] Jihyong Oh and Munchurl Kim. Demfi: Deep joint deblurring and multi-frame interpolation with flow-guided attentive correlation and recursive boosting. *arXiv preprint arXiv:2111.09985*, 2021. 2

[22] 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 *CVPR*, 2019. 1, 2, 3, 6, 7

[23] Federico Paredes-Vallés and Guido CHE de Croon. Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3446–3455, 2021. 2

[24] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. ESIM: an open event camera simulator. In *CoLR*, 2018. 5

[25] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019. 4, 6, 8

[26] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *CVPR*, 2019. 4, 6

[27] Wenqi Ren, Jingang Zhang, Xiangyu Xu, Lin Ma, Xiaochun Cao, Gaofeng Meng, and Wei Liu. Deep video dehazing with semantic segmentation. *IEEE transactions on image processing*, 28(4):1895–1908, 2018. 1

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 4

[29] 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 *ICCV*, 2021. 7

[30] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. Blurry video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5114–5123, 2020. 2

[31] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015. 5, 7

[32] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. *Advances in neural information processing systems*, 30, 2017. 5, 7

[33] Timo Stoffregen, Cedric Scheerlinck, Davide Scaramuzza, Tom Drummond, Nick Barnes, Lindsay Kleeman, and Robert Mahony. Reducing the sim-to-real gap for event cameras. In *ECCV*, 2020. 6

[34] Lei Sun, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, and Luc Van Gool. Event-based fusion for motion deblurring with cross-modal attention. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 5, 6, 7

[35] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018. 7

[36] Stepan Tulyakov, Alfredo Bochicchio, Daniel Gehrig, Stamatios Georgoulis, Yuanyou Li, and Davide Scaramuzza. Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17755–17764, 2022. 1, 2, 3, 4, 6

[37] 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*, pages 16155–16164, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[38] Patricia Vitoria, Stamatios Georgoulis, Stepan Tulyakov, Alfredo Bochicchio, Julius Erbach, and Yuanyou Li. Event-based image deblurring with dynamic motion awareness. *arXiv preprint arXiv:2208.11398*, 2022. 5

[39] Lin Wang, S. Mohammad Mostafavi I., Yo-Sung Ho, and Kuk-Jin Yoon. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *CVPR*, 2019. 2

[40] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 416–431, 2018. 1

[41] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[42] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022. 7

[43] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021. 7

[44] 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*, pages 17765–17774, 2022. 2, 3, 6

[45] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)*, 23(3):600–608, 2004. 1

[46] Liad Pollak Zuckerman, Eyal Naor, George Pisha, Shai Bagon, and Michal Irani. Across scales and across dimensions: Temporal super-resolution using deep internal learning. In *European Conference on Computer Vision*, pages 52–68. Springer, 2020. 2