Training Weakly Supervised Video Frame Interpolation with Events

Zhiyang Yu[†] [1, 2], Yu Zhang[*] [2, 3], Deyuan Liu[†2, 5], Dongqing Zou[2, 4],
Xijun Chen[*1], Yebin Liu[3], and Jimmy Ren[2, 4]

[1]Harbin Institute of Technology, [2]SenseTime Research and Tetras.AI, [3]Tsinghua University
[4]Qing Yuan Research Institute, Shanghai Jiao Tong University, [5]Peking University

Abstract

Event-based video frame interpolation is promising as event cameras capture dense motion signals that can greatly facilitate motion-aware synthesis. However, training existing frameworks for this task requires high frame-rate videos with synchronized events, posing challenges to collect real training data. In this work we show event-based frame interpolation can be trained without the need of high frame-rate videos. This is achieved via a novel weakly supervised framework that 1) corrects image appearance by extracting complementary information from events and 2) supplants motion dynamics modeling with attention mechanisms. For the latter we propose subpixel attention learning, which supports searching high-resolution correspondence efficiently on low-resolution feature grid. Though trained on low frame-rate videos, our framework outperforms existing models trained with full high frame-rate videos (and events) on both GoPro dataset and a new real event-based dataset. Codes, models and dataset will be made available at: https://github.com/YU-Zhiyang/WEVI.

1. Introduction

Modern dedicated cameras are now capable of capturing high frame rate videos (*e.g.* 240 FPS for Sony GoPro series), allowing users to create professional slow motion effect. However, most prevailing devices, like smartphones, still cannot compete with them before overcoming various challenges on hardware and software designing. It is thus desired to develop computational techniques to synthesize high temporal resolution videos from lower resolution ones.

Foremost among the challenges of video interpolation is the loss of motion caused by the insufficient temporal sampling rate of the input video. Many previous works “hallucinate” the missing motion by assuming a parameterized motion model (*e.g.* linear or quadratic flows [14, 48], phase models [25, 24]) or data-driven models [13, 7, 34, 18, 32]. However, despite the rapid advances on end-to-end learning video interpolation, the task is inherently ill-posed, with large ambiguity that cannot be trivially addressed from only the sparse set of input frames.

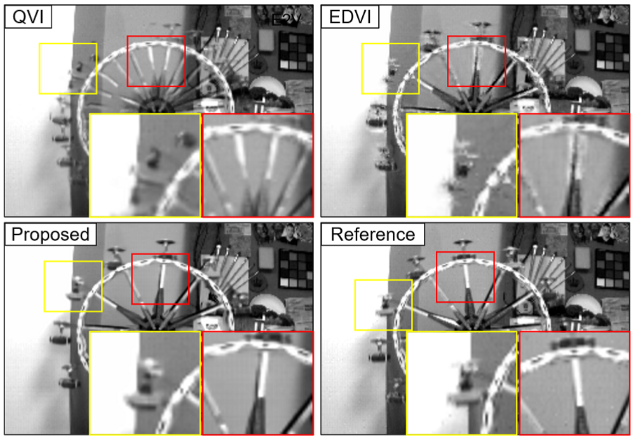


Figure 1. Motivation of this work. For interpolating challenging real-world videos, even the state-of-the-art Quadratic Video Interpolation (QVI) [48] fails to inferring correct motion. The event-based method (EDVI) [21] generates better but still sub-optimal reconstructions due to gap between training and testing. Capable of being trained directly on the raw low frame-rate videos, our approach possesses best generalization behavior. Due to lack of ground truth inbetweens, an input frame is shown as reference.

Characteristically for this age, event-based sensors [20] start to play roles in solving ill-posed low-level tasks such as deblurring [15, 33] and frame interpolation [47, 21]. Event cameras capture per-pixel change of intensities at high temporal resolution and limited power cost, making them ideal supplement to low frame-rate image sensors with the capability to capture dense motion signals [2]. Despite its potential, event signals have distinct mode discrepancy when working with video frames. In recent works [15, 47, 21], it is largely addressed with modern deep networks by translating events to image-space representations at dense temporal sites. Nevertheless, collecting synchronized training events and high frame-rate videos requires complicated hardware

[†] The work is done during an internship at SenseTime Research.
[*] Corresponding authors: Yu Zhang (zhangyulb@gmail.com) and Xijun Chen (chenxijun@hit.edu.cn).

calibration of dedicated cameras; this is why recent methods [15, 47, 21] mostly adopt synthesized training data.

In this work, we propose a weakly supervised framework for video frame interpolation that bypasses the need of high frame-rate training videos with events. Instead of synthetic training, our framework is dedicatedly designed to be trained on low frame-rate videos with event streams, improving generalization on real data (see Fig. 1 for example). For interpolation at intermediate time instants, we first warp input frames with coarse motion models. Such generated immediate reconstructions are then corrected by fusing complementary appearance cues extracted from events at multiple scales. We further leverage temporal context to improve the first-stage estimation, with a lightweight transformer architecture [45, 50]. This supplants the need of densely modeling motion dynamics, which is difficult in case of low frame-rate training, with attention mechanisms. We develop novel attention modules learning subpixel offsets from low-resolution feature grid to efficiently extract accurate motion correspondences without the cost on processing high-resolution features. Though with low frame-rate training, the proposed framework surpasses the state-of-the-art image-based and event-based models trained with full high frame-rate videos, on both the GroPro dataset [28] and a new dataset captured by DAVIS240C camera [1].

In summary, the contributions of this paper include: 1) A novel framework for weakly supervised video interpolation with events, which surpasses state-of-the-art fully supervised models and shows better generalization; 2) Complementary appearance fusion that adaptively aggregates image and event appearance at multiple scales; 3) Subpixel attention mechanism that supports high-resolution correspondence learning on the low-resolution grid; Finally, 4) a new real event dataset and benchmarking results on it to facilitate future research on event-based frame interpolation.

2. Related Work

Video frame interpolation is typically solved by inferring plausible immediate motion from sparse input frames. Much of the recent research concentrates on inferring one single immediate frame [30, 23, 24, 29, 4, 9]. Theoretically, applying single frame interpolation recursively can reach to any desired frame-rate, yet is inefficient and suffers the risk of accumulating errors [14]. In contrast, dense video interpolation requires continuous motion representations. It could be achieved by computing optical flows and fitting linear [14, 30, 34], quadratic [48] or cubic [7] trajectory models. Another crucial topic is to fix occlusions caused by the variation of scene geometry. Various proposals emerge including the Gaussian resampling of flows [48], soft splatting [30], flow refinement [22], contextual feature incorporation [29]. There were also motion representations learned in data-driven manner, *e.g*. voxel flows [23], deep phase model [24], feature flows [11], pixel-varying kernels [31, 32, 18, 41], task-specific flows [49, 16]. Depth, semantic and scene-adaptive cues were explored to boost accuracy [3, 51, 8]. In [39], cycle consistency is explored as free self-supervision to alleviate the need of high frame-rate training videos. However, above methods all address an ill-posed setting of frame interpolation that does not observe intermediate motion among input frames.

Event-based sensors [20] capture temporally dense signals that represent the change of local pixel intensities at microsecond level. It gives the opportunity to counter the ill-poseness of video interpolation by supplying low frame-rate image sensors with a synchronized event camera, which has already industrial models [1, 2]. Solutions were proposed for event-based deblurring [33, 47, 15] and frame interpolation [21]. In these methods, events contribute to final image results by modeling the physical relations between images and events, while deep networks were explored in [47, 15, 21] to learn data-driven reconstructions. To train such networks, high frame-rate videos and synchronized events are required, which are difficult to collect in practice and largely bypassed with synthetic data.

What worthies to mention is that events themselves could reconstruct videos without the need of images [27, 40, 38, 52]. However, such contrast-based reconstructions do not look naturally. Similar with [15, 21], we are interested in synthesizing natural look videos likely to be produced by a high frame-rate image sensor, using events as guidance.

3. Approach

3.1. Overview

Given consecutive video frames $\mathcal{I}_0$ and $\mathcal{I}_1$, we are interested in interpolating any intermediate frame $\mathcal{I}_t$ where $t \in (0, 1)$ is a normalized fractional time instant. Following [47, 15, 21], we assume the availability of dense spatiotemporal events simultaneously captured for the same input scene. For a frame $\mathcal{I}_t$ at time t, it gives a set of events $\mathbb{E}_t$ incurred at a local time window. We propose a two-stage framework that supports training on triplets of consecutive, temporally sparse frames $\mathcal{I}_0$, $\mathcal{I}_1$, $\mathcal{I}_2$, but applies to arbitrary time instant during inference. As shown in in Fig. 2. It consists of a Complementary Appearance Fusion (CAF) network and a Subpixel Motion Transformer (SMT).

In CAF, we first warp $\mathcal{I}_0$ and $\mathcal{I}_2$ to the middle frame with optical flows, yielding coarsely aligned reconstructions with potential errors around where flow estimation is unreliable. CAF corrects such errors by exploring complementary cues from events $\mathbb{E}_1$. To this end, a two-branch UNet separately consumes the images and events, fuses their decoder outputs at multiple scales with a Adaptive Appearance Fusion Blocks (AAFB), and outputs the refined interpolation result $\hat{\mathcal{I}}_1$. Unlike previous works [14, 48, 7] that *corrects interme-*

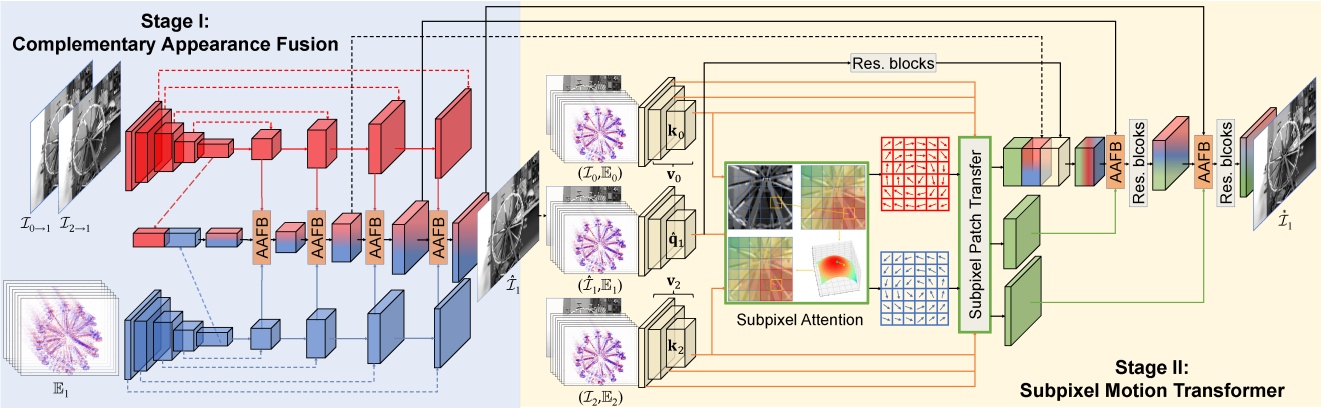


Figure 2. Pipeline of our framework, consisting of two stages: complementary appearance fusion and subpixel motion transfer. See Sect. 3.1 for elaborations. Due the limit of space, we refer the detailed layer configurations to the supplementary material. Best viewed with color.

diate motion, we cast CAF as *correcting intermediate appearance* with events. It ensures generalization on unseen time instant by eliminating the need of motion modeling.

To further explore motion context, the second stage of our framework is implemented as a transformer [45, 50, 6]. It treats $(\hat{\mathcal{I}}_1, \mathbb{E}_1)$ as the query and surrounding observations $(\mathcal{I}_0, \mathbb{E}_0)$, $(\mathcal{I}_2, \mathbb{E}_2)$ as support memory. A subpixel attention module finds accurate correspondence among the query and memory, by which the relevant information from the memory is retrieved and aggregated with subpixel patch transfer, resuting into multi-scale context features. As a final step, we fuse these features and the decoded features produced by the first stage with AAFB and residual blocks, to produce the final refined interpolation result. SMT leverages motion context with attention mechanisms, in contrast with previous works [15, 21] that explicitly model motion dynamics evolved along dense temporal sites. Doing so largely closes the gap between training and testing, with no need of unrealistic high frame-rate video synthesis at training stage.

3.2. Complementary Appearance Fusion (CAF)

To create the input of CAF network, we compute forward optical flows [43] from $\mathcal{I}_0$ or $\mathcal{I}_2$ to $\mathcal{I}_1$, by which input images are warped with *forward rendering* [44] to achieve $\mathcal{I}_{0\to1}$ and $\mathcal{I}_{2\to1}$. Using forward instead of backward warping, we eliminate the need for occlusion filling while leaving it for later processing. Alongside with warped input images are stacked frame representations [46] (details explained in Sect. 3.4) of events $\mathbb{E}_1$. As illustrated in Fig. 2, CAF is a two-branch UNet, each of whose branches handles a particular modality. To effectively arrange cross-modality information, we adaptively fuse features of images and events at multiple levels among the decoder outputs.

Multi-scale adaptive fusion. Our fusion module is inspired from recent high-fidelity image synthesis that gradually modulates immediate features with transferred statistics in coarse-to-fine manner [35, 17, 19]. The fused features at the sth scale, denoted with $\mathbf{x}^s$, are recursively produced as:

$$\mathbf{x}^s = g\left(\mathbf{x}_{\uparrow}^{s-1}; \mathbf{f}^s, \mathbf{e}^s\right), \ s \in \{1,2,3,4,5\}, \qquad (1)$$

where $\mathbf{x}_{\uparrow}^{s-1}$ denotes the 2x upsampled version of $\mathbf{x}^{s-1}$ to match resolution, $\mathbf{f}^s$ and $\mathbf{e}^s$ are decoder outputs at the sth scale of image and event branches, respectively. For initialization, $\mathbf{x}^0$ is obtained by concatenating the deepest encoder outputs of both branches followed by 1×1 convolution.

To effectively modulate $\mathbf{x}^s$ with image and event features at the current scale, we regard $\mathbf{f}^s$, $\mathbf{e}^s$ as two different views of the underlying reconstructions. We follow learned feature renormalization [12, 19] that aligns the feature distributions of different views while preserving fine-grained spatial details. For either $\mathbf{f}^s$ or $\mathbf{e}^s$, we process them with separate convolution layers to learn spatial-varying pixel-wise scalings and biases $\mathbf{s}^{\mathbf{f}}$ and $\mathbf{b}^{\mathbf{f}}$, or $\mathbf{s}^{\mathbf{e}}$ and $\mathbf{b}^{\mathbf{e}}$. We transfer these statistics to the fused features as follows, *i.e.*

$$\mathbf{y}^{\mathbf{e}} = \left(\frac{\mathbf{x}_{\uparrow}^s - \mu(\mathbf{x}_{\uparrow}^s)}{\sigma(\mathbf{x}_{\uparrow}^s)}\right) \odot \mathbf{s}^{\mathbf{e}} + \mathbf{b}^{\mathbf{e}}, \qquad (2)$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are statistical means and standard deviations of $\mathbf{x}_{\uparrow}^s$ computed on spatial dimensions, operator $\odot$ denotes Hadamard product. With doing so, $\mathbf{y}^{\mathbf{e}}$ overwrites $\mathbf{x}_{\uparrow}^s$ with event-induced information. We can obtain $\mathbf{y}^{\mathbf{f}}$ analogously, by substituting $\mathbf{s}^{\mathbf{f}}$ and $\mathbf{b}^{\mathbf{f}}$ into (2).

Generally events are sensitive to physical motion boundaries due to the fast illuminance change, where image optical flows are often less reliable. For textureless area events are less active and reliable than optical flows. Such complementary cues are combined with an adaptive soft mask $\mathbf{m}$ produced from $\mathbf{x}_{\uparrow}^{s-1}$ by a convolution and a sigmoid layer:

$$\mathbf{y} = \mathbf{y}^{\mathbf{e}} \odot \mathbf{m} + \mathbf{y}^{\mathbf{f}}(1 - \mathbf{m}). \qquad (3)$$

Steps (2) and (3) complete a single fusion pass, which are summarized in Fig. 3. We gain non-linearity by stacking 2 fusion passes, interleaved with a 3×3 convolution followed by LeakyReLU non-linearity. All these operations constitute to our Adaptive Appearance Fusion Block (*i.e.* the AAFB in Fig. 2 and the function $g(\cdot)$ in (1)).

3.3. Subpixel Motion Transformer (SMT)

We adopt a lightweight transformer to capture the context cues to improve the estimation of CAF. As shown in Fig. 2, SMT starts by taking as input concatenated image and event representations $(\mathcal{I}, \mathbb{E})$, then feeds them into a shared encoder with three convolutional blocks, yielding 3-scale features $\{\mathbf{v}^s | s \in \{0,1,2\}\}$. The deepest scale and lowest resolution features $\mathbf{v}^2$ are also cloned and denoted with $\mathbf{k}$. For $(\mathcal{I}_0, \mathbb{E}_0)$ or $(\mathcal{I}_2, \mathbb{E}_2)$, the obtained $\mathbf{v}_0^s$ or $\mathbf{v}_2^s$ is named *values*, while $\mathbf{k}_0$ or $\mathbf{k}_2$ is named *keys*. For $(\hat{\mathcal{I}}_1, \mathbb{E}_1)$, the computed $\hat{\mathbf{k}}_1$ is named *query*. Keys, values and query constitute the ingredients of attention modules in a transformer, used frequently for memory retrieval [42, 26].

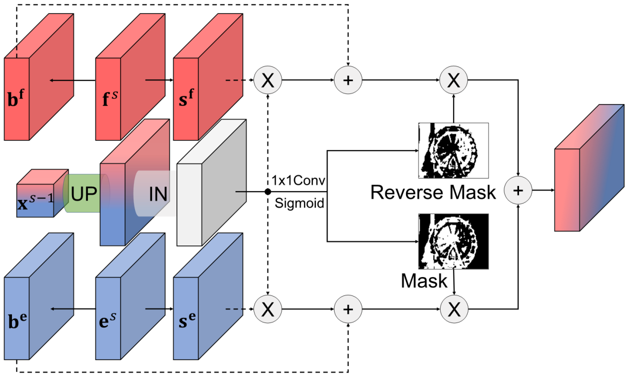
To retrieve memory stored in the values, we search correspondence for each pixel of the query map $\hat{\mathbf{k}}_1$ on both key maps, which we take $\mathbf{k}_0$ as example. Since that we operate with key maps with $\frac{1}{8}$ of input resolution, a limited offset indicates large pixel motion in original image. Thus we restrict correspondence search within a $(2m+1)^2$ local window ($m = 3$) around each pixel. Given pixel site $\mathbf{i}$ on $\hat{\mathbf{k}}_1$ and a spatial offset $\mathbf{p} \in [-m, m]^2$, the relevance is measured as Euclidean distance on ℓ_2 normalized features:

$$\mathbf{D}_0(\mathbf{i}, \mathbf{p}) = \left\| \frac{\hat{\mathbf{k}}_1(\mathbf{i})}{\|\hat{\mathbf{k}}_1(\mathbf{i})\|_2} - \frac{\mathbf{k}_0(\mathbf{i}+\mathbf{p})}{\|\mathbf{k}_0(\mathbf{i}+\mathbf{p})\|_2} \right\|_2^2, \tag{4}$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm. The correlation matrix $\mathbf{D}_0$ can be utilized to aggregate information from memory values $\mathbf{v}_0^s$. Conventional transformer achieves it via *soft coding*, which performs softmax normalization on such correlation matrix and transfers knowledge as the weighted sum of values at all locations. For image synthesis, it may blur immediate features and degenerate final quality. This issue is addressed by [53, 50] with *hard coding*, that computes hard locations of maximal affinity (minimal distance here) and gathers values only at those locations. However, as the offset $\mathbf{p}$ is defined on $\frac{1}{8}$ resolution, even the optimal offsets may not align the higher resolution features in $\{\mathbf{v}^s\}$ well.

Subpixel attention learning. We introduce a solution that computes *subpixel-level* offsets on a low resolution image grid, which indicates improved accuracy when upsampled to high resolution. For a feature pixel $\mathbf{i}$ on the $\hat{\mathbf{k}}_1$, hard attention computation gives us its matched pixel $\mathbf{j}$ on the $\mathbf{k}_0$, *i.e.* $\mathbf{j} = \mathbf{i} + \mathbf{p}^*$ where $\mathbf{p}^* = \arg\min_{\mathbf{p}} \mathbf{D}_0(\mathbf{i}, \mathbf{p})$. In proper manner, the row elements $\{\mathbf{D}_0(\mathbf{i}, \mathbf{p}) | \mathbf{p} \in [-m, m]^2\}$ could be organized into a $(2m+1)^2$ patch of distances, where $\mathbf{p}^*$ corresponds to the index of its minimum.

To reach at subpixel level, we make inductive bias that the local distance field centered around $\mathbf{p}^*$ can be well approximated by continuous representation parametrized by second-order polynomials [36, 10], whose global minimum is achievable in closed-form. By plugging polynomial fitting in learning, it gives the chance to regularize the shape of distance field and provides subpixel-level accuracy. Specifically, we sample a smaller local $(2n+1)^2$ window ($n = 1$ in our implementation) centered at $\mathbf{p}^*$ from the distance patch, denoted with $\mathbf{d}$. We define the local distance field as



$\mathbf{b}^{\mathbf{f}}$
$\mathbf{f}^s$
$\mathbf{s}^{\mathbf{f}}$
X
+
X
$\mathbf{x}^{s-1}$
UP
IN
1x1Conv
Sigmoid
Reverse Mask
+
Mask
$\mathbf{b}^{\mathbf{e}}$
$\mathbf{e}^s$
$\mathbf{s}^{\mathbf{e}}$
X
+
X

Figure 3. One-pass process of the proposed adaptive fusion, see text for details. In this figure, “UP” and “IN” represent 2x bilinear upsampling and instance normalization, respectively.

$$\mathbf{d}(\mathbf{u}) = \mathbf{D}_0(\mathbf{i}, \mathbf{p}^* + \mathbf{u}), \mathbf{u} \in \mathbb{Z}^2 \cap [-n, n]^2. \tag{5}$$

To make this field continuously defined on $[-n, n]^2$, we fit a local quadratic surface as follows:

$$\mathbf{d}(\mathbf{u}) \approx \hat{\mathbf{d}}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{A}\mathbf{u} + \mathbf{b}^{\mathrm{T}}\mathbf{u} + c, \tag{6}$$

where $\mathbf{A}$ is assumed a 2×2 positive definite matrix, $\mathbf{b}$ is a 2×1 vector, and c is a bias constant. These conditions render (6) a valid quadratic surface with global minimum.

To estimate the unknown parameters $\mathbf{A}$, $\mathbf{b}$ and c we use weighted least squares, according to the $(2n+1)^2$ known mappings between $\mathbf{u}$ and $\mathbf{d}(\mathbf{u})$:

$$\min_{\mathbf{A}, \mathbf{b}, c} \sum_{\mathbf{u}} \mathbf{w}(\mathbf{u}) \left\| \hat{\mathbf{d}}(\mathbf{u}) - \mathbf{d}(\mathbf{u}) \right\|^2, \tag{7}$$

where the weights $\mathbf{w}(\mathbf{u})$ can be defined with various ways, *e.g.* a spatial Gaussian $\mathbf{w}(\mathbf{u}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\mathbf{u}^{\mathrm{T}}\mathbf{u}}{2\sigma^2}\right)$.

It can be proved[1] that for constant weights $\mathbf{w}$, the elements of $\mathbf{A}$, $\mathbf{b}$ and c all can be simply estimated via the form $\mathbf{c}^{\mathrm{T}}\mathrm{vec}(\mathbf{d})$, where $\mathbf{c}$ is constant vector depending on the element, $\mathrm{vec}(\cdot)$ denotes vectorization. This makes the polynomial fitting a differentiable layer friendly to be implemented and plugged into the network. However, the estimated $\mathbf{A}$ is not guaranteed positive definite, which we address simply. We assume off-diagonal elements of $\mathbf{A}$ be zeros, only optimize the diagonal ones, and half-rectify them

[1] Please check our supplementary material for detailed derivations.

with $\max(0, \cdot)$ if they are negative. Disgarding off-diagonal parameters makes (6) only capable of fitting isotropic surfaces; however, by integrating subpixel fitting into network training, shape of the distance field (5) could be regularized with backpropagation to remedy this limitation.

The optimal minimum of (6) takes the form

$$\mathbf{u}^* = \left(-\frac{\mathbf{b}^{(0)}}{\mathbf{A}^{(0,0)} + \epsilon}, -\frac{\mathbf{b}^{(1)}}{\mathbf{A}^{(1,1)} + \epsilon} \right)^{\mathrm{T}}, \tag{8}$$

where ϵ is a small constant to avoid dividing by zero. After estimating $\mathbf{u}^*$, we shift the inital matched position by $\mathbf{j}^* \leftarrow \mathbf{j} + \mathbf{u}^*$ to inject the learned subpixel information.

Subpixel patch transfer. Via previous steps we obtain for each pixel $\mathbf{i}$ on $\hat{\mathbf{k}}_1$ a matched subpixel position $\mathbf{j}^*$ on $\mathbf{k}_0$, by which the multiscale values $\{\mathbf{v}_0^s\}$ are warped. Assume the value map of sth scale $\mathbf{v}_0^s$ is t times the size of $\mathbf{k}_0$ in each border. We crop a $t \times t$ patch centered at $\mathbf{j}^*$ on the $\mathbf{v}_0^s$ and address the subpixel patch indices via bilinear interpolation. This yields a $N \times t^2$ tensor $\mathbf{z}_0^s$ after looping over all $\mathbf{i}$s, which is then reshaped to the size of $\mathbf{v}_0^s$ by organizing the patches spatially on the N sites of $\hat{\mathbf{k}}_1$. It can be seen as a subpixel extension of patch swapping on integer lattice [53, 50].

In practice we apply subpixel fitting and patch transfer to both $\mathbf{k}_0$ and $\mathbf{k}_2$, yielding transferred values $\mathbf{z}_0^s$ and $\mathbf{z}_2^s$. We perform hard selection of patches, depending on distances:

$$\mathbf{z}_1^s(\mathbf{i}) = \begin{cases} \mathbf{z}_0^s(\mathbf{i}), \text{ if } \mathbf{D}_0(\mathbf{i}, \mathbf{p}_0^*) < \mathbf{D}_2(\mathbf{i}, \mathbf{p}_2^*), \\ \mathbf{z}_2^s(\mathbf{i}), \text{ otherwise.} \end{cases} \tag{9}$$

It exploits the fact that a pixel on an intermediate frame often finds correspondence from at least one input frame [14].

Cross-stage fusion. As shown in Fig. 2, the retrieved temporal context is incorporated to enhance first stage estimation. Specifically, $\mathbf{z}_1^s$ at the sth scale is first reshaped to the size of $\mathbf{v}_1^s$, yielding multiscale warped context values $\{\tilde{\mathbf{v}}_1^s\}$. The multiscale features fused from the two decoder branches of the first stage CAF are further aggregated with $\{\tilde{\mathbf{v}}_1^s\}$, using another adaptive fusion process enhanced with residual blocks. The fused features at the highest resolution are decoded to produce the refined residuals $\mathcal{R}_1$ of the input $\hat{\mathcal{I}}_1$, giving the result $\mathcal{I}_1^* = \hat{\mathcal{I}}_1 + \mathcal{R}_1$.

3.4. Implementation Details

Event representation. For a time instant t, we quantize the local time window $(t-\tau, t+\tau)$ to 20 bins, where τ is half of the time interval between consecutive frames. Polarities of events falling into each bin are summed pixel-wisely, and clipped to the range $[-10, 10]$ to form a 20-channel tensor $\mathbb{E}_t$. It resembles to the stacked event representations [46].

Architecture. For CAF we build a two-branch UNet with 4 scales, whose encoder of each branch expands the features to $32, 64, 128, 256, 256$ channels with convolution blocks. The first block maintains resolution, while the others sequentially downsample the features by 2x. The decoder is set up symmetrically with skip connections. After multiscale branch fusion, the highest resolution features go through two convolution blocks with 32 output channels to generate final output. For SMT, we directly inherit the same feature extractor of [50] to generate the key and value maps.

Loss functions. We first train CAF to convergence then fix its weights to train SMT. For both stages, the Charbonnier error [5] between the prediction and groundtruth of the middle frame in a training triplet is the only loss function.

Inference. Given the input video frames and an intermediate time instant to interpolate, we locate the nearest two frames and warp them to the target time instant with forward rendering to form the inputs of CAF. To warp intermediate results, we compute optical flows [43] among input frames and fit a quadratic motion model for each pixel, so that intermediate forward flows to the target time could be estimated. Readers are referred to [48] for more details. However, as shall be shown in Sect. 4.3, CAF is robust to the choice of motion models thanks to the guidance of events.

4. Experiments

4.1. Experimental settings

Datasets. We evaluate the proposed framework on two datasets. The GoPro dataset introduced by Nah *et al.* [28] consists of 720p high frame-rate videos with 240FPS. We follow the official dataset split, using 22 videos for training and 11 for testing. The evaluation policies of previous works on GoPro dataset have inconsistency: recent event-based methods [15, 21] adopt 10x interpolation, while many image-based approaches (*e.g.* [14, 48]) adopt 7x. For fairness we unify the evaluations with 10x setting. To this end we sample training sequences with 21 consecutive frames, using the 1th, 11th and 21th frames to form a sparse training triplet to train our approach, while dense frames to train previous works accordingly. In total there are 4304 sequences for training and 1190 for testing. We follow [15] and adopt ESIM simulator [37] to synthesize event streams.

Besides GoPro dataset, we introduce a real dataset captured with DAVIS240C camera [1], named SloMo-DVS. It consists of 60 staged slow-motion videos with synchronized video frames and event streams, covering indoor, outdoor and lab scenes such as standard test charts. To provide quantitative comparisons we create a synthetic 4x interpolation setting, by sampling 9 consecutive frames in which the 1th, 5th and 9th are used to form the training triplet of our approach, and the complete sequence to train fully supervised approaches. In total there are 24500 sequences for training, and 5115 for testing. On this dataset we also evaluate generalization behaviors on real data, through qualitative comparisons on 20 additionally captured videos without down-

Table 1. Comparing models on GoPro dataset, measured in PSNR and SSIM. Bold indicates the top place while underline the second.

Supervision	High FPS videos					High FPS videos + events					Low FPS videos + events	
Methods	SloMo[14]	QVI[48]	DAIN[3]	TAMI†[7]	FLAVR[16]	ETV[38]	SloMo*[14]	QVI*[48]	EMD[15]	EDVI[21]	BHA[33]	Proposed
PSNR	27.79	29.54	27.30	32.91	31.10	32.25	32.79	33.07	29.67	30.90	28.49	**33.33**
SSIM	0.838	0.872	0.836	**0.943**	0.917	0.925	0.940	**0.943**	0.927	0.905	0.920	0.940

† TAMI also adopts external private datasets for training. ∗ Enhanced variants with events added into the inputs of network.

Table 2. Comparing models on SloMo-DVS dataset, measured in PSNR and SSIM. Bold indicates the top place while underline the second.

Supervision	High FPS videos				High FPS videos + events				Low FPS videos + events	
Methods	SloMo[14]	QVI[48]	DAIN[3]	FLAVR[16]	ETV[38]	SloMo*[14]	QVI*[48]	EDVI[21]	BHA[33]	Proposed
PSNR	30.69	30.93	30.38	30.79	32.06	33.46	33.70	33.60	22.95	**34.17**
SSIM	0.915	0.920	0.914	0.917	0.936	0.950	**0.953**	0.948	0.828	0.952

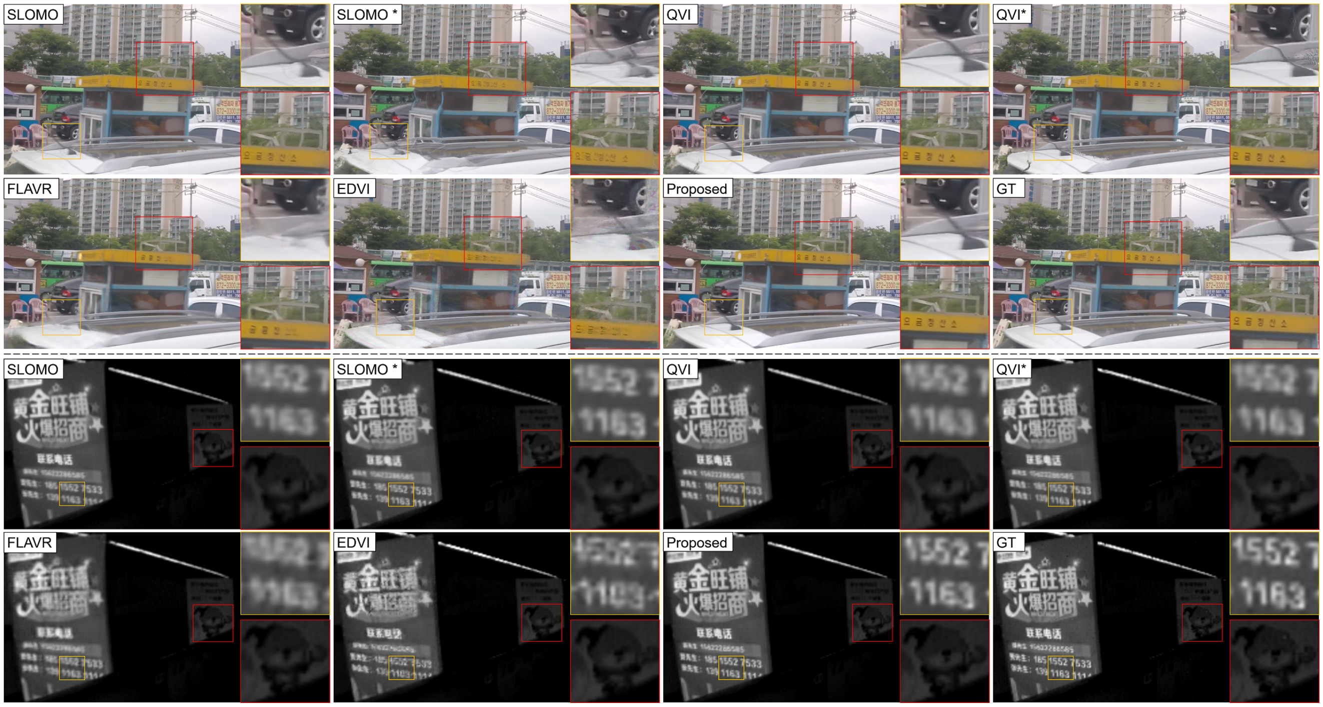


Figure 4. Representative results generated from different approaches on GoPro (top) and Slomo-DVS (bottom) datasets. Best compared in the electronic version of this paper with zoom.

sampling frame rate.

State-of-the-art methods. We report benchmarking results of 9 representative methods for dense video frame interpolation, falling into two groups. The *image-based* group consists of SloMo [14], DAIN [3], QVI [48], FLAVR [16], and TAMI [7], trained by high-frame videos without events. The *event-based* group includes EMD [15], EDVI [21] and ETV [38] trained on high frame-rate videos and events, and the learning-free approach BHA [33]. As the original ETV is purely event based, its reconstructions are not comparable with the lack of dataset-specific appearance. To this end the model is adjusted so that at each of its inference step, the temporally nearest 2 frames in the input video are fed in along with the events, and that finetuned on both datasets.

To evaluate 10x interpolation, we retrain SloMo, DAIN and QVI with the released code, use the pretrained model of FLAVR on GoPro and retrain it on SloMo-DVS. For them we guarantee the reproduction of original results, and refer the details to our supplementary material. For TAMI and EMD, we copy the original results due to the unavailability of code/model. EDVI is retrained on both datasets for fair comparisons (we obtain the code from the authors).

Training details. For each stages we train 100 and 600 epochs respectively on GoPro, 200 and 1000 epochs on SloMo-DVS, both with initial learning rate 5e-4, using exponential decay policy. On GoPro dataset we use a batch of 16 images cropped to 640 × 480, while on SloMo-DVS a batch of 128 images without cropping. No data augmentation is performed. Xavier initialization is adopted for all learnable weights. Training is distributed on 16 NVIDIA GTX1080 TI GPUs, taking about 50 GPU hours.

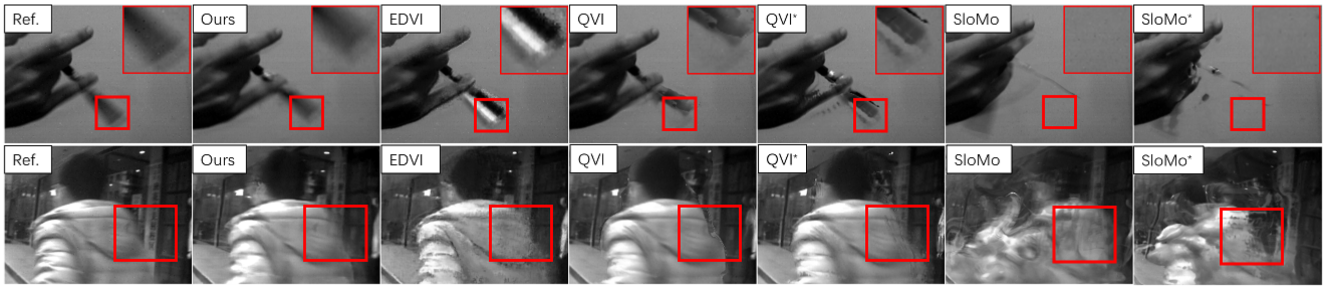


Figure 5. Qualitative comparisons on real data. In the first column (Ref.) we visualize the nearest input frame as reference since there is no groundtruth. We suggest the readers to watch our supplementary video for more qualitative comparisons on real-world video interpolation.

Table 3. Performance in PSNR with low frame-rate training.

Method	SloMo*[14]		QVI*[48]		Proposed
Frame rate	High	Low	High	Low	Low
GoPro	32.79	31.40	33.07	29.88	33.33
SloMo-DVS	33.46	32.76	33.70	31.80	34.17

Table 4. Analysing the performance of CAF network.

Setting	PSNR	SSIM
Replacing AFFB with cat.+conv.	32.27	0.930
Using image branch only	29.43	0.882
Using event branch only	31.37	0.927
Full model	32.47	0.929

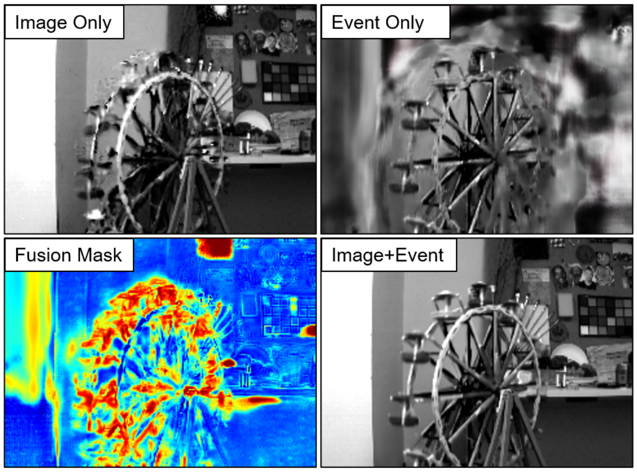


Figure 6. Visualizing the impact of adaptively fusing image and event appearance features in the CAF network.

4.2. Comparisons with State-of-the-Art Models

Benchmarking results. We summarize the results in Table 1 and 2, respectively, in which the proposed framework surpasses all the others in PSNR, while performing comparably with the leading ones in SSIM. To show that the improvement does not fully attribute to the incorporation of events, we train enhanced variants of SloMo and QVI by feeding the same event representations $\mathbb{E}_t$ for interpolation at time instant t. In Fig. 4 we show visual comparisons of different approaches. Our approach recovers correct scene geometry (top), preserves object structures in case of fast motion (top), and restores fine details (bottom).

Comparisons with low frame-rate training. Most previous methods are trained with high-frame rate videos, while our training framework only observes low frame-rate videos. To show the advantage of our framework under low frame-rate training, we retrain two of the top performing methods, SloMo* and QVI*, with the triplets used to train our approach, and report the results in Table 3. It shows a notable performance drop for SloMo*, while interestingly, a significant drop for QVI*. We suspect that since QVI adopts a more powerful and thus flexible motion model, it requires denser video frames for necessary regularization. This experiment illustrates the advantage of our framework that learns motion from low frame-rate videos.

Generalization behavior on real data. The biggest advantage of our approach is that it could be trained on the low frame-rate videos without synthesizing high frame-rate training data, making it generalize better when applied on real-world video interpolation. To illustrate this we test various models on the additional real sequences from SloMo-DVS, improving further their original frame-rate by 4x. We show in Fig. 5 that existing methods trained on synthesized data generate more artifacts particularly on fast motion videos, while our approach does not.

4.3. Performance Analysis

In this section we analyse the proposed framework via a series of experiments, conducted on the GoPro dataset.

Analysing the CAF network. To justify several key designs of the Complementary Appearance Fusion (CAF) network, we report the results under several settings in Table 4. First, we evaluate the effectiveness of the proposed fusion mechanisms by replacing AFFB in Fig. 2 with simple concatenation of features followed by convolution blocks. This clearly makes final results degenerated, demonstrating the advantage of AFFB. Second, we also evaluate contributions of the image and event branch. Eliminating either branch would lead to a loss of performance, showing that image and event cues are complementary to each other.

Visualizing the learned fusion mask. We further illus-

Table 5. Analysing the performance of SMT network.

ID	Key type	Value type	Att. type	Fused stage	PSNR
1	img.+evt.	image	subpix.	both	32.72
2	img.+evt.	event	subpix.	both	32.91
3	image	img.+evt.	subpix.	both	33.01
4	event	img.+evt.	subpix.	both	33.03
5	img.+evt.	img.+evt.	subpix.	first	33.00
6	img.+evt.	img.+evt.	subpix.	second	32.56
7	img.+evt.	img.+evt.	hard	both	33.02
8	img.+evt.	img.+evt.	soft	both	32.50
9	img.+evt.	img.+evt.	subpix.	both	33.33

trate the complementary effect in Fig. 6. Excluding event cues, the foreground windmill is not reconstructed well due to its complicated motion, which poses difficulty to existing motion estimation models. Using event cues only addresses the foreground motion well, yet the background is blurry for the lack of event evidence under static background movement. The fusion mask clearly expresses such adaptiveness, identifying regions to be explained for either modality.

Analysing the SMT network. To justify the key designs of the Subpixel Motion Transformer (SMT) network, we vary several important building blocks and summarize the final results in Table 5. In the first group of experiments (ID 1 $\sim$ 4), we aim to see the contributions of image and event cues in query-key matching and value transfer. We find that isolating any modality in either key or value representations would lead to suboptimal results. In the second group (ID 5 $\sim$ 6), we analyse the contribution of the information extracted from the first and second stages by removing either one from the AAFB fusion. The results show that using only the second stage context information does not achieve good results, demonstrating the effectiveness of the first-stage appearance correction. In the last group (ID 7 $\sim$ 9), subpixel attention is replaced with hard or soft attention. We empirically find that soft attention does not work well as also commented in [50]. Subpixel attention improves over hard attention by roughly 0.3dB, showing the effectiveness of integrating subpixel fitting into learning.

Visualizing patch transfer. In Fig. 7 we visualize the warping results of an input frame to the reference frame with the warping fields learned from different types of attention. Soft attention leads to blurry result, while hard attention generates sharper one but with block artifact due to the large stride of patch positions. Learning subpixel offsets of patches renders more accurate transfer with less artifact.

Robustness to the choice of motion model. Initializing our CAF network requires a heuristic motion model for input frame warping. By default we adopt quadratic model as in [48], yet in Fig. 8 we showcase the results of less accurate models. We estimate intermediate flows with linear models, and alpha blending them with the results estimated from quadratic models. We evaluate the proposed CAF and QVI, and a variant of CAF that excludes events in the input. The PSNR of the full CAF degrades much slower than those of others, showing that event guidance brings robustness to the inaccuracy of motion model.

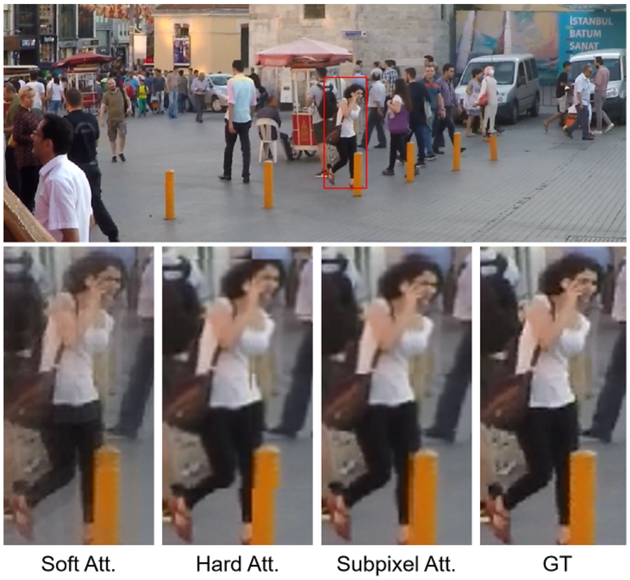


Figure 7. Patch transfer results with different types of attention.

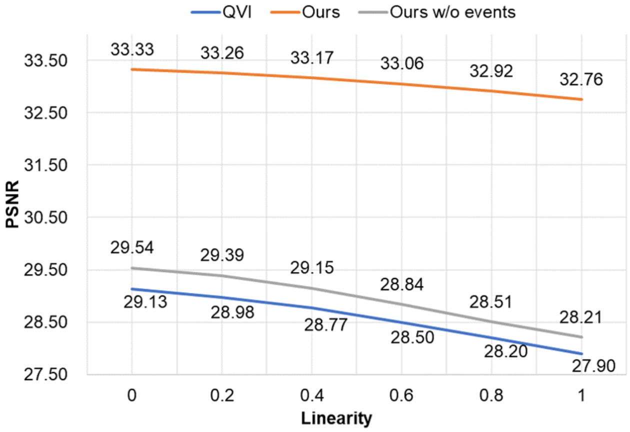


Figure 8. Performance of our approach and QVI as functions of the linearity of the motion model that warps input frames.

5. Conclusion

We propose in this work a novel framework for weakly supervised video interpolation with events. We equip it with complementary appearance fusion blocks and motion transformer with subpixel attention. Trained with low frame-rate videos only, it achieves the state-of-the-art results on two benchmarks and generalizes better to real-world videos.

Via this work we aim to provide a new routine of training event-based vision other than data simulation, through better exploring event cues to“weakly supervised learning” the task objective. It could be further extended to address more tasks, such as deblurring and depth/motion from events.

References

[1] Davis 240 specs. https://inivation.com/wp-content/uploads/2019/08/DAVIS240.pdf. Accessed: 2021-02-24. 2, 5

[2] Sony and prophesee develop a stacked event-based vision sensor with the industry's smallest pixels and highest hdr performance. https://www.sony.net/SonyInfo/News/Press/202002/20-0219E/. Accessed: 2021-02-24. 1, 2

[3] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang. Depth-aware video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3703–3712, 2019. 2, 6

[4] W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(3):933–948, 2021. 2

[5] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *International Conference on Image Processing (ICIP)*, pages 168–172, 1994. 5

[6] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao. Pre-trained image processing transformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[7] Z. Chi, R. M. Nasiri, Z. Liu, J. Lu, J. Tang, and K. N. Plataniotis. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *European Conference on Computer Vision (ECCV)*, volume 12372, pages 107–123, 2020. 1, 2, 6

[8] M. Choi, J. Choi, S. Baik, T. H. Kim, and K. M. Lee. Scene-adaptive video frame interpolation via meta-learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9441–9450, 2020. 2

[9] M. Choi, H. Kim, B. Han, N. Xu, and K. M. Lee. Channel attention is all you need for video frame interpolation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 10663–10671, 2020. 2

[10] G. Farnebäck. *Polynomial expansion for orientation and motion estimation*. PhD thesis, Linköping University, Sweden, 2002. 4

[11] S. Gui, C. Wang, Q. Chen, and D. Tao. Featureflow: Robust video interpolation via structure-to-texture generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14001–14010, 2020. 2

[12] X. Huang and S. J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1510–1519, 2017. 3

[13] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou. RIFE: Real-Time Intermediate Flow Estimation for Video Frame Interpolation. Arxiv preprint, 2011.06294v2 [cs.CV], 2020. 1

[14] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. G. Learned-Miller, and J. Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9000–9008, 2018. 1, 2, 5, 6, 7

[15] Z. Jiang, Y. Zhang, D. Zou, S. J. Ren, J. Lv, and Y. Liu. Learning event-based motion deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3317–3326, 2020. 1, 2, 3, 5, 6

[16] T. Kalluri, D. Pathak, M. Chandraker, and D. Tran. FLAVR: flow-agnostic video representations for fast frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 6

[17] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019. 3

[18] H. Lee, T. Kim, T.-Y. Chung, D. Pak, Y. Ban, and S. Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5315–5324, 2020. 1, 2

[19] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen. Advancing high fidelity identity swapping for forgery detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5073–5082, 2020. 3

[20] P. Lichtsteiner, C. Posch, and T. Delbrück. A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Jounral of Solid State Circuits*, 43(2):566–576, 2008. 1, 2

[21] S. Lin, J. Zhang, J. Pan, Z. Jiang, D. Zou, Y. Wang, J. Chen, and S. J. Ren. Learning event-driven video deblurring and interpolation. In *European Conference on Computer Vision (ECCV)*, volume 12353, pages 695–710, 2020. 1, 2, 3, 5, 6

[22] Y.-L. Liu, Y.-T. Liao, Y.-Y. Lin, and Y.-Y. Chuang. Deep video frame interpolation using cyclic frame generation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 8794–8802, 2019. 2

[23] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. In *IEEE International Conference on Computer Vision, (ICCV)*, pages 4473–4481, 2017. 2

[24] S. Meyer, A. Djelouah, B. McWilliams, A. Sorkine-Hornung, M. H. Gross, and C. Schroers. Phasenet for video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 498–507, 2018. 1, 2

[25] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung. Phase-based frame interpolation for video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1410–1418, 2015. 1

[26] A. H. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston. Key-value memory networks for directly reading documents. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1400–1409, 2016. 4

[27] G. Munda, C. Reinbacher, and T. Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *International Journal on Computer Vision (IJCV)*, 126(12):1381–1393, 2018. 2

[28] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 257–265, 2017. 2, 5

[29] S. Niklaus and F. Liu. Context-aware synthesis for video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1710, 2018. 2

[30] S. Niklaus and F. Liu. Softmax splatting for video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5436–5445, 2020. 2

[31] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive convolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2270–2279, 2017. 2

[32] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. In *IEEE International Conference on Computer Vision (ICCV)*, pages 261–270, 2017. 1, 2

[33] L. Pan, C. Scheerlinck, X. Yu, R. Hartley, M. Liu, and Y. Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6820–6829, 2019. 1, 2, 6

[34] J. Park, K. Ko, C. Lee, and C.-S. Kim. BMBC: bilateral motion estimation with bilateral cost volume for video interpolation. In *European Conference on Computer Vision (ECCV)*, volume 12359, pages 109–125, 2020. 1, 2

[35] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2337–2346, 2019. 3

[36] T. Q. Pham, L. J. van Vliet, and K. Schutte. Robust fusion of irregularly sampled data using adaptive normalized convolution. *Journal of Advanced Signal Processing*, 2006, 2006. 4

[37] H. Rebecq, D. Gehrig, and D. Scaramuzza. ESIM: an open event camera simulator. In *Annual Conference on Robot Learning (CoRL)*, volume 87, pages 969–982, 2018. 5

[38] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3857–3866, 2019. 2, 6

[39] F. A. Reda, D. Sun, A. Dundar, M. Shoeybi, G. Liu, K. J. Shih, A. Tao, J. Kautz, and B. Catanzaro. Unsupervised video interpolation using cycle consistency. In *IEEE International Conference on Computer Vision (ICCV)*, pages 892–900, 2019. 2

[40] C. Scheerlinck, N. Barnes, and Konrad Schindler R. E. Mahony. Continuous-time intensity estimation using event cameras. In *Asian Conference on Computer Vision (ACCV)*, volume 11365, pages 308–324, 2018. 2

[41] Z. Shi, X. Liu, K. Shi, L. Dai, and J. Chen. Video interpolation via generalized deformable convolution. Arxiv preprint, 2008.10680 [cs.CV], 2020. 2

[42] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-to-end memory networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2440–2448, 2015. 4

[43] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018. 3, 5

[44] S. Tulsiani, R. Tucker, and N. Snavely. Layer-structured 3d scene inference via view synthesis. In *European Conference on Computer Vision (ECCV)*, volume 11211, pages 311–327, 2018. 3

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. 2, 3

[46] L. Wang, S. M. M. I., Y.-S. Ho, and K.-J. Yoon. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10081–10090, 2019. 3, 5

[47] Z. W. Wang, W. Jiang, K. He, B. Shi, A. K. Katsaggelos, and O. Cossairt. Event-driven video frame synthesis. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 4320–4329, 2019. 1, 2

[48] X. Xu, S. Li, W. Sun, Q. Yin, and M.-H. Yang. Quadratic video interpolation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1645–1654, 2019. 1, 2, 5, 6, 7, 8

[49] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019. 2

[50] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo. Learning texture transformer network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5790–5799, 2020. 2, 3, 4, 5, 8

[51] L. Yuan, Y. Chen, H. Liu, T. Kong, and J. Shi. Zoom-in-to-check: Boosting video interpolation via instance-level discrimination. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12183–12191, 2019. 2

[52] S. Zhang, Y. Zhang, Z. Jiang, D. Zou, J. S. J. Ren, and B. Zhou. Learning to see in the dark with events. In *European Conference on Computer Vision (ECCV)*, volume 12363, pages 666–682, 2020. 2

[53] Z. Zhang, Z. Wang, Z. L. Lin, and H. Qi. Image super-resolution by neural texture transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7982–7991, 2019. 4, 5