

Learning Spatial-Temporal Implicit Neural Representations for Event-Guided Video Super-Resolution

Yunfan Lu^{1*} Zipeng Wang^{1*} Minjie Liu^{1†} Hongjian Wang^{2†} Lin Wang^{1,3‡}
¹AI Thrust, HKUST(GZ) ²Shenzhen International Graduate School, Tsinghua University
³Dept. of Computer Science and Engineering, HKUST

{yflu066, zwang253, mliu942}@connect.hkust-gz.edu.cn, whj20@mails.tsinghua.edu.cn, linwang@ust.hk

Abstract

Event cameras sense the intensity changes asynchronously and produce event streams with high dynamic range and low latency. This has inspired research endeavors utilizing events to guide the challenging video super-resolution (VSR) task. In this paper, we make the **first** attempt to address a novel problem of achieving VSR at random scales by taking advantages of the high temporal resolution property of events. This is hampered by the difficulties of representing the spatial-temporal information of events when guiding VSR. To this end, we propose a novel framework that incorporates the spatial-temporal interpolation of events to VSR in a unified framework. Our key idea is to learn implicit neural representations from queried spatial-temporal coordinates and features from both RGB frames and events. Our method contains three parts. Specifically, the Spatial-Temporal Fusion (STF) module first learns the 3D features from events and RGB frames. Then, the Temporal Filter (TF) module unlocks more explicit motion information from the events near the queried timestamp and generates the 2D features. Lastly, the Spatial-Temporal Implicit Representation (STIR) module recovers the SR frame in arbitrary resolutions from the outputs of these two modules. In addition, we collect a real-world dataset with spatially aligned events and RGB frames. Extensive experiments show that our method significantly surpasses the prior-arts and achieves VSR with random scales, e.g., 6.5. Code and dataset are available at <https://vllis2022.github.io/cvpr23/egvsr>.

1. Introduction

Video super-resolution (VSR) is a task of recovering high-resolution (HR) frames from successive multiple low-resolution (LR) frames. Unlike LR videos, HR videos contain more visual information, e.g., edge and texture, which

*These authors are co-first authors.

†These authors are co-second authors.

‡Corresponding author

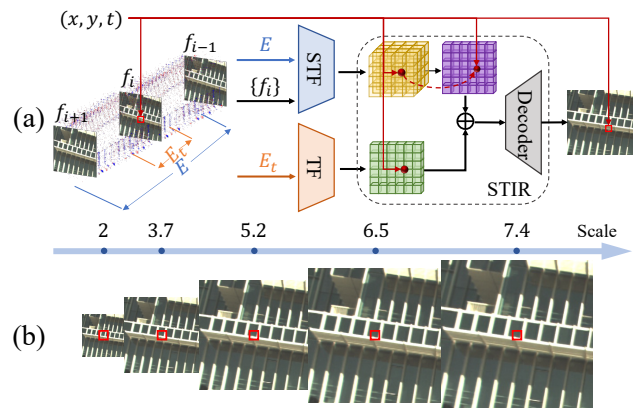


Figure 1. (a) Our method learns implicit neural representations (INR) from the queried spatial-temporal coordinates (STF) and temporal features (TF) from RGB frames and events. (b) An example of VSR with random scale factors, e.g., 6.5, by our method.

can be very helpful for many tasks, e.g., metaverse [52], surveillance [15] and entertainment [4]. However, VSR is a highly ill-posed problem owing to the loss of both spatial and temporal information, especially in the real-world scenes [5, 6, 26, 45]. Recently, deep learning-based algorithms have been successfully applied to learn the intra-frame correlation and temporal consistency from the LR frames to recover HR frames, e.g., DUF [19], EDVR [48], RBPN [14], BasicVSR [5], BasicVSR++ [6]. However, due to the lack of inter-frame information, these methods are hampered by the limitations of modeling the spatial and temporal dependencies and may fail to recover HR frames in complex scenes.

Event cameras are bio-inspired sensors that can asynchronously detect the per-pixel intensity changes and generate event streams with low latency (1 μ s) and high dynamic range (HDR) compared with the conventional frame-based cameras (140dB vs. 60dB) [35, 51]. This has sparked extensive research in reconstructing image/video from events [12, 29, 42, 44, 46, 53]. However, the reconstructed results are less plausible due to the loss of visual

details, *e.g.*, structures, and textures. As a result, a recent work has utilized events for guiding VSR [18], trying to ‘inject’ energy from the event-based to the frame-based cameras. It leverages the high-frequency event data to synthesize neighboring frames, so as to find correspondences between consecutive frames. However, it only treats video frames in discrete ways with 2D arrays of pixels and up-samples them at a fixed up-scaling factor *e.g.*, $\times 2$ or $\times 4$. This causes inconvenience and inflexibility in the applications of SR videos, which often require arbitrary resolutions, *i.e.*, random scales.

Recently, some works tried to learn continuous image representations with arbitrary resolutions, *e.g.*, LIIF [7], taking 2D queried coordinates and 2D features as input to learn an implicit neural representation (INR). VideoINR [8], on the other hand, decodes the LR videos into arbitrary spatial resolutions and frame rates by learning from the spatial coordinates and temporal coordinates, respectively. However, it is still unclear how to leverage events to guide learning spatial-temporal INRs for VSR. This is hampered by two challenges. Firstly, although event data can benefit VSR with its high-frequency temporal and spatial information, the large modality gap between the events and video frames makes it challenging to use INRs to represent the 3D spatial-temporal coordinates with event data. Moreover, there lacks HR real-world dataset with spatially well-aligned events and frames.

In this paper, we make the **first** attempt to address a novel problem of achieving VSR at random scales by taking advantage of the high-temporal resolution property of events. Accordingly, we propose a novel framework that subtly incorporates the spatial-temporal interpolation from events to VSR in a unified framework, as shown in Fig. 1. Our key idea is to learn INRs from the queried spatial-temporal coordinates and features from both the RGB frames and events. Our framework mainly includes three parts. The Spatial-Temporal Fusion (**STF**) branch learns the spatial-temporal information from events and RGB frames (Sec. 3.2). The shallow feature fusion and deep feature fusion are employed to narrow the modality gap and fuse the events and RGB frames into 3D global spatial-temporal representations. Then, the Temporal Filter (**TF**) branch further unlocks more explicit motion information from events. It learns the 2D event features from events nearing the queried timestamp (Sec. 3.3). With the features from the STF and TF branches, the Spatial-Temporal Implicit Representation (**STIR**) module decodes the features and recovers SR frames with arbitrary spatial resolutions (Sec. 3.4). That is, given the arbitrary queried coordinates, we apply 3D sampling and 2D sampling to the fused 3D features and event data separately. Finally, the sampling features are added and fed into a decoder, and generate targeted SR frames. In addition, we collect a real-world dataset with a spatial reso-

lution of 3264×2248 , in which the events and RGB frames are spatially aligned. Extensive experiments on two real-world datasets show that our method surpasses the existing methods by **1.3 dB**.

In summary, the main contributions of this paper are five-fold: **(I)** Our work serves as the **first** attempt to address a non-trivial problem of learning INRs from events and RGB frames for VSR at random scales. **(II)** We propose the STF branch and the TF branch to model the spatial and temporal dependencies from events and RGB frames. **(III)** We propose the STIR module to reconstruct RGB frames with arbitrary spatial resolutions. **(IV)** We collect a high-quality real-world dataset with spatially aligned events and RGB frames. **(V)** Our method significantly surpasses the existing methods and achieves SR with random-scales, *e.g.*, 6.5.

2. Related Work

Event-guided Video and Image SR Recently, event data has shown the potential to guide the image or video SR. eSL-Net [43] and EvIntSR [13] focus on employing events to guide the single image SR. Specifically, eSL-Net [43] feeds both the events and LR image to a sparse learning framework to recover an SR image. EvIntSR [13] first reconstructs the latent frames from the events and LR image, which are then fused to reconstruct the SR image. Differently, event-guided VSR takes consecutive frames and events as inputs and models both the spatial and temporal information. A recent work [18] proposed a two-stage method by 1) utilizing events to interpolate the LR video to get a high-frequency video and 2) rebuilding HR key frames. However, it encodes video frames in discrete ways and only up-samples videos at a fixed upscale factor, *e.g.*, $\times 2$. We make the first attempt to achieve VSR at random scales by learning the spatial-temporal implicit representations from events and video frames.

Video Super-Resolution (VSR) The dominant research for VSR mainly focuses on designing learning pipelines [24], concerning the feature learning, frame alignment and multi-frame fusion [3, 16, 22, 23, 30, 40, 47, 49, 50]. For example, Bao *et.al* [2] employed the motion compensation to achieve the frame alignment, while EDVR [48] proposes the deformable convolutions after extracting the features from the input frames. To address the feature propagation and alignment problem effectively, BasicVSR [5] and BasicVSR++ [6] proposed a succinct pipeline based on the bidirectional propagation and optical flow. To better exploit the temporal information of video frames, RBPN [14] extracts and propagates the spatial and temporal information of consecutive frames in a recurrent back-projection manner. Inspired by the random image upsampling, VideoINR [9] represents the frames with implicit representations and thus makes it possible to learn random-scale VSR. However, these works focus on a single modality. Differently, in our work, we

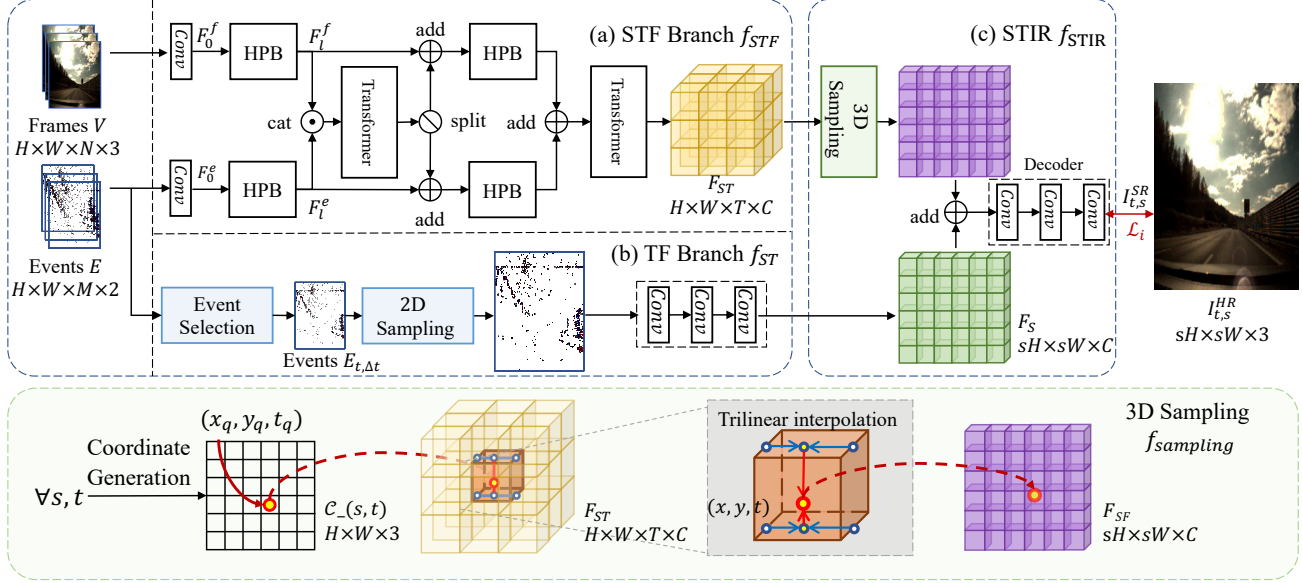


Figure 2. Overview of our proposed framework. Our method consists of three parts, the Spatial-Temporal Fusion (STF), the Temporal Filter (TF) and Spatial-Temporal Implicit Representation (STIR). These three parts are shown in (a) (b) (c) of this figure, respectively. Details of STF, TF and STIR are described in this Sec.3.2, Sec.3.3 and Sec.3.4, respectively.

make the first attempt to address a novel problem of achieving VSR at random scales by taking advantage of the high-temporal resolution property of events.

Implicit Neural Representation (INR) It also called the coordinate-based representation, aims at parameterizing signals, *e.g.*, images and audio, in a continuous way via neural networks [36]. INR has been widely applied to 3D scene representation [28,33] and generative models [17,32], etc. Recently, INRs have been extensively studied for image and video SR. For instance, LIIF [7] achieves image SR with random scales given the 2D queried coordinates and 2D features. JIIF [39] uses HR images to guide the interpolation weights and values of LR depth maps. VideoINR [8] decodes the LR and low-frame-rate videos into an arbitrary spatial resolution and frame rate with three INRs. It adopts two INR functions to learn the spatial coordinates and the temporal coordinates, respectively. These two INR functions are then used to generate a motion flow field, which is applied back to warp the encoded features. Then, the warped features are decoded to recover the SR frame by the spatial INR function. Differently, considering the large modality gap between the events and video frames, we accordingly propose a novel INR module to directly represent the 3D spatial-temporal coordinates from event data.

3. The Proposed Framework

Event Representation As event streams are sparse points, we first describe how to stack them into the fixed-size representations as inputs of our framework. Events are produced by detecting variations in the log intensity of each pixel. An

event $e = (x, y, t, p)$ is triggered and recorded when the logarithmic brightness change exceeds a certain threshold θ at pixel (x, y) . This process can be described as Eq.1, where $\Delta L = \log(I_t^l + n) - \log(I_{t-\Delta t}^l + n)$, n is noise, I_t^l and $I_{t-\Delta t}^l$ are intensity values in linear domain at timestamps t and $t - \Delta t$, respectively.

$$p = \begin{cases} +1, \Delta L > \theta \\ -1, \Delta L < -\theta \end{cases} \quad (1)$$

According to Eq.1, the relation between frames $I_{t_0}^l(x, y)$ and $I_{t_1}^l(x, y)$ at timestamps t_0 and t_1 can be formulated as Eq.2, where p is the polarity of event at pixel (x, y) .

$$I_{t_1}^l(x, y) = I_{t_0}^l(x, y) \times \exp\left(\theta \int_{t_0}^{t_1} p dt\right) \quad (2)$$

Events record the intensity changes with higher temporal resolution than frames, which is advantageous for VSR task [18]. We split events into M moment segments [42] with a shape of $H \times W \times M \times 2$ as the input to our framework. Each segment keeps events that take place within time window. The window size is set to be a small constant to preserve the temporal information of events.

3.1. Overview

The overall framework of our method is depicted in Fig.2, which consists of three parts: (I) spatial-temporal fusion (STF) branch, (II) temporal filter (TF) branch, and (III) spatial-temporal implicit representation (STIR) module. The input of our framework are spatially aligned

RGB video frames $V = \{V_0 \dots V_i \dots V_n\}$ and events $E \in \mathbb{R}^{H \times W \times M \times 2}$, where $H \times W$ denotes the frame size. The output of our framework is a super-resolved video frame $I_{s,t}^{SR}$ with the up-sampling scale s and the timestamp t . Note that the values of s and t can be freely adjusted. In practice, s is a real number greater than 1, and t can take the timestamps of all frames. The spatial resolution of the SR frame $I_{s,t}^{SR}$ is $sW \times sH$.

Our framework consists of three major components. The STF branch learns the holistic spatial-temporal information from events and RGB frames (Sec.3.2). Then, the TF branch unlocks more explicit motion information from events. It learns 2D event features from events nearing the queried timestamp t (Sec.3.3). Lastly, the STIR module decodes the features and recovers SR frames with arbitrary spatial resolutions (Sec.3.4). We now describe these components in detail in the following sections.

3.2. Spatial-Temporal Fusion (STF) Branch

This module aims to extract spatial and temporal information from events E and RGB frames V to obtain a global spatial-temporal representation F_{ST} . The representation is a 3D feature map of $H \times W \times T \times C$, where T is the temporal dimension, and C is the number of representation channels. The output F_{ST} of STF branch f_{STF} can be described as:

$$F_{ST} = f_{STF}(V, E) \quad (3)$$

Specifically, as depicted in Fig. 2(a), we first employ two 1×1 convolutional layers to obtain the initial frame feature map F_0^f and the event feature map F_0^e with same dimension. As stated in [13], shallow features preserve sharper details and local structural information while deeper features preserve more semantic information. Thus, we design fusion blocks to aggregate both the shallow and deep features.

Shallow feature fusion: As explored in [27], the residual architecture improves the model’s capacity for representation and lessens the gradient vanishing issue. For this reason, we first employ two high preserving blocks (HPB) as the basic feature extractors to extract two shallow feature maps F_1^f and F_1^e from the initial feature maps F_0^f and F_0^e , respectively. After that, F_1^f and F_1^e are concatenated and then fed into a transformer-based fusion model to obtain a fused feature map so as to bridge the modality gap. The intuition behind this is that transformer has a larger perception field [1, 25, 25, 31, 41], which can be potentially used to model the global spatial and temporal dependencies from the frames and events. Then, the fused feature map is split into two parts in the channel dimension and added to F_1^f and F_1^e as the input for deep feature fusion.

Deep feature fusion: After the shallow feature fusion, we again use two HPBs to extract deep features, which are then added and passed to the transformer-based fusion model to attain a 3D feature map F_{ST} . Through shallow and deep

feature fusion, we can better learn the temporal and spatial information from the events and frames.

3.3. Temporal Filter (TF) Branch

Through the STF branch, the spatial-temporal feature information from both the frames and events can be effectively learned. However, STF branch is insufficient to take full advantage of the high temporal resolution of event data. Therefore, we design the TF branch to explore more detailed motion information solely from events, which turns out to be effective in further enhancing the VSR performance, as demonstrated in our experiments (See Table 5).

Intuitively, we design the TF branch f_{TF} to capture the detailed motion information from events $E_{t,\Delta t}$ near the key frame at timestamp t , where Δt is a small time interval. TF branch first selects events from $t - \Delta t$ to $t + \Delta t$ (Fig. 2(c)). The selected events are interpolated and sent to three convolutional layers to learn the temporal features. Overall, the output F_T of TF branch f_{TF} can be described as:

$$F_T = f_{TF}(E_{t,\Delta t}) \quad (4)$$

In Sec.4.4, we show that STF branch can capture pixel intensities, while the TF branch can capture the motion details, e.g., edges and corners.

3.4. Spatial-Temporal Implicit Representation

In this section, our goal is to learn continuous INRs for VSR based on the spatial-temporal feature map F_{ST} and temporal feature map F_T . The INRs are then used to decode coordinates at time t with scale factor s into RGB values. In this paper, we introduce the Spatial-Temporal Implicit Representation (STIR) module to accomplish the spatial-temporal VSR, as shown in Eq.5. We employ a simple-yet-effective 3D feature sampling and trilinear interpolation scheme to upsample F_{ST} and F_T to a desired resolution. A decoder, parameterized as a multi-layer CNN, is used to convert the interpolated features into RGB values. Fig. 2(c) depicts the detailed design of STIR. The output SR frame $I_{t,s}^{SR}$ of the STIR f_{STIR} can be formulated as:

$$I_{t,s}^{SR} = f_{STIR}(F_{ST}, F_T), \forall s, t \quad (5)$$

3D Feature Sampling: Here, we aim at generating a coordinate to make a query in grid form F_{ST} . We uniformly sample a 3D coordinate grid, which can be expressed as $C_{t,s}$ with the dimension of $sH \times sW \times 3$. Formally, for any query q , the corresponding element p_q in the 3D coordinate grid $C_{t,s}$ can be described as $p_q = (x_q, y_q, t_q)$, where $x_q \in [0, H]$, $y_q \in [0, W]$, $t_q \in [t_s, t_e]$, t_s is the start time and t_e is the end time of input. For each coordinate $p_q = (x_q, y_q, t_q)$, we choose features of the nearest eight points around this coordinate in the 3D spatial-temporal feature F_{ST} for interpolation.

Feature Interpolation: Then, we compute the feature of a queried coordinate p_q by using 3D interpolation techniques,

such as trilinear interpolation. Inspired by the representation theory [37], complex signals in the low-dimensional space *e.g.*, images, can be transformed as linear representations in the high-dimensional space *e.g.*, features. From this theory, we can observe that the spatial-temporal feature F_{ST} is indeed the high-dimensional feature representation for the low-dimensional image. We use linear interpolation to obtain features at the queried coordinate p_q . In the experiments of Table 6, we have compared several interpolation methods, *e.g.*, nearest sampling, and the results show that linear interpolation shows the best performance.

In summary, for any scale s and timestamp t , the feature interpolation (*i.e.*, 3D to 2D) process can be formulated by Eq. 6, given the 3D coordinate grid $C_{t,s}$ and spatial-temporal feature F_{ST} .

$$F_{SF} = f_{sample}(F_{ST}, C_{t,s}) \quad (6)$$

Implicit Representation Decoding: Finally, the sampled 2D feature map F_{SF} and the temporal feature map F_T are added together and fed into the decoder. For simplicity and efficiency, we design three-layers CNN structure as the decoder. This is supported by the empirical experiment in Table 6, showing that a simple CNN block can achieve good results with low complexity.

3.5. Loss Function

We employ the *Charbonnier loss* [21] as our VSR supervision loss \mathcal{L}_{SR} between the ground truth (GT) HR frame $I_{t,s}^{HR}$ and the output SR frame $I_{t,s}^{SR}$ at timestamp t with up-sampling scale s , as shown in Eq. 7, where ϵ is $1e - 3$.

In training, the value range of t is all timestamps of key frames $\{t_0, t_1 \dots t_n\}$. s could be a real number in the range $[1.0, s_{max}]$, where s_{max} is the maximum up-sampling scale during training depending on the resolution of training data. The loss function \mathcal{L} is shown in Eq. 8. For example, when the resolution of input LR frame is set to be 128×128 and the GT HR frame is 1024×1024 , the s_{max} is 8.

$$\mathcal{L}_{SR} = \sqrt{(I_{t,s}^{SR} - I_{t,s}^{HR})^2 + \epsilon^2} \quad (7)$$

$$\mathcal{L} = \sum_{t \in \{t_0, t_1 \dots t_n\}} \left(\sum_{s \in [1.0, s_{max}]} \mathcal{L}_{SR}(I_{t,s}^{SR}, I_{t,s}^{HR}) \right) \quad (8)$$

3.6. Real-world Dataset Collection

Existing datasets, *e.g.*, CED [35], suffer from the limited resolution (346×260) and severe noise, as shown in Fig.7. Although CED dataset provides the active pixel sensor (APS) frames, they are in low quality because they are simply demosaiced by OpenCV [34] from RAW data. Therefore, collecting HR and high-quality datasets with spatially aligned frames and events is important to inspire more research for the event-guided VSR problem. In this paper, we collected a new real-world dataset, called ALPIX-

VSR, using a ALPIX-Eiger event camera¹. The camera outputs well aligned RGB frames and events. The RGB frames enjoy a resolution of 3264×2448 and are generated by a carefully designed image signal processor(ISP) from RAW data with the Quad Bayer pattern [10], and the events have a resolution with 1632×1224 .

Our ALPIX-VSR dataset consists of 26 video sequences with 5388 frames and well-aligned events in total. These sequences include diverse scenes, *e.g.*, streets, buildings, flowers, textures, and machines. To avoid motion blur and low-light noise, we collect the dataset in bright indoor and sunny outdoor scenes. *For more details about our real-world dataset, please refer to the supplementary material.*

4. Experiments

4.1. Experiments Setting

Implementation Details and Datasets: For all experiments, we use the Adam optimizer [20] with a learning rate of $1e - 4$ for CED dataset and $5e - 5$ for our ALPIX-VSR dataset. We train our framework for 100 epochs with a batch size of 2 using two NVIDIA RTX A30 GPU cards.

Evaluation Metrics: We statistically assess the effectiveness of our approach using the peak-signal-to-noise ratio (PSNR) and structural similarity (SSIM).

Datasets: We use the CED [35] and the ALPIX-VSR dataset for experiments. **1) CED Dataset.** It includes a collection of color events and video sequences in many scenes, *e.g.*, indoor, outdoor, driving, human, calibration. The resolution of the frames and events is 346×260 . We follow the setting of E-VSR [18] to preprocess this dataset. Note that the RGB frames provided by CED are obtained by demosaicing [34] from raw frames and suffer from severe noise. **2) ALPIX-VSR Dataset.** We select 20 videos for training and 6 videos for testing. The training and testing sets include 4212 and 1176 frames with aligned events, respectively. Note that we apply data augmentation strategy, such as random crop, to ALPIX-VSR dataset for all compared methods to avoid memory overflow during training.

4.2. Comparison with SoTA Methods

We compare our method with six SoTA methods under three VSR settings: (I) one SoTA event-guided, fixed-scale VSR method E-VSR [18] (II) one SoTA method of frame-based arbitrary-scale VSR method VideoINR [8] (III) five SoTA methods of frame-based, fixed-scale VSR methods: BasicVSR++ [6], RBPN [14], SOF [45], TDAN [40], DUF [19]. We report $\times 2$ and $\times 4$ super-resolution results of our method and all 6 comparison methods on CED dataset. We also compare our method with E-VSR and BasicVSR++ on our ALPIX-VSR dataset. Moreover, we compare our method with VideoINR on out-of-distribution

¹<https://www.alpsentek.com/product>

Clip Name	DUF [19]*	TDAN [40]	SOF [45]	RBPN [14]	VideoINR [8]*	E-VSR [18]	Ours
people_dynamic_wave	32.02 / 0.9333	35.83 / 0.9540	33.32 / 0.9360	40.07 / 0.9868	27.47 / 0.8229	41.08 / 0.9891	38.78 / 0.9794
indoors_foosball_2	30.55 / 0.9262	32.12 / 0.9339	30.86 / 0.9253	34.15 / 0.9739	26.03 / 0.7766	34.77 / 0.9775	38.68 / 0.9750
simple_wires_2	30.08 / 0.9387	31.57 / 0.9466	30.12 / 0.9326	33.83 / 0.9739	26.77 / 0.8321	34.44 / 0.9773	38.67 / 0.9815
people_dynamic_dancing	31.64 / 0.9369	35.73 / 0.9566	32.93 / 0.9388	39.56 / 0.9869	27.36 / 0.8202	40.49 / 0.9891	39.06 / 0.9798
people_dynamic_jumping	31.57 / 0.9334	35.42 / 0.9536	32.79 / 0.9347	39.44 / 0.9859	27.24 / 0.8183	40.32 / 0.9880	38.93 / 0.9792
simple_fruit_fast	37.46 / 0.9442	37.75 / 0.9440	37.22 / 0.9390	40.33 / 0.9782	27.21 / 0.8456	40.80 / 0.9801	41.96 / 0.9821
outdoor_jumping_infrared_2	25.33 / 0.8162	28.91 / 0.9062	26.67 / 0.8746	30.36 / 0.9648	26.88 / 0.8226	30.70 / 0.9698	38.03 / 0.9755
simple_carpet_fast	31.43 / 0.8811	32.54 / 0.9006	31.83 / 0.8774	34.91 / 0.9502	24.21 / 0.5909	35.16 / 0.9536	36.14 / 0.9635
people_dynamic_armroll	31.38 / 0.9311	35.55 / 0.9541	32.79 / 0.9345	40.05 / 0.9878	27.26 / 0.8193	41.00 / 0.9898	38.84 / 0.9787
indoors_kitchen_2	29.92 / 0.9273	30.67 / 0.9323	29.61 / 0.9192	31.51 / 0.9551	26.44 / 0.7502	31.79 / 0.9586	37.68 / 0.9726
people_dynamic_sitting	30.62 / 0.9331	35.09 / 0.9561	32.13 / 0.9367	39.03 / 0.9862	27.63 / 0.8230	39.97 / 0.9884	38.86 / 0.9810
average PSNR/SSIM	31.09 / 0.9183	33.74 / 0.9398	31.84 / 0.9226	36.66 / 0.9754	26.77 / 0.7938	37.32 / 0.9783	38.69 / 0.9771

Table 1. Quantitative results (PSNR/SSIM) of the proposed our framework and other methods on the CED dataset for $\times 2$. Because the official training code is not available, * denoted values were acquired from the pre-trained model that the authors have released.

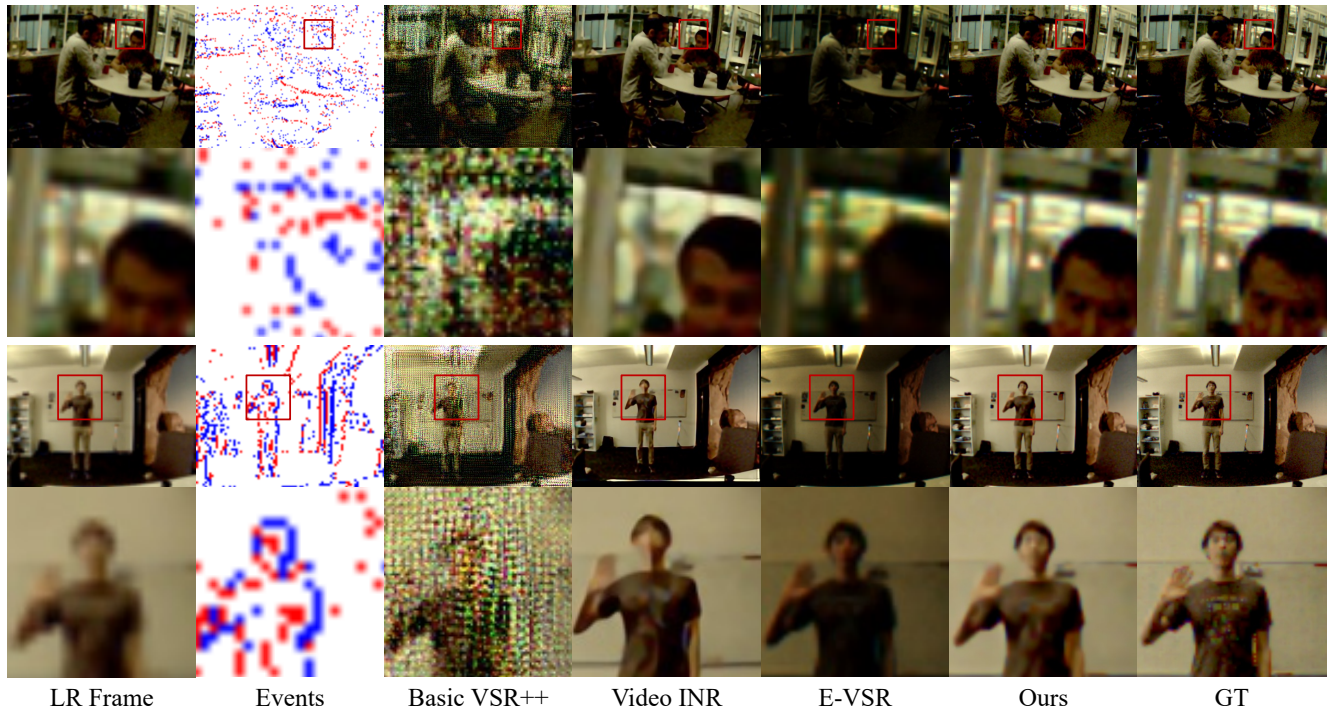


Figure 3. Visual results of $\times 4$ VSR on the CED dataset. Our method recovers more details, *e.g.*, textual, edge than the SoTA event-guided VSR method E-VSR [18] and the random up-sampling VSR method Video INR [8].

scales to demonstrate our method’s ability for arbitrary-scale VSR. Note that E-VSR only supports $\times 2$ and $\times 4$ SR, and BasicVSR++ only supports $\times 4$ SR.

Evaluation on CED Dataset Table 1 and Table 2 present the quantitative results for $\times 2$ and $\times 4$ VSR, respectively. Our model clearly outperforms other methods in terms of PSNR, and shows a comparable performance in SSIM with E-VSR. The qualitative results in Fig. 3 demonstrate that our model is capable of recovering fine details, like sharp edges and detailed textures. We can see that event-guided methods (E-VSR and our method) yield better performances than the frame-based counterparts, showing the complementary effects of event data for VSR. Furthermore,

VideoINR performs noticeably worse than other methods, indicating that frame-based implicit neural representations struggle with the low-resolution and high-noise input. Contrarily, our approach benefits from additional event information and generates satisfactory INRs.

Table 2 shows quantitative results in $\times 4$ SR on the CED dataset, where our model achieves SoTA performance while remaining lightweight and efficient. Specifically, our model only accounts for one-200th of the E-VSR. BasicVSR++, the SoTA method of frame-based VSR, fails to perform well on CED dataset as the upsampling scale increases, which demonstrates its less robustness on the highly noisy dataset.

Evaluation on the ALPIX-VSR Dataset The quantitative

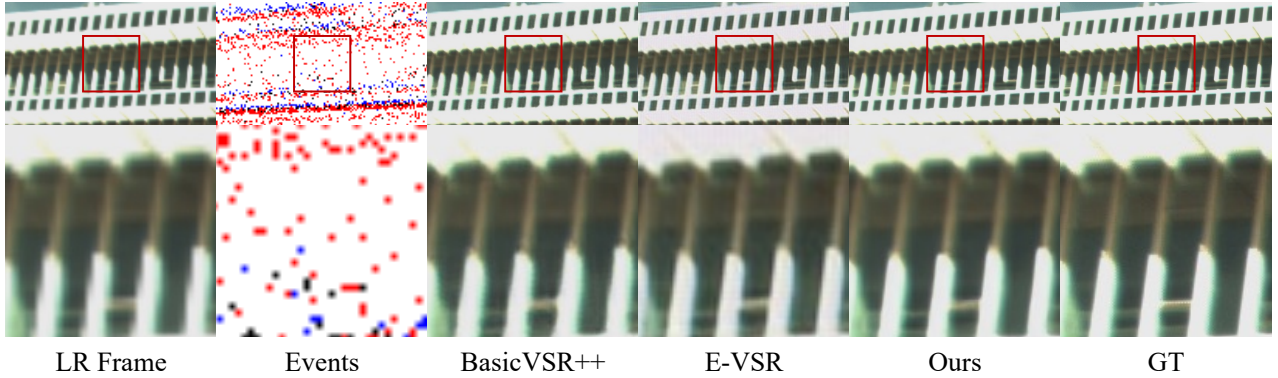


Figure 4. Results of $\times 4$ VSR on the ALPIX-VSR dataset.

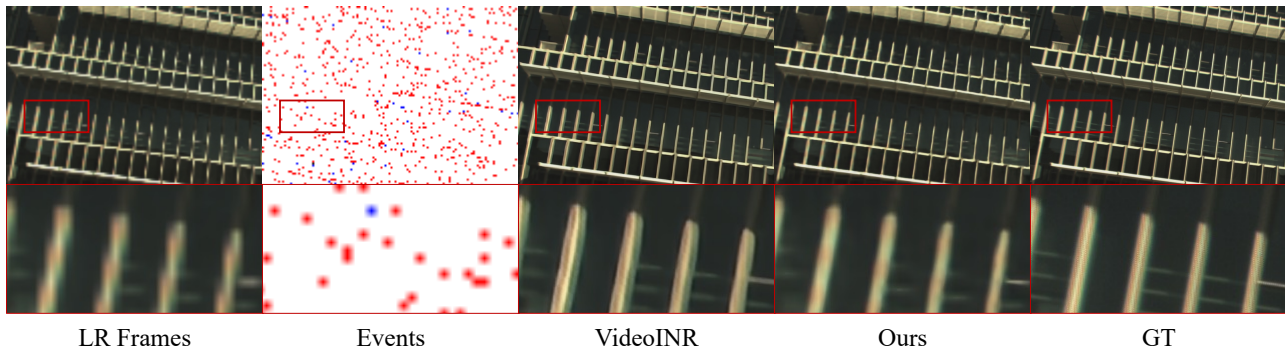


Figure 5. Results of $\times 8$ VSR on the ALPIX-VSR dataset.

Methods	Model Size(M)	PSNR	SSIM
DUF [19]	1.90	24.43	0.8177
TDAN [40]	1.97	27.88	0.8231
SOF [45]	1.00	27.00	0.8050
RBPN [14]	12.18	29.80	0.8975
BasicVSR++ [6]	7.30	14.76	0.1641
VideoINR [8]*	11.31	25.53	0.7871
E-VSR [18]	412.42	30.15	0.9052
Ours	2.45	31.12	0.9211

Table 2. **Quantitative results on CED dataset** for $\times 4$. * denotes the values obtained from the official pre-trained models.

	Methods	PSNR	SSIM
$\times 2$	E-VSR	36.10	0.9761
	Ours	38.25	0.9822
$\times 4$	E-VSR	32.54	0.9163
	BasicVSR++	35.30	0.9353
	Ours	37.12	0.9503
$\times 6$	VideoINR*	31.15	0.9084
	Ours	31.85	0.9267
$\times 8$	VideoINR*	28.11	0.8625
	Ours	28.53	0.8901

Table 3. **Quantitative comparison** (PSNR/SSIM) of our methods and other methods on the ALPIX-VSR dataset. * denotes the values obtained from the official pre-trained models.

and qualitative results are shown in Table 3 and Fig. 5. We present comparison results of our method with BasicVSR++ and E-VSR on our collected real-world dataset.

4.3. Random Scale Up-sampling

Results of random scale upsampling are shown in Table 3. We upsample the video frames to $\times 2$, $\times 4$, $\times 6$

	$\times 1.8$	$\times 2.6$	$\times 5.6$
Ours	39.2508 / 0.9803	37.3408 / 0.9589	31.2549 / 0.9135
	$\times 6.6$	$\times 7.1$	$\times 7.8$
Ours	28.3182 / 0.8772	28.3188 / 0.87762	28.3198 / 0.8783

Table 4. **Quantitative results**(PSNR/SSIM) of random-scale comparison on the **ALPIX-VSR dataset**.

, $\times 8$, and compare with the SoTA models, E-VSR and BasicVSR++. Though these two models are strictly constrained to a specific upsampling scale, our method presents superior performance on these settings.

We also compare our method with VideoINR, the SoTA random-scale VSR method quantitatively in $\times 6$, $\times 8$. Results in Table 3 show that our model surpasses VideoINR in all evaluated scales of $\times 6$, $\times 8$. Furthermore, to evaluate the performance of our method on arbitrary scales, we conduct experiments of 6 random-chosen float scales. The results are shown in Table 4 and indicate our model’s robustness across arbitrary random scales. It is easy to find that values of performance, *e.g.*, PSNR, SSIM, and upsampling scales are not strictly monotonic.

4.4. Ablation Studies and Discussion

The following ablation experiments investigate the importance of each of our proposed modules. As it takes more than 120 hours to train a model on the complete CED dataset, we uniformly select 1/5 of CED as the dataset for ablation experiments.

	TF Branch	Shallow Feature Fusion	Model Size(M)	PSNR	SSIM
1	w	w	2.4513	38.14	0.9820
2	w/o	w	2.4482	37.96	0.9812
3	w	w/o	1.7360	37.76	0.9729

Table 5. Ablation of the TF branch and shallow feature fusion.

	Interpolation	Decoder	LF Channels	PSNR	SSIM
1	Linear	CNN	16	38.14	0.9820
2	Nearest	CNN	16	28.91	0.9131
3	Linear	SIREN	16	10.30	0.2997
4	Linear	MLP	16	37.94	0.9811
5	Linear	CNN	8	36.82	0.9763
6	Linear	CNN	24	38.25	0.9825

Table 6. Impacts of interpolation methods, decoder designs and channel size of the spatial-temporal features F_{ST} .

Efficiency of TF branch and Shallow Feature Fusion:

Table 5 validates the contribution of TF branch and shallow feature fusion in STF branch. The removal of TF branch reduces both PSNR and SSIM scores, where PSNR drops by 0.18dB. In Fig. 6, we use PCA [11] to visualize the output of TF branch. As can be seen, F_T focuses on edge and corner information, which is very helpful for VSR, especially texture recovery. We also find that removing shallow feature fusion results in a large performance drop (PSNR drops by 0.38 dB and SSIM drops by nearly 0.1 dB). This finding indicates that fusion on shallow features is critical since shallow features carry rich local structure information which may be missing in deep features.

Feature Interpolation: We apply a feature-based interpolation strategy, *i.e.* interpolating features near a coordinate and sending the interpolated feature into the decoder to reconstruct HR frames. Such strategy has been studied in a prior work on implicit neural representation learning for 3D objects [38] and shown to be able to recover clearer details and sharper edges. We further study the influence of interpolation methods, as shown in Table 6. The trilinear manner of interpolation yields better PSNR and SSIM scores compared with the nearest interpolation.

Feature Decoder: We also compare the performance of different decoder designs, including MLP, SIREN [36] and CNN. Table 6 shows that decoding with CNN has the best performance among these methods, while non-convergence occurs with SIREN. We argue the reason CNN performs well is that decoding in STIR is only a dimensional reduction process, so no complicated design is required.

Robustness to Noise: In comparison to BasicVSR++, our method not only performs SR but also removes noise on the CED dataset, as shown in Fig.7. Note that BasicVSR++ is a SoTA frame-based method in VSR task. We analyze that the poor performance of BasicVSR++ on CED is caused by the dependence for only frame modality and excessive emphasis on the frame’s high-frequency information. High frequencies are often present in the image as edges, cor-

	PSNR	SSIM
3 to 1	38.14	0.9820
5 to 3	38.04	0.9818

Table 7. Ablation for the number of input and output frames.

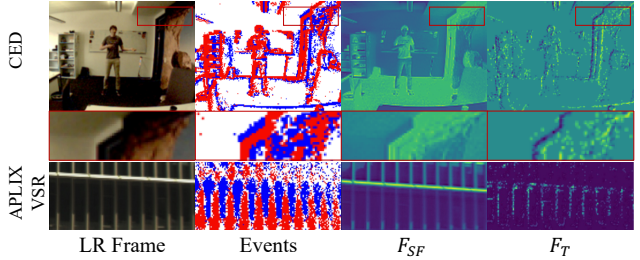


Figure 6. Feature visualization of F_{SF} (Eq. 6) and F_T (Eq. 4).

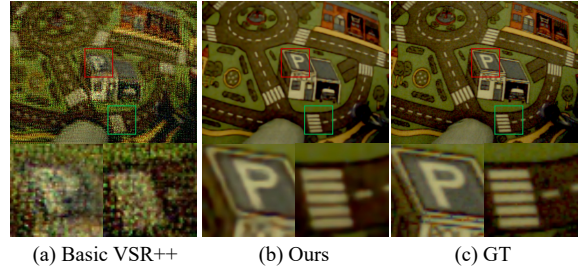


Figure 7. Comparison of noise removal capacity of BasicVSR++ [6] with our methods, with respect to HR GT.

ners and noise. Therefore BasicVSR++ is easily affected by serious noise. In comparison to BasicVSR++, our framework has more robustness. Benefiting from the guidance of events, *e.g.*, edges, corners, our method can effectively reduce the adverse effects of noise on frames.

5. Conclusion

In this paper, we proposed a novel framework which jointly learns INRs from RGB frames and events, and enables arbitrary scale VSR. Our method effectively uses high temporal resolution property of events to complement RGB frames with STF and TF. A simple yet effective STIR is used to recover frames at arbitrary scales. Extensive experiments on two real-world datasets validate our method enjoys better performance over current related SoTA methods with significantly lower model size.

Limitation and Future Work: In this work we tackle the problem of spatial VSR. Since our work learns INRs of videos, it naturally enables frame interpolation, which will be explored in our future work. We hope our work can shed light on more research about enhancing implicit neural representations with non-photorealistic data.

6. Acknowledgment

This work was supported by the Research Project Fund of AlpsenTek and the National Natural Science Foundation of China (NSFC) under Grant No. NSFC22FYT45.

References

- [1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021.
- [2] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):933–948, 2021.
- [3] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4778–4787, 2017.
- [4] Daniel Canedo and António JR Neves. Facial expression recognition using computer vision: A systematic review. *Applied Sciences*, 9(21):4678, 2019.
- [5] Kelvin C.K. Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021.
- [6] Kelvin C.K. Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. 2022.
- [7] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021.
- [8] Zeyuan Chen, Yinbo Chen, Jingwen Liu, Xingqian Xu, Vidit Goel, Zhangyang Wang, Humphrey Shi, and Xiaolong Wang. Videoinr: Learning video implicit neural representation for continuous space-time super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2047–2057, 2022.
- [9] Zeyuan Chen, Yinbo Chen, Jingwen Liu, Xingqian Xu, Vidit Goel, Zhangyang Wang, Humphrey Shi, and Xiaolong Wang. Videoinr: Learning video implicit neural representation for continuous space-time super-resolution. 2022.
- [10] Minhyeok Cho, Haechang Lee, Hyunwoo Je, Kijeong Kim, Dongil Ryu, Jinsu Kim, Jonghyun Bae, and Albert No. Pynet-qxq: A distilled pynet for qxq bayer pattern demosaicing in cmos image sensor. *arXiv preprint arXiv:2203.04314*, 2022.
- [11] Andreas Daffertshofer, Claudine JC Lamoth, Onno G Meijer, and Peter J Beek. Pca in studying coordination and variability: a tutorial. *Clinical biomechanics*, 19(4):415–428, 2004.
- [12] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conrath, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.
- [13] Jin Han, Yixin Yang, Chu Zhou, Chao Xu, and Boxin Shi. Evintsr-net: Event guided multiple latent frames reconstruction and super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4882–4891, 2021.
- [14] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3892–3901, 2019.
- [15] Earnest Paul Ijjina, Dhananjai Chand, Savvasachi Gupta, and K Goutham. Computer vision-based accident detection in traffic surveillance. In *2019 10th International conference on computing, communication and networking technologies (ICCCNT)*, pages 1–6. IEEE, 2019.
- [16] Takashi Isobe, Songjiang Li, Xu Jia, Shanxin Yuan, Gregory Slabaugh, Chunjing Xu, Ya-Li Li, Shengjin Wang, and Qi Tian. Video super-resolution with temporal group attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8008–8017, 2020.
- [17] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022.
- [18] Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Turning frequency to resolution: Video super-resolution via event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7772–7781, 2021.
- [19] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3224–3232, 2018.
- [20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [21] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2599–2613, 2018.
- [22] Sheng Li, Fengxiang He, Bo Du, Lefei Zhang, Yonghao Xu, and Dacheng Tao. Fast spatio-temporal residual network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10522–10531, 2019.
- [23] Ding Liu, Zhaowen Wang, Yuchen Fan, Xianming Liu, Zhangyang Wang, Shiyu Chang, Xinchao Wang, and Thomas S Huang. Learning temporal dynamics for video super-resolution: A deep learning approach. *IEEE Transactions on Image Processing*, 27(7):3432–3445, 2018.
- [24] Hongying Liu, Zubo Ruan, Peng Zhao, Chao Dong, Fanhua Shang, Yuanyuan Liu, Linlin Yang, and Radu Timofte. Video super-resolution based on deep learning: a comprehensive survey. In *Artif Intell Rev.*, 2022.

- [25] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022.
- [26] Yunfan Lu, Yiqi Lin, Hao Wu, Yunhao Luo, Xu Zheng, and Lin Wang. All one needs to know about priors for deep image restoration and enhancement: A survey. *arXiv preprint arXiv:2206.02070*, 2022.
- [27] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tiejong Zeng. Transformer for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 457–466, 2022.
- [28] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- [29] Mohammad Mostafavi, Yeongwoo Nam, Jonghyun Choi, and Kuk-Jin Yoon. E2sri: Learning to super-resolve intensity images from events. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):6890–6909, 2021.
- [30] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
- [31] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3163–3172, 2021.
- [32] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.
- [33] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.
- [34] Vadim Pisarevsky. Introduction to opencv. *Agenda*, 42:433–434, 2010.
- [35] Cedric Scheerlinck, Henri Rebecq, Timo Stoffregen, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Ced: Color event camera dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
- [36] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.
- [37] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.
- [38] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022.
- [39] Jiaxiang Tang, Xiaokang Chen, and Gang Zeng. Joint implicit image function for guided depth super-resolution. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4390–4399, 2021.
- [40] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3360–3369, 2020.
- [41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [42] Bishan Wang, Jingwei He, Lei Yu, Gui-Song Xia, and Wen Yang. Event enhanced high-quality image recovery. In *European Conference on Computer Vision*, pages 155–171. Springer, 2020.
- [43] Bishan Wang, Jingwei He, Lei Yu, Gui-Song Xia, and Wen Yang. Event enhanced high-quality image recovery. In *European Conference on Computer Vision*, pages 155–171. Springer, 2020.
- [44] Lin Wang, Yujeong Chae, Sung-Hoon Yoon, Tae-Kyun Kim, and Kuk-Jin Yoon. Evidistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 608–619, 2021.
- [45] Longguang Wang, Yulan Guo, Li Liu, Zaiping Lin, Xinpu Deng, and Wei An. Deep video super-resolution using hr optical flow estimation. *IEEE Transactions on Image Processing*, 29:4323–4336, 2020.
- [46] Lin Wang, Tae-Kyun Kim, and Kuk-Jin Yoon. Eventsr: From asynchronous events to image reconstruction, restoration, and super-resolution via end-to-end adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8315–8325, 2020.
- [47] Wei Wang, Haochen Zhang, Zehuan Yuan, and Changhu Wang. Unsupervised real-world super-resolution: A domain adaptation perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4318–4327, 2021.
- [48] Xintao Wang, Ke Yu, Kelvin C.K. Chan, Chao Dong, and Chen Change Loy. Basicsr. <https://github.com/xinntao/BasicSR>, 2020.
- [49] Xi Yang, Wangmeng Xiang, Hui Zeng, and Lei Zhang. Real-world video super-resolution: A benchmark dataset and a decomposition based learning scheme. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4781–4790, 2021.
- [50] Huanjing Yue, Zhiming Zhang, and Jingyu Yang. Realrawvsr: Real-world raw video super-resolution with a benchmark dataset. *arXiv preprint arXiv:2209.12475*, 2022.
- [51] Xu Zheng, Yexin Liu, Yunfan Lu, Tongyan Hua, Tianbo Pan, Weiming Zhang, Dacheng Tao, and Lin Wang. Deep learning for event-based vision: A comprehensive survey and benchmarks. *arXiv preprint arXiv:2302.08890*, 2023.

- [52] Pengyuan Zhou, Jinjing Zhu, Yiting Wang, Yunfan Lu, Zixiang Wei, Haolin Shi, Yuchen Ding, Yu Gao, Qinglong Huang, Yan Shi, et al. Vetaverse: Technologies, applications, and visions toward the intersection of metaverse, vehicles, and transportation systems. *arXiv preprint arXiv:2210.15109*, 2022.
- [53] Yunhao Zou, Yinqiang Zheng, Tsuyoshi Takatani, and Ying Fu. Learning to reconstruct high speed and high dynamic range videos from events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2024–2033, 2021.