Event Stream Super-Resolution via Spatiotemporal Constraint Learning

Siqi Li[1], Yutong Feng[1], Yipeng Li[2], Yu Jiang[3], Changqing Zou[4], Yue Gao[1]*
[1]BNRist, THUICBS, KLISS, School of Software, Tsinghua University, China
[2]Department of Automation, Tsinghua University, China
[3]College of Computer Science and Technology, Jilin University, China
[4]Huawei Technologies Canada Co., Ltd
{lsq19, fyt19}@mails.tsinghua.edu.cn, aaronzou1125@gmail.com, jiangyu2011@jlu.edu.cn, {liep, gaoyue}@tsinghua.edu.cn

Abstract

Event cameras are bio-inspired sensors that respond to brightness changes asynchronously and output in the form of event streams instead of frame-based images. They own outstanding advantages compared with traditional cameras: higher temporal resolution, higher dynamic range, and lower power consumption. However, the spatial resolution of existing event cameras is insufficient and challenging to be enhanced at the hardware level while maintaining the asynchronous philosophy of circuit design. Therefore, it is imperative to explore the algorithm of event stream super-resolution, which is a non-trivial task due to the sparsity and strong spatio-temporal correlation of the events from an event camera. In this paper, we propose an end-to-end framework based on spiking neural network for event stream super-resolution, which can generate high-resolution (HR) event stream from the input low-resolution (LR) event stream. A spatiotemporal constraint learning mechanism is proposed to learn the spatial and temporal distributions of the event stream simultaneously. We validate our method on four large-scale datasets and the results show that our method achieves state-of-the-art performance. The satisfying results on two downstream applications, i.e. object classification and image reconstruction, further demonstrate the usability of our method. To prove the application potential of our method, we deploy it on a mobile platform. The high-quality HR event stream generated by our real-time system demonstrates the effectiveness and efficiency of our method.

1. Introduction

Event cameras, *e.g.* Dynamic Vision Sensor (DVS) [6], are bio-inspired vision sensors. Different from traditional frame-based sensors that capture images at a fixed rate, the event cameras respond to the pixel-wise brightness changes of the scene asynchronously. Specifically, let $I(x, y, t)$ be the brightness intensity of the spatial coordinates (x, y) at time t. When the change of its logarithm reaches a threshold value C, *i.e.* $|\Delta \log I(x, y, t)| > C$, an ***event*** will be triggered [34]. Each event is denoted as $e_i = (x_i, y_i, t_i, p_i)$, where $p_i \in \{1, -1\}$ is the polarity, indicating whether the brightness is increased or decreased. Therefore, the output of the event camera is a tuple list called ***event stream***, denoted as $\mathcal{E} = \{e_i\}_{i=1}^{N}$. Figure 1 compares the output of an event camera and a traditional frame-based camera.

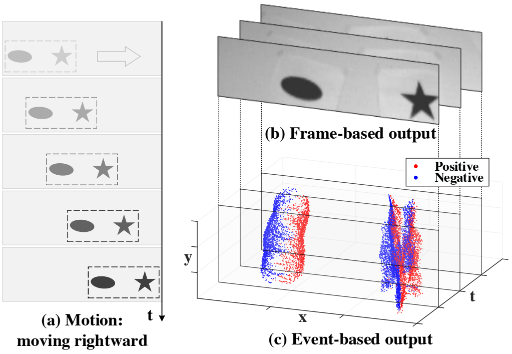


Figure 1. A comparison of the outputs of an event camera and a frame-based camera while capturing (a) an ellipse and a star moving rightward. Different from the frame-based camera (b) that captures the scene at a fixed rate, the event camera asynchronously responds to the brightness changes of the scene and outputs a spatiotemporal event stream (c), where the red and blue points represent the positive and negative events, respectively, *i.e.* the increase and decrease of the brightness. Figure inspired by [32].

Compared with traditional cameras, event cameras own many outstanding properties: high dynamic range (140 dB *vs*. 60 dB for traditional cameras), high temporal resolution, and free from motion blur, which make them widely used in many applications, *e.g.* object recognition [31, 41], gesture recognition [1, 2], optical flow estimation [47, 33], high frame rate video reconstruction [32, 36, 16], visual-inertial odometry [46, 24], and 3D reconstruction [17, 8].

However, the spatial resolution of existing event cameras is insufficient. Many influential works are based on the 128 × 128 pixels DVS128 [34] or 240 × 180 pixels

*Corresponding author

DAVIS240 [6]. If we aim at improving the spatial resolution at the hardware level, *i.e*. reducing the pixel size, the asynchronous circuit design philosophy will tough to be maintained [13]. Therefore, under the difficulty of increasing the spatial resolution with hardware design, it is imperative to explore the algorithm of event stream super-resolution.

The concerned event stream super-resolution is to increase the spatial resolution of the input low-resolution (LR) event stream without changing the data modality, *i.e*. directly generating high-resolution (HR) event stream instead of HR grid-based representation of the event stream, *e.g*. intensity image [27] and event frame [42]. This is because the grid-based representations of the event stream are incomplete with a lower temporal resolution, leading to a deficiency of the temporal information of the original event stream. Some high-performance algorithms [39, 25] also directly deal with the event stream instead of the grid-based representation of it, and achieve a higher computing efficiency. So we claim that it is more significant to generate HR event streams instead of its grid-based representation.

The event stream is a type of spatiotemporal event cloud containing high-precision timestamp information compared with traditional frame-based vision. The core question of the event stream super-resolution is how to effectively describe the spatial and temporal distribution of the event stream. To the best of our knowledge, there are few studies on this task. Li *et al*. [22] use a sparse signal representation based method [44] to learn the event stream's spatial distribution and use a non-homogeneous Poisson processes to simulate the event stream at each pixel. The timestamp of each event in the output HR event stream is predicted by a sampling method according to the Poisson intensity, which causes a fatal error in temporal dimension due to the lack of effective supervised learning on the temporal distribution.

In this paper, we propose an end-to-end event stream super-resolution method based on the spiking neural network, which could better preserve the temporal component of the event stream. A spatiotemporal constraint learning mechanism is proposed to describe both the spatial and temporal distribution of the event stream. The proposed method could generate the HR event stream by precisely predicting each output event's timestamp, instead of predicting a grid-based representation. We evaluate our method on four large-scale datasets, *i.e*. N-MNIST [30], CIFAR10-DVS [23], ASL-DVS [2], and Event Camera Dataset [28]. Experimental results show that our method outperforms the state-of-the-art method by a large margin. For further testing the quality of super-resolution results, we evaluate our generated HR event streams on two downstream applications, *i.e*. object classification and image reconstruction. The marvelous performance shows the usability of the proposed method. We also deploy our method on a mobile system and show that our method can perform high-quality event stream super-resolution in real-time.

Our main contributions are summarized as follows:

- An end-to-end SNN-based model is proposed for the event stream super-resolution task. Compared with the previous method, a spatiotemporal constraint learning mechanism is proposed to learn the temporal and spatial distribution of the event stream simultaneously.
- Satisfying performances on two downstream applications with the generated HR event streams as input, *i.e*. object classification and image reconstruction, demonstrate the usability of the proposed method.
- An embedded deployment of our method is implemented proving that the proposed method can generate high-quality HR event streams in real-time, which shows the potential of deploying the proposed method on mobile systems.

2. Related Work

2.1. Spiking Neural Network

Spiking Neural Networks (SNNs) are bio-inspired Artificial Neural Networks (ANNs), using biomimetic spiking neurons as computing units. Different from the neurons in traditional ANNs with a static, continuous-valued, and differentiable activation function, the spiking neurons in SNNs use discrete spike trains, *i.e*. temporal series of spikes, to transmit information, which is similar to biological neurons. An internal variable, called membrane potential, is defined to describe the hidden state of the spiking neuron. When the membrane potential exceeds a specified threshold, an output spike will be generated. Immediately after the spike, the membrane potential will be inhibited. Such a self-suppression mechanism is called **refractory response**. Each input spike train will generate a **Post Synaptic Potential** (PSP). The membrane potential of the neuron is the sum of all PSPs and the refractory responses. There are various spiking neural models in neuroscience, including Leaky Integrate and Fire neuron [11], Hodgkin-Huxley neuron [14], Izhikevich model [15], and Spike Response Model [10].

One of the main limitations of SNNs is that the spike function that triggers output spikes in the spiking neuron model is non-differentiable, which restricts the utilization of backpropagation for training. Nevertheless, there have been some efforts in recent years to design specific backpropagation for SNNs, *e.g*. methods with event-based backpropagation [4, 5, 12], methods that use a differentiable continuous function to approximate the spike function [45, 29], methods that reassign error along temporal dimension [43, 40].

2.2. Event Stream Super-resolution

The event stream super-resolution task is complicated and challenging due to the special spatiotemporal property

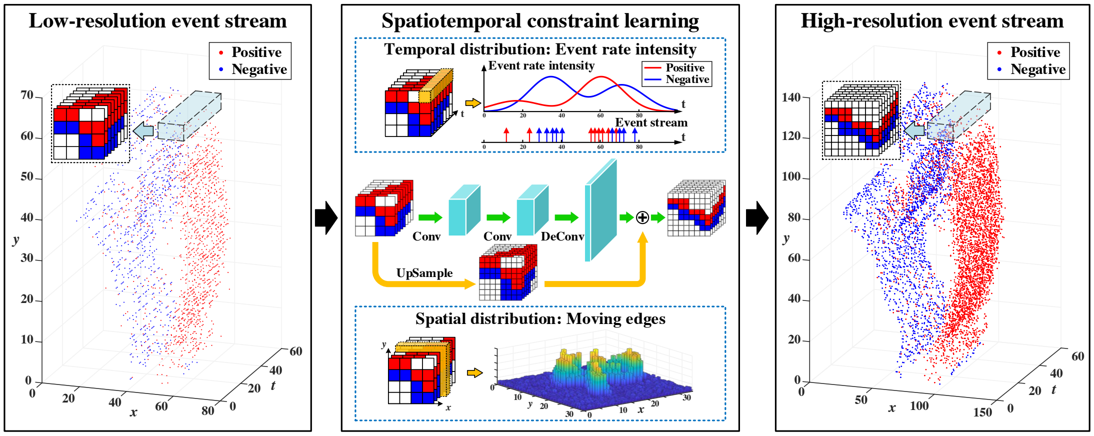

Low-resolution event stream
Positive
Negative
Spatiotemporal constraint learning
Temporal distribution: Event rate intensity
Event rate intensity
Event stream
Conv
Conv
DeConv
UpSample
Spatial distribution: Moving edges
High-resolution event stream

Figure 2. Pipeline of the proposed method. Taking the LR event stream obtained from the event camera as input, the proposed method generates the HR event stream by predicting the timestamp of each output event. A spatiotemporal constraint learning mechanism is applied to simultaneously learn the spatial and temporal distribution of the event stream, which can effectively boost the performance.

of the event stream. To the best of our knowledge, the first and only work on this task is [22], which uses an event count map (ECM) to describe the event stream's spatial distribution and a nonhomogeneous Poisson process to simulate the event stream at each pixel. Specifically, in the spatial dimension, the LR ECM is obtained by counting the number of events in the input LR event stream at each pixel, and the HR ECM is predicted using an image super-resolution method [44]. In the temporal dimension, the intensity of the nonhomogeneous Poisson process at each pixel could be computed from the input LR event stream, and then a fixed convolutional kernel is applied to get the Poisson intensity of the predicted HR event stream. Finally, a thinning-based sample algorithm [21] is applied to obtain the timestamps of output events, according to the predicted HR ECM and Poisson intensity. Though the event stream's spatial distribution is well described in this work, the timestamp of each output event is obtained by a sampling-based method in the temporal dimension, resulting in low accuracy of the event stream super-resolution task.

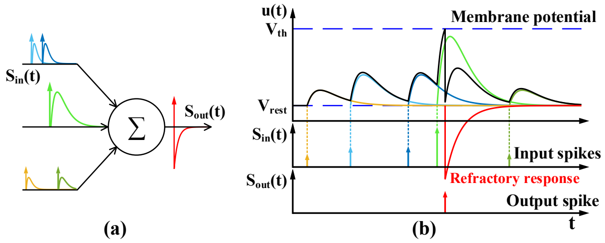


Figure 3. Illustration of spiking neuron dynamics described by the Spike Response Model. (a) A spiking neuron modeled by SRM. (b) The neuron is stimulated by input spikes with different synaptic weights, leading to an increment of the membrane potential $u(t)$. As soon as the membrane potential exceeds the threshold $\mathrm{V_{th}}$, an output spike will be emitted, and the membrane potential will decrease due to the refractory response. See Section 3.1 for more detail. (Best viewed in color.)

3. Method

We start with an overview of our pipeline for event stream super-resolution, as illustrated in Figure 2. Our method takes the LR event stream as input and treats each event as an input spike. A convolutional SNN based on spatiotemporal constraint learning is proposed to learn the spatial and temporal distribution of the event stream simultaneously, and the HR event stream is generated with precisely predicted timestamps of output events.

3.1. Spiking Neuron and SNN Model

In this section, we introduce the spiking neural model, *i.e.* the Spike Response Model (SRM) [10], and the SNN model used in our method. Denote $\boldsymbol{s}_{in}(t)$ and $s_{out}(t)$ as the input and output of a neuron in SRM, consisting of n and 1 spike trains, respectively. An internal variable $u(t)$, named membrane potential, is used to maintain the hidden state of the neuron and generate an output strike once it exceeds a specified threshold $\mathrm{V_{th}}$. And $u(t)$ is defined as the sum of all the Post Synaptic Potentials (PSPs) and the refractory responses. Among them, the PSP is obtained by convolving the input $\boldsymbol{s}_{in}(t)$ with a spike response kernel $\varepsilon(\cdot)$, and then multiplied by a corresponding synapse weight $\boldsymbol{w}$. Similarly, the refractory response is defined as the convolution of $s_{out}(t)$ and a refractory response kernel $\gamma(\cdot)$. Therefore, the membrane potential $u(t)$ is calculated as:

$$u(t) = (\gamma(t) * s_{out}(t)) + \sum_i w^i \left(\varepsilon(t) * s_{in}^i(t)\right), \quad (1)$$

where $s_{in}^i(t) = \sum_k \delta\left(t - t_k^i\right)$ is the i-th input spike train and t_k^i is the timestamp of the t-th spike.

Figure 3 shows an illustration of the spiking neuron dynamics described by SRM. Each input spike train $s_{in}^i(t)$ has a corresponding synapse weight w^i, which is visualized as the length of spikes in the figure. The PSP generated by each spike is shown as a curve in the corresponding color. The membrane potential $u\left(t\right)$, colored in black, is the sum of all PSPs and refractory responses. When $u\left(t\right)$ exceeds the threshold $\mathrm{V_{th}}$, an output spike is emitted, and a refractory response (colored in red) is triggered to reduce the current membrane potential.

Consider a feed-forward SNN architecture with L layers and N_l neurons in the l-th layer. The forward propagation can be defined as:

$$
\begin{aligned}
\boldsymbol{u}^{(l+1)}\left(t\right) &= \mathbf{W}^{(l)}\left(\varepsilon\left(t\right) * \boldsymbol{s}^{(l)}\left(t\right)\right) + \left(\gamma\left(t\right) * \boldsymbol{s}^{(l+1)}\left(t\right)\right) \\
\boldsymbol{s}^{(l+1)}\left(t\right) &= \sum_{t_k \in \left\{t \mid \boldsymbol{u}^{(l+1)}\left(t\right)=\mathrm{V_{th}}\right\}} \delta\left(t-t_k\right) \\
\boldsymbol{s}^{(1)}\left(t\right) &= \boldsymbol{s}_{in}\left(t\right) \\
\boldsymbol{s}_{out}\left(t\right) &= \boldsymbol{s}^{(N)}\left(t\right)
\end{aligned} \tag{2}
$$

where the $\mathbf{W}^{(l)} \in \mathbb{R}^{N_{l+1} \times N_l}$ is the synaptic weight, $\boldsymbol{s}_{in}(t)$ and $\boldsymbol{s}_{out}(t)$ are the input LR spike trains and the output HR spike trains, respectively, and $\mathrm{V_{th}}$ is the membrane potential threshold. In this paper, the spike response kernel and the refractory response kernel are defined as follows:

$$
\begin{aligned}
\varepsilon\left(t\right) &= \frac{t}{\tau_s} e^{1-\frac{t}{\tau_s}} \Theta\left(t\right) \\
\gamma\left(t\right) &= -2\mathrm{V_{th}} e^{-\frac{t}{\tau_r}} \Theta\left(t\right)
\end{aligned} \tag{3}
$$

where τ_s and τ_r are the time constants of spike response kernel and refractory response kernel, respectively, and $\Theta\left(t\right)$ is the Heaviside step function.

3.2. Network Architecture

The pipeline of our method is shown in Figure 2. Our network architecture is a convolutional SNN with 2 convolution layers and 1 deconvolution layer. The input and output of the model contain 2 channels, corresponding to the positive and negative events, respectively. The first convolutional layer extracts the feature using 8 kernels of size $5 \times 5 \times 2$ with a padding of 2 pixels. The second convolutional layer has 8 kernels of size $3 \times 3 \times 8$ with a padding of 1 pixel. The final deconvolutional layer has 2 kernels of size $2 \times 2 \times 8$ with a stride of 2. The PSP of the input LR event stream is interpolated to the desired size and added to the output PSP of the last layer to generate the final HR event stream. In practice, the time constants of the spike response kernel and refractory response kernel are set as $\tau_s = \tau_r = 1, 2, 4$ for the three layers, respectively. For the depth of the proposed model, our experimental results show that increasing the number of layers will not improve the performance. See Section 4.6 for more details.

3.3. Spatiotemporal Constraint Learning

Inferring HR event streams from LR event streams is an underdetermined task. The output HR event stream is expected to have the same statistical characteristics as the LR input in both spatial and temporal dimensions. The event stream obtained from the event camera is a spatiotemporal event cloud, and the statistics along its temporal dimension t and spatial dimension (x, y) represent different meanings. For the temporal dimension, we learn the temporal distribution of the event stream at each pixel, representing the distribution of the event rate intensity. For the spatial dimension, we learn the holistic spatial distribution of the event stream, representing the boundary and shape of moving edges in the scene, which is the principle of event triggering.

Temporal dimension. We learn the temporal distribution by minimizing the error between the predicted and the ground truth events at each pixel of each timestamp. Specifically, the temporal loss is defined as:

$$
\mathcal{L}^T = \frac{1}{2}\sum_i \int_0^T \left(s_i^N\left(t\right) - \widehat{s}_i\left(t\right)\right)^2 dt \tag{4}
$$

where $s_i^N\left(t\right)$ and $\widehat{s}_i\left(t\right)$ denote the output and ground truth spike trains, respectively, at the pixel with coordinate i.

Spatial dimension. We sum the output spike trains triggered within a time bin $[T_0, T_1]$ along the temporal dimension to obtain the Peristimulus Time Histogram (PSTH) as the representation of the spatial distribution, *i.e.*, $\mathrm{PSTH}_i = \int_{T_0}^{T_1} s_i^N\left(t\right) dt$. Specifically, the spatial distribution is learned by minimizing the error between the PSTH of the output spike train and the ground truth spike train, and the corresponding spatial loss is defined as:

$$
\mathcal{L}^S = \frac{1}{2}\sum_i \left(\mathrm{PSTH}_i - \widehat{\mathrm{PSTH}}_i\right)^2 \tag{5}
$$

where PSTH_i and $\widehat{\mathrm{PSTH}}_i$ are the PSTH of the output spike train and ground truth spike train, respectively, at the pixel with coordinate i. In practice, the length of time bin is manually setting to 50 milliseconds without overlapping.

Finally, the total loss function is defined as:

$$
\mathcal{L} = \alpha \cdot \mathcal{L}^T + \beta \cdot \mathcal{L}^S \tag{6}
$$

where α and β are hyperparameters for balancing the weights of different loss items.

4. Experiments

4.1. Datasets

We evaluate our method on four public datasets, *i.e.*, N-MNIST [30], CIFAR10-DVS [23], ASL-DVS [2], and Event Camera Dataset [28]. The N-MNIST dataset is obtained by capturing samples from the MNIST [20] dataset through a moving ATIS sensor [35], which contains 60,000 samples for training and 10,000 samples for test. The CIFAR10-DVS dataset is collected by moving images from the CIFAR10 [19] dataset in front of a fixed DVS128 [34] sensor, which contains 8,500 samples for training and 1,500 samples for test. The ASL-DVS dataset is a handshape dataset recorded in the real scene, containing 100,800 samples, among which 80% are selected for training and the remaining 20% for test. The Event Camera Dataset contains sequences captured by a moving DAVIS240C camera in various environments. We split each sequence into samples of 50 milliseconds length. See Section 4.5 for more details. Following the method in [22], we use the original DVS recordings as the ground truth HR event streams and down-sample the data by merging the events in each 2×2 kernel with a stride of 2 to obtain the LR event streams.

To match the simulation precision of SNN, we adjust the temporal resolution of the event stream to millisecond, *i.e.* merging events within each 1 millisecond. This process will result in partial event loss that more than one event triggered in the same pixel within 1 millisecond will be merged. According to our statistics, such loss of events is 0.14%, 1.96%, 3.41%, and 0.21% on N-MNIST, CIFAR10-DVS, ASL-DVS, and Event Camera Dataset, respectively.

4.2. Implementation Details

Since SNN is a time-continuous system, it must be discretized for simulation on GPUs. Considering the duration of samples from different datasets, we set the simulation step size as 1 millisecond and restrict the simulation time to 350, 200, 1500, and 50 milliseconds on N-MNIST, ASL-DVS, CIFAR10-DVS, and Event Camera Dataset, respectively. Our implementation is based on a publicly available SNN library named SLAYER [40].

The hyperparameters of the spatiotemporal constraint learning are chosen as $\alpha = 1$, $\beta = 5$, respectively. The proposed model is trained for 30 epochs with a batch size of 16. The optimization method is Adam [18] with a learning rate of 0.1, multiplied by 0.1 after 15 epochs.

4.3. Criterion

The criterion used to evaluate the performance of the event stream super-resolution is the root mean squared error (RMSE) along both the temporal and spatial dimensions:

$$\begin{aligned}
\mathrm{RMSE}_{\mathrm{ST}} &= \sqrt{\frac{1}{(T_1 - T_0)}\left(\mathrm{MSE}_{\mathrm{S}} + \mathrm{MSE}_{\mathrm{T}}\right)} \\
\mathrm{MSE}_{\mathrm{T}} &= \frac{1}{N_p}\sum_{i,j}\int_{T_0}^{T_1}\left(\mathrm{Spike}^{h}_{i,j}(t) - \mathrm{Spike}^{gt}_{i,j}(t)\right)^2 dt \qquad (7) \\
\mathrm{MSE}_{\mathrm{S}} &= \frac{1}{N_p}\sum_{k=1}^{N_b}\sum_{i,j}\left\|\mathrm{PSTH}^{h}_{i,j}(k) - \mathrm{PSTH}^{gt}_{i,j}(k)\right\|_2^2
\end{aligned}$$

where T_0 and T_1 are timestamps of the first and last events, and N_p is the number of pixels that contain at least one event in the ground truth event stream, and N_b is the number of time bins. $\mathrm{Spike}^{h}_{i,j}$ and $\mathrm{Spike}^{gt}_{i,j}$ are the output and ground truth event stream at pixel (i, j), and $\mathrm{PSTH}^{h}_{i,j}(k)$ and $\mathrm{PSTH}^{gt}_{i,j}(k)$ are the corresponding PSTH calculated in the k-th time bin. We set the length of time bins as 50 milliseconds without overlapping for the calculation of PSTH.

	N-MNIST	CIFAR10-DVS	ASL-DVS
Li et al. [22]	0.757	0.404	0.550
Ours	**0.272**	**0.179**	**0.229**
Ours (w/o s-loss)	0.280	0.183	0.234
Ours (w/o t-loss)	0.273	0.180	0.230

Table 1. Super-resolution results on N-MNIST, CIFAR10-DVS, and ASL-DVS datasets. Bold numbers represent the best scores.

Here we describe our motivation for designing the criterion in detail. Since the events are triggered by moving edges in the scene, the events at different timestamps in the same pixel have less correlation than the events in a spatial neighborhood at the same moment. Meanwhile, the spatial and the temporal dimensions of the event stream have different physical meanings. This is the reason why we do not adopt some of the existing spike train similarity metrics, *e.g.* [37] and [38], which are designed for neuronal spike trains and are not suitable for the event stream. Summing the event stream along the temporal dimension can depict the texture and boundaries of moving objects in the scene. Therefore, considering the principle of the event triggering, we design the $\mathrm{MSE}_{\mathrm{S}}$ to measure the consistency of the event stream with the moving edges. Similarly, we design the $\mathrm{MSE}_{\mathrm{T}}$ to measure the error of events' timestamps in the temporal dimension.

4.4. Super-resolution Results

Quantitative comparison. Table 1 and Table 3 show the quantitative results on the N-MNIST, CIFAR10-DVS, ASL-DVS, and Event Camera Dataset. $\mathrm{RMSE}_{\mathrm{ST}}$ is chosen as the criterion to evaluate the super-resolution performance. The proposed method is compared with the only existing method [22] and achieves satisfying improvement on all datasets. As shown in tables, the proposed method reduces $\mathrm{RMSE}_{\mathrm{ST}}$ by 64.1%, 55.7%, 58.4%, and 61.2% on the N-MNIST, CIFAR10-DVS, ASL-DVS, and Event Camera Dataset, respectively. Different from [22], which uses the non-homogeneous Poisson process to simulate the event stream and predicts the timestamp of each event in a sampling way, the proposed method utilizes a supervised learning mechanism on the spatial and temporal distributions of the event stream. And our performance is attributed to such a strategy that could predict more accurate spatial structures as well as timestamps of the output HR event stream.

Qualitative analysis. Figure 5 shows the qualitative results generated by the proposed method and [22] on the N-MNIST dataset. It can be intuitively seen that our method achieves better performance than the previous work [22]. For example, the proposed method can generate a clearer

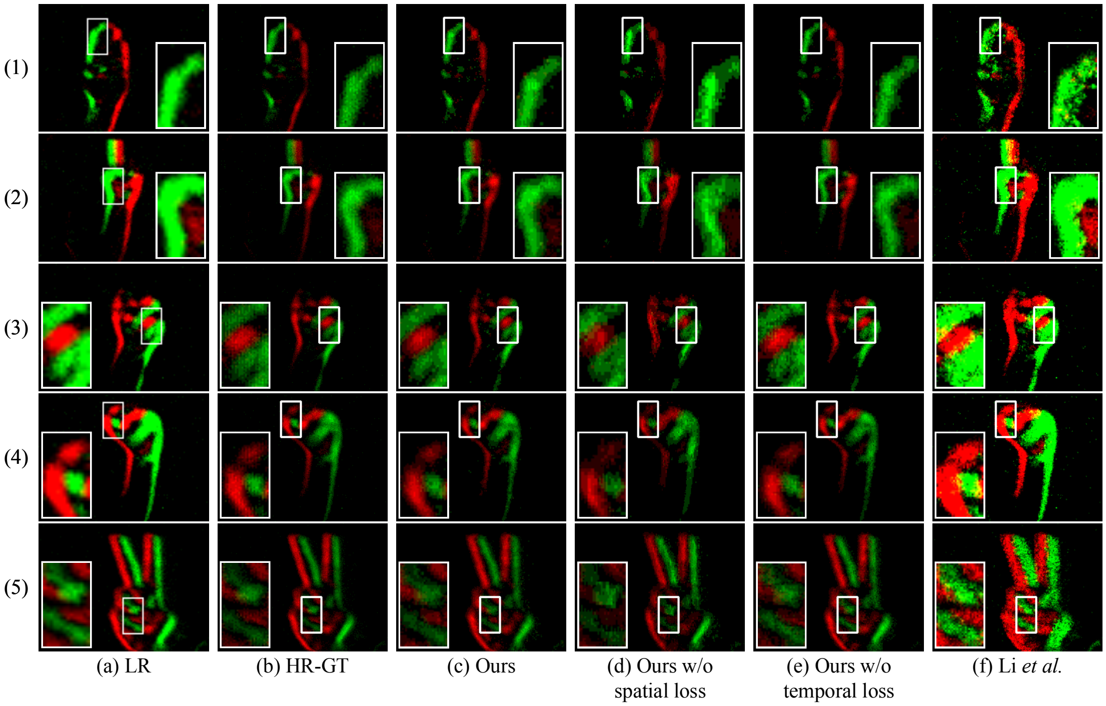


Figure 4. Visualization results of super-resolution on ASL-DVS dataset [2]. From left to right: the input LR event streams, the ground truth HR event streams, results generated by our method, our method without spatial distribution learning, our method without temporal distribution learning, and Li *et al.* [22]. See Section 4.4 and Section 4.6 for the detailed analysis. (Best viewed in color.)

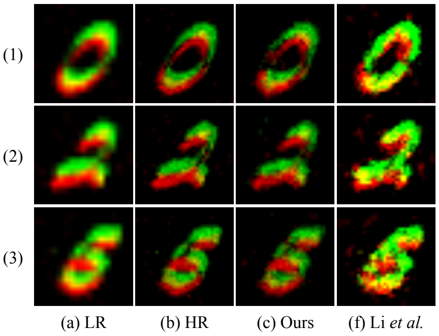


Figure 5. Visualization results on N-MNIST dataset [30]. From left to right: the input LR event streams, the ground truth HR event streams, results generated by our method, and Li *et al.* [22].

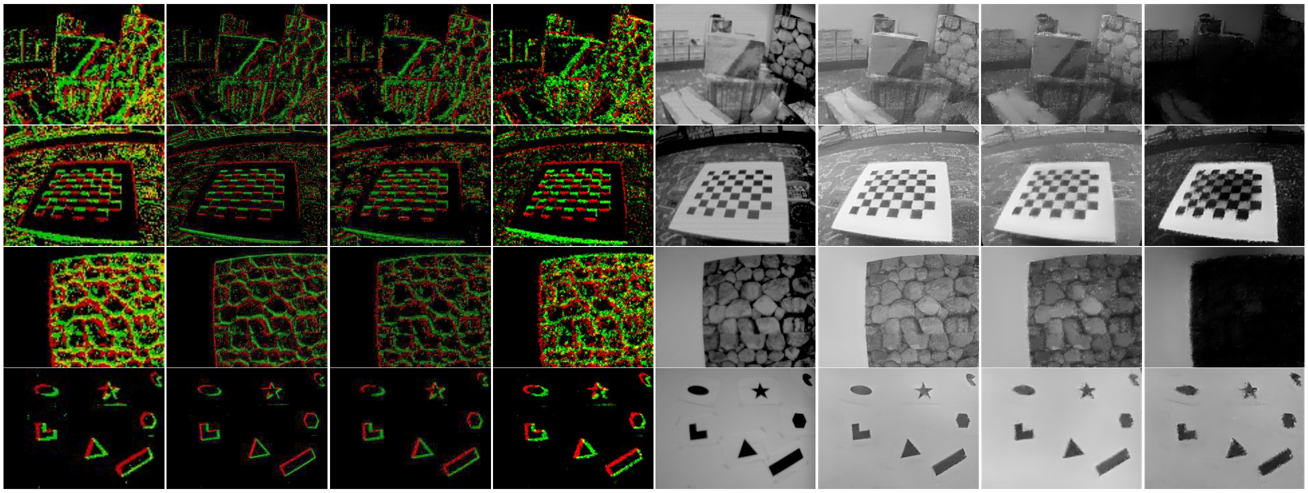
shape boundary, as shown in row (1). Row (2) shows that the proposed method performs better on reconstructing tiny details in pixel-level, *e.g.* the middle part of the digital number 2. In contrast, the result generated by [22] is blurry. Figure 4 shows more visualization results on the ASL-DVS dataset. Our method can generate relatively sparser output with less noise, *e.g.* row (1) and row (4), which is more similar to the ground truth. More visualization results are provided in the supplementary material.

4.5. Downstream Applications

To demonstrate the usability of the proposed method and evaluate the quality of generated HR event streams, we test on two downstream applications: object classification and image reconstruction. First, we test the classification performance on the N-MNIST, CIFAR10-DVS, and ASL-DVS datasets. We use the classifier proposed in [9] to classify the LR event streams, the ground truth HR event streams, the HR event streams generated by our method, and by [22], respectively. To prevent data leakage, the split of training and test sets for classification is the same as the super-resolution task. The results in Table 2 show that the HR event streams generated by our model can achieve a comparable classification accuracy with the ground truth HR event streams, and outperform the LR event streams and those generated by [22], which prove that our method can effectively improve the accuracy of the downstream classification task. More details of the experimental setting are provided in the supplementary material.

	N-MNIST	ASL-DVS	CIFAR10-DVS
LR events	99.0%	99.7%	50.8%
Li *et al.* [22]	97.8%	98.0%	59.6%
Ours	**99.1%**	**99.9%**	**76.8%**
GT events (ref.)	99.1%	99.9%	78.7%

Table 2. Classification results on N-MNIST, CIFAR10-DVS, and ASL-DVS datasets. The classification accuracy of the HR event streams generated by our method outperforms the LR event streams and those generated by [22]. The classification accuracy of the ground truth HR event streams is also provided for reference. Bold numbers represent the best scores.



(a) LR Events (b) HR Events-GT (c) HR Events-Ours (d) HR Events-Li *et al.* (e) Frame-GT (f) Frame reconstructed by (b) GT events (g) Frame reconstructed by (c) ours events (h) Frame reconstructed by (d) Li *et al.*'s events

Figure 6. Results on Event Camera Dataset. (a) Input LR event streams. (b) Ground truth HR event streams. (c-d) HR event streams generated by the proposed method and Li *et al*. [22]. (e) Ground truth frames. (f-h) The images reconstructed from ground truth HR event streams, HR event streams generated by the proposed method and by [22], respectively (*i.e*. corresponding to column (b-d)).

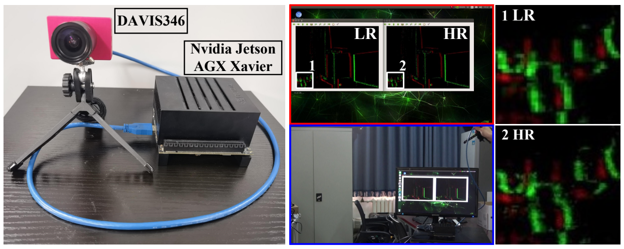
Dataset	$\mathrm{RMSE}_{\mathrm{ST}}$ ↓		SSIM ↑			MSE ↓		
	Li *et al.*	Ours	Li *et al.*	Ours	GT Event (ref.)	Li *et al.*	Ours	GT Event (ref.)
dynamic_6dof	0.937	**0.356**	**0.62**	**0.62**	0.68	0.07	**0.05**	0.03
boxes_6dof	0.786	**0.312**	0.38	**0.46**	0.65	**0.06**	0.07	0.03
poster_6dof	0.902	**0.344**	0.44	**0.54**	0.68	0.07	**0.05**	0.03
shapes_6dof	0.924	**0.356**	0.40	**0.47**	0.45	0.19	**0.13**	0.18
office_zigzag	0.852	**0.325**	0.57	**0.63**	0.71	**0.04**	**0.04**	0.03
slider_depth	0.667	**0.274**	**0.58**	0.52	0.65	**0.06**	0.10	0.07
calibration	0.794	**0.306**	**0.49**	0.47	0.55	**0.03**	**0.03**	0.03
Mean	0.837	**0.325**	0.497	**0.530**	0.624	0.074	**0.067**	0.057

Table 3. Results of event stream super-resolution ($\mathrm{RMSE}_{\mathrm{ST}}$) and image reconstruction (SSIM, MSE) on Event Camera Dataset. Bold numbers represent the best results. Our method achieves a 61.2% decrease of $\mathrm{RMSE}_{\mathrm{ST}}$ in the event stream super-resolution task. The HR event streams generated by our method outperform those generated by [22] in the image reconstruction task with an average 6.6% increase of SSIM and 9.5% decrease of MSE. The results of ground truth event streams are also provided for reference.

We also evaluate our super-resolution result with the image reconstruction task. Reconstructing images from event streams is a challenging task, which requires our method to be extremely robust. We test on the Event Camera Dataset [28] containing more complex motions and scenes to demonstrate the robustness of our method. Following the experimental settings in [36], we choose the same 7 sequences for testing, *e.g*. boxes_6dof, office_zigzag, *etc*., and the other sequences for training, *e.g*. boxes_rotation, office_spiral, *etc*. The split of training and test sets are the same for super-resolution and image reconstruction tasks. For image reconstruction, we use the pre-trained model provided in [36] and reconstruct images with events in a window with a fixed duration of 50 ms. We use the mean square error (MSE) and structural similarity (SSIM) as criteria and choose the same ground truth frames as [36] for evaluation. Table 3 presents the main quantitative results. Our super-resolution result outperforms the state-of-the-art by a large margin, with an average of 61.2% decrease in $\mathrm{RMSE}_{\mathrm{ST}}$. Meanwhile, the HR event streams generated by our method can achieve a better performance in the image reconstruction task than those generated by [22] with an average of 6.6% increase in SSIM and 9.5% decrease in MSE. Figure 6 shows more qualitative results. The images reconstructed by the HR event streams generated by our method (column (g)) are comparable to those reconstructed by the ground truth HR event streams (column (f)) and outperform those reconstructed by HR event streams generated by [22] (column (h)). Meanwhile, the images reconstructed by the HR event streams generated by [22] result in a large error of intensity, *e.g*. row 1 and 3, which is caused by the fact that the method in [22] will mis-trigger lots of events and lead to a blurred boundary, as column (d) shown. We also observe that the HR event stream generated by our method outperforms the ground truth event stream on shapes_6dof sequence. This is due to that our model learns precise pat-

	3 layers	4 layers	5 layers	5 layers
$\mathrm{RMSE}_{\mathrm{ST}}$	**0.272**	0.278	0.295	0.309

Table 4. Super-resolution results with different network architecture on N-MNIST dataset. Bold numbers represent the best scores.



(a) Hardware system (b) Indoor scene: office table

Figure 7. (a) Picture of our embedded implementation. The event camera, DAVIS346, is connected to the edge computing processor, Nvidia Jetson AGX Xavier. (b) The real-time super-resolution results generated by our embedded system as well as the operating screenshots. Some details are zoomed for a better comparison. The complete video is presented in the supplementary material.

terns and generates HR event stream contain less noise.

4.6. Ablation Study

We conduct ablation experiments to demonstrate the effectiveness of the proposed method. Table 1 shows the quantitative results of our method without spatial distribution learning, denoted as Ours (w/o s-loss), and without temporal distribution learning, denoted as Ours (w/o t-loss). The experimental results show that the removal of either the spatial or temporal distribution learning will lead to a decline in performance. As shown in Figure 4, the outputs of the proposed method without spatial distribution learning contain fuzzier boundaries and more noise, *e.g*. row (3).

For the network architecture, we test the performance of models with different numbers of layers on the N-MNIST dataset. As shown in Table 4, the quantitative results show that the performance decline as the model becomes deeper. This may attribute to the difficulty of training the deep SNN model. More details of the experimental setting are provided in the supplementary material.

4.7. Limitations

Since the implementation of our SNN model is a numerical simulation on GPUs, the precision of events' timestamps is decreased to millisecond to match the simulation precision. This can be improved by deploying the model on an SNN chip, *e.g*. Loihi [7], during the prediction stage.

5. Embedded Implementation

To further prove the efficiency and usability of the proposed method, we deploy our method on a mobile platform, as shown in Figure 7 (a). The event camera we used is the DAVIS346, which can achieve a dynamic range of 120 dB and a maximum bandwidth of 12M events/s. The computing processor we choose is the Nvidia Jetson AGX Xavier, a powerful edge computing processor. In our implementation, all the computing is performed on Xavier, and a monitor is connected only for displaying the result. It should be noted that our method is implemented in a numerical simulation way based on GPU, so the power consumption is relatively high (about 30 W for Xavier). If we use some neuromorphic computing chips specifically designed for SNN, *e.g*. Loihi [7] or TrueNorth [26], the power consumption could be reduced by about 95% [3], which would be more suitable for deployment on mobile devices.

We test our embedded system indoor, and the result shows that our system can generate superior HR event streams in real-time and owns strong robustness to complex motions. Figure 7 (b) shows our operation and super-resolution results. We hold the DAVIS346 in hand and move it to capture the original data. The movement involves from simple low-speed shifting to complex high-speed spinning. The upper-left part in the red box shows the LR event stream obtained by the DAVIS346 and the HR event stream generated by our system. The bottom-left part in the blue box is the screenshot of our operation. It can be seen that our system could generate HR event streams with a high-quality recovery of detail. The complete video of the super-resolution result and our operation screenshot are presented in the supplementary material.

6. Conclusion

In this paper, we introduce an event stream super-resolution method based on spiking neural network to directly predict the HR event stream from the input LR event stream. The major contributions of the proposed method include the spatiotemporal constraint learning mechanism that learns both the spatial and temporal distribution of the event stream simultaneously. The experimental results show that the proposed method outperforms the state-of-the-art method on four large-scale datasets by a large margin ($>55\%$ improvement). Ablation study and visualization results further prove the effectiveness of the proposed learning mechanism. The marvelous performance on two downstream applications, *i.e*. object classification and image reconstruction, demonstrates the usability of our method. An embedded deployment shows that our method could generate high-quality HR event streams in real-time, which is suitable to be deployed on mobile systems.

7. Acknowledgment

This work was supported by National Natural Science Funds of China (U1701262, U1801263, 62088102, 62021002).

References

[1] Arnon Amir, Brian Taba, David Berg, et al. A Low Power, Fully Event-Based Gesture Recognition System. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7243–7252, 2017. 1

[2] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-Based Object Classification for Neuromorphic Vision Sensing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 491–501, 2019. 1, 2, 4, 6

[3] Peter Blouw, Xuan Choo, Eric Hunsberger, and Chris Eliasmith. Benchmarking Keyword Spotting Efficiency on Neuromorphic Hardware. In *Annual Neuro-inspired Computational Elements Workshop*, pages 1–8, 2019. 8

[4] Sander M. Bohte, Joost N. Kok, and Han La Poutre. Error-Backpropagation in Temporally Encoded Networks of Spiking Neurons. *Neurocomputing*, 48(1-4):17–37, 2002. 2

[5] Olaf Booij and Hieu tat Nguyen. A Gradient Descent Rule for Spiking Neurons Emitting Multiple Spikes. *Information Processing Letters*, 95(6):552–558, 2005. 2

[6] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 1, 2

[7] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro*, 38(1):82–99, 2018. 8

[8] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3867–3876, 2018. 1

[9] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. End-to-End Learning of Representations for Asynchronous Event-Based Data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5633–5643, 2019. 6

[10] Wulfram Gerstner. Time Structure of the Activity in Neural Network Models. *Physical review E*, 51(1):738, 1995. 2, 3

[11] Wulfram Gerstner and Werner M. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002. 2

[12] Samanwoy Ghosh-Dastidar and Hojjat Adeli. A New Supervised Learning Algorithm for Multiple Spiking Neural Networks with Application in Epilepsy and Seizure Detection. *Neural Networks*, 22(10):1419–1431, 2009. 2

[13] Gallego Guillermo, Delbruck Tobi, Michael Orchard Garrick, Bartolozzi Chiara, Taba Brian, Censi Andrea, Leutenegger Stefan, Davison Andrew, Conradt Jorg, Daniilidis Kostas, and Scaramuzza Davide. Event-based Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[14] Alan L. Hodgkin and Andrew F. Huxley. A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve. *The Journal of physiology*, 117(4):500, 1952. 2

[15] Eugene M. Izhikevich. Simple Model of Spiking Neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003. 2

[16] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning Event-Based Motion Deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2020. 1

[17] Hanme Kim, Stefan Leutenegger, and Andrew J. Davison. Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera. In *Proceedings of the European Conference on Computer Vision*, pages 349–364, 2016. 1

[18] Diederik P. Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. 5

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning Multiple Layers of Features from Tiny Images. *Technical report*, 2009. 4

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4

[21] P. A. W. Lewis and G. S. Shedler. Simulation of Nonhomogeneous Poisson Processes by Thinning. *Naval Research Logistics Quarterly*, 26(3):403–413, 1979. 3

[22] Hongmin Li, Guoqi Li, and Luping Shi. Super-Resolution of Spatiotemporal Event-Stream Image. *Neurocomputing*, 335:206–214, 2019. 2, 3, 5, 6, 7

[23] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. CIFAR10-DVS: An Event-Stream Dataset for Object Classification. *Frontiers in Neuroscience*, 11:309, 2017. 2, 4

[24] Daqi Liu, Alvaro Parra, and Tat-Jun Chin. Globally Optimal Contrast Maximisation for Event-based Motion Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6349–6358, 2020. 1

[25] Gehrig Mathias, Shrestha Sumit, Bam, Mouritzen Daniel, and Scaramuzza Davide. Event-Based Angular Velocity Regression with Spiking Networks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4195–4202, 2020. 2

[26] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface. *Science*, 345(6197):668–673, 2014. 8

[27] S. Mohammad Mostafavi I., Jonghyun Choi, and Kuk-Jin Yoon. Learning to Super Resolve Intensity Images from Events. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2765–2773, 2020. 2

[28] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The Event-Camera Dataset and Simulator: Event-Based Data for Pose Estimation, Visual Odometry, and SLAM. *The International Journal of Robotics Research*, 36(2):142–149, 2017. 2, 4, 7

[29] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate Gradient Learning in Spiking Neural Networks. *IEEE Signal Processing Magazine*, 36:61–63, 2019. 2

[30] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades. *Frontiers in Neuroscience*, 9:437, 2015. 2, 4, 6

[31] Garrick Orchard, Cedric Meyer, Ralph Etienne-Cummings, Christoph Posch, Nitish Thakor, and Ryad Benosman. HFirst: A Temporal Approach to Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2028–2040, 2015. 1

[32] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a Blurry Frame Alive at High Frame-Rate with an Event Camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6820–6829, 2019. 1

[33] Federico Paredes-Vallés, Kirk Yannick Willehm Scheper, and Guido Cornelis Henricus Eugene De Croon. Unsupervised Learning of a Hierarchical Spiking Neural Network for Optical Flow Estimation: From Events to Global Motion Perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):2051–2064, 2020. 1

[34] Lichtsteiner Patrick, Posch Christoph, and Delbruck Tobi. A 128× 128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. 1, 4

[35] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS. *IEEE Journal of Solid-State Circuits*, 46(1):259–275, 2010. 4

[36] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High Speed and High Dynamic Range Video with an Event Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1, 7

[37] MCW van Rossum. A Novel Spike Distance. *Neural Computation*, 13(4):751–763, 2001. 5

[38] Susanne Schreiber, Jean-Marc Fellous, D Whitmer, P Tiesinga, and Terrence J Sejnowski. A New Correlation-Based Measure of Spike Timing Reliability. *Neurocomputing*, 52:925–931, 2003. 5

[39] Yusuke Sekikawa, Kosuke Hara, and Hideo Saito. EventNet: Asynchronous Recursive Event Processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2019. 2

[40] Sumit Bam Shrestha and Garrick Orchard. Slayer: Spike Layer Error Reassignment in Time. In *Advances in Neural Information Processing Systems*, pages 1412–1421, 2018. 2, 5

[41] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. HATS: Histograms of Averaged Time Surfaces for Robust Event-based Object Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018. 1

[42] Zihao Wang, Peiqi Duan, Oliver Cossairt, Aggelos Katsaggelos, Tiejun Huang, and Boxin Shi. Joint Filtering of Intensity Images and Neuromorphic Events for High-Resolution Noise-Robust Imaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1609–1619, 2020. 2

[43] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-Temporal Backpropagation for Training High-Performance Spiking Neural Networks. *Frontiers in Neuroscience*, 12:331, 2018. 2

[44] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image Super-Resolution as Sparse Representation of Raw Image Patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2, 3

[45] Friedemann Zenke and Surya Ganguli. SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks. *Neural Computation*, 30(6):1514–1541, 2018. 2

[46] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-Based Visual Inertial Odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5391–5399, 2017. 1

[47] Alex Zihao Zhu, LiangFzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised Event-Based Learning of Optical Flow, Depth, and Egomotion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019. 1