EDeF-Net: Spatio-temporal Association Network for Flicker Removal in Event Streams

Jin Han*
The University of Tokyo
Tokyo, Japan
hanjin@pku.edu.cn

Yixin Yang†
Peking University
Beijing, China
yangyixin93@pku.edu.cn

Zhan Zhan*
The University of Tokyo
Tokyo, Japan
zhanzhanchn@gmail.com

Boxin Shi†‡
Peking University
Beijing, China
shiboxin@pku.edu.cn

Imari Sato*
National Institute of Informatics
Tokyo, Japan
imarik@nii.ac.jp

Abstract

Event cameras with bio-inspired neuromorphic sensors are highly sensitive to brightness changes. When there are moving objects in a scene under constant lighting, event cameras only record motion information and output a sequence of events asynchronously. However, the common flickering light sources, such as fluorescent or LED lamps powered by alternating current exist in various real-world scenarios. When operating under a flickering light source, event cameras output numerous redundant event signals that are triggered by the flickering effect, which overwhelm the useful signals that encode motion information. In this paper, we propose EDeF-Net, an Event streams DeFlickering Network that effectively leverages the spatio-temporal correlation of event streams by modeling both the inter-channel temporal attention and inter-patch spatial attention. To facilitate network training and evaluation, we synthesize the first dataset containing paired flickering and flicker-free event streams. Moreover, we demonstrate that event streams filtered by EDeF-Net yield performance improvements on downstream applications such as event-based optical flow estimation and object tracking.

CCS Concepts

• **Computing methodologies** → **Computational photography**.

Keywords

Event camera; Flicker removal; Signal filtering

*Jin Han, Zhan Zhan, and Imari Sato are with National Institute of Informatics and The University of Tokyo, Tokyo, Japan.
†Yixin Yang and Boxin Shi are with State Key Laboratory of Multimedia Information Processing and National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, Beijing, China.
‡Corresponding author.
Project page: https://github.com/hjynwa/EDeF-Net

MM '25, Dublin, Ireland.

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3754995

ACM Reference Format:
Jin Han, Yixin Yang, Zhan Zhan, Boxin Shi, and Imari Sato. 2025. EDeF-Net: Spatio-temporal Association Network for Flicker Removal in Event Streams. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3746027.3754995

1 Introduction

Neuromorphic signals from event cameras [4, 25, 30] encode the brightness changes in a scene, which have emerged as a promising solution for high-speed and high dynamic range (HDR) visual sensing, exhibiting unique advantages over traditional frame-based cameras [11]. Thanks to these advantages, event cameras have been imposed to versatile tasks ranging from computer vision to robotics fields [12, 18, 28, 31, 34, 39]. Majority of the existing event-based algorithms assume a constant lighting condition when capturing event data, and rely on the events only triggered by motion. Researchers usually regard the motion-triggered events as the valid signals in these applications, because they encode the critical information of moving objects in a scene. However, real-world scenarios frequently involve flickering light sources, such as fluorescent or LED lights powered by alternating current (AC). When operating in such scenarios, event cameras' high sensitivity to brightness changes becomes a double-edged sword. On the one hand, the high sensitivity makes event cameras highly effective to capture fast motion, on the other hand, it also makes them prone to flickering effect of some indoor lighting [37]. Consequently, this results in a mass of redundant and ambiguous event data triggered by the flickering light source. The flickering issue introduces substantial noise into the event streams, which degrades the performance of event-based algorithms.

As show in Figure 1 (a), the event signals record a waving white board with uniform reflection under a 100 Hz flickering light. The rapid changes of light intensity trigger a mass of positive and negative event signals in one flicker cycle. In Figure 1 (b), we can clearly see the portion of positive events and that of negative events from the time axis. In the first row of Figure 1 (c), we accumulate and visualize the events triggered in each flickering cycle into two images (*i.e.*, light energy increasing portion and decreasing portion). The following three rows show that some classical filters (*e.g.*, erosion filter and median filter) perform poorly since they do not take the crucial temporal pattern of event signals into account. The comb

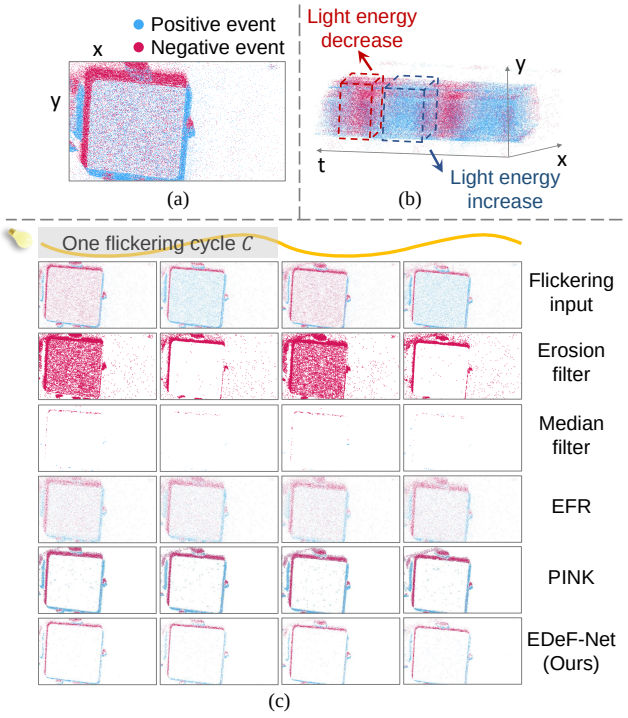

Figure 1: (a) A white board moving in front of a flickering light source. (b) Visualization of the events stream from the temporal domain. (c) The first row is a flickering event stream. The following rows show the comparison of the flicker removal results from classical filter algorithms, including erosion filter, median filter, comb filter (EFR [36]), Gaussian filter (PINK [16]), and the proposed EDeF-Net.

filter-based method (EFR [36]) is restricted to specific scenarios where the flickering light source is static in the camera's field of view. When there are objects moving under a flickering light, events are triggered by both motion and flicker. It complicates the task of distinguishing whether an event signal is triggered by motion (should be preserved) or flicker (should be discarded). Details of these filter-based methods are presented in Sec. 3.1. The Gaussian filter-based method (PINK [16]) integrates of period of event signals. It makes the edges of objects thicker and cannot be redistributed back to streams.

In this paper, we propose EDeF-Net that leverages the correlation of event signals in both temporal and spatial domains to solve the ambiguity of events triggering, as shown in the last row of Figure 1 (c). We model the temporal repetitive pattern and spatial sparsity in the flickering event streams. For temporal correlation, we design a module that learns the inter-channel attention along the temporal axis by computing the correlation among events triggered in different channels. To ensure spatial consistency of the event signals, we further use another module that takes patch-wise tokens to learn an spatial attention residual.

For network training, we synthesize a dataset with corresponding flickering and non-flickering event streams using Blender [7] and an event simulator [17]. We conduct extensive experiments on both synthetic data and real-world data, demonstrating the proposed method can effectively filter out the redundant events triggered by flickering light sources while preserving the valid signals triggered only by motion. We further verify that the event streams after flicker removal yield performance improvement in several down-stream applications, including event-based optical flow estimation and object tracking. The contributions of this paper are summarized as follows:

- We propose EDeF-Net, a light-weighted network with specific attention modules for flicker removal in event streams, which incorporates the inter-channel temporal correlation and inter-patch spatial consistency.
- We build the first synthetic dataset with flickering events and corresponding flicker-free ground truth for flicker removal in event streams.
- Event streams filtered by EDeF-Net yield better performance when applied to several down-stream applications, including event-based optical flow estimation and object tracking, which demonstrates the significance and effectiveness of the proposed method.

2 Related Works

2.1 Video deflickering using conventional cameras

Flickering artifacts are a prevalent issue in conventional RGB videos, stemming from various factors. For instance, capturing scene information with a high-speed camera under flickering light sources can result in a noticeable global intensity shift among consecutive frames. Delon *et al.* [8] introduced a local stabilization operator that acts on frame patches and relies on a similarity measurement. Kanj *et al.* [19] proposed a local method for flicker removal based on super-pixels segmentation. Blind video consistency algorithms [3, 10, 22, 23] attempted to enhance the temporal consistency of videos without the need for specific flickering guidance. Event cameras possess a significantly high temporal resolution and sensitivity to brightness changes with stream-like data format. Therefore, developing deflickering algorithms specifically tailored for event cameras is of vital importance.

2.2 Event signals filtering

We firstly give a brief review on noise removal in event signals. For background activity noise [20, 27], some approaches such as the development of in-chip filters [26] and efficient noise models for real-time processing [13], neural network-based techniques [2] for event classification, and the utilization of motion association [35] to enhance noise filtering accuracy. Under alternating current-powered light sources, the flickering effect is obvious and triggers numerous redundant event signals. To eliminate flicker in event streams, EFR [36] applied a linear comb filter that considers the frequency of events triggering at each pixel. However, this approach primarily focuses on flicker removal in static positions (*e.g.*, ceiling lights). ELIR [33] is a two-stage filtering pipeline that eliminates light interference. However, it needs additional inertial measurement units (IMUs) to measure the motion polarity. The proposed EDeF-Net

can successfully deal with the ambiguity of events that are triggered by motion associated with light source flickering without any additional priors.

3 Proposed Method

In this section, we describe the proposed method, which begins with formulating the event camera's signal triggering mechanism and flicker model of light source power by AC, followed by demonstrating that a simple polarity offset-based operation cannot solve the problem.

3.1 Preliminary

Event camera model. The sensor of event camera detects the brightness changes in a scene asynchronously, instead of recording the absolute intensity like conventional cameras. The data output by an event camera is in a stream-like format $\mathbb{E} = \{\mathbf{e}_i\}_{i=1}^{N}$, where $\mathbb{E}$ represents a sequence of event signals and $\mathbf{e}_i = (t_i, x_i, y_i, p_i)$ is the i-th event, encoding the timestamp t_i, coordinates (x_i, y_i), and brightness changing polarity p_i. The process of event triggering is:

$$p = \begin{cases} +1, & \Delta I_t^{(x,y)} \geq \theta \\ \text{none}, & \Delta I_t^{(x,y)} \in (-\theta, \theta) \\ -1, & \Delta I_t^{(x,y)} \leq -\theta \end{cases} \tag{1}$$

where $\Delta I = \log I_{t+\Delta t}^{(x,y)} - \log I_t^{(x,y)}$ is the intensity changes in log-scale during a short time slot Δt. Once the intensity changes exceed a pre-defined threshold θ, an event signal will be fired. The polarity $p \in \{-1, +1\}$ indicates the increase or decrease of intensity values.

Flicker model of alternating current. The AC with a zero-mean and a stable peak outlet amplitude exhibits a quasi-periodic nature [32]. We define the time interval between two successive zero-crossings of AC as ΔT. Then the frequency of AC is $1/(2\Delta T)$, because there are two ΔTs (*i.e.*, positive and negative voltage) in one period of AC. A bulb powered by AC goes dark whenever the AC voltage reaches zero. As a result, it would flicker at twice the AC frequency. We refer to this flickering period as a *cycle* C, whose frequency is $1/\Delta T$. Consider a static scene with a flickering light source, for each pixel, the event signals triggered by flicker follow a pattern of consecutive "ON-OFF", where "ON" represents positive event and "OFF" represents negative event, respectively. Once motion occurs, the events triggered by moving objects will intervene in such a flickering pattern in the event stream of each pixel, which makes it difficult to distinguish which factor the event signals are triggered from.

Flicker removal by polarity offset and filters. It is straightforward to think about using the polarity offset to achieve flicker removal, *i.e.*, counting the number of positive and negative events during a flickering cycle and making subtraction. However, the number of positive and negative event signals triggered in one flickering cycle are different due to the inconsistency in the increasing and decreasing portions, as illustrated in Figure 1 (b). Existing event denoising methods [2, 13] mainly consider the background activity noise from current leakage and temporal fluctuation, which cannot be directly applied to flicker that appears in a repetitive pattern. Besides, we have also tried several classical morphological filtering algorithms to filter the events accumulated over a small time window, as compared in Figure 1 (c). The middle three rows show the results of filter-based algorithms, which cannot effectively remove flickering events (erosion filter), mistakenly remove the valid motion-triggered events (median filter), or blur the events in the object edges (EFR [36]), resulting in poor performance. In contrast, the proposed EDeF-Net not only removes the flickering events, but also preserves the motion-triggered events.

3.2 EDeF-Net

Event cameras output stream-like data with dense temporal information and sparse spatial representation. We propose to analyze the spatio-temporal correlation of event streams by learning an inter-channel temporal attention and an inter-patch spatial attention. The overview of EDeF-Net is illustrated in Figure 2, which is composed of the temporal attention module (TAtt-M) and the spatial attention module (SAtt-M). To make the output tensors easy to redistribute back into event streams, we use a stack $\mathbf{S}(\mathbb{E}) \in \mathbb{R}^{h \times w \times c}$ to represent events triggered in one flickering cycle C. EDeF-Net takes an flickering event stack $\mathbf{S}(\mathbb{E}_f)$ as input, and outputs its corresponding flicker-free stack $\mathbf{S}(\mathbb{E}_m)$ with only motion-triggered events.

Inter-channel temporal attention. From the time axis, we can see the obvious flickering pattern along, which could be modeled by learning a channel-wise attention map. Inspired by the transposed self-attention [38], we first map the input stack $\mathbf{S}(\mathbb{E}_f)$ to the embedding features $\mathbf{F}(\mathbb{E}_f)$ by the channel embedding operation $\text{Embed}(\cdot)$, which is implemented by the 1×1 convolutions:

$$\mathbf{F}(\mathbb{E}_f) = \text{Embed}(\mathbf{S}(\mathbb{E}_f)). \tag{2}$$

Then the feature maps in each channel are regarded as a sequence of c tokens with dimension of $h \times w$. To compute the self-attention map in channel-wise, we generate the query $\mathbf{Q}$, key $\mathbf{K}$ and value $\mathbf{V}$ from each token by a depth-wise convolution. The channel-wise attention $\mathbf{A} \in \mathbb{R}^{c \times c}$ can be obtained by the matrix multiplication operation of the reshaped $\mathbf{Q}$ and $\mathbf{K}$. The output feature maps after channel-wise attention is:

$$\begin{aligned} \mathbf{X} &= \text{Conv}(\mathbf{V} \otimes \mathbf{A}) + \mathbf{F}(\mathbb{E}_f), \\ \hat{\mathbf{F}}(\mathbb{E}_f) &= \text{Conv}(\text{LN}(\mathbf{X})) + \mathbf{X}, \end{aligned} \tag{3}$$

where $\mathbf{X}$ is the middle features. $\text{Conv}(\cdot)$ and $\text{LN}(\cdot)$ represent convolution and layer normalization [1], respectively. The channel-wise attention mechanism computes the temporal correlation of an event stack to get a weighting map, without affecting the spatial consistency in each channel. The final operation of TAtt-M is the Hadamard product of the weighting map and input event stack, which is represented as:

$$\mathbf{L} = \mathbf{S}(\mathbb{E}_f) \odot \text{UnEmbed}(\hat{\mathbf{F}}(\mathbb{E}_f)), \tag{4}$$

where $\mathbf{L}$ is the latent features output from TAtt-M, and $\text{UnEmbd}(\cdot)$ represents the unembedding operation that maps the output features back to the same dimension of the input event stack.

Inter-patch spatial attention. Event streams not only contain the temporal information with very high resolution, but also encodes the sparse spatial representations of the scene. Given the latent

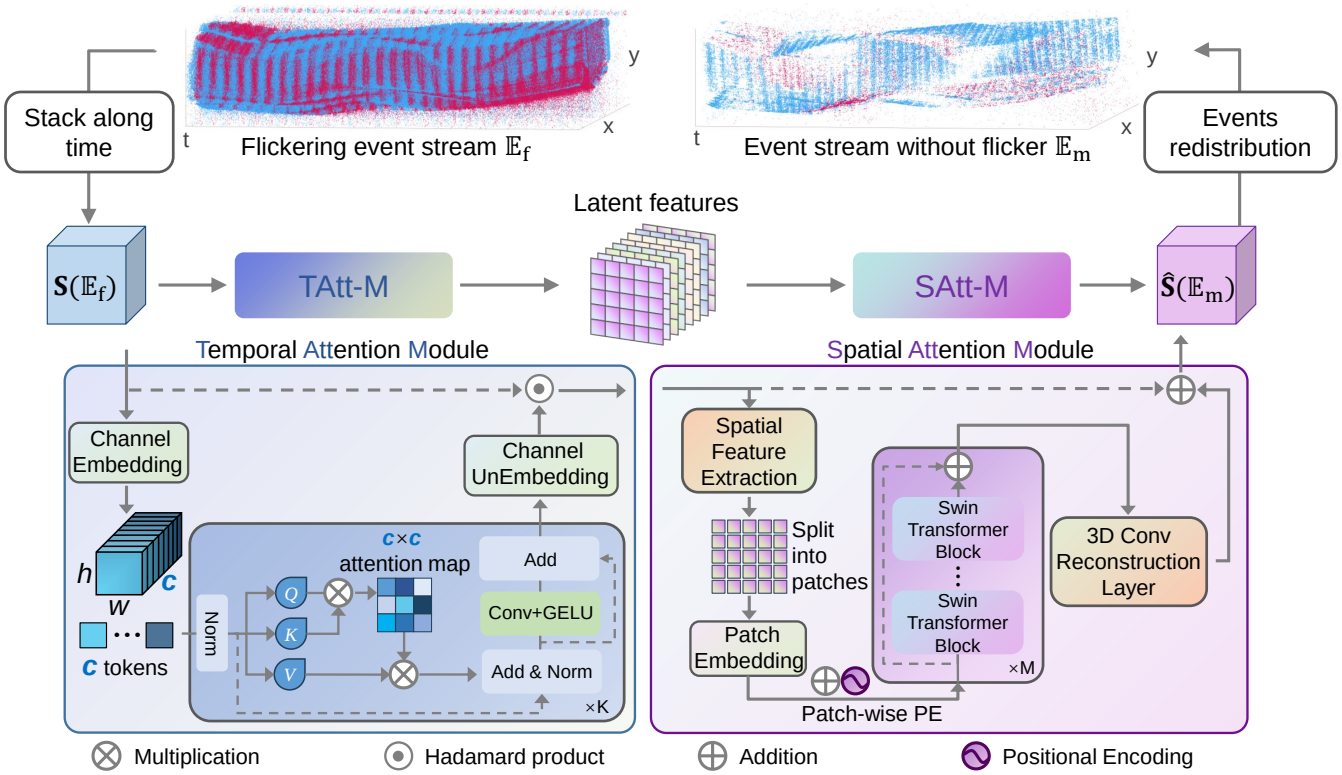

Figure 2: The overall pipeline and detailed architecture of EDeF-Net. The input flickering event stream is firstly stacked along the time axis. Then the event stack is passed through K temporal attention modules (TAtt-M) and M spatial attention modules (SAtt-M), which learn the intra-pixel temporal attention and inter-patch spatial attention. Finally, the output event stack is redistributed back to the event stream with flickering effect being removed.

features $\mathbf{L}$ after being processed with temporal attention in TAtt-M, we aim to model the connections among event signals triggered in different pixels by learning a spatial attention residual.

As shown in Figure 2, the SAtt-M mainly consists of a spatial feature extraction layer, followed by several residual swin transformer blocks [24] and a reconstruction layer. Since 2D convolution fuses all the channels together, we therefore apply 3D convolutions in the spatial feature extraction layer and the final reconstruction layer, which will not impact the temporal attention figured out in the previous TAtt-M. The extracted features from $\mathbf{L}$ is:

$$\mathbf{F}(\mathbf{L}) = \text{3DConv}(\mathbf{L}), \tag{5}$$

where $\text{3DConv}(\cdot)$ represents 3D convolution. After the spatial feature extraction, we split the tensor into patches, and conduct patch embedding as well as patch-wise positional encoding, which are the same as the operations in classical ViTs [5, 9, 14]. The following operation module is composed of several residual swin transformer blocks [24] with powerful and efficient modelling ability for visual information. Then the learned spatial attention residual is added back to the feature tensor and reconstructed by the 3D convolutional layers. The full operations in SAtt-M is formulated as:

$$\hat{\mathbf{S}}(\mathbb{E}_m) = \mathbf{L} + \text{3DConv}(\mathcal{R}(\mathbf{F}(\mathbf{L}) + pe_p)), \tag{6}$$

where $\mathcal{R}(\cdot)$ represents the residual swin transformer block, and pe_p is the patch-wise positional encoding. The values in event stack $\hat{\mathbf{S}}(\mathbb{E}_m)$ output from the SAtt-M is rounded to integers and redistributed back into a stream by assigning a timestamp for each event signal. Details of the redistribution method are introduced in the supplementary material.

3.3 Loss functions

Considering the sparse nature of event signals, which results in sparse input and output stacks for the network, we define the loss function as a combination of two parts:

$$\mathcal{L} = \mathcal{L}_{\ell_1} + \lambda \mathcal{L}_s, \tag{7}$$

where λ is a weight parameter that balances the two parts. The first part $\mathcal{L}_{\ell_1}$ is the ℓ_1 loss, which directly computes the mean absolute error (MAE) between the output and ground truth stacks. It helps to avoid the output stacks becoming excessively smooth, which violates the sparse characteristics of accumulated event stacks. The second part $\mathcal{L}_s$ represents the sparsity loss, which is designed to quantify the discrepancy in the number of valid (*i.e.*, non-zero) voxels between the prediction and the ground truth, which is defined as:

$$\mathcal{L}_s = \left\| N - \hat{N} \right\|_1, \tag{8}$$

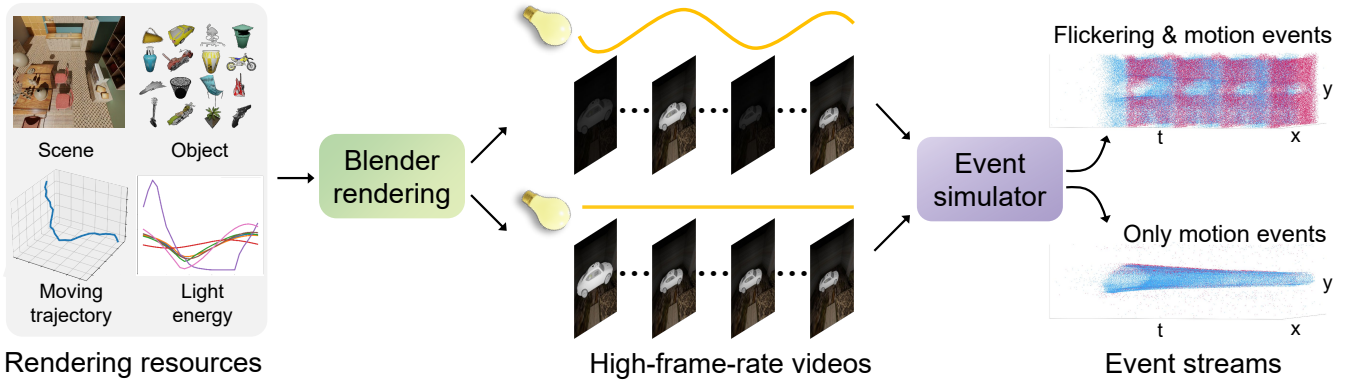

Figure 3: Our synthetic dataset creation pipeline consists of a rendering engine in Blender [7] and an event signals simulator [17]. The Blender renders HFR videos with different scenes, objects, moving trajectories, and light energies. Then the event signals simulator generates event streams according to the input HFR videos.

where N and $\hat{N}$ represent the number of valid voxels in $\mathrm{S}(\mathbb{E}_m)$ and $\hat{\mathrm{S}}(\mathbb{E}_m)$, respectively. $\|\cdot\|_1$ is the ℓ_1 distance.

3.4 Dataset preparation

There is no existing large-scale event stream pairs for training flicker removal models. Therefore, we synthesize a dataset with flickering event streams as input and the corresponding non-flickering event streams as the ground truth. The dataset simulation pipeline is illustrated in Figure 3, which is composed of two stages: HFR videos rendering and event data simulation.

For the first stage, we use the open source 3D rendering tool Blender [7] to synthesize HFR videos. The rendering is composed of four main resources: Scene (*i.e.*, environment), object, moving trajectory, and light source. To guarantee the diversity of the proposed dataset, we randomly set the environment with texture as a static background from the SceneNet [15], and randomly select objects from ShapeNet [6]. The random moving trajectories are set as the translation and rotation path of different objects. For the light sources, the flickering energy curves come from the database [32], which measured the response functions of various bulbs that dominated indoor and nocturnal outdoor lighting. We choose 6 different indoor bulbs and set 26 key frames in one flickering cycle. Each video is rendered with 4 flickering cycles, containing 832 frames with the resolution of 256×256. The flickering frequency of bulbs occurs in 100 Hz. Since the proposed EDeF-Net accumulates the events triggered in one flickering cycle as an event stack, it can deal with scenarios with light sources in different frequencies. For the corresponding non-flicker lighting, the only difference is the light energy, which is set to a constant value without fluctuation.

For the second stage, we use an event data simulator [17] to generate event streams given the HFR videos as input. To improve the generalization ability of our model to real event signals, we randomly set the logarithm thresholds with a mean of 0.3 and standard deviation of 0.05. We synthesize 4420 pairs of flickering / non-flickering event stacks in total, including 3508 event stacks for training and the other 912 for testing.

Table 1: Quantitative evaluation of flicker removal in the format of event stack. ↑ (↓) means the higher (lower) the better results throughout this paper. The champion results are marked in bold.

	MSE↓	MAE↓	SNR↑	PSNR↑
Erosion filter	0.373	0.215	1.107	18.16
Median filter	0.222	0.153	1.331	22.97
EFR [36]	0.195	0.091	0.531	15.61
EDeF-Net (Ours)	**0.157**	**0.081**	**2.591**	**23.13**

3.5 Implementation details

During the training process, we split each continuous event stream into small slices according to the flickering cycles (*e.g.*, under 100 Hz light source, we split 0.01s events into one stack). For each cycle, both of the flickering and non-flickering event streams are binned into an 8-channel stack. We have tried stacks with 4 and 16 channels, and found that 4-channel stacks sacrifice more temporal resolution, leading the output streams to look discrete; 16-channel stacks can hardly contain sufficient events in each bin, making it hard for the model to converge. In total, the synthetic 1105 event streams are split into 4420 event stacks, including 3508 pairs as the training set and 912 pairs for testing. The proposed EDeF-Net is a light-weighted network with only 1.051 M parameters, which is implemented by Pytorch [29]. We train it with a batch size of 2 on an NVIDIA 4090 GPU. 60 epochs make the model converge, which take around 48 hours. We use the Adam optimizer [21] with an initial learning rate of 10^{-3}, which linearly decays to zero after the first 50 epochs.

4 Experiments

4.1 Quantitative evaluation on synthetic dataset

Since we have the non-flickering event streams as the ground truth in the synthetic dataset, we can conduct quantitative evaluation on

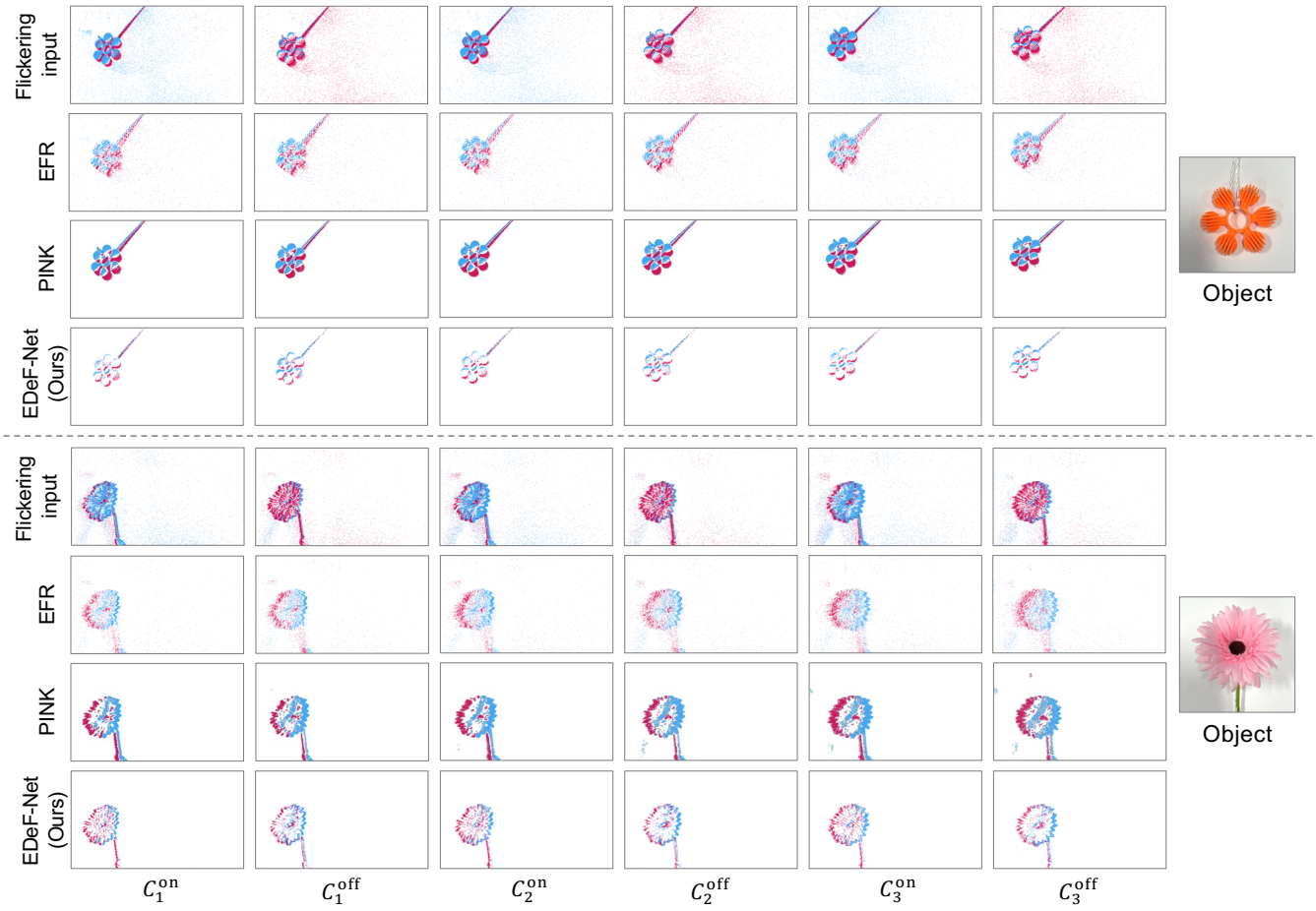

Figure 4: Qualitative comparison of two real flickering sequences and their deflickering results. The deflickering sequences from the state-of-the-art methods (EFR [36] and PINK [16]) and the proposed EDeF-Net (Ours) are shown below the flickering input sequences.

the event stacks. Several recognized metrics are applied for evaluation, including mean square error (MSE), mean absolute error (MAE), signal-to-noise ratio (SNR), and peak signal-to-noise ratio (PSNR). In SNR metrics, the noise part is computed from the difference between ground truth and the predicted event stacks. The results are listed in Tab. 1. We compare with two morphological filtering algorithms (*i.e.*, erosion filter and median filter) and the state-of-the-art method EFR [36]. The proposed EDeF-Net achieves better performance on the synthetic dataset.[1]

4.2 Qualitative comparison on real event sequences

We visualize the event streams as frame sequences by integrating the event signals triggered in each semi-flickering cycle (*i.e.*, C_i^{on} and C_i^{off}). There are three real-data examples shown in Figure 4. The input sequences are severely affected by the flickering effect with lots of positive events accumulated during C_i^{on} and negative events during C_i^{off}. The comparing algorithm EFR [36] focuses on flickering pattern in static pixels, which hampers its performance in scenarios where both flicker- and motion-triggered events appear on the same object. It is difficult for EFR [36] to clearly figure out whether the events should be filtered or not, which introduces severe blurry artifacts as shown in the second column of each sample in Figure 4. PINK [16] needs to integrate events along time that makes it difficult to redistribute back to event streams. The integration also introduce thick edges.

The proposed EDeF-Net can effectively filter out redundant events triggered by flickering light source and only preserve valid events triggered by motion, which solves the ambiguity of event triggering. The preserved events contain clear edges of the moving objects without interfered by flickering light source. Moreover, the noise on the background caused by flickering light is visible in the flickering input and results from EFR [36], but nicely removed in our results. In the following experiments, we conduct two event-based down-stream applications, including optical flow estimation and object tracking, to verify that after removing the flickering events,

[1]Due to the integration along time axis in PINK [16], we can only conduct qualitative evaluation on it.

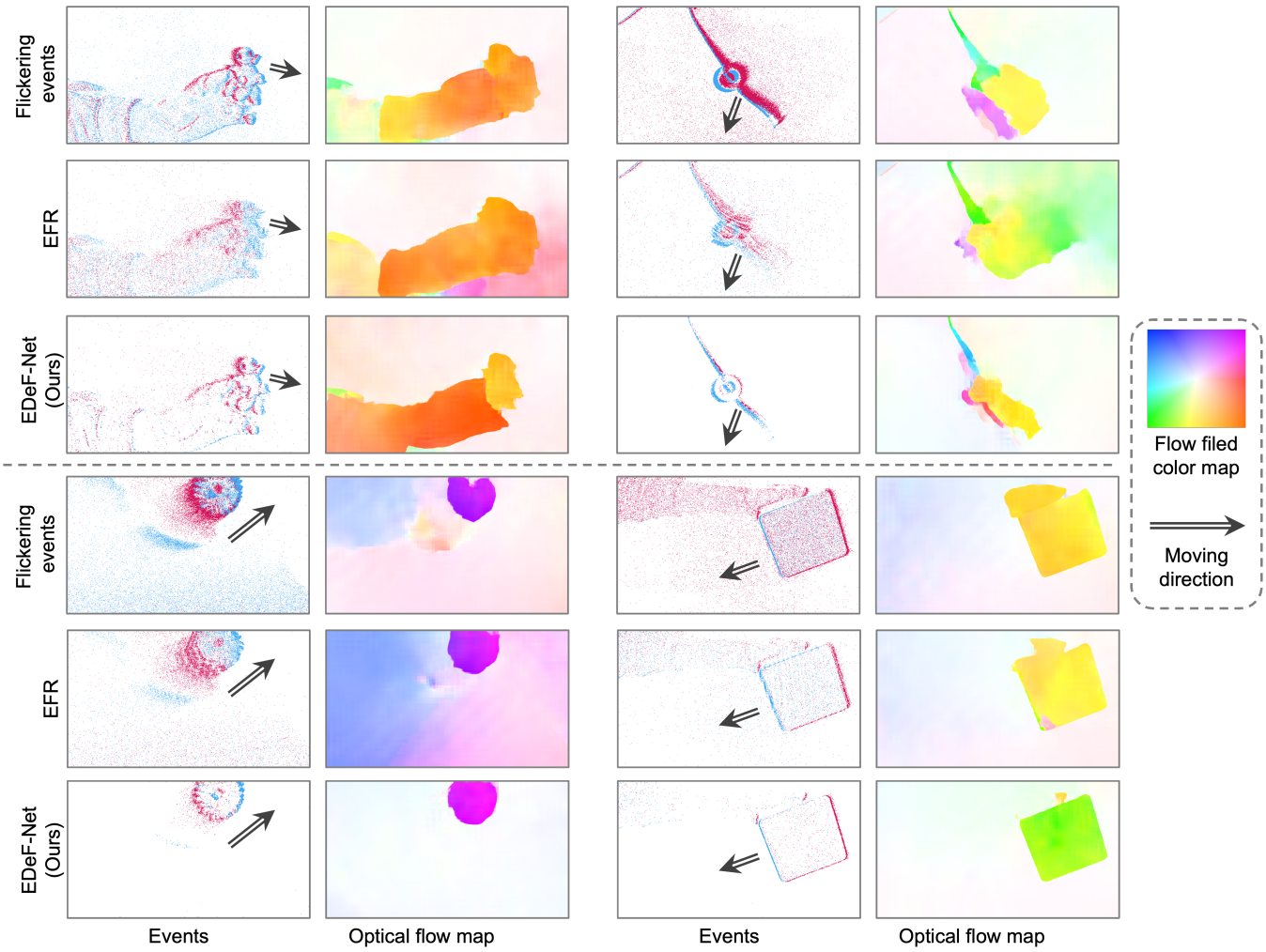

Figure 5: The results of dense optical flow estimation from BFlow [12]. The gray arrows indicate the actual moving directions in each case.

we can improve the performance of existing algorithms on scenarios with flickering light source. Please refer to the supplementary material for more visual results.

4.3 Downstream applications

Optical flow estimation. We compare the dense optical flow estimation results from the raw flickering events, the event streams filtered by EFR [36], and the streams output from the proposed EDeF-Net on several real-captured event sequences, as shown in Figure 5. The BFlow [12] model is used as the method for dense optical flow estimation. We can infer from the flow maps in Figure 5 that the results from EDeF-Net show sharper edges (the top two cases) of the objects, which is crucial to identify the moving objects and their moving direction. Besides, the events triggered in the background have influence on the optical flow estimation model (the bottom left case). Our method effectively removes the flickering events in the background, which significantly improve the accuracy of the flow estimation. In the last case, the results from flickering events and EFR [36] made wrong estimation, while the proposed EDeF-Net preserves the motion-triggered events only, resulting in correct flow estimation from the model [12].

Event-based object tracking. We choose an event-based object tracking method: STNet [39] as the benchmark tracker, which directly outputs the bounding box in an end-to-end manner. We use both synthetic data and real data to evaluate the performance of single object tracking on the raw flickering event streams and the corresponding streams after flicker removal. On synthetic data, we manually label the bounding box for each frame and evaluate by the intersection over union (IoU) and the center location distance (DIS) between the predicted and ground truth bounding boxes. The qualitative and quantitative evaluation results are shown in Figure 6 and Tab. 2, respectively. Both of the visual examples and evaluation metrics demonstrate the improvement of the performance of object tracker [39] on the filtered event streams, where the redundant flickering events have been removed by the proposed EDeF-Net.

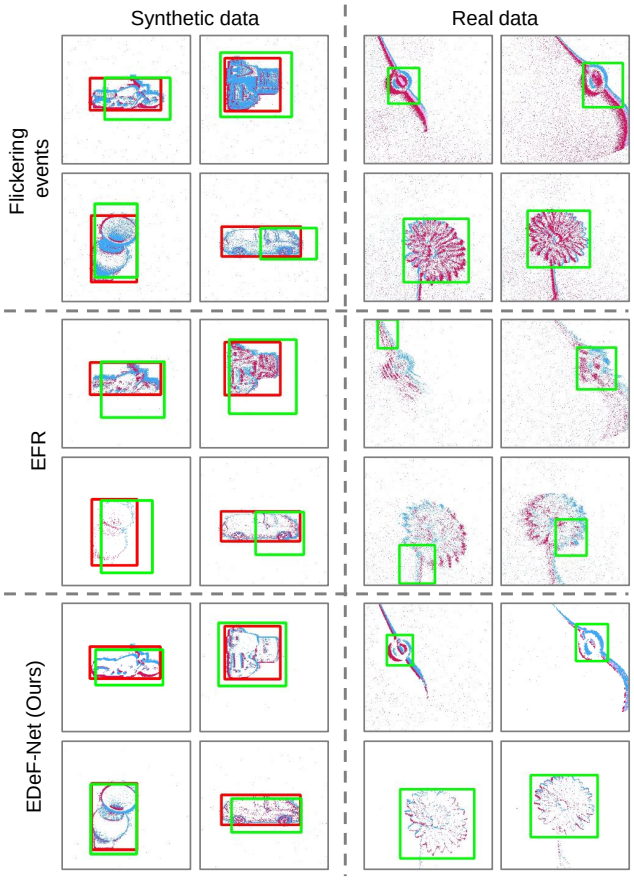

Figure 6: The tracking results on synthetic and real event sequences. The red and green bounding boxes represent the ground truth and prediction, respectively (There is no ground truth of bonding box for real data).

Table 2: Quantitative evaluation of event-based object tracking on 20 synthetic samples.

	Flickering events	EFR [36]	EDeF-Net (Ours)
IoU↑	0.640	0.534	**0.658**
DIS↓	21.93	30.75	**16.82**

However, the comparing method EFR [36] mixed the temporal order of the flickering event streams, which introduces blurry effect that has impact on the accuracy of object tracking. Figure 6 shows the real data results of object tracking. It is obvious that the event streams filtered by EDeF-Net can yield higher tracking accuracy compared to that from the raw flickering events and events filtered by EFR [36].

4.4 Ablation study

To evaluate the effectiveness of modules and design in the proposed EDeF-Net, we conduct experiments on different model variants, as summarized in Tab. 3. In the first two variants, we remove the temporal attention module (W/o TAtt-M) or spatial attention module (W/o SAtt-M) from EDeF-Net, respectively. The first variant only computes the spatial attention among patches without considering the intra-pixel temporal correlation. Therefore, it is hard to learn the flickering pattern along the temporal axis. The variant without SAtt-M does not preserve the spatial consistency, resulting in performance degradation. Removing the SAtt-M affects more on the evaluation metrics. Because the output of SAtt-M is residual addition (+) that is easier for network to adapt. We further validate the effectiveness of 3D convolution by substituting them with 2D convolutions in SAtt-M. The 3D convolutions assign filters with different weights to different channels, which play an important role in processing tensors with important temporal information. To demonstrate the contribution of sparsity loss, we validate train a network variant without $\mathcal{L}_s$, whose performance degrades compared to the complete model.

Table 3: Quantitative evaluation of ablation studies.

	MSE↓	MAE↓	SNR↑	PSNR↑
W/o TAtt-M	0.192	0.096	2.329	23.84
W/o SAtt-M	0.200	0.098	2.315	23.75
2D Conv	0.164	0.084	2.488	24.40
W/o $\mathcal{L}_s$	0.161	0.084	2.534	24.42
Complete model	**0.158**	**0.082**	**2.591**	**24.45**

5 Conclusion

In this paper, we propose to solve the problem of the ambiguity that whether an event is triggered from motion or flickering light source, and remove the redundant flickering event signals. We introduce EDeF-Net, a network with specially designed temporal attention and spatial attention modules. It effectively removes the redundant events triggered by flickering light sources and preserves the valid events triggered only by motion. Besides, we synthesize a dataset with flickering and non-flickering event streams for training and evaluation. We have conducted extensive experiments on the performance of flickering events filtering and several down-stream applications. The quantitative and qualitative evaluations demonstrated that the EDeF-Net can effectively remove the flickering events. The flicker-free event streams filtered by EDeF-Net yield performance improvement in the down-stream applications.

Limitations. Compared to the classical filter-based methods, the proposed EDeF-Net requires a training process. Besides, the event redistribution may have impact on the temporal resolution of the filtered event streams if the time window in a stack is too long.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant No. 62088102, 62136001); Beijing Natural Science Foundation (Grant No. L233024); Beijing Municipal Science & Technology Commission, Administrative Commission of Zhongguancun Science Park (Grant No. Z241100003524012); and JST-Mirai Program (Grant No. JPMJM123G1).

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] R Baldwin, Mohammed Almatrafi, Vijayan Asari, and Keigo Hirakawa. 2020. Event probability mask (EPM) and event denoising convolutional neural network (EDnCNN) for neuromorphic cameras. In *Proc. of Computer Vision and Pattern Recognition*.

[3] Nicolas Bonneel, James Tompkin, Kalyan Sunkavalli, Deqing Sun, Sylvain Paris, and Hanspeter Pfister. 2015. Blind video temporal consistency. *ACM Transactions on Graphics (TOG)* (2015).

[4] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. 2014. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *Journal of Solid-State Circuits* (2014).

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Proc. of European Conference on Computer Vision*.

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012* (2015).

[7] Blender Online Community. [n. d.]. Blender: A 3D modelling and rendering package. http://www.blender.org.

[8] Julie Delon and Agnes Desolneux. 2010. Stabilization of flicker-like effects in image sequences through local contrast correction. *SIAM Journal on Imaging Sciences* (2010).

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. of International Conference on Learning Representations*.

[10] Gabriel Eilertsen, RafałMantiuk, and Jonas Unger. 2019. Single-frame Regularization for Temporally Stable CNNs. In *Proc. of Computer Vision and Pattern Recognition*.

[11] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. 2020. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[12] Mathias Gehrig, Manasi Muglikar, and Davide Scaramuzza. 2024. Dense Continuous-Time Optical Flow from Event Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[13] Shasha Guo and Tobi Delbruck. 2022. Low cost and latency event camera background activity denoising. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[14] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. 2022. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[15] A Handa, V Patraucean, V Badrinarayanan, S Stent, and R Cipolla. 2015. SceneNet: Understanding real world indoor scenes with synthetic data. *arXiv preprint arXiv:1511.07041* 3 (2015).

[16] Gyubeom Im, Keunjoo Park, Junseok Kim, Bongki Son, Seungchul Shin, and Haechang Lee. 2023. Live Demonstration: PINK: Polarity-based Anti-flicker for Event Cameras. In *Proc. of Computer Vision and Pattern Recognition Workshops*.

[17] Damien Joubert, Alexandre Marcireau, Nic Ralph, Andrew Jolley, André van Schaik, and Gregory Cohen. 2021. Event camera simulator improvements via characterized parameters. *Frontiers in Neuroscience* (2021).

[18] Dachun Kai, Jiayao Lu, Yueyi Zhang, and Xiaoyan Sun. 2024. EvTexture: Event-driven Texture Enhancement for Video Super-Resolution. In *Proc. of International Conference on Machine Learning*.

[19] Ali Kanj, Hugues Talbot, and Raoul Rodriguez Luparello. 2017. Flicker removal and superpixel-based motion tracking for high speed videos. In *Proc. of International Conference on Image Processing*.

[20] Alireza Khodamoradi and Ryan Kastner. 2018. O(N)-Space Spatiotemporal Filter for Reducing Noise in Neuromorphic Vision Sensors. *IEEE Transactions on Emerging Topics in Computing* (2018).

[21] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[22] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. 2018. Learning Blind Video Temporal Consistency. In *Proc. of European Conference on Computer Vision*.

[23] Chenyang Lei, Yazhou Xing, and Qifeng Chen. 2020. Blind video temporal consistency via deep video prior. In *Proc. of Conference on Neural Information Processing Systems*.

[24] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. SwinIR: Image restoration using swin transformer. In *Proceedings of International Conference on Computer Vision Workshops*.

[25] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. 2008. A 128×128 120 dB 15μs Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits* (2008).

[26] Hongjie Liu, Christian Brandli, Chenghan Li, Shih-Chii Liu, and Tobi Delbruck. 2015. Design of a spatiotemporal correlation filter for event-based sensors. In *Proc. of International Symposium on Circuits and Systems*.

[27] Yuji Nozaki and Tobi Delbruck. 2017. Temperature and parasitic photocurrent effects in dynamic vision sensors. *IEEE Transactions on Electron Devices* (2017).

[28] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. 2019. Bringing a blurry frame alive at high frame-rate with an event camera. In *Proc. of Computer Vision and Pattern Recognition*.

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proc. of Conference on Neural Information Processing Systems*.

[30] Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. 2020. Learning to detect objects with a 1 megapixel event camera. In *Proc. of Conference on Neural Information Processing Systems*.

[31] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. 2019. Events-to-video: Bringing modern computer vision to event cameras. In *Proc. of Computer Vision and Pattern Recognition*.

[32] Mark Sheinin, Yoav Y Schechner, and Kiriakos N Kutulakos. 2017. Computational imaging on the electric grid. In *Proc. of Computer Vision and Pattern Recognition*.

[33] Chenyang Shi, Yuzhen Li, Ningfang Song, Boyi Wei, Yibo Zhang, Wenzhuo Li, and Jing Jin. 2023. Identifying Light Interference in Event-Based Vision. *IEEE Transactions on Circuits and Systems for Video Technology* (2023).

[34] Jiaxu Wang, Junhao He, Ziyi Zhang, Mingyuan Sun, SUN Jingkai, and Renjing Xu. 2024. EvGGS: A Collaborative Learning Framework for Event-based Generalizable Gaussian Splatting. In *Proc. of International Conference on Machine Learning*.

[35] Yanxiang Wang, Bowen Du, Yiran Shen, Kai Wu, Guangrong Zhao, Jianguo Sun, and Hongkai Wen. 2019. EV-Gait: Event-based robust gait recognition using dynamic vision sensors. In *Proc. of Computer Vision and Pattern Recognition*.

[36] Ziwei Wang, Dingran Yuan, Yonhon Ng, and Robert Mahony. 2022. A linear comb filter for event flicker removal. In *Proc. of International Conference on Robotics and Automation*.

[37] Lexuan Xu, Guang Hua, Haijian Zhang, Lei Yu, and Ning Qiao. 2023. "Seeing" Electric Network Frequency From Events. In *Proc. of Computer Vision and Pattern Recognition*.

[38] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. 2022. Restormer: Efficient Transformer for High-Resolution Image Restoration. In *Proc. of Computer Vision and Pattern Recognition*.

[39] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. 2022. Spiking transformers for event-based single object tracking. In *Proc. of Computer Vision and Pattern Recognition*.

EDeF-Net: Spatio-temporal Association Network for Flicker Removal in Event Streams –Supplementary Material–

Jin Han*
The University of Tokyo
Tokyo, Japan
hanjin@pku.edu.cn

Yixin Yang†
Peking University
Beijing, China
yangyixin93@pku.edu.cn

Zhan Zhan*
The University of Tokyo
Tokyo, Japan
zhanzhanchn@gmail.com

Boxin Shi†‡
Peking University
Beijing, China
shiboxin@pku.edu.cn

Imari Sato*
National Institute of Informatics
Tokyo, Japan
imarik@nii.ac.jp

ACM Reference Format:
Jin Han, Yixin Yang, Zhan Zhan, Boxin Shi, and Imari Sato. 2025. EDeF-Net: Spatio-temporal Association Network for Flicker Removal in Event Streams –Supplementary Material–. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3746027.3754995

6 Visualization of Attention Maps

We present visualizations of the attention maps calculated by our temporal attention module (TAtt-M) and spatial attention module (SAtt-M) to demonstrate the effectiveness of these two modules in modeling the temporal correlation and spatial consistency of event streams. As shown in Figure 8, we visualize the channel-wise temporal attention and spatial attention maps of a real-world example. The flickering stack that covers a whole flickering cycle is split into 8 channels, which clearly show the global events triggered by the flickering cycle at that time window (*e.g.*, negative events in channel 2-4, and positive events in channel 5-7). The average value of each temporal attention map is highly related to the dominant events at that channel. For example, the temporal attention maps at the central channels (*i.e.*, channels 2-7) are darker compared to the sided channels (*i.e.*, channels 1 and 8), which indicates that those temporal attention maps try to **refrain** the flicker-triggered global events at those channels. It shows that the TAtt-M has learned to give different attention weights to different channels in one event stack. Meanwhile, in the temporal attention maps, the pixels in the object edges keep high values among all the channels, which means the temporal attention does not affect the spatial consistency. In essence, the visualization of temporal attention validates the proficiency of TAtt-M in discerning and leveraging the temporal correlation within event streams.

For the spatial attention map in Figure 8, it exhibits **higher uniformity** among the pixels where events exist, enhancing the preservation of spatial structure and consistency. Through the strategic allocation of varied attention weights to distinct event patches, it highlights critical features while diminishing the prominence of non-essential regions. This selective focus significantly maintains the spatial consistency within event streams.

7 Events Redistribution

In this section, we introduce post-processing of the network's output and how to redistribute the events back into the streams from the stacks. Since the output of the network is an 8-channel stack with floating-point values, our initial step is to **round off** these floating-point values, which filters out the flickering part and preserves the valid events triggered by motion. Then we conduct events redistribution in a channel-wise manner. As illustrated in Figure 11, the proposed EDeF-Net removes flicker-triggered events (*i.e.*, the dotted circles) in each channel of the event stack. For the remaining valid events in each channel, each of them is assigned with a **random timestamp** within the time window. This procedure ensures that the sequential integrity of event signals is maintained during the conversion from stack format to stream. Although we cannot restore the exactly original timestamps of the preserved events, this approach significantly enhances the temporal consistency of the event signals. This method strikes a balance between maintaining the structural integrity of event streams and addressing the challenge of timestamp precision loss.

8 Robustness of EDeF-Net

The proposed EDeF-Net is designed to remove the flickering events while preserving the motion-triggered events in scenarios with

*Jin Han, Zhan Zhan, and Imari Sato are with National Institute of Informatics and The University of Tokyo, Tokyo, Japan.
†Yixin Yang and Boxin Shi are with State Key Laboratory of Multimedia Information Processing and National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, Beijing, China.
‡Corresponding author.
Project page: https://github.com/hjynwa/EDeF-Net

MM '25, Dublin, Ireland.

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3754995

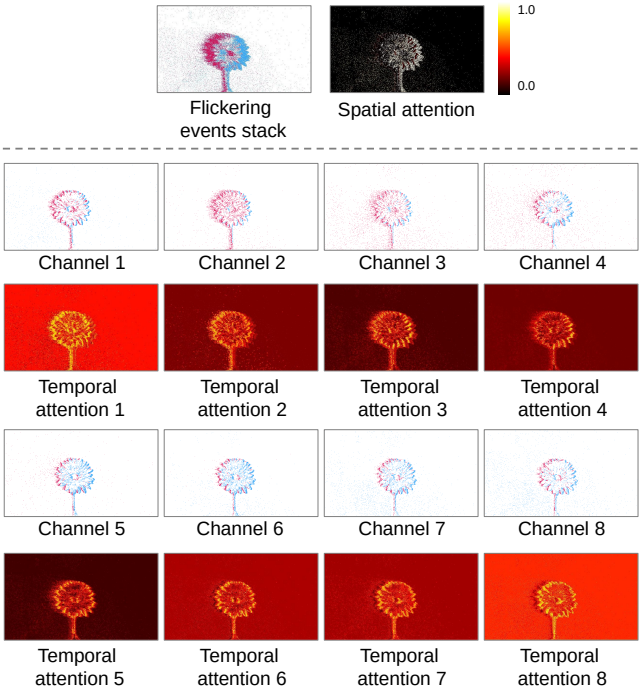

Figure 8: Visualization of the temporal attention and spatial attention of a real-world example. The flickering events stack (top left) is split to display each individual channel. The corresponding temporal residual maps are shown under each channel of the flickering stack. The spatial attention map is shown under the flickering events stack.

flickering light sources. However, in scenarios with constant lighting, there is no flickering events triggered. Therefore, it is necessary to keep the original event signals unaffected. The experiments demonstrate that EDeF-Net performs robustly on such kind of scenarios and ensures that meaningful motion events are retained while minimizing the impact of any other remaining noise. As the results of non-flickering scenarios shown in the Figure 9, the input events streams are captured under constant lighting. The comparing EFR [2] and PINK [1] cannot adapt to this scenario and introduce blurry artifacts in the results. The proposed EDeF-Net can effectively preserve the events on the moving objects and remove the noise in the background. This adaptability is crucial for applications in various scenarios with different lighting conditions, as it ensures that relevant event signals are well-preserved for post-processing. Besides, the proposed EDeF-Net can be easily adapted to flickering light sources with different frequencies. Since the flickering cycles can be estimated by sampling the positive and negative events in temporal axis, such as Xu *et al.* [3] did in their paper, we can stack the event signals based on the estimated flickering cycle and feed it into the EDeF-Net for flicker removal.

9 Computational Cost

The inference time and FLOPs for an event stack (with shape of 256 × 256 × 8) are 67.92 G 0.37 s on an NVIDIA V100 GPU. We agree

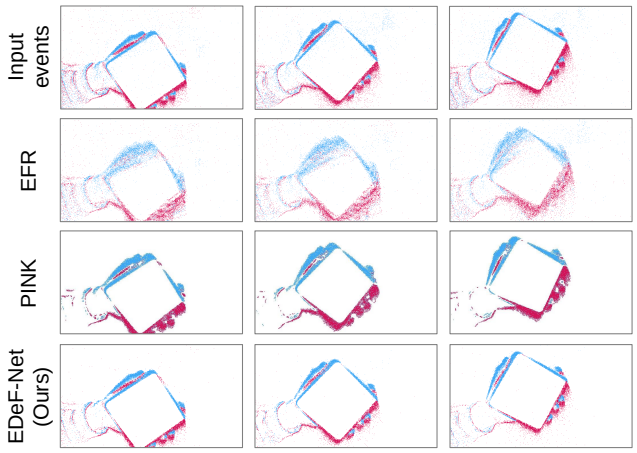

Figure 9: The qualitative comparison of the filtered results on non-flickering scenarios. The first column shows the input event streams, and the second and third columns show the results from EFR [2], PINK [1], and the proposed EDeF-Net, respectively.

that the self-attention mechanisms may affect runtime on edge devices. However, we can develop optimized transformer operators tailored for edge devices to significantly reduce latency and make EDeF-Net more suitable for real-time embedded applications.

10 Unknown Flicker Frequency

In our current setup, we assume that the AC flicker frequency (*e.g.*, 100 Hz) is known or can be measured from short calibration. For example, by applying E-ENF [3] to the first small period of events, the flickering frequency can be estimated by sampling the positive and negative events in temporal axis. Then we can stack the event signals based on the estimated flickering frequency, and the proposed EDeF-Net can be easily adapted to different flickering light sources. This stacking process is decoupled from the design of EDeF-Net itself, which operates on the assumption that the input stack contains a complete flicker cycle.

11 Additional Results of EDeF-Net

In Figure 10, we show further qualitative outcomes of deflickering on real-world event data. Furthermore, we present a **supplementary video** to demonstrate the effectiveness of EDeF-Net. The supplementary video provides the results of event stream deflickering and extending to two applied scenarios: event-based optical flow estimation and object tracking.

References

[1] Gyubeom Im, Keunjoo Park, Junseok Kim, Bongki Son, Seungchul Shin, and Haechang Lee. 2023. Live Demonstration: PINK: Polarity-based Anti-flicker for Event Cameras. In *Proc. of Computer Vision and Pattern Recognition Workshops.*

[2] Ziwei Wang, Dingran Yuan, Yonhon Ng, and Robert Mahony. 2022. A linear comb filter for event flicker removal. In *Proc. of International Conference on Robotics and Automation.*

[3] Lexuan Xu, Guang Hua, Haijian Zhang, Lei Yu, and Ning Qiao. 2023. "Seeing" Electric Network Frequency From Events. In *Proc. of Computer Vision and Pattern Recognition.*

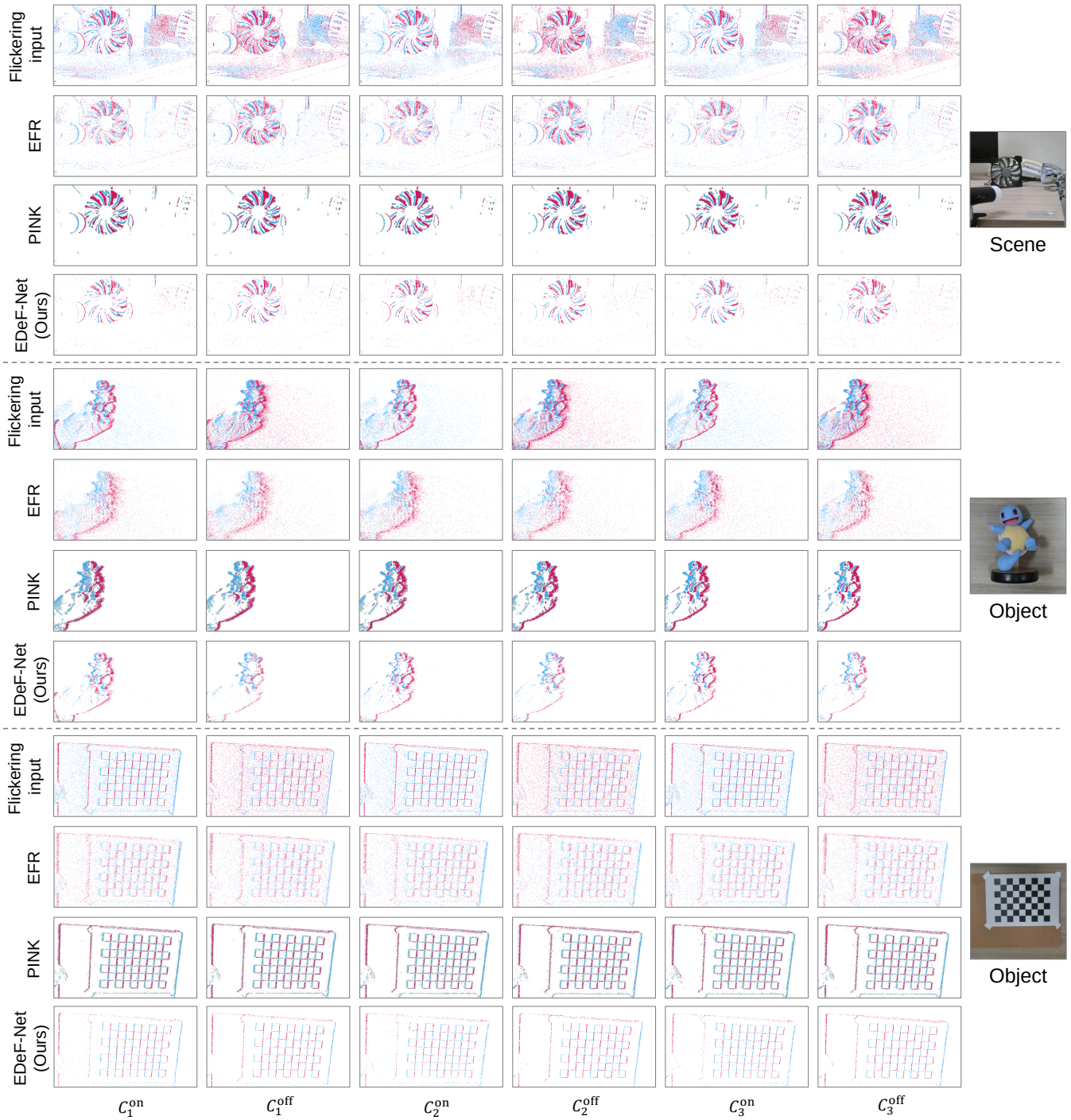

Figure 10: Qualitative comparison of real flickering sequences and their deflickering results. Please zoom in for more details.

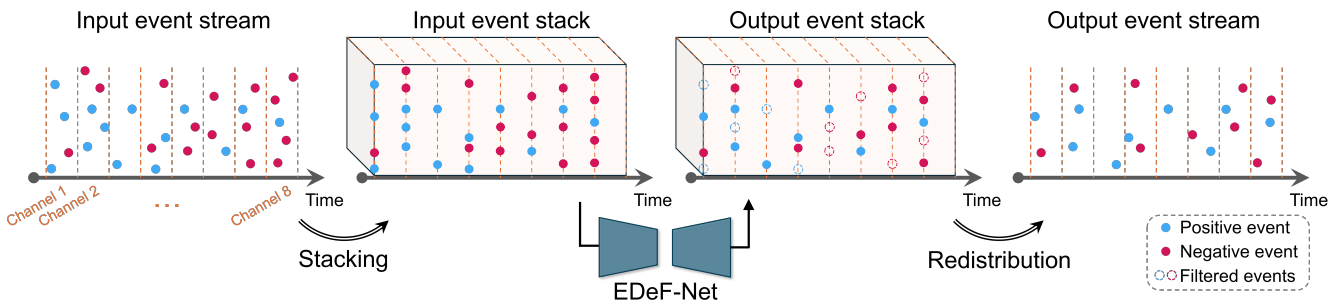

Figure 11: The process of event stream stacking and redistribution. The dotted circles represent the events that are filtered out by the proposed EDeF-Net.