Video Frame Prediction from a Single Image and Events

Juanjuan Zhu* **Zhexiong Wan*** **Yuchao Dai †**

School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China
Shaanxi Key Laboratory of Information Acquisition and Processing
{ juanjuanzhu2022, wanzhexiong }@mail.nwpu.edu.cn daiyuchao@nwpu.edu.cn

Abstract

Recently, the task of Video Frame Prediction (VFP), which predicts future video frames from previous ones through extrapolation, has made remarkable progress. However, the performance of existing VFP methods is still far from satisfactory due to the fixed framerate video used: **1)** they have difficulties in handling *complex dynamic scenes*; **2)** they cannot predict future frames with *flexible prediction time intervals*. The event cameras can record the intensity changes asynchronously with a very high temporal resolution, which provides rich dynamic information about the observed scenes. In this paper, we propose to predict video frames from *a single image and the following events*, which can not only handle *complex dynamic scenes* but also predict future frames with *flexible prediction time intervals*. First, we introduce a symmetrical cross-modal attention augmentation module to enhance the complementary information between images and events. Second, we propose to jointly achieve optical flow estimation and frame generation by combining the motion information of events and the semantic information of the image, then inpainting the holes produced by forward warping to obtain an ideal prediction frame. Based on these, we propose a lightweight pyramidal coarse-to-fine model that can predict a 720P frame within 25 ms. Extensive experiments show that our proposed model significantly outperforms the state-of-the-art frame-based and event-based VFP methods and has the fastest runtime. Code is available at https://npucvr.github.io/VFPSIE.

Introduction

Video frame prediction (VFP) aims to predict future frames from previous frames, which has broad applications in autonomous driving, robotics planning and weather forecasting. Existing VFP methods usually take previous image sequences as input to predict a sequence of future frames with the same framerate as the inputs (simplified diagram in Fig. 1). By exploiting various network architectures, the performance of VFP has been significantly improved. However, due to the limitations of frame-based cameras in capturing complex scenes, it is still challenging by only exploiting the previous frames. In Fig. 4 and Fig. 5, even the state-of-the-art VFP methods, their performances deteriorate quickly as the motion complexity increases. This is mainly due to the fixed framerate inputs, which limit the ability to capture complex dynamics beyond the sampling frequency. Another major issue is that they cannot predict future frames at *flexible time intervals*, *i.e.*, consecutive in-between frames or across multiple frames. These factors limit the performance and real-world applications of existing VFP methods.

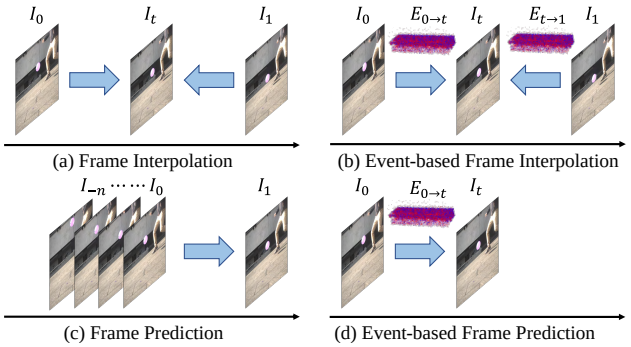

Figure 1: **Differences between VFI, event-based VFI, VFP and event-based VFP.** VFI methods require the image (and events) after the target time t, while image-based VFP is subject to the limited complexity of motion expressed in fixed frame rate inputs. Event-based VFP can solve the above problems, and this paper focuses on the minimal setup of combining a single frame with the following events.

The event camera, as a new kind of bio-inspired sensor, can capture the asynchronous brightness change of each pixel. As the event camera captures high-speed motion with a low data bandwidth and a high temporal resolution with millisecond accuracy, it provides critical and complementary information to images and is widely used in motion estimation (Ding et al. 2022; Wan et al. 2023), object tracking (Wang et al. 2023) and *etc*. In this paper, we investigate a new and minimal setup for event-enhanced video frame prediction, where we predict the next frames from *a single RGB image* (providing contextual information) and *the following events* (providing rich motion information).

Apparently, this task is closely related to video frame interpolation (VFI) with events, where the intermediate frames

*These authors contributed equally.
†Corresponding author.

are interpolated by exploiting two frames and the events in-between. In Fig. 1, subject to the interpolation formulation, these methods require image and event data after time t to generate the frame at time t. This violates the causality constraint in frame generation and limits their applications in any practical systems. With this precedent of successfully introducing events to VFI, we believe that combining events also has a significant effect on improving the usability of VFP in complex dynamic scenarios.

In this paper, we propose a lightweight network that can predict a 720P frame within 25ms on an RTX2080Ti GPU. We first use two respective encoders to extract pyramid features from the input reference image and event representation. To complementarily utilize the characteristics of image and event data, we design a symmetrical cross-modal attention module to augment these two features. Then we refine the synthesized feature and optical flow in a coarse-to-fine joint estimation way. To resolve the holes arising from forward warping, we present an inpainting module that can repair the holes without bringing lots of extra computation. Finally, we adopt a weighted fusion to output the final frame prediction from the synthesized and warped frames. Thanks to the sparse events that can be divided into multiple time segments, the training data we can use covers various motion ranges and time intervals. Therefore, by adjusting the end times of input events, our model can predict high-framerate frames as well as frames for a long time, whereas the frame-based VFP models cannot because their predicted framerates need to be consistent with the input framerates. We conduct experiments on both synthetic and real datasets, and the PSNR is improved by over **3.5dB** on GoPro compared to the state-of-the-art frame-based and event-based VFP methods, which demonstrates the effectiveness of our model in solving the VFP problem.

Our main contributions are summarized as follows:

1) We introduce a minimal practical configuration to introduce events for the VFP tasks, *i.e.*, predicting future frames from a single image and events.
2) We propose a lightweight model with symmetrical cross-attention augmentation and hole inpainting module, which can predict a future frame from a single image and events within real-time requirements.
3) Experiments on both synthetic and real-captured datasets prove the effectiveness and efficiency of our approach in predicting flexible future video frames.

Related Work

Video Frame Prediction

VFP aims to predict future frames from past frames. Existing works have exploited different architectures such as CNN (Liu et al. 2017; Huo et al. 2020; Choi and Baji 2021), RNN (Finn, Goodfellow, and Levine 2016; Fan, Zhu, and Yang 2019; Wang et al. 2022), GAN (Liang et al. 2017; Kwon and Park 2019; Chang et al. 2022) and *etc*. Due to future motion uncertainty, some studies obtain predictions by estimating the distribution of future pixels, optical flow and latent space (Choi and Baji 2021; Liu et al. 2021; Chang et al. 2022). Meanwhile, some studies (Villegas et al. 2017; Gao et al. 2019) decompose the scenes into two parts to build a more accurate motion model. Despite this progress, frame-only methods still cannot handle complicated scenes for lack of motion information. Thus semantic map (Wu et al. 2020; Bei, Yang, and Soatto 2021) and depth map (Qi et al. 2019) resort to incorporating additional data to alleviate the difficulty. The event-enhanced solution, EDI (Pan et al. 2022), designed for simultaneous deblurring and video reconstruction by an optimization algorithm, has explained the significance of combining a single image with the following events for frame prediction, but they are time-consuming and vulnerable to noise.

Image-based Frame Interpolation

Image-based VFI is to increase the temporal resolution of frame sequences. It can be simply divided into two categories: namely kernel-based (Niklaus, Mai, and Liu 2017a,b; Choi et al. 2020; Khalifeh et al. 2022; Shi et al. 2022) and flow-based (Jiang et al. 2018; Liu et al. 2019; Kong et al. 2022; Hu et al. 2022; Huang et al. 2022) approachs. Kernel-based methods generate latent pixels for the interpolated frames by local convolutions and can only handle the limited motion range. The flow-based VFI produces the intermediate frames by estimating the optical flow and can adapt to various motion ranges. Since flow-based methods rely on linear motion assumptions, most of them cannot model complex scenes accurately. Although quadratic (Xu et al. 2019; Dutta, Subramaniam, and Mittal 2022) and cubic (Chi et al. 2020) motion models are proposed to address these problems, these methods still cannot solve the performance degradation when facing difficult situations.

Event-based Frame Interpolation

Event-based VFI utilizes the information of the image and event stream to generate the intermediate frames. Existing methods can be divided into kernel-based (Lin et al. 2020; Zou et al. 2021; Yu et al. 2021; Zhang and Yu 2022; Kılıç, Akman, and Alatan 2023), flow-based (He et al. 2022; Wu et al. 2022) and composite methods (Tulyakov et al. 2021, 2022). Kernel-based methods generate the latent frames by convolution network, while flow-based ones produce the intermediate frames by estimating optical flow. Composite methods are a mixture of these two methods and compromise the merits of two of these methods. Despite the significant performance, the event-based VFI suffers from the same problem as the image-based VFI, that is, it still requires future frames relative to the generated frames as input, which leads to significant latency in practice.

Method

Given a single input image I_{t_0} at time t_0 and the following events $E_{t_0 \to t_n} = \{e_i\}^M = \{x_i, y_i, t_i, p_i\}^M, i \in [1, M], t_i \in [t_0, t_n]$ with position (x_i, y_i) at image plane, brightness change timestamp t_i and polarity p_i, M is the number of events, event-based video frame prediction aims to predict future frames $\{I_{t_1}, I_{t_2}, \cdots, I_{t_n}\}^{(M-1)}_{t_0<t_1<t_2<\cdots<t_n}$, where n is the number of predicted frames. In this section, we introduce our proposed frame prediction model (see

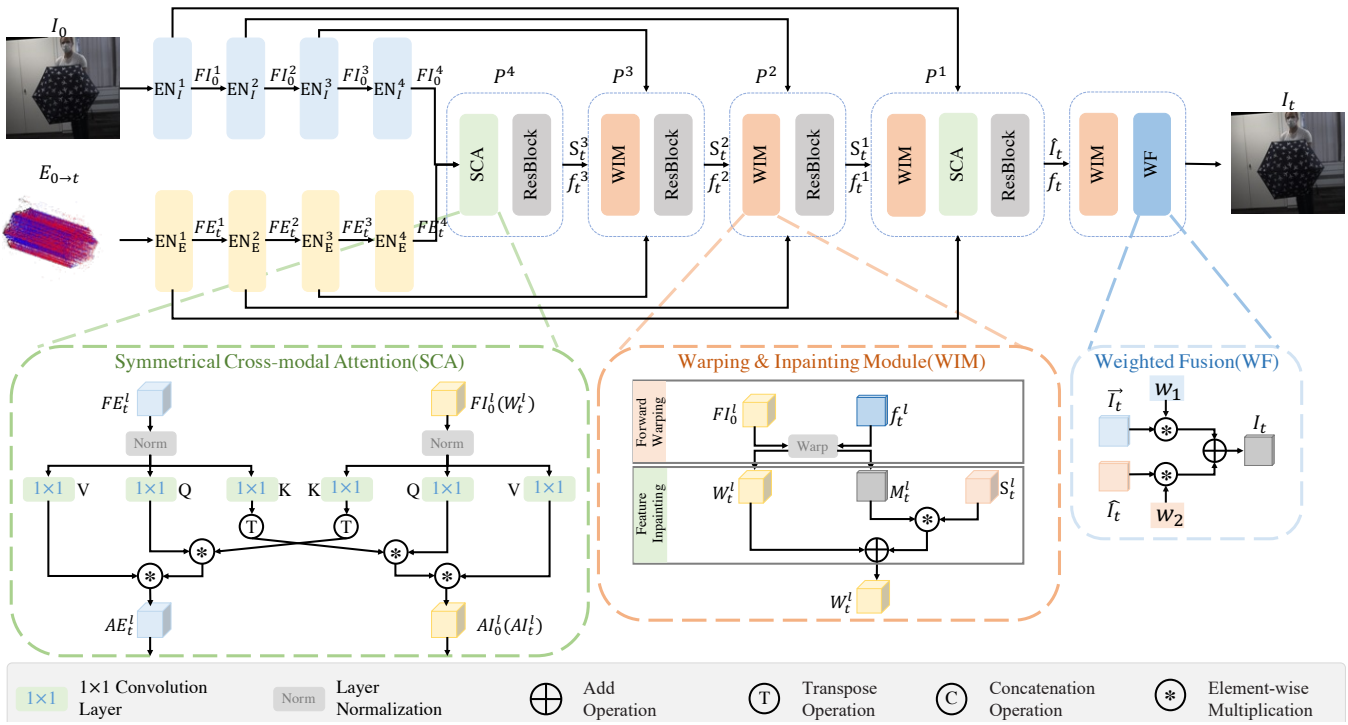

Figure 2: **Overview of our proposed network model.** In our framework, we first use two encoders to extract pyramid features for the image and events. Then we apply a coarse-to-fine joint decoder to get the synthesized feature and optical flow at each pyramid layer. In the decoder, we utilize *Symmetrical Cross-modal Attention* to augment both image and event features. We also introduce *Warping and Inpainting Module* to repair the holes caused by forward warping and get spatially-aligned image features. Finally, we adopt *Weighted Fusion* to output the final frame prediction from the synthesized and warped frames.

Fig. 2) from a single image and events, including event representation and feature encoding, symmetrical cross-modal attention, and joint flow and frame decoder with inpainting.

Event Representation and Feature Encoding

Due to the special space-time format of the event stream, we need to first convert the origin events $E_{t_0 \to t_n}$ to the event voxel $EV_{t_0 \to t_n}$ before inputting it into the subsequent models. Following (Zihao Zhu et al. 2018; Rebecq et al. 2019), we divide $\{e_i\}^M$ into B temporal bins and sum the normalized timestamps for each pixel position $(x, y) \in \{(x_i, y_i)\}^M$ in each temporal bin $b \in [1, B]$ as follows:

$$E(x,y,b) = \sum_{i=1}^{M} p_i \max\left(0, 1 - \left|b - (B-1)\frac{t_i - t_0}{t_0 - t_n}\right|\right). \quad (1)$$

To reduce the computational cost, we apply two lightweight feature encoders to extract the pyramid features for the image and events separately. Each encoder consists of residual convolution blocks, which comprise two convolutions and the PReLU (He et al. 2015) activation. The channel number of the pyramid features are set to 24, 36, 54 and 72 from the shallow to deep pyramid layer.

Symmetrical Cross-modal Attention

Due to special perception mechanism, event stream lacks the competence to capture the motion in the areas where the brightness change is implicit. By contrast, image can provide dense and rich context information, but cannot encode motion information. To compensate for the disadvantages of these two data sources, we introduce a cross-modal attention feature augmentation module to symmetrically enhance the context and motion feature. This module is an adaptation of self-attention(Vaswani et al. 2017), which includes two symmetrical attention enhancement branches: Image-to-Event (I2E) attention and Event-to-Image (E2I) attention. Note that, unlike the attention fusion in EFNet (Sun et al. 2022) to obtain one fused feature, we aim to augment each other to get two enhanced features and apply this module only at the 1st and 4th pyramid layers. As the image and event features are gradually spatially aligned by the estimated optical flow, the augmented features in the 1st layer are further augmented by adding them with the original features, while the augmented features in the 4th layer are not.

The I2E attention determines the importance matrix of the image feature by counting the similarity between the image feature and the event feature. The image feature is enhanced by multiplying the image feature and the normalized weight obtained from cross similarity:

$$Attention(Q_E, K_I, V_I) = V_I \cdot softmax\left(\frac{{Q_E}^T K_I}{\sqrt{d_k}}\right), \quad (2)$$

where K_I and V_I are the keys and values obtained from im-

age feature, Q_E are the queries extracted from event feature and $\sqrt{d_k}$ means the dimension of K_E.

The E2I attention obtains the weight of the event feature by normalizing the similarity matrix between the event feature and the image feature. We can obtain the augmented event feature by reweighting the event feature:

$$Attention(Q_I, K_E, V_E) = V_E \cdot softmax\left(\frac{{Q_I}^T K_E}{\sqrt{d_k}}\right), \tag{3}$$

where K_E and V_E are the keys and values obtained from event feature, Q_I are the queries extracted from image feature and $\sqrt{d_k}$ means the dimension of K_I.

Joint Flow and Frame Decoder with Inpainting

To simplify the procedure and reduce the computational cost, we apply an integrated decoder to estimate the optical flow and generate the target frame in a coarse-to-fine manner. Our model includes four pyramid layers. For the bottom layer, namely P_4, we input the event and image feature FI_0^4, FE_t^4 to symmetrical cross-modal attention and predict the optical flow f_t^3 and synthesized frame feature S_t^3 using the augmented feature AI_0^4, AE_t^4. For the middle layers P^3 and P^2, we first warp the extracted image feature FI_0^l to the target time t and get W_t^l. Then we concatenate the warped feature W_t^l, event feature FE_t^l and synthesized frame feature S_t^l and optical flow f_t^l from the last layer together as the input of the decoder. For the top layer P^1, we augment the event and inpainted feature from the 2nd layer, feed into the decoder and get optical flow f_t and synthesized frame $\widehat{I}_t$ at the top layer. Then we directly warp the input reference image I_0 to $\overrightarrow{I_t}$. Different from the frame-based VFI methods that perform a bi-directional check to deal with the occlusions, our VFP setting has a unique problem in dealing with the holes generated by forward warping. To relieve this problem, we introduce an efficient *hole inpainting module* to inpaint the warped frame at the top layer and inpaint the warped feature at the bottom layers. First, we modify the commonly used CUDA accelerated implementation of forward warping (Niklaus and Liu 2020) to get the occlusion mask $\mathbf{Occ}$ by a fixed threshold. Then we use synthesized frame feature S_t^l to inpaint the holes:

$$W_t^l(x,y) = \begin{cases} S_t^l(x,y) & ,\mathbf{Occ}(x,y) > 0, \\ Warp(FI_0^l, f_t^l)(x,y) & ,\mathbf{Occ}(x,y) \leq 0, \end{cases} \tag{4}$$

where $Warp$ is the forward warping operator.
Compared with existing methods that design a new module to inpaint the holes (Gao et al. 2019), our efficient module can inpaint the holes with the synthesized features at every pyramid layer without increasing large computational cost.

Although holes in the warped frame $\overrightarrow{I_t}$ can be inpainted by the synthesized frame $\widehat{I}_t$, the outline of the hole may be evident and affect the harmony of the final prediction. In addition, the accuracy of the synthesized frame $\widehat{I}_t$ and the warped frame $\overrightarrow{I_t}$ is different when confronting different motion scenes. Accordingly, we propose a weighted fusion refinement module, which uses a network to learn an allocation weight $w \in [0, 1]$. Lastly, we fuse the synthesized frame $\widehat{I}_t$ and warped frame $\overrightarrow{I_t}$ with the weight w to obtain the final output, *i.e.*, frame prediction:

$$I_t = w \cdot \overrightarrow{I_t} + (1 - w) \cdot \widehat{I}_t. \tag{5}$$

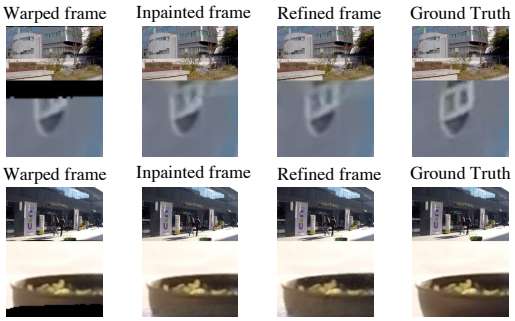

Figure 3: **Visualization schematics** of Inpainting Module and Fusion Refinement Module.

Fig. 3 illustrates the visual comparisons of the inpainting module and the fusion refinement module. The inpainting module can generate the filling pixels which are consistent with the surrounding area in absence of the subsequent frame, while the weighted fusion refinement module can make the border more accurate and smooth.

Loss Function

To supervise the final predicted frames in training, we first apply our reconstruction loss, which consists of the Charbonnier loss(Charbonnier et al. 1994) L_{Cha}, the Census loss(Meister, Hur, and Roth 2018) L_{Cen}, and the LPIPS loss(Zhang et al. 2018) L_{LPIPS}:

$$\begin{aligned} L_{rec}(I_t) =& L_{Cha}(I_t - I_{gt}) + \alpha_1 L_{Cen}(I_t - I_{gt}) \\ &+ \alpha_2 L_{LPIPS}(I_t - I_{gt}), \end{aligned} \tag{6}$$

where $L_{Cha}(x) = \sqrt{x^2 + 10^{-6}}$ and $\alpha_1 = 1.0$, $\alpha_2 = 1.0$. To ensure the quality of inpainting padding, we also apply the reconstruction loss to synthesized frames,*i.e.*, $L_{rec}(\widehat{I}_t)$.

We adopt the pseudo optical flow generated by RAFT (Teed and Deng 2020) to supervise the optical flow using the task-oriented flow loss (Kong et al. 2022), which can adjust the loss weight dynamically and is defined as follows:

$$\begin{aligned} R &= e^{-\beta \|f_t - f_p\|_2}, \\ L_{flow} &= \sum_{l=1}^{L-1} \left((f_t^l - f_p)^2 + \epsilon^2\right)^{\frac{r}{2}} + \|f_t^l - f_p\|, \end{aligned} \tag{7}$$

where$\|.\|$ is the L_2 norm between estimated optical flow f_t and pseudo optical flow f_p, $r = R(u, v)$ is the robustness weight at position (u, v),$\epsilon = 10^{(10r-1)/3}$ and $\beta = 0.3$.

We use the Census loss as feature consistency loss to supervise the synthesized feature as follows:

$$L_{feat} = \sum_{l=1}^{L-1} L_{Cen}(S_t^l - FI_{gt}^l), \tag{8}$$

Table 1: **Performance comparison** on the GoPro and HS-ERGB datasets. The results refer to the PSNR/SSIM metrics. * means inpainting the holes caused by forward warping with the synthesized frames generated by our model.

Method	Setting	Input	GoPro		HS-ERGB	Model Size	Time
			7 frames	15 frames	7 frames	(MB)	(s)
IFRNet	Frame Interpolation	2 Images	29.27/0.92	24.78/0.82	27.35/0.83	19.0	**0.038**
EVDI		2 Images + Event	25.13/0.75	22.62/0.66	26.10/0.77	**1.6**	0.200
Time Lens		2 Images + Event	**32.66/0.94**	**29.81/0.90**	**32.12/0.86**	454.0	0.290
E2VID	Reconstruction	Event	14.46/0.59	-	8.84/0.40	41.0	0.054
HyperE2VID		Event	15.37/0.61	-	10.92/0.44	39.0	0.140
DCEIFlow	Flow Estimation	1 Image + Event	26.45/0.92	23.36/0.85	26.29/0.80	28.0	0.130
DCEIFlow*		1 Image + Event	29.21/0.93	26.15/0.87	27.87/0.83		
OVP	Frame Prediction	2 Images	26.15/0.89	22.90/0.68	25.47/0.76	33.0	327.860
DMVFN		2 Images	25.48/0.84	21.46/0.73	27.59/0.82	14.0	0.013-0.038
EDI		1 Image + Event	20.11/0.62	18.43/0.55	22.64/0.70	-	-
Our model		1 Image + Event	**29.73/0.93**	**27.71/0.89**	**28.07/0.83**	**8.3**	**0.024**

where FI^l_{gt} means ground-truth feature extracted from the ground-truth frame I_{gt} using the image encoder.

Based on the above analysis, the final training loss is formulated as:

$$L = L_{rec}(I_t) + \lambda_1 L_{rec}(\widehat{I}_t) + \lambda_2 L_{flow} + \lambda_3 L_{feat}, \quad (9)$$

where the weighting parameters are set to $\lambda_1 = 1.0, \lambda_2 = 0.5, \lambda_3 = 0.1$ in our experiments.

Experiments

Implementation Details

All experiments are conducted with PyTorch. We employ an AdamW optimizer for 50 epochs training with batch size 4 on two NVIDIA RTX3090 GPUs. The learning rate is decayed from 1×10^{-4} to 1×10^{-5} with a cosine learning rate scheduler. To obtain reliable motion priors, we first pretrain our model only under the supervision of task-oriented flow loss in the first 15 epochs, followed by training the model with full loss in the remaining 35 epochs. To augment the training data, we make vertical and horizontal flipping with 50% probability and crop 384 $\times$ 384 patches randomly. We simulate events using the Vid2E (Gehrig et al. 2020) simulator. To enhance the model's ability to extract temporal information, we apply two training modes: predict the target frame using the first frame and predict the target frame using the last prediction, and switch two modes with 50 % probability in training.

Our experiments are conducted on both synthetic and real datasets. PSNR and SSIM are adopted for quantitative evaluation. Consistent with the setting of existing event-based VFI (Tulyakov et al. 2021), we pre-simulate the events of Vimeo90k septuplet dataset (Xue et al. 2019) and GoPro dataset (Nah, Hyun Kim, and Mu Lee 2017), then train our model on Vimeo90k and evaluate on the GoPro test set. For experiments with real-captured data, we choose the HS-ERGB dataset (Tulyakov et al. 2021) for evaluation, which records the data with 1280$\times$ 720 resolution at 160 fps and contains diverse scenes. Besides, we also perform quantitative comparisons on DSEC (Gehrig et al. 2021), a dataset of events for driving scenarios.

Evaluation with Synthetic Events

We first evaluate our Vimeo90k pretrained model on the GoPro dataset. We conduct a quantitative comparison between our method and several existing methods with different input settings in Table 1. The methods we compare are state-of-the-art models with open source code in the fields of **1)** VFI with two images, *i.e.*, IFRNet (Kong et al. 2022), **2)** VFI with two images and events, *i.e.*, Time Lens (Tulyakov et al. 2021), **3)** Frame reconstruction with events, *i.e.*, E2VID (Rebecq et al. 2019), hyperE2VID (Ercan et al. 2023), **4)** Flow estimation with single image and events and get the predicted frame by warping, *i.e.*, DCEIFlow (Wan, Dai, and Mao 2022), **5)** VFP with two images, *i.e.*, OVP (Hu et al. 2022) and DMVFN (Hu et al. 2023) **6)** VFP with single image and events, *i.e.*, EDI (Pan et al. 2022) and our model. Note that EDI is an optimization method, E2VID and hyperE2VID do not provide training codes, DCEIFlow is originally used to estimate optical flow, thus we use their publicly available parameters and model weights. As shown in Table 1, we evaluate the above methods for 7 frames and 15 frames respectively, which indicates that the prediction (interpolation) methods predict (interpolate) 7 and 15 following (intermediate) frames.

Compared with VFI methods, we achieve competitive results with the premise that only the first frame is input. Time Lens integrates two frames and events and achieves better results, which shows that event data is of great help to solve long-term motion than using only images. Since we do not use the second frame, our results are inferior to Time Lens. Nonetheless, compared with IFRNet which employs two frames to interpolate intermediate frames, our model achieves a PSNR improvement of up to 2.93dB for 15 frames prediction.

Compared to the VFP methods, they leverage multiple preceding images to predict the frame, while our model utilizes a single frame along with events. Our method improves the PSNR by up to 3dB than OVP. Furthermore, we also compare the visualization results of each model in Fig. 4. Following OVP, we present the outcomes of the 1st, 3rd and

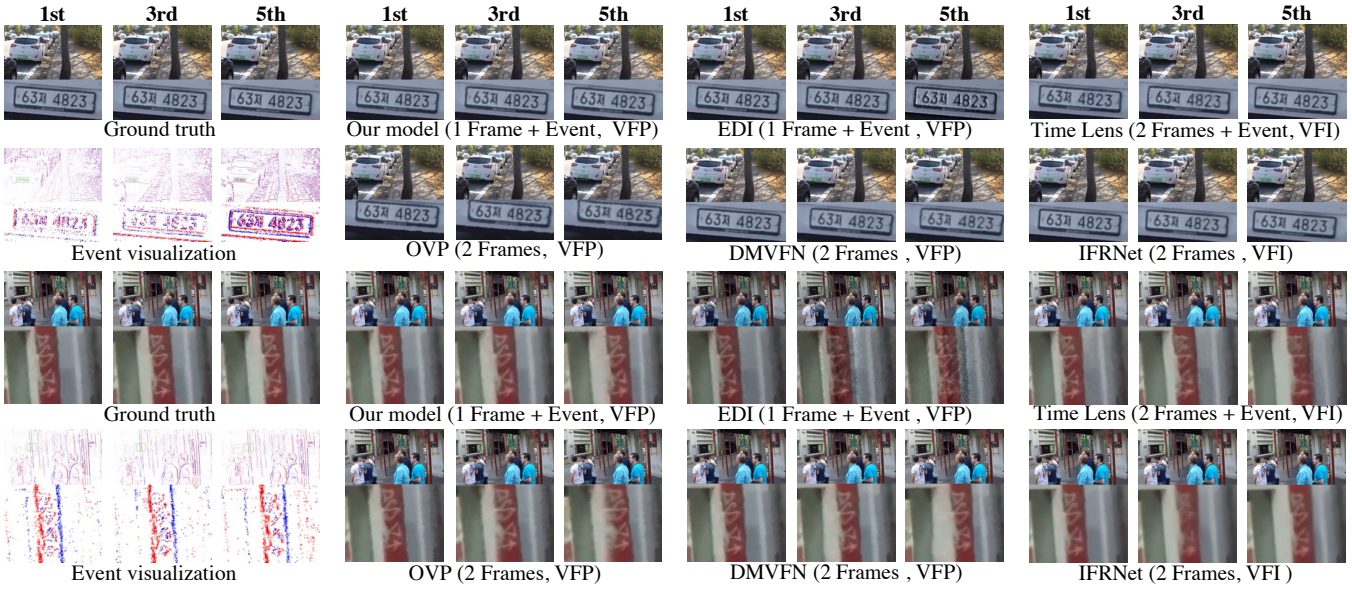

Figure 4: **Visual comparisons** on the GoPro dataset with synthetic events.

Table 2: **Performance comparisons** on the DSEC dataset for 3 frames VFP when 1 image and events are input.

Method	Setting	PSNR(dB)	SSIM
DCEIFlow	Flow Estimation	25.18	0.81
DCEIFlow*		25.59	0.82
EDI	Frame prediction	20.59	0.62
Our model		**26.61**	**0.85**

5th frames under the 7-frames evaluation setting. The visual comparisons show that our model can predict more accurate frames than the existing image-based VFP methods. Under the same input setting, our model also shows better performance than EDI and DCEIFlow in frame estimation. This superior performance validates the efficacy of incorporating events and our proposed framework into VFP.

In addition, we present a comprehensive report of the model size and runtime for each model in Table 1, where the runtime is measured by generating a 720P image on a 2080Ti GPU. For EVDI, we assume that its efficacy is inferior to that of our model because its model parameters are too small to handle intricate dynamic scenarios. Compared with DCEIFlow, we attribute our model's superior performance and efficiency to the inclusion of the inpainting module and the integrated architecture that obviates iterations. For DMVFN, its runtime ranges from 0.013s to 0.038s for its dynamic routing mechanism, which takes longer time to deal with large motion.

In summary, our proposed model exhibits optimal runtime performance, possessing the second smallest model size in comparison to competing methods and it stands out as the only approach satisfying real-time demand.

Table 3: **Ablation studies** on attention augmentation, flow estimation, loss function, $\widehat{I}_t$ estimation target and training mechanism.

Ablations	Variations	PSNR	SSIM
Attention Augmentation	W/o Attention	28.58	0.91
	4th layer	29.32	0.92
	3th layer	29.08	0.92
	2nd layer	29.15	0.92
	1st layer	29.21	0.93
	1st and 4th layer	**30.80**	**0.95**
Flow Estimation	Backward Flow	**30.84**	0.94
	Forward Flow &W/o Inpainting	26.14	0.91
	Forward Flow &W/ Inpainting	30.80	**0.95**
Loss Function	W/o Flow Loss	29.75	0.93
	W/o Feature Loss	28.45	0.90
	W/o Charbonnier	28.69	0.89
	W/o LPIPS	30.52	0.93
	Full Losses	**30.80**	**0.95**
$\widehat{I}_t$ Estimation Target	Residual Intensity	**30.83**	**0.95**
	Absolute Intensity	30.80	**0.95**
Training Mechanism	W/o flow pretrain	29.80	0.93
	W/ flow pretrain	**30.80**	**0.95**

Evaluation with Real-captured Events

We conduct experiments on the HS-ERGB dataset in Table 1. The reported PSNR and SSIM results are averaged over the two subsets. Compared to VFI methods, our proposed model only performs inferiorly to Time Lens for lack of event and image information after t. Compared with VFP

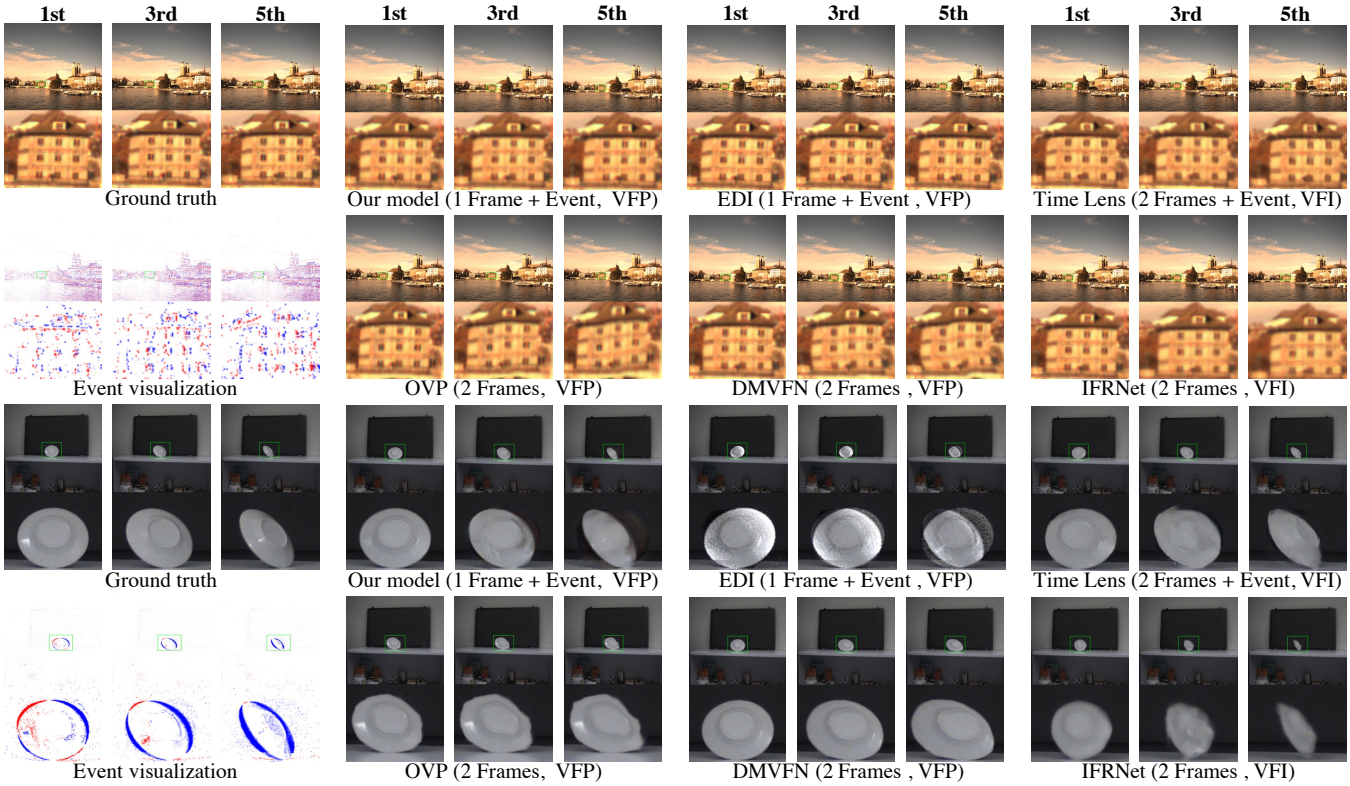

Figure 5: **Visual comparisons** on the real-captured HS-ERGB dataset.

methods, our model outperforms the above methods, which is consistent with the observation on the GoPro and confirms our model's superiority. Visual comparisons in Fig. 5 also verify this observation. In Table 2, we perform quantitative comparisons of the real-captured DSEC dataset collected in driving scenarios, which further illustrate our model's ability for VFP to generalize to practical complex scenarios.

Ablation Studies

To verify the contribution of each module, we conduct ablations in Table 3 from five aspects: attention augmentation, flow estimation, loss function, estimation target and training mechanism. Due to the large data volume of Vimeo90K dataset, we choose to train these ablation models on the GoPro training set with 100 epochs and evaluate them on the test set for comparison.

Attention Augmentation. To verify the effectiveness of our cross-modal attention augmentation, we first remove the attention module, resulting in a decrease of over 2dB in PSNR. Then we apply it to four pyramid layers respectively. From Table 3, the attention mechanism contributes most on the first and fourth layers. Thus we strike a balance between computation cost and performance and apply the augmentation module to the first and fourth layers.

Flow Estimation. We conduct experiments on flow estimation and the result indicates that the model estimating forward flow with the inpainting module achieves higher SSIM while the model estimating backward flow has higher PSNR. Considering the ghost effect introduced by backward warping, we ultimately select the former approach.

Loss Function. To evaluate the contributions of task-oriented flow loss, feature loss and reconstruction loss, we conduct experiments wherein we train the model without them separately. Table 3 shows that the model's performance significantly decreases without any of them.

Estimation Target. Since the initial frame is provided, it is intuitive to estimate residual intensity instead of absolute intensity, which is also proved in Table 3. However, for consistency with existing event-based methods, we still use the absolute intensity as the target of $\widehat{I}_t$ in our model.

Training Mechanism. Since jointly learning optical flow and frame is a "chicken-and-egg" problem, we employ a two-stage training approach. This strategy results in a 1.0 dB PSNR performance improvement.

Conclusion

In this paper, we have studied the problem of video frame prediction (VFP) from a single RGB image and the following events. By introducing events to VFP, we can achieve *flexible frame prediction for complex dynamic scenes*, where the temporal interval between the predicted frames can be long or short. Based on our proposed network, we can significantly exceed the performance of existing VFP methods and meet the requirements of real-time frame prediction. We believe that event-based VFP, which combines events with images, is more practical than image-based VFI and VFP.

Acknowledgements

This research was supported in part by the National Natural Science Foundation of China (62271410, 62001394), Zhejiang Lab (NO.2021MC0AB05), the Fundamental Research Funds for the Central Universities, and the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University (CX2023013).

References

Bei, X.; Yang, Y.; and Soatto, S. 2021. Learning semantic-aware dynamics for video prediction. In *CVPR*, 902–912.

Chang, Z.; Zhang, X.; Wang, S.; Ma, S.; and Gao, W. 2022. STRPM: A spatiotemporal residual predictive model for high-resolution video prediction. In *CVPR*, 13946–13955.

Charbonnier, P.; Blanc-Feraud, L.; Aubert, G.; and Barlaud, M. 1994. Two deterministic half-quadratic regularization algorithms for computed imaging. In *CVPR*, 168–172.

Chi, Z.; Mohammadi Nasiri, R.; Liu, Z.; Lu, J.; Tang, J.; and Plataniotis, K. N. 2020. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *ECCV*, 107–123.

Choi, H.; and Baji, I. V. 2021. Affine transformation-based deep frame prediction. *TIP*, 30: 3321–3334.

Choi, M.; Kim, H.; Han, B.; Xu, N.; and Lee, K. M. 2020. Channel attention is all you need for video frame interpolation. In *AAAI*, 10663–10671.

Ding, Z.; Zhao, R.; Zhang, J.; Gao, T.; Xiong, R.; Yu, Z.; and Huang, T. 2022. Spatio-temporal recurrent networks for event-based optical flow estimation. In *AAAI*, 525–533.

Dutta, S.; Subramaniam, A.; and Mittal, A. 2022. Non-linear motion estimation for video frame interpolation using space-time convolutions. In *CVPR*, 1726–1731.

Ercan, B.; Eker, O.; Saglam, C.; Erdem, A.; and Erdem, E. 2023. HyperE2VID: Improving Event-Based Video Reconstruction via Hypernetworks. *arXiv preprint arXiv:2305.06382*.

Fan, H.; Zhu, L.; and Yang, Y. 2019. Cubic LSTMs for video prediction. In *AAAI*, 8263–8270.

Finn, C.; Goodfellow, I.; and Levine, S. 2016. Unsupervised learning for physical interaction through video prediction. In *NeurIPS*, 64–72.

Gao, H.; Xu, H.; Cai, Q.-Z.; Wang, R.; Yu, F.; and Darrell, T. 2019. Disentangling propagation and generation for video prediction. In *ICCV*, 9006–9015.

Gehrig, D.; Gehrig, M.; Hidalgo-Carrió, J.; and Scaramuzza, D. 2020. Video to events: Recycling video datasets for event cameras. In *CVPR*, 3586–3595.

Gehrig, M.; Aarents, W.; Gehrig, D.; and Scaramuzza, D. 2021. DSEC: A stereo event camera dataset for driving scenarios. *RA-L*, 6(3): 4947–4954.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 1026–1034.

He, W.; You, K.; Qiao, Z.; Jia, X.; Zhang, Z.; Wang, W.; Lu, H.; Wang, Y.; and Liao, J. 2022. TimeReplayer: Unlocking the potential of event cameras for video interpolation. In *CVPR*, 17804–17813.

Hu, P.; Niklaus, S.; Sclaroff, S.; and Saenko, K. 2022. Many-to-many splatting for efficient video frame interpolation. In *CVPR*, 3553–3562.

Hu, X.; Huang, Z.; Huang, A.; Xu, J.; and Zhou, S. 2023. A dynamic multi-scale voxel flow network for video prediction. In *CVPR*, 6121–6131.

Huang, Z.; Zhang, T.; Heng, W.; Shi, B.; and Zhou, S. 2022. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*, 624–642.

Huo, S.; Liu, D.; Li, B.; Ma, S.; Wu, F.; and Gao, W. 2020. Deep network-based frame extrapolation with reference frame alignment. *TCSVT*, 31(3): 1178–1192.

Jiang, H.; Sun, D.; Jampani, V.; Yang, M.-H.; Learned-Miller, E.; and Kautz, J. 2018. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, 9000–9008.

Khalifeh, I.; Blanch, M. G.; Izquierdo, E.; and Mrak, M. 2022. Multi-encoder network for parameter reduction of a kernel-based interpolation architecture. In *CVPR*, 725–734.

Kılıç, O. S.; Akman, A.; and Alatan, A. A. 2023. E-VFIA: Event-Based video frame interpolation with attention. In *ICRA*, 8284–8290.

Kong, L.; Jiang, B.; Luo, D.; Chu, W.; Huang, X.; Tai, Y.; Wang, C.; and Yang, J. 2022. IFRNet: Intermediate feature refine network for efficient frame interpolation. In *CVPR*, 1969–1978.

Kwon, Y.-H.; and Park, M.-G. 2019. Predicting future frames using retrospective cycle gan. In *CVPR*, 1811–1820.

Liang, X.; Lee, L.; Dai, W.; and Xing, E. P. 2017. Dual motion GAN for future-flow embedded video prediction. In *ICCV*, 1744–1752.

Lin, S.; Zhang, J.; Pan, J.; Jiang, Z.; Zou, D.; Wang, Y.; Chen, J.; and Ren, J. 2020. Learning event-driven video deblurring and interpolation. In *ECCV*, 695–710.

Liu, Y.-L.; Liao, Y.-T.; Lin, Y.-Y.; and Chuang, Y.-Y. 2019. Deep video frame interpolation using cyclic frame generation. In *AAAI*, 8794–8802.

Liu, Z.; Nie, Y.; Long, C.; Zhang, Q.; and Li, G. 2021. A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *ICCV*, 13588–13597.

Liu, Z.; Yeh, R. A.; Tang, X.; Liu, Y.; and Agarwala, A. 2017. Video frame synthesis using deep voxel flow. In *ICCV*, 4463–4471.

Meister, S.; Hur, J.; and Roth, S. 2018. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, 7251–7259.

Nah, S.; Hyun Kim, T.; and Mu Lee, K. 2017. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 3883–3891.

Niklaus, S.; and Liu, F. 2020. Softmax splatting for video frame interpolation. In *CVPR*, 5437–5446.

Niklaus, S.; Mai, L.; and Liu, F. 2017a. Video frame interpolation via adaptive convolution. In *CVPR*, 670–679.

Niklaus, S.; Mai, L.; and Liu, F. 2017b. Video frame interpolation via adaptive separable convolution. In *ICCV*, 261–270.

Pan, L.; Hartley, R.; Scheerlinck, C.; Liu, M.; Yu, X.; and Dai, Y. 2022. High frame rate video reconstruction based on an event camera. *TPAMI*, 44(5): 2519–2533.

Qi, X.; Liu, Z.; Chen, Q.; and Jia, J. 2019. 3D motion decomposition for RGBD future dynamic scene synthesis. In *CVPR*, 7673–7682.

Rebecq, H.; Ranftl, R.; Koltun, V.; and Scaramuzza, D. 2019. Events-to-video: Bringing modern computer vision to event cameras. In *CVPR*, 3857–3866.

Shi, Z.; Xu, X.; Liu, X.; Chen, J.; and Yang, M.-H. 2022. Video frame interpolation transformer. In *CVPR*, 17482–17491.

Sun, L.; Sakaridis, C.; Liang, J.; Jiang, Q.; Yang, K.; Sun, P.; Ye, Y.; Wang, K.; and Van Gool, L. 2022. Event-based fusion for motion deblurring with cross-modal attention. In *ECCV*, 412–428.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 402–419.

Tulyakov, S.; Bochicchio, A.; Gehrig, D.; Georgoulis, S.; Li, Y.; and Scaramuzza, D. 2022. Time Lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion. In *CVPR*, 17755–17764.

Tulyakov, S.; Gehrig, D.; Georgoulis, S.; Erbach, J.; Gehrig, M.; Li, Y.; and Scaramuzza, D. 2021. Time Lens: Event-based video frame interpolation. In *CVPR*, 16155–16164.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 6000–6010.

Villegas, R.; Yang, J.; Hong, S.; Lin, X.; and Lee, H. 2017. Decomposing motion and content for natural video sequence prediction. In *ICLR*.

Wan, Z.; Dai, Y.; and Mao, Y. 2022. Learning dense and continuous optical flow from an event camera. *TIP*, 31: 7237–7251.

Wan, Z.; Mao, Y.; Zhang, J.; and Dai, Y. 2023. RPEFlow: Multimodal Fusion of RGB-PointCloud-Event for Joint Optical Flow and Scene Flow Estimation. In *ICCV*, 10030–10040.

Wang, D.; Jia, X.; Zhang, Y.; Zhang, X.; Wang, Y.; Zhang, Z.; Wang, D.; and Lu, H. 2023. Dual Memory Aggregation Network for Event-Based Object Detection with Learnable Representation. In *AAAI*, 2492–2500.

Wang, Y.; Wu, H.; Zhang, J.; Gao, Z.; Wang, J.; Yu, P.; and Long, M. 2022. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *TPAMI*, 45(2): 2208–2225.

Wu, S.; You, K.; He, W.; Yang, C.; Tian, Y.; Wang, Y.; Zhang, Z.; and Liao, J. 2022. Video interpolation by event-Driven anisotropic adjustment of optical flow. In *ECCV*, 267–283.

Wu, Y.; Gao, R.; Park, J.; and Chen, Q. 2020. Future video synthesis with object motion prediction. In *CVPR*, 5539–5548.

Xu, X.; Siyao, L.; Sun, W.; Yin, Q.; and Yang, M.-H. 2019. Quadratic video interpolation. In *NeurIPS*, 1647–1656.

Xue, T.; Chen, B.; Wu, J.; Wei, D.; and Freeman, W. T. 2019. Video enhancement with task-Oriented flow. *IJCV*, 1106–1125.

Yu, Z.; Zhang, Y.; Liu, D.; Zou, D.; Chen, X.; Liu, Y.; and Ren, J. S. 2021. Training weakly supervised video frame interpolation with events. In *ICCV*, 14589–14598.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 586–595.

Zhang, X.; and Yu, L. 2022. Unifying motion deblurring and frame interpolation with events. In *CVPR*, 17765–17774.

Zihao Zhu, A.; Yuan, L.; Chaney, K.; and Daniilidis, K. 2018. Unsupervised event-based optical flow using motion compensation. In *ECCV Workshops*, 711–714.

Zou, Y.; Zheng, Y.; Takatani, T.; and Fu, Y. 2021. Learning to reconstruct high speed and high dynamic range videos from events. In *CVPR*, 2024–2033.