MENet: a Memory-Based Network with Dual-Branch for Efficient Event Stream Processing

Linhui Sun[1,2], Yifan Zhang[1,2, ⋆], Ke Cheng[1,2], Jian Cheng[1,2], and Hanqing Lu[1,2]

[1] Institute of Automation, Chinese Academy of Sciences, 100190, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, 100049, Beijing, China
{sunlinhui2018,chengke2017}@ia.ac.cn
{yfzhang, jcheng, luhq}@nlpr.ia.ac.cn

Abstract. Event cameras are bio-inspired sensors that asynchronously capture per-pixel brightness change and trigger a stream of events instead of frame-based images. Each event stream is generally split into multiple sliding windows for subsequent processing. However, most existing event-based methods ignore the motion continuity between adjacent spatiotemporal windows, which will result in the loss of dynamic information and additional computational costs. To efficiently extract strong features for event streams containing dynamic information, this paper proposes a novel memory-based network with dual-branch, namely MENet. It contains a base branch with a full-sized event point-wise processing structure to extract the base features and an incremental branch equipped with a light-weighted network to capture the temporal dynamics between two adjacent spatiotemporal windows. For enhancing the features, especially in the incremental branch, a point-wise memory bank is designed, which sketches the representative information of event feature space. Compared with the base branch, the incremental branch reduces the computational complexity up to 5 times and improves the speed by 19 times. Experiments show that MENet significantly reduces the computational complexity compared with previous methods while achieving state-of-the-art performance on gesture recognition and object recognition.

Keywords: Event-based model, Dual-branch structure, Memory bank

1 Introduction

Event cameras [4, 30, 49] are novel sensors that represent visual information by sparse and asynchronous events. Different from traditional cameras that record synchronized frames at a fixed low-rate (typically less than 60 frames per second), event cameras trigger an individual event asynchronously when the brightness

[⋆] Corresponding author

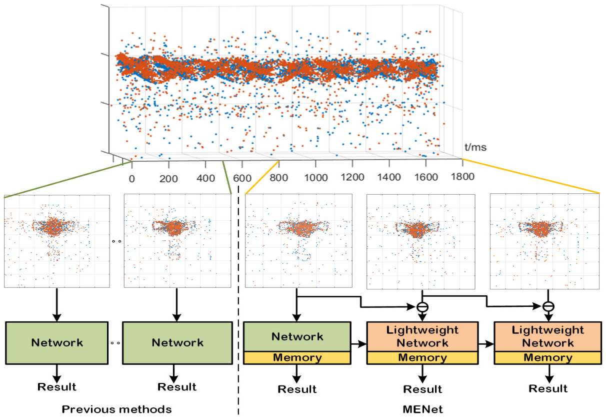

Fig. 1. Top: The event stream of an arm roll gesture is shown in the time-width-height space, in which the red dots represent that the polarity of an event is positive and the blue dots represent negative polarity. **Bottom:** The way to process adjacent windows of previous methods and that of the proposed MENet, respectively. Note that for intuitively representing the events contained in sliding windows, we transform it into a frame-based form by compressing the time dimension.

change on a pixel exceeds a preset threshold at a high rate. Each event encodes the pixel location, trigger time, and polarity of the brightness change. Compared with traditional cameras, event cameras exhibit four attractive properties. Firstly, event cameras are low latency, because they trigger event immediately when the intensity change exceeds the threshold. Secondly, event cameras only transmit changed information, thus they are low power. Thirdly, the high temporal resolution (µs) of event-based data can avoid motion blur. Fourthly, event cameras have a high dynamic range (140 dB vs 60 dB of traditional cameras), thus they can acquire information under challenging lighting conditions. These characteristics bring advantages to event cameras over traditional cameras when facing tasks that require low latency, low power, robustness to high-speed motion and variant illumination. Therefore, event cameras are widely used in many applications, such as object recognition [7, 43, 58], gesture recognition [1, 3, 64], pose relocalization [40, 55], 3D reconstruction [11, 23, 52], autonomous driving [10, 33], optical flow estimation [46, 67], video reconstruction [19, 45, 54], etc.

To take the advantage of event-based data in downstream tasks, extracting meaningful features efficiently and effectively from the event stream is one of the key steps. Some previous methods [1, 31, 55] proposed to operate on the event-based data through event-by-event processing. However, processing each event serially will accumulate a large time consumption and an event alone can not provide enough information. Therefore, following other methods [8, 12, 33, 40, 59], this paper operates on groups of events contained in sliding windows. In this way, the accumulated events are processed in parallel, which can extract sufficient information and improve processing efficiency.

However, as shown in Fig. 1, the information contained in adjacent windows is continuous and some of it is redundant. And most previous methods [8, 12, 33, 40, 59] ignored the correlation between adjacent inputs and process them independently and equally. It will cause useless computational costs and the loss of dynamic information. This paper proposes a novel memory-based network with dual-branch, namely MENet, which utilizes the dynamic correlation between adjacent windows and avoids repeated extraction of redundant information.

For extracting meaningful information from sliding windows effectively and efficiently, a base branch and an incremental branch are designed to form a dual-branch structure. The base branch with a full-sized event point-wise processing structure, aims at extracting base features. The incremental branch is equipped with a light-weighted structure, which inputs the differences between two spatiotemporal windows to capture temporal dynamics. Furthermore, for obtaining high-quality differences between adjacent windows, a double polarities calculation method is proposed, which only records the changed event information between two windows and retains the low power characteristic of event-based data. Thanks to the proposed dual-branch structure and the double polarities calculation method, the inference accuracy and efficiency are both improved.

For utilizing the information extracted by the base branch to enhance features, a point-wise memory bank is proposed, which aims at sketching the representative information of the event feature space. For gesture recognition, the memory bank records the motion pattern of each action, while for object recognition, it records discrimination information among categories. Therefore, through adaptively recalling the information stored in the memory, both the base branch and incremental branch can perform feature enhancement to improve accuracy.

The contributions of this paper are summarized as follows:

1. We propose a MENet with dual-branch to utilize the correlation between adjacent spatiotemporal windows to improve feature extraction efficiency and prediction accuracy. The base branch with a full-sized event point-wise processing structure extracts base features, while the light-weighted incremental branch captures temporal dynamics between adjacent windows.
2. We propose a double polarities calculation method that calculates the high-quality differences between adjacent windows with little time consumption.
3. We introduce a point-wise memory bank to MENet for recording representative information of the event feature space, which can be recalled adaptively to further enhance features for improving estimation accuracy.
4. Experiments show that the MENet achieves state-of-the-art results on gesture recognition and object recognition while significantly reducing the computational complexity with respect to previous methods.

2 Related Work

2.1 Event-Based Representations

According to the number of events processed simultaneously, event-based methods can be divided into two categories. The first type operates on an event-by-event basis, which can update the estimation upon the arrival of a single event. For event-by-event processing, event-based data can be compressed into a 2D map, namely time surface (TS) [27], in which each pixel records the timestamp of the most recent event. Although the representation of TS is applied in many tasks[32, 39, 58, 66], their ability will degrade on dealing with textured scenes [38], in which pixels spike frequently. In addition to the TS-based methods, Spiking Neural Networks (SNNs) [31, 41, 43] are adopted to process a single event, which is also bio-inspired designed. However, the training phase of SNNs is difficult because the output spikes are non-differentiable. Besides, Li *et al.* [62] proposed a graph-based method to process single event asynchronously. Sekikawa *et al.* [55] designed a recursive and event-wise manner to process event streams. Although these methods can respond immediately when a new event arrives, the serial processing of event data will accumulate a large time consumption due to the high time dimension characteristic of events.

The other type of method operates on sliding windows containing groups of events, which are obtained by splitting event streams with a fixed time interval or event number. For utilizing existing methods based on deep neural networks (DNNs), some methods [8, 40, 53] compressed sliding windows into 2D frames. Such intuitive expression retains the spatial information about scene edges, thus it can be applied in low-level and mid-level problems [2, 13, 44]. However, these methods discard the sparsity nature of events and quantify the timestamps. For improving the temporal resolution, events are converted into 3D voxel grids [6, 35, 67]. However, the computation of 3D convolution is expensive. Different from these methods that use alternative representations, some methods treated groups of events as event clouds [3, 9, 37, 54] to retain the high temporal resolution characteristic of them. Benosman *et al.* [21] computed the dense visual flow by introducing plane fitting. Wang *et al.* [59] utilized PointNet++ [50] for gesture recognition, which aggregated local and global features. However, these methods ignore the correlation between adjacent windows and just process them independently and equally, which will cause useless computational costs and the loss of dynamic information. Therefore, the proposed MENet introduces a dual-branch structure to utilize the relationship between adjacent windows.

2.2 Memory-Based Networks

Recently, memory networks have been introduced in many computer vision tasks, such as anomaly detection [14, 47], few-shot learning [5, 20, 68], video captioning [48], video prediction [29], etc. For a memory module, how to update important information to memory and how to recall effective content from memory are critical issues. Weston *et al.* [60] firstly proposed an additional memory component to deal with the task of question answer, which overcomes the drawback of limited memory of recurrent networks (RNNs). Huang *et al.* [17] introduced a self-supervised memory module to record the prototypical patterns of rain degradations for image deraining. To utilize long-term context for short-term image prediction, Lee *et al.* [29] introduced a long-term motion context memory (LMC-Memory) with an additional matrix, which is updated through back-propagation.

For understanding the unstructured documents, the Key-Value Memory Network [36] utilized key memory to infer the weight of the corresponding value memory to obtain the fused features. For sketching the representative information of the event feature space, we introduce a memory bank to event-based data, which only utilizes information extracted by the base branch to update memory and can be adaptively recalled to improve the prediction accuracy.

3 Event Camera Model

Event cameras capture the change in logarithmic brightness signal $L(u_i, t_i) = logI(u_i, t_i)$ of each pixel $u_i = (x_i, y_i)$. Let ΔL denotes the change at pixel location (x_i, y_i) between timestamp t_i and t_{i-1}:

$$\Delta L = L(u_i, t_i) - L(u_i, t_{i-1}) \tag{1}$$

When ΔL exceeds a preset threshold C, an event will be triggered asynchronously. Each event $e_i = (x_i, y_i, t_i, p_i)$ encodes the pixel location (x_i, y_i), trigger time t_i, and polarity of the brightness change $p_i \in \{-1, 1\}$.

$$p = \begin{cases} 1, \Delta L \geqslant C \\ -1, \Delta L \leqslant -C \end{cases} \tag{2}$$

An asynchronous event stream can be split into multiple sliding windows with a fixed time interval T or event number N_{num}:

$$\begin{cases} S_k^T = \{e_i | i = j, ..., j + n(j)\} \\ S_k^N = \{e_i | i = j, ..., j + N_{num}\} \end{cases} \tag{3}$$

where S_k^T and S_k^N represent kth sliding window based on T or N_{num}, respectively. $n(j)$ represents the number of events between the time of the first event in the kth sliding window t_j and the time $(t_j + T)$. In this paper, sliding windows are obtained based on a fixed time interval, and the step size is set as $T/2$.

4 Method

For effectively and efficiently extracting meaningful information contained in event streams, this paper proposes a novel memory-based network with dual-branch, namely MENet, as illustrated in Fig. 2. In order to utilize the correlation between adjacent spatiotemporal windows, a dual-branch structure (Section 4.1) is introduced. In addition, we propose a double polarities calculation method to obtain high-quality differences between two adjacent windows (Section 4.2). Furthermore, a point-wise memory bank is introduced to sketch the representative information of the event feature space, which can be adaptively recalled to perform feature enhancement (Section 4.3). In Section 4.4, the details of the training and testing strategies will be described.

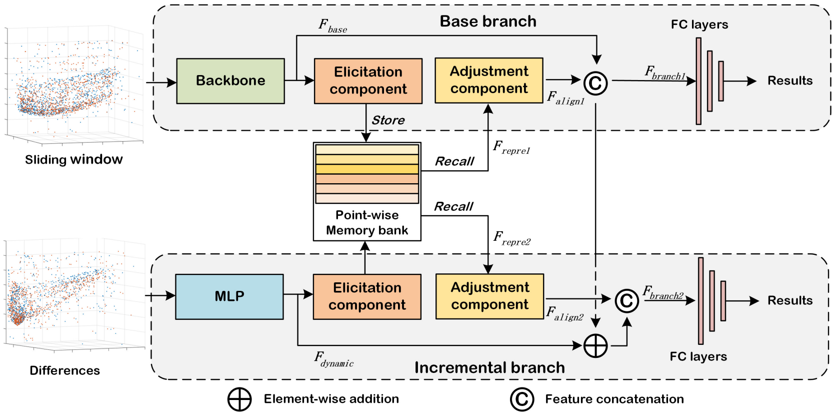

Fig. 2. Overview of the proposed MENet. The upper path is the base branch utilized to extract base features. The lower branch is the incremental branch used for capturing the temporal dynamics between adjacent spatiotemporal windows. The point-wise memory bank is introduced to sketch representative information of the event feature space.

4.1 Dual-Branch Structure

The proposed MENet adopts a dual-branch structure, including a base branch and an incremental branch, as illustrated in Fig. 2. Taking PointNet++ [50] as the backbone, the base branch introduces an elicitation component and an adjustment component for memory feature extraction and feature alignment, as illustrated in the upper path of Fig. 2. Considering that each sliding window contains rich dynamic information and the information contained in adjacent spatiotemporal windows is correlated, a light-weighted incremental branch is proposed to capture the temporal dynamics between two adjacent windows while avoiding repeated extraction of redundant information. As illustrated in the lower path of Fig. 2, the incremental branch adopts a multi-layer-perceptron (MLP) block consisting of four layers to extract dynamic features. And the elicitation component and adjustment component are also introduced.

The base branch takes a sliding window as input. The base features F_{base} are extracted by the backbone, which is then input into the elicitation component for memory feature extraction. Through memory bank, the representative features F_{repre1} are recalled and will be aligned with F_{base} by the adjustment component to obtain the aligned features F_{align1}. Features $F_{branch1}$ used for predicting is obtained by concatenating the base features and the aligned features:

$$F_{branch1} = F_{base} \textcircled{c} F_{align1} \tag{4}$$

where $\textcircled{c}$ is the feature concatenation operation. The incremental branch takes the differences win_{diff} between the previous window and the current input window as input. The temporal dynamics $F_{dynamic}$ is captured by MLP block. Then through the elicitation component and memory bank, the representative features

F_{repre2} are adaptively recalled. The adjustment component is also introduced to obtain aligned features F_{align2}. Finally, features $F_{branch2}$ of the incremental branch utilized to predict results can be obtained:

$$F_{branch2} = F_{align2} \copyright (F_{pre} \oplus F_{dynamic}) \tag{5}$$

where F_{pre} represents the features of the previous window used for predicting and $\oplus$ is the element-wise addition operation.

In addition, in inference, N_{win} sliding windows are regarded as a processing package. Only $1/N_{win}$ of sliding windows will go through the base branch, and the rest will be processed by the light-weighted incremental branch, which reduces the computational complexity.

4.2 Double Polarities Calculation Method

A direct way to obtain the differences between two adjacent windows is to directly subtract the unordered event clouds of the previous window win_{pre} from the clouds of the current one win_{curr}. However, this approach ignores the location and timestamp of each event, which will obtain low-quality differences. Besides, due to the high time resolution of events, obtaining the differences strictly according to the timestamp and space location will bring huge time consumption and memory usage. Therefore, in order to efficiently and effectively calculate the differences win_{diff} between two adjacent windows, a double polarities calculation method is proposed, which can be divided into four steps.

1. The previous sliding window win_{pre} is compressed into an edge frame ($W \times H \times 2$). The two channels of each pixel record the number of corresponding events with positive polarity or negative polarity, respectively. Therefore, the edge frame records the histograms of positive events and negative events:

$$h^{+}(x, y) = \sum_{e_k \in win_{pre}, p_k = +1} \delta(x - x_k, y - y_k) \tag{6}$$

   where $h^{+}(x, y)$ represents the histogram of positive events. e_k means a single event belongs to win_{pre}. δ is the Kronecker delta. The histogram h^{-} of negative events can be obtained through a similar way with $p_k = -1$. Stacking the h^{+} and h^{-} will obtain the edge frame.
2. Based on the first step, an edge frame ($W \times H \times 2$) are produced for the current input window win_{curr}. And a time frame ($W \times H \times 2$) is produced, in which each pixel records the timestamp of the most recent event.
3. The edge frame of win_{pre} is subtracted from that of win_{curr}. The position of the results whose values larger than 0 are recorded. According to the position and results, the location (x, y) and cumulative polarity p are obtained. And the corresponding timestamp t can be obtained from the time frame.
4. Finally, based on the location, cumulative polarity, and timestamp of events, the differences win_{diff} between two adjacent windows in the form of event clouds will be obtained, which contains the additional events that occur in win_{curr}, compared with win_{pre}.

Through the double polarities calculation method, high-quality differences between adjacent windows only recording changed event information is obtained efficiently and effectively, which retains the low power characteristic of event-based data. Taking such differences as input, the incremental branch can capture temporal dynamics of two windows, which provides guidance for prediction.

4.3 Point-Wise Memory Bank

Features extracted by the base branch represent the complete high-level semantics of the input window. Based on these features, the representative information of event feature space can be sketched by introducing a memory bank, which can be adaptively recalled to enhance features. Considering the form of memory features extracted by the elicitation component, we propose a point-wise memory bank with a matrix form, $M \in \mathbb{R}^{N \times C^m}$ with N points and C^m channels, which can be updated through back-propagation. For only recording the representative information of the base features, the memory bank is stored and recalled by the base branch, while the incremental branch only involves the recall operation.

Let $m_i \in \mathbb{R}^{C^m}$ represent an item of the memory bank M and $f_j^{mem1} \in \mathbb{R}^{C^m}$ is a row vector of features $F_{mem1} \in \mathbb{R}^{N_1 \times C^m}$ extracted from the elicitation component. For the base branch, the addressing vector $a^{ddr} \in \mathbb{R}^N$will be firstly calculated, in which each scalar a_i^{ddr} represents an attention weight for the corresponding memory item m_i:

$$a_i^{ddr} = \frac{exp((f_j^{mem1})^T, m_i)}{\sum_{k=1}^{N} exp((f_j^{mem1})^T, m_k)} \tag{7}$$

where $exp(\cdot)$ represents softmax function. For each query f_j^{mem1}, the relevant representative information can be recalled from memory by weighting the item m_i with the corresponding weight a_i^{ddr}:

$$f_j^{repre1} = \sum_{i=1}^{N} a_i^{ddr} m_i \tag{8}$$

The representative features $F_{repre1} = \{f_j^{repre1}\}_{j=1}^{N_1} \in \mathbb{R}^{N_1 \times C^m}$ can be obtained by positioning each feature f_j^{repre1}.

For the incremental branch, the same operations as the base branch will be performed, including using the elicitation component to extract memory features, calculating the addressing vector for recalling the memory, and utilizing the adjustment component to align features, except performing the back-propagate to the memory. By only using the results of the base branch to perform back-propagation on the memory, the representative information of the event feature space can be adaptively recorded into the memory bank.

4.4 Training And Testing Strategies

In the training process, two adjacent windows win_{pre} and win_{curr} will be input into MENet. win_{pre} is input into the base branch to perform prediction, and the estimation error is used to update the parameters of the base branch and memory bank. The incremental branch takes the differences win_{diff} between win_{pre} and win_{curr} as input to capture temporal dynamics between two windows. For gesture recognition and object recognition, the cross-entropy loss function with label smoothing is adopted. To improve training efficiency, the result prediction and parameters updating of the two branches are performed in parallel.

In the testing stage, N_{win} consecutive windows belonging to the same event stream are regarded as a processing package. For taking advantage of the dual-branch structure, only the first window of the package will go through the base branch, while the rest windows will calculate the differences with the previous one and use the incremental branch for prediction. In this way, the incremental branch further exerts its advantages, and the captured temporal dynamics can provide guidance for subsequent prediction.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate our methods on four commonly used datasets, DVS128 Gesture Dataset [1], N-Cars [58], CIFAR10-DVS [15], and MNIST-DVS [42]. The DVS128 Gesture Dataset is collected from 29 subjects under 3 kinds of light conditions and records 1342 instances of 11 gestures. The N-Cars dataset is a benchmark for car recognition, which contains 12,336 car samples and 11693 background samples. Different from the first two datasets, CIFAR10-DVS and MNIST-DVS are converted from the frame-based datasets. The CIFAR10-DVS dataset collects 10000 samples for 10 categories, which is converted from CIFAR10 [25]. In MNIST-DVS, 10000 samples chosen from MNIST [28] are displayed at three different scales, thus it contains 30000 samples in total.

Implementation Details. The base branch adopts the first three set abstraction levels of PointNet++ [50] as the backbone, with three fully connected (FC) layers $[512, 256, K]$ for prediction (K means the number of categories). For efficiency, the elicitation component adopts a simplified set abstraction level [50] ($SA(32, 0.2, [512, 256, 64])$), which selects 32 points from input, forms 32 local regions with ball radius 0.2 and encodes local regions into features by three FC layers. The adjustment component utilizes a lightweight MLP only containing two layers ($[128, 256]$). For the incremental branch, a MLP consisting of four layers ($[64, 256, 512, 1024]$) is adopted for feature extraction, and a MLP containing 2 layers ($[256, 512]$) is utilized for the adjustment component. The structure of the elicitation component and FC layers of the incremental branch are the same as the base branch. The matrix size of the point-wise memory bank is 16×64. The proposed method is implemented by PyTorch, which is trained on a Tesla K80 GPU. The batch size is set as 24 and the Adam [24] optimizer is adopted with an initial learning rate of 0.001 multiplied by 0.5 after 20 epochs.

Metrics. For object recognition and gesture recognition, prediction accuracy is adopted as the evaluation metric. In addition, the giga floating-point operations of the network (GFLOPs), the million floating-point operations per sliding window (MFLOPs/win), and the million floating-point operations per event (MFLOPs/event) are used for evaluating the computational complexity.

5.2 Ablation Study

For verifying the improvement of accuracy brought by the proposed adjustment component, point-wise memory bank, and double polarities calculation method, as well as the reduction of computational complexity and time consumption brought by the dual-branch structure, we conduct ablation experiments on the DVS128 Gesture Dataset [1]. Since the average duration of each event stream is $6s$, the fixed time interval is set as $T = 0.5s$ for producing sliding windows. In addition, for improving processing efficiency and considering that meaningful events have the characteristic of aggregation, each window is sampled 512 events randomly in the time dimension for processing.

Table 1. Contribution of the proposed adjustment component and the point-wise memory bank, evaluated on the DVS128 Gesture Dataset.

Method	Adjustment component	Point-wise memory bank	Accuracy %
MENet	×	✓	97.34
	✓	×	96.96
	✓	✓	**98.86**

Adjustment component and Point-wise memory bank. For verifying the effectiveness of adjustment component and memory bank, we conduct experiments on two additional structures. The first structure removes the adjustment component from MENet, and the second removes the memory bank but retains elicitation component and adjustment component. As shown in Table 1, removing either the adjustment component or memory bank will both decrease accuracy. The results confirm that the memory bank can sketch representative information of event feature space for feature enhancement, and the adjustment component can align recalled memory features with features extracted by branch.

Double Polarities Calculation Method. Table 2 confirms the effect of the proposed calculation method, and both the models adopt memory bank and adjustment component. In Table 2, Sub-diff represents that the differences between two windows are obtained by directly subtracting the unordered event clouds of the previous window from the clouds of the current input one. The results show that even though taking the rough differences calculated by Sub-diff as input, the incremental branch can still capture useful information and obtain an accuracy of 97.34%. When higher quality differences calculated by the proposed calculation method are obtained, the accuracy is increased by 1.52%. The results confirm that the double polarities calculation method can obtain

Table 2. Contribution of the proposed double polarities calculation method, evaluated on the DVS128 Gesture Dataset.

Method	Double Polarities Calculation	Sub-diff	Accuracy %
MENet	×	✓	97.34
	✓	×	**98.86**

Table 3. The average time and MFLOPs of the base branch and the incremental branch for processing a single sliding window.

Method	Branch	MFLOPs/win	Time (ms)
MENet	Base branch	4732	288
	Incremental branch	**1045**	**15**

meaningful differences to assist incremental branch to obtain temporal dynamics between two adjacent spatiotemporal windows.

Computational complexity and time consumption. For utilizing the correlation between two spatiotemporal windows and avoiding repeated extraction of redundant information, MENet adopts a dual-branch structure. For verifying the efficiency improvement brought by the incremental branch, we record the average time and MFLOPs required by the base branch and incremental branch to process a sliding window, as shown in Table 3. Compared with the base branch, the incremental branch reduces the MFLOPs/win by nearly 5 times and speeds up by 19 times. In testing, N_{win} sliding windows are treated as a processing package, as mentioned in Section 4.4. Table 4 evaluates the impact of choosing different N_{win} on MFLOPs/win, inference time of processing a single window, and accuracy. As shown in Table 4, as N_{win} increases, both MFLOPs/win and inference time are reduced due to the low computational complexity of the incremental branch. The best result is achieved by setting the N_{win} as 4, and when $N_{win} = 6$, MENet also achieves a competitive accuracy.

Table 4. The MFLOPs, inference time, and accuracy of choosing different N_{win}.

N_{win}	MFLOPs/win	Time(ms)	Accuracy(%)
2	2889	151.5	98.11
4	2003	85.0	**98.86**
6	1708	64.1	98.48
8	1561	53.2	96.59
10	1450	45.0	95.07
12	1376	39.5	94.31

It is worth noting that the accuracy shows a trend of rising first and then falling, with the increase of N_{win}. There are reasons for this phenomenon. Tak-

Table 5. Comparison with different methods on the MNIST-DVS dataset and N-Cars dataset. Red and blue represent the best and the second best result, respectively.

		MNIST-DVS		N-Cars	
Methods	Representation	Accracy%	MFLOPs/event	Accuracy%	MFLOPs/event
Shi *et al.* [56]	Spike	78.1	-	-	-
H-First [43]	Spike	59.5	-	56.1	-
HATS [58]	TimeSurface	98.4	-	90.2	-
HOTS [27]	TimeSurface	80.3	26	62.4	14.0
DART [51]	TimeSurface	98.5	-	-	-
LIAF-Net [61]	Frame	99.1	-	-	-
YOLE [7]	VoxelGrid	96.1	-	92.7	328.1
Asynet [35]	VoxelGrid	**99.4**	112	**94.4**	21.5
Bi *et al.* [3]	Graph	98.6	-	91.4	-
EvS-S [62]	Graph	99.1	15.2	93.1	6.1
Dominic *et al.* [18]	Point-clouds	99.1	-	-	-
Single-nomem	Point-clouds	98.8	**9.2**	93.4	**4.6**
MENet-single	Point-clouds	**99.57**	**9.2**	**95.32**	**4.6**

ing differences between adjacent windows as input, the incremental branch can model the motion information contained in the two windows. Then, features of the previous window will be used for prediction of the current one, which accumulates the motion information. The accumulation of motion context in a short time period can provide guidance for estimation. Therefore, as N_{win} increases, the accuracy first shows an upward trend. However, in the long-term accumulation of motion context, events in the front window have a weak connection with the events in the back. When N_{win} is too large, part of the accumulated motion context may even introduce noise for prediction, thus the accuracy decreases.

5.3 Object Recognition

The experiments are conducted on three commonly used object recognition datasets. Since the average duration of each event stream in the CIFAR10-DVS dataset is $1.2s$, the fixed time interval is set as $200ms$. Each window is randomly sampled 4096 events. In addition, for verifying the effect of the proposed point-wise memory bank on object recognition, two additional structures are proposed, including MENet-single without incremental branch and Single-nomem further removing the memory bank. These two structures are evaluated on MNIST-DVS and N-Cars, in which each stream is sampled 512 or 1024 events, respectively.

Comparison with State of the Art. Table 5 compares the proposed Single-nomem and MENet-single with previous methods on MNIST-DVS and N-Cars. Firstly, compared with Single-nomem, MENet-single improves accuracy by 0.77% and 1.92% with almost the same computational complexity. The results confirm that the proposed memory bank stores representative information of event feature space, and the features recalled from it contains discrimination information of categories, which improves the accuracy with very low computational complexity. Secondly, previous methods have achieved high accuracy on

Table 6. Comparison with different methods on the CIFAR10-DVS dataset. Red and blue represent the best and the second best result, respectively.

Methods	Representation	Accuracy (%)	MFLOPs/event	GFLOPs
STBP-tdBN [65]	Spike	67.8	-	-
HOTS [27]	TimeSurface	27.1	26	-
HATS [58]	TimeSurface	52.4	-	-
DART [51]	TimeSurface	65.8	-	-
Asynet [35]	VoxelGrid	66.3	103	-
Kugele *et al.* [26]	Frame	66.7	-	8.8
LIAF-Net [61]	Frame	70.4	-	**7.1**
TA-SNN [64]	Frame	**72.0**	-	-
EvS-S [62]	Graph	68.0	33.2	-
Dominic *et al.* [18]	Point-clouds	56.6	-	-
MENet	Point-clouds	**74.1**	**0.9**	**3.7**

these two datasets, but the MENet-single further improves performance with the lowest computational complexity. Compared with two competitive methods, Asynet [35] and EvS-S [62] which process events asynchronously, MENet-single improves accuracy by 0.17% and 0.47% while reducing MFLOPs/event by nearly 12 times and 1.6 times on MNIST-DVS, respectively. On N-Cars, the accuracy is improved by 0.92% and 2.22% with reducing MFLOPs/event by 4.7 times and 1.3 times. The results confirm that by taking event clouds as input, the proposed method can process multiple events in parallel and extract the rich information contained in events effectively. Moreover, compared with the other methods in Table 5, MENet-single uses fewer events and achieves higher accuracy, which also confirms that the proposed method is effective and efficient.

In Table 6, compared with other methods on CIFAR10-DVS, MENet achieves the best performance with the lowest computational complexity. Compared with two competitive methods LIAF-Net[61] and TA-SNN[64], MENet improves accuracy by 3.7% and 2.1%, respectively. These two methods both compressed event stream into frames and utilized SNN-based model. Experimental results prove that MENet retains the high time resolution characteristic of event-based data and the incremental branch can effectively capture the rich temporal dynamics contained in event data. Therefore, MENet can greatly improve accuracy while reducing computational complexity.

5.4 Gesture Recognition

As set in the ablation study, in DVS128 Gesture Dataset [1], the fixed time interval is $T = 0.5s$. For each sliding window, 512 events are sampled for processing.

Comparison with State of the Art. In Table 7, compared with MENet-single, MENet achieves a better result (98.86% vs 98.11%), while the speed is increased by 3 times and computational complexity is reduced by 2 times. In addition, compared with other methods that have achieved high accuracy, MENet achieves a new state-of-the-art result while significantly reducing the computational complexity. Although TA-SNN adopts a small time interval $dt = 10ms$ to generate the frames and processes all events contained in the window, MENet also achieves a better result (98.86% vs 98.61%). In addition, compared with LIAF-Net [61], MENet improves accuracy by 1.3% and reduces the GFLOPs by 7.8 times. These experimental results prove once again that MENet can effectively and efficiently utilize the dynamic information contained in event streams.

Table 7. Comparison with different methods on the DVS128 Gesture Dataset. Red and blue represent the best and the second best result, respectively.

Methods	Representation	Accuracy (%)	GFLOPs
Slayer [57]	Spike	93.64	-
Amir *et al.* [1]	Spike	94.59	-
SpArNet [22]	Spike	95.10	-
STBP-tdBN [65]	Spike	96.87	-
Bi *et al.* [3]	Graph	97.20	13.7
Wang *et al.* [59]	Point-clouds	95.32	-
PAT [63]	Point-clouds	96.00	-
Kugele *et al.* [26]	Frame	95.56	15.0
Massa *et al.* [34]	Frame	89.64	-
LIF-Net [16]	Frame	93.40	-
LIAF-Net [61]	Frame	97.56	13.6
TA-SNN [64]	Frame	**98.61**	-
MENet-single	Point-clouds	98.11	**4.73**
MENet	Point-clouds	**98.86**	**2.00**

6 Conclusion

This paper proposes a novel memory-based network with dual-branch for efficiently and effectively processing event-based data, namely MENet. For utilizing the correlation between adjacent windows and avoiding repeated extraction of redundant information, MENet contains two branches. The first one is the base branch which aims at extracting base features, while the second one is the incremental branch with a light-weighted structure for capturing temporal dynamics between two adjacent spatiotemporal windows. In addition, for calculating the differences between two adjacent windows to capture meaningful information, a double polarities calculation method is proposed. Furthermore, a point-wise memory bank is introduced to sketch the representative information of event feature space for feature enhancement. Experimental results show that the proposed dual-branch structure can reduce computational complexity and time consumption while improving accuracy, and the proposed double polarities calculation method and the point-wise memory bank can play their roles.

Acknowledgments. This work was supported in part by the National Key Research and Development Program of China under Grant 2020AAA0103402, Jiangsu Key Research and Development Plan (No.BE2021012-2), and NSFC 61876182, 61906195.

References

1. Amir, A., Taba, B., Berg, D.J., Melano, T., McKinstry, J.L., di Nolfo, C., Nayak, T.K., Andreopoulos, A., Garreau, G., Mendoza, M., Kusnitz, J., DeBole, M., Esser, S.K., Delbrück, T., Flickner, M., Modha, D.S.: A low power, fully event-based gesture recognition system. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 7388–7397. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.781, https://doi.org/10.1109/CVPR.2017.781
2. Bardow, P., Davison, A.J., Leutenegger, S.: Simultaneous optical flow and intensity estimation from an event camera. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 884–892. IEEE Computer Society (2016). https://doi.org/10.1109/CVPR.2016.102, https://doi.org/10.1109/CVPR.2016.102
3. Bi, Y., Chadha, A., Abbas, A., Bourtsoulatze, E., Andreopoulos, Y.: Graph-based spatial-temporal feature learning for neuromorphic vision sensing. CoRR **abs/1910.03579** (2019), http://arxiv.org/abs/1910.03579
4. Brandli, C., Berner, R., Yang, M., Liu, S., Delbrück, T.: A 240 × 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. IEEE J. Solid State Circuits **49**(10), 2333–2341 (2014). https://doi.org/10.1109/JSSC.2014.2342715, https://doi.org/10.1109/JSSC.2014.2342715
5. Cai, Q., Pan, Y., Yao, T., Yan, C., Mei, T.: Memory matching networks for one-shot image recognition. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 4080–4088. Computer Vision Foundation / IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00429
6. Cannici, M., Ciccone, M., Romanoni, A., Matteucci, M.: Asynchronous convolutional networks for object detection in neuromorphic cameras. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 1656–1665. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPRW.2019.00209
7. Cannici, M., Ciccone, M., Romanoni, A., Matteucci, M.: Attention mechanisms for object recognition with event-based cameras. In: IEEE Winter Conference on Applications of Computer Vision, WACV 2019, Waikoloa Village, HI, USA, January 7-11, 2019. pp. 1127–1136. IEEE (2019). https://doi.org/10.1109/WACV.2019.00125, https://doi.org/10.1109/WACV.2019.00125
8. Cannici, M., Ciccone, M., Romanoni, A., Matteucci, M.: A differentiable recurrent surface for asynchronous event-based data. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XX. Lecture Notes in Computer Science, vol. 12365, pp. 136–152. Springer (2020). https://doi.org/10.1007/978-3-030-58565-5_9, https://doi.org/10.1007/978-3-030-58565-5_9
9. Chen, J., Meng, J., Wang, X., Yuan, J.: Dynamic graph CNN for event-camera based gesture recognition. In: IEEE International Symposium on Circuits and Systems, ISCAS 2020, Sevilla, Spain, October 10-21, 2020. pp. 1–5. IEEE (2020). https://doi.org/10.1109/ISCAS45731.2020.9181247, https://doi.org/10.1109/ISCAS45731.2020.9181247

10. Cheng, W., Luo, H., Yang, W., Yu, L., Chen, S., Li, W.: DET: A high-resolution DVS dataset for lane extraction. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 1666–1675. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPRW.2019.00210
11. Gallego, G., Rebecq, H., Scaramuzza, D.: A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 3867–3876. Computer Vision Foundation / IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00407
12. Gehrig, D., Loquercio, A., Derpanis, K.G., Scaramuzza, D.: End-to-end learning of representations for asynchronous event-based data. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 5632–5642. IEEE (2019). https://doi.org/10.1109/ICCV.2019.00573, https://doi.org/10.1109/ICCV.2019.00573
13. Gehrig, D., Rebecq, H., Gallego, G., Scaramuzza, D.: Eklt: Asynchronous photometric feature tracking using events and frames. International Journal of Computer Vision **128**, 601–618 (2019)
14. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., van den Hengel, A.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. pp. 1705–1714. IEEE (2019). https://doi.org/10.1109/ICCV.2019.00179, https://doi.org/10.1109/ICCV.2019.00179
15. H, L., H, L., X, J., G, L., L, S.: Cifar10-dvs: An event-stream dataset for object classification. front neurosci. (2017)
16. He, W., Wu, Y., Deng, L., Li, G., Wang, H., Tian, Y., Ding, W., Wang, W., Xie, Y.: Comparing snns and rnns on neuromorphic vision datasets: Similarities and differences. CoRR **abs/2005.02183** (2020), https://arxiv.org/abs/2005.02183
17. Huang, H., Yu, A., He, R.: Memory oriented transfer learning for semi-supervised image deraining. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 7732–7741. Computer Vision Foundation / IEEE (2021)
18. Jack, D., Maire, F., Denman, S., Eriksson, A.P.: Sparse convolutions on continuous domains for point cloud and event stream networks. In: Ishikawa, H., Liu, C., Pajdla, T., Shi, J. (eds.) Computer Vision - ACCV 2020 - 15th Asian Conference on Computer Vision, Kyoto, Japan, November 30 - December 4, 2020, Revised Selected Papers, Part I. Lecture Notes in Computer Science, vol. 12622, pp. 400–416. Springer (2020). https://doi.org/10.1007/978-3-030-69525-5_24, https://doi.org/10.1007/978-3-030-69525-5_24
19. Jiang, Z., Zhang, Y., Zou, D., Ren, J.S.J., Lv, J., Liu, Y.: Learning event-based motion deblurring. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 3317–3326. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.00338, https://openaccess.thecvf.com/content_CVPR_2020/html/Jiang_Learning_Event-Based_Motion_Deblurring_CVPR_2020_paper.html

20. Kaiser, L., Nachum, O., Roy, A., Bengio, S.: Learning to remember rare events. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017), https://openreview.net/forum?id=SJTQLdqlg
21. Khairallah, M.Z., Bonardi, F., Roussel, D., Bouchafa, S.: PCA event-based optical flow for visual odometry. CoRR **abs/2105.03760** (2021), https://arxiv.org/abs/2105.03760
22. Khoei, M.A., Yousefzadeh, A., Pourtaherian, A., Moreira, O., Tapson, J.: Sparnet: Sparse asynchronous neural network execution for energy efficient inference. In: 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems, AICAS 2020, Genova, Italy, August 31 - September 2, 2020. pp. 256–260. IEEE (2020). https://doi.org/10.1109/AICAS48895.2020.9073827, https://doi.org/10.1109/AICAS48895.2020.9073827
23. Kim, H., Leutenegger, S., Davison, A.J.: Real-time 3d reconstruction and 6-dof tracking with an event camera. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI. Lecture Notes in Computer Science, vol. 9910, pp. 349–364. Springer (2016). https://doi.org/10.1007/978-3-319-46466-4_21, https://doi.org/10.1007/978-3-319-46466-4_21
24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980
25. Krizhevsky, A.: Learning multiple layers of features from tiny images pp. 32–33 (2009), https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf
26. Kugele, A., Pfeil, T., Pfeiffer, M., Chicca, E.: Efficient processing of spatio-temporal data streams with spiking neural networks. Frontiers in Neuroscience **14**, 439 (2020). https://doi.org/10.3389/fnins.2020.00439, https://www.frontiersin.org/article/10.3389/fnins.2020.00439
27. Lagorce, X., Orchard, G., Galluppi, F., Shi, B.E., Benosman, R.: HOTS: A hierarchy of event-based time-surfaces for pattern recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(7), 1346–1359 (2017). https://doi.org/10.1109/TPAMI.2016.2574707, https://doi.org/10.1109/TPAMI.2016.2574707
28. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
29. Lee, S., Kim, H.G., Choi, D.H., Kim, H., Ro, Y.M.: Video prediction recalling long-term motion context via memory alignment learning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. pp. 3054–3063. Computer Vision Foundation / IEEE (2021)
30. Lichtsteiner, P., Posch, C., Delbrück, T.: A 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor. IEEE J. Solid State Circuits **43**(2), 566–576 (2008). https://doi.org/10.1109/JSSC.2007.914337, https://doi.org/10.1109/JSSC.2007.914337
31. Liu, Q., Ruan, H., Xing, D., Tang, H., Pan, G.: Effective AER object classification using segmented probability-maximization learning in spiking neural networks. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI

2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. pp. 1308–1315. AAAI Press (2020), https://aaai.org/ojs/index.php/AAAI/article/view/5486

32. Manderscheid, J., Sironi, A., Bourdis, N., Migliore, D., Lepetit, V.: Speed invariant time surface for learning to detect corner points with event-based cameras. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 10245–10254. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPR.2019.01049
33. Maqueda, A.I., Loquercio, A., Gallego, G., García, N., Scaramuzza, D.: Event-based vision meets deep learning on steering prediction for self-driving cars. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 5419–5427. Computer Vision Foundation / IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00568, http://openaccess.thecvf.com/content_cvpr_2018/html/Maqueda_Event-Based_Vision_Meets_CVPR_2018_paper.html
34. Massa, R., Marchisio, A., Martina, M., Shafique, M.: An efficient spiking neural network for recognizing gestures with a DVS camera on the loihi neuromorphic processor. CoRR **abs/2006.09985** (2020), https://arxiv.org/abs/2006.09985
35. Messikommer, N., Gehrig, D., Loquercio, A., Scaramuzza, D.: Event-based asynchronous sparse convolutional networks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VIII. Lecture Notes in Computer Science, vol. 12353, pp. 415–431. Springer (2020). https://doi.org/10.1007/978-3-030-58598-3_25, https://doi.org/10.1007/978-3-030-58598-3_25
36. Miller, A.H., Fisch, A., Dodge, J., Karimi, A., Bordes, A., Weston, J.: Key-value memory networks for directly reading documents. In: Su, J., Carreras, X., Duh, K. (eds.) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016. pp. 1400–1409. The Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/d16-1147, https://doi.org/10.18653/v1/d16-1147
37. Mitrokhin, A., Hua, Z., Fermüller, C., Aloimonos, Y.: Learning visual motion segmentation using event surfaces. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 14402–14411. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.01442
38. Mueggler, E., Bartolozzi, C., Scaramuzza, D.: Fast event-based corner detection. In: British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017. BMVA Press (2017), https://www.dropbox.com/s/vicqrsz0yicq65c/0070.pdf?dl=1
39. Munda, G., Reinbacher, C., Pock, T.: Real-time intensity-image reconstruction for event cameras using manifold regularisation. Int. J. Comput. Vis. **126**(12), 1381–1393 (2018). https://doi.org/10.1007/s11263-018-1106-2, https://doi.org/10.1007/s11263-018-1106-2
40. Nguyen, A., Do, T., Caldwell, D.G., Tsagarakis, N.G.: Real-time 6dof pose relocalization for event cameras with stacked spatial LSTM networks. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 1638–1645. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPRW.2019.00207

41. Orchard, G., Benosman, R., Etienne-Cummings, R., Thakor, N.V.: A spiking neural network architecture for visual motion estimation. In: 2013 IEEE Biomedical Circuits and Systems Conference (BioCAS), Rotterdam, The Netherlands, October 31 - Nov. 2, 2013. pp. 298–301. IEEE (2013). https://doi.org/10.1109/BioCAS.2013.6679698, https://doi.org/10.1109/BioCAS.2013.6679698
42. Orchard, G., Jayawant, A., Cohen, G., Thakor, N.: Converting static image datasets to spiking neuromorphic datasets using saccades (2015)
43. Orchard, G., Meyer, C., Etienne-Cummings, R., Posch, C., Thakor, N.V., Benosman, R.: Hfirst: A temporal approach to object recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(10), 2028–2040 (2015). https://doi.org/10.1109/TPAMI.2015.2392947, https://doi.org/10.1109/TPAMI.2015.2392947
44. Pan, L., Liu, M., Hartley, R.: Single image optical flow estimation with an event camera. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 1669–1678. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.00174
45. Pan, L., Scheerlinck, C., Yu, X., Hartley, R., Liu, M., Dai, Y.: Bringing a blurry frame alive at high frame-rate with an event camera. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 6820–6829. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPR.2019.00698
46. Paredes-Vallés, F., Scheper, K.Y.W., de Croon, G.C.H.E.: Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception. IEEE Trans. Pattern Anal. Mach. Intell. **42**(8), 2051–2064 (2020). https://doi.org/10.1109/TPAMI.2019.2903179, https://doi.org/10.1109/TPAMI.2019.2903179
47. Park, H., Noh, J., Ham, B.: Learning memory-guided normality for anomaly detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 14360–14369. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.01438, https://openaccess.thecvf.com/content_CVPR_2020/html/Park_Learning_Memory-Guided_Normality_for_Anomaly_Detection_CVPR_2020_paper.html
48. Pei, W., Zhang, J., Wang, X., Ke, L., Shen, X., Tai, Y.: Memory-attended recurrent network for video captioning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 8347–8356. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPR.2019.00854, http://openaccess.thecvf.com/content_CVPR_2019/html/Pei_Memory-Attended_Recurrent_Network_for_Video_Captioning_CVPR_2019_paper.html
49. Posch, C., Matolin, D., Wohlgenannt, R.: A QVGA 143 db dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. IEEE J. Solid State Circuits **46**(1), 259–275 (2011). https://doi.org/10.1109/JSSC.2010.2085952, https://doi.org/10.1109/JSSC.2010.2085952
50. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in

Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 5099–5108 (2017)

51. Ramesh, B., Yang, H., Orchard, G., Thi, N.A.L., Zhang, S., Xiang, C.: DART: distribution aware retinal transform for event-based cameras. IEEE Trans. Pattern Anal. Mach. Intell. **42**(11), 2767–2780 (2020). https://doi.org/10.1109/TPAMI.2019.2919301, https://doi.org/10.1109/TPAMI.2019.2919301
52. Rebecq, H., Gallego, G., Mueggler, E., Scaramuzza, D.: EMVS: event-based multi-view stereo - 3d reconstruction with an event camera in real-time. Int. J. Comput. Vis. **126**(12), 1394–1414 (2018). https://doi.org/10.1007/s11263-017-1050-6, https://doi.org/10.1007/s11263-017-1050-6
53. Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D.: Events-to-video: Bringing modern computer vision to event cameras. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 3857–3866. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPR.2019.00398
54. Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D.: High speed and high dynamic range video with an event camera. IEEE Trans. Pattern Anal. Mach. Intell. **43**(6), 1964–1980 (2021). https://doi.org/10.1109/TPAMI.2019.2963386, https://doi.org/10.1109/TPAMI.2019.2963386
55. Sekikawa, Y., Hara, K., Saito, H.: Eventnet: Asynchronous recursive event processing. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 3887–3896. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPR.2019.00401
56. Shi, C., Li, J., Wang, Y., Luo, G.: Exploiting lightweight statistical learning for event-based vision processing. IEEE Access **6**, 19396–19406 (2018). https://doi.org/10.1109/ACCESS.2018.2823260, https://doi.org/10.1109/ACCESS.2018.2823260
57. Shrestha, S.B., Orchard, G.: SLAYER: spike layer error reassignment in time. CoRR **abs/1810.08646** (2018), http://arxiv.org/abs/1810.08646
58. Sironi, A., Brambilla, M., Bourdis, N., Lagorce, X., Benosman, R.: HATS: histograms of averaged time surfaces for robust event-based object classification. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 1731–1740. Computer Vision Foundation / IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00186
59. Wang, Q., Zhang, Y., Yuan, J., Lu, Y.: Space-time event clouds for gesture recognition: From RGB cameras to event cameras. In: IEEE Winter Conference on Applications of Computer Vision, WACV 2019, Waikoloa Village, HI, USA, January 7-11, 2019. pp. 1826–1835. IEEE (2019). https://doi.org/10.1109/WACV.2019.00199, https://doi.org/10.1109/WACV.2019.00199
60. Weston, J., Chopra, S., Bordes, A.: Memory networks. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1410.3916
61. Wu, Z., Zhang, H., Lin, Y., Li, G., Wang, M., Tang, Y.: Liaf-net: Leaky integrate and analog fire network for lightweight and efficient spatiotemporal information processing. CoRR **abs/2011.06176** (2020), https://arxiv.org/abs/2011.06176

62. Y, L., H, Z., B, Y.: Graph-based asynchronous event processing for rapid object recognition. ICCV pp. 934–943 (2021)
63. Yang, J., Zhang, Q., Ni, B., Li, L., Liu, J., Zhou, M., Tian, Q.: Modeling point clouds with self-attention and gumbel subset sampling. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 3323–3332. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPR.2019.00344
64. Yao, M., Gao, H., Zhao, G., Wang, D., Lin, Y., Yang, Z., Li, G.: Temporal-wise attention spiking neural networks for event streams classification. CoRR **abs/2107.11711** (2021), https://arxiv.org/abs/2107.11711
65. Zheng, H., Wu, Y., Deng, L., Hu, Y., Li, G.: Going deeper with directly-trained larger spiking neural networks. CoRR **abs/2011.05280** (2020), https://arxiv.org/abs/2011.05280
66. Zhou, Y., Gallego, G., Shen, S.: Event-based stereo visual odometry. IEEE Trans. Robotics **37**(5), 1433–1450 (2021). https://doi.org/10.1109/TRO.2021.3062252, https://doi.org/10.1109/TRO.2021.3062252
67. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 989–997. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPR.2019.00108, http://openaccess.thecvf.com/content_CVPR_2019/html/Zhu_Unsupervised_Event-Based_Learning_of_Optical_Flow_Depth_and_Egomotion_CVPR_2019_paper.html
68. Zhu, L., Yang, Y.: Inflated episodic memory with region self-attention for long-tailed visual recognition. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 4343–4352. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.00440