

Effective AER Object Classification Using Segmented Probability-Maximization Learning in Spiking Neural Networks

Qianhui Liu,¹ Haibo Ruan,¹ Dong Xing,¹ Huajin Tang,^{1,2} Gang Pan^{1,2*}

¹College of Computer Science and Technology, Zhejiang University, Hangzhou, China

²Zhejiang Lab, Hangzhou, China

{qianhui.liu, hbruan, dongxing, htang, gpan}@zju.edu.cn

Abstract

Address event representation (AER) cameras have recently attracted more attention due to the advantages of high temporal resolution and low power consumption, compared with traditional frame-based cameras. Since AER cameras record the visual input as asynchronous discrete events, they are inherently suitable to coordinate with the spiking neural network (SNN), which is biologically plausible and energy-efficient on neuromorphic hardware. However, using SNN to perform the AER object classification is still challenging, due to the lack of effective learning algorithms for this new representation. To tackle this issue, we propose an AER object classification model using a novel *segmented probability-maximization* (SPA) learning algorithm. Technically, 1) the SPA learning algorithm iteratively maximizes the probability of the classes that samples belong to, in order to improve the reliability of neuron responses and effectiveness of learning; 2) a *peak detection* (PD) mechanism is introduced in SPA to locate informative time points segment by segment, based on which information within the whole event stream can be fully utilized by the learning. Extensive experimental results show that, compared to state-of-the-art methods, not only our model is more effective, but also it requires less information to reach a certain level of accuracy.

Introduction

Address event representation (AER) cameras are neuromorphic devices imitating the mechanism of human retina. Contrary to traditional frame-based cameras, which record the visual input from all pixels as images at a fix rate, with AER cameras, each pixel individually emits events when it monitors sufficient changes of light intensity in its receptive field. The final output of the camera is a stream of events collected from each pixel, forming an asynchronous and sparse representation of the scene. AER cameras naturally respond to moving objects and ignore static redundant information, resulting in significant reduction of memory usage and energy consumption. Moreover, AER cameras capture visual information at a significantly higher temporal resolution than traditional frame-based cameras, achieving accuracy down to

sub-microsecond levels under optimal conditions (Orchard et al. 2015b). Commonly used AER cameras include the asynchronous time-based image sensor (ATIS) (Posch, Matolin, and Wohlgenannt 2011), dynamic vision sensor (DVS) (Lichtsteiner, Posch, and Delbruck 2008; Leñero-Bardallo, Serrano-Gotarredona, and Linares-Barranco 2011), dynamic and active pixel vision sensor (DAVIS) (Brandli et al. 2014).

This event-based representation is inherently suitable to coordinate with the spiking neural network (SNN) since SNN also has the event-based property. SNN is generally more solid on biological plausibility and more powerful on processing both spatial and temporal information. It may also very help to build cyborg intelligent systems (Wu, Pan, and Zheng 2013; Wu et al. 2014). Moreover, SNN has the advantage of energy efficiency, for example, current implementations of SNN on neuromorphic hardware use only a few nJ or even pJ for transmitting a spike (Diehl and Cook 2015).

However, the novelty of this representation also poses several challenges to AER object classification using SNN. Firstly, the event streams from AER cameras are not stable, compared with the video streams from tradition cameras. AER cameras are sensitive to the dynamic information within the visual receptive field. Along with the events relevant to the objects, factors like camera shaking and subtle changes of environmental light will generate a large quantity of noisy events, which will impact the reliability of neuron responses and the learning performance of SNN. Secondly, the event stream from AER camera records massive information of a period of time, and a mechanism is required to make full use of the information for training and reach a competitive level of accuracy despite the testing information is not complete. We will make steps towards improving the effectiveness of classification using SNN.

We propose an AER object classification model, which consists of an event-based spatio-temporal feature extraction and a new *segmented probability-maximization* (SPA) learning algorithm of SNN. Firstly, the feature extraction obtains the representative features of the output of AER cameras, reducing the effect of noisy events to some extent and maintaining the precise timing of the output events. Feature extraction employs the spiking neurons to utilize the pre-

*Corresponding author

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

cise timing information inherently present in the output of AER cameras and uses spike timing to represent the spatio-temporal features (Orchard et al. 2015b). Then, the SPA supervised learning algorithm constructs an objective function based on the probability of the classes that samples belong to and iteratively updates the synaptic weights with gradient-based optimization of the objective function. To fully utilize the massive information in the event stream covering a period of time, we introduce a *peak detection* (PD) in SPA to trigger the weight updating procedure based on the informative time points located segment by segment. The SPA learning algorithm enables the trained neurons to respond actively to their representing classes. Therefore, the classification decision can be determined by the firing rate of every trained neuron. We perform extensive experiments to verify our model, and the results show that our model is more effective when compared with state-of-the-art methods, and requires less information to reach a certain level of accuracy.

Related Work

Event-Based Features and Object Classification

(Pérez-Carrasco et al. 2013) presented a methodology by training a frame-driven convolutional neural network (ConvNet) with images (frames) by collecting events during fixed time intervals and mapping the frame-driven ConvNet to an event-driven ConvNet. (Neil and Liu 2016) introduced various deep network architectures including a deep fusion network composed of convolutional neural networks and recurrent neural networks to jointly solve recognition tasks. Different from the existing deep learning based methods, (Lagorce et al. 2017) proposed the spatio-temporal features based on recent temporal activity within a local spatial neighborhood called time-surfaces and a hierarchy of event-based time-surfaces for pattern recognition (HOTS). (Sironi et al. 2018) introduced the Histograms of Averaged Time Surfaces (HATS) for feature representation of event-based object recognition. In addition, there are some existing works using SNN for event-based features and object classification. (Zhao et al. 2015) presented an event-driven HMAX network for feature extraction and a tempotron classifier of SNN for classification. Further, (Orchard et al. 2015b) proposed an HMAX inspired SNN for object recognition (HFirst). HFirst does not require extra coding and consistently uses precise timing of spikes for feature representation and learning process. (Cohen et al. 2016) presented an implementation of Synaptic Kernel Inverse Method (SKIM), which is a learning method based on principles of dendritic computation, in order to perform a large-scale AER object classification task. (Liu et al. 2019) proposed a multiscale spatio-temporal feature (MuST) representation of AER events and an unsupervised recognition approach.

SNN Learning Algorithm

SpikeProp (Bohte, Kok, and La Poutre 2002) is one of the most classical SNN learning algorithm. It constructs an error function by the difference between the desired and actual output spikes, then updates the synaptic weights based

on gradient descent. Other learning algorithms that also define the desired spike sequences are ReSuMe (Ponulak and Kasiński 2010), SPAN (Mohammed et al. 2012), PSD (Yu et al. 2013), etc. Recently, membrane voltage-driven methods have emerged in an attempt to improve the learning efficiency and accuracy of spiking neurons. MPD-AL (Zhang et al. 2019) proposed a membrane-potential driven aggregate-label learning algorithm, which constructs an error function based on the membrane potential trace and the fixed firing threshold of the neuron. It dynamically determines the number of desired output spikes instead of enforcing a fixed number of desired spikes. However, these algorithms need to output a corresponding spike sequence for classification. Note that, for the AER object classification task, it is desired to give a result in time instead of waiting until the output sequence has been fully generated.

Tempotron (Gütig and Sompolinsky 2006; Qi et al. 2018) is also a voltage-driven learning algorithm and aims to train the output neuron to fire a single spike or not according to its class label. If the neuron is supposed to fire (or not fire, on the other hand) but it actually fails to do so (or does fire, vice versa), then the weights should be modified. Tempotron implements a gradient descent dynamics that minimizes the error defined as the difference between the maximal membrane potential and the firing threshold. This kind of “single spike” classifier tends to be affected by noise and thus is not suitable for the task of AER object classification.

The proposed SPA learning algorithm for AER object classification aims to enable the trained neurons to respond actively to their representing classes. The model gives the classification decision by choosing the class with the highest average firing rate. In this way, we do not need to wait for the output sequence to complete and can directly give the results based on the firing rates at the current time. Therefore, our proposed SPA algorithm is more robust and flexible for AER object classification task.

Method

In this section, we will introduce the proposed AER object classification model, which mainly consists of an event-based spatio-temporal feature extraction and a novel SNN learning algorithm. The flow chart of the model is shown in Figure 1.

Event-Based Feature Extraction

Feature extraction follows the manner of Hierarchical Model and X (HMAX) (Riesenhuber and Poggio 1999), a popular bio-inspired model mimicking the information processing of visual cortex. We employ a hierarchy of $S1$ layer and $C1$ layer, corresponding to the simple and complex cells in primary visual cortex V1 respectively. The simple cells combine the input with a bell-shaped tuning function to increase feature selectivity and the complex cells perform the maximum operation to increase feature invariance (Riesenhuber and Poggio 1999). We model the simple and complex cells using spiking neurons which generate spikes to represent the features of the output events of AER cameras. Given an AER camera with pixel grid size $N \times M$, the i -th event is de-

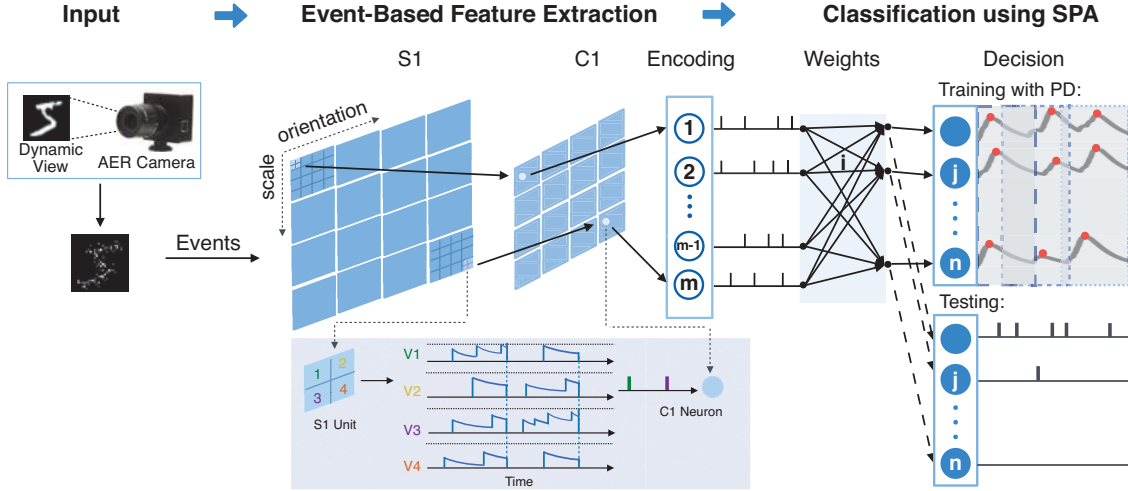


Figure 1: The flow chart of the proposed AER object classification. The events from the AER camera are firstly sent to $S1$ layer. Neurons in $S1$ layer have their own receptive scale and respond best to a certain orientation. Neurons of the same receptive scale and orientation are organized into one feature map (denoted by blue squares). $S1$ feature maps are divided into adjacent non-overlapping 2×2 cell regions, namely $S1$ units. If the membrane voltage of $S1$ neuron exceeds its threshold, the neuron will fire a spike to the $C1$ layer and all neurons in the same unit are reset to V_{reset} . $C1$ neurons then transmit the feature spikes to the encoding layer. During training, the weights of synapses from encoding neurons to decision neurons are updated by the proposed SPA learning algorithm, which locates the informative time points t_{peak} segment by segment with PD and maximizes the corresponding probability based on the t_{peak} . The gray bands and red dots denote the segments and informative time points respectively. After training is done, the synaptic weights are fixed. During testing, the final classification decision is determined by averaging the firing rates of decision layer neurons per class and choosing the class with the highest average firing rate. Note that we separately show the decision layer during training and testing because of the different readout way during training and testing.

scribed as:

$$e_i = [e_{x_i}, e_{y_i}, e_{t_i}, e_{p_i}], i \in \{1, 2, \dots, I\} \quad (1)$$

where $(e_{x_i}, e_{y_i}) \in \{1, 2, \dots, N\} \times \{1, 2, \dots, M\}$ is the position of the pixel generating the i -th event, $e_{t_i} \geq 0$ the timestamp at which the event is generated, $e_{p_i} \in \{-1, 1\}$ the polarity of the event, with $-1, 1$ meaning respectively OFF and ON events, and I the number of events. Figure 2 shows the visualization of an AER event stream representing the “heart” symbol in Cards dataset (Serrano-Gotarredona and Linares-Barranco 2015).

From Input to $S1$ Layer The output events of the AER camera are sent as input to the $S1$ layer, in which each event is convolved with a group of Gabor filters (Zhao et al. 2015). Each filter models a neuron cell that has a certain scale s of receptive field and responds best to a certain orientation θ . The function of Gabor filter can be described with the following equation:

$$G(\Delta x, \Delta y) = \exp\left(-\frac{X^2 + \gamma^2 Y^2}{2\sigma^2}\right) \cos\left(\frac{2\pi}{\lambda} X\right) \quad (2)$$

$$X = \Delta x \cos \theta + \Delta y \sin \theta$$

$$Y = -\Delta x \sin \theta + \Delta y \cos \theta$$

where Δx and Δy are the spatial offsets between the pixel position (x, y) and the event address (e_x, e_y) , γ is the aspect ratio. The wavelength λ and effective width σ are parameters

determined by scale s . We choose four orientations ($0^\circ, 45^\circ, 90^\circ, 135^\circ$) and a range of sizes from 3×3 to 9×9 pixels with strides of two pixels for Gabor filters. Other parameters in the Gabor filter are inherited from (Zhao et al. 2015).

Each spiking neuron corresponds to one pixel in the camera and neurons of the same receptive scale and orientation are organized into one feature map. The membrane voltages of neurons in feature maps are initialized as zeros, then updated by adding each element of the filters to the maps at the position specified by the address of each event. At the same time, the decay mechanism of the spiking neuron is maintained to eliminate the impact of very old events on the current response. The membrane voltage of the neuron at position (x, y) and time t in the map of specific scale s and orientation θ can be described as:

$$V(x, y, t; s, \theta) = \sum_i \mathbb{1}\{x \in \mathcal{X}(e_{x_i})\} \mathbb{1}\{y \in \mathcal{Y}(e_{y_i})\} G(x - e_{x_i}, y - e_{y_i}; s, \theta) \exp\left(-\frac{t - e_{t_i}}{\tau_m}\right) \quad (3)$$

where $\mathbb{1}\{\cdot\}$ is the indicator function, $\mathcal{X}(e_{x_i}) = [e_{x_i} - s, e_{x_i} + s]$ and $\mathcal{Y}(e_{y_i}) = [e_{y_i} - s, e_{y_i} + s]$ denote the receptive field of the neuron, and τ_m denotes the decay time constant. The function $G(\cdot; s, \theta)$ represents the filters grouped by s and θ . When the membrane voltage of neuron in $S1$ layer exceeds its threshold V_{thr}^{s1} , the neuron will fire a spike. The threshold V_{thr}^{s1} is set as 2 in this paper.

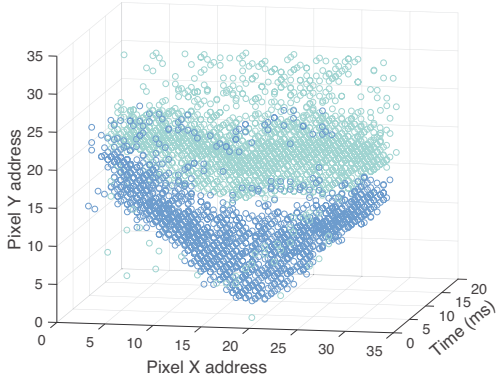


Figure 2: Visualization of the AER event stream representing the “heart” symbol in Cards dataset (Serrano-Gotarredona and Linares-Barranco 2015). ON and OFF events are represented by cyan and blue circles respectively.

From $S1$ Layer to $C1$ Layer Feature maps in $S1$ layer are divided into adjacent non-overlapping 2×2 cell regions, namely $S1$ units. When a neuron in $S1$ layer emits a spike, other neurons in the same unit will be inhibited, and all neurons in this unit will be forced to reset to V_{reset} , which is typically set as 0. This lateral inhibition mechanism ensures that only maximal responses in $S1$ units can be propagated to the subsequent layers. Therefore, we only need to observe which neuron generates an output spike to the $C1$ neuron first instead of comparing the neuron responses.

Classification Using Segmented Probability-Maximization (SPA) Learning Algorithm

After extracting the spatio-temporal features, we introduce the SPA learning algorithm of SNN to enable the trained neurons to respond actively to their representing classes. During classification, the model will give the result based on the firing rate of each trained neuron.

Neuron Model In this paper, we employ the Leaky Integrate-and-Fire (LIF) model, which has been used widely to simulate spiking neurons and is good at processing temporal information, to describe the neural dynamics. In LIF model, each incoming spike induces a postsynaptic potential (PSP) to the neuron. For an incoming spike received at t_i , the normalized PSP kernel is defined as follows:

$$K(t - t_i) = V_0 \left(\exp\left(-\frac{(t - t_i)}{\tau_m}\right) - \exp\left(-\frac{(t - t_i)}{\tau_s}\right) \right) \quad (4)$$

where τ_m and τ_s denote decay time constants of membrane integration and synaptic currents respectively, and the ratio between them is fixed at $\tau_m/\tau_s = 4$. The coefficient V_0 normalizes PSP so that the maximum value of the kernel is 1. The membrane voltage $V(t)$ of the decision neuron is described as:

$$V(t) = \sum_i w_i \sum_{t_i} K(t - t_i) + V_{rest} \quad (5)$$

where w_i and t_i are the synaptic weight and the firing time of the afferent i . V_{rest} denotes the resting potential of the neuron, which is typically set as 0.

SPA Learning Algorithm We adopt a biologically-inspired activation function called *Noisy Softplus*, which is well-matched to the response function of LIF spiking neurons (Liu and Furber 2016). The mean input current in (Liu and Furber 2016) can be approximated to the effective input current, which can be reflected by peak voltage $V(t_{peak})$ of the receptive neuron. Here t_{peak} denotes the time at which the postsynaptic voltage reaches its peak value. In the following, we use V_{peak} to represent $V(t_{peak})$ for short. We rewrite the *Noisy Softplus* function in (Liu and Furber 2016) as:

$$f_{out} = \log(\exp(V_{peak}) + 1) \quad (6)$$

where f_{out} is the normalized output firing rate. For a C -class categorization task, we need C decision neurons, each representing one class. The output firing rate of the neuron representing class j is:

$$f_{out}^j = \log(\exp(V_{peak}^j) + 1) \quad (7)$$

where V_{peak}^j is the peak voltage of the neuron representing class j .

The classification decisions are made based on the firing rates of spiking neurons in decision layer, so the aim of the learning is to train the decision neurons to respond actively to the input patterns of the class they represent. To improve the reliability of neuron responses and effectiveness of learning, we introduce the probability of the class that the sample actually belongs to, and constantly increase the probability. We define the probability that k -th sample belongs to class j as:

$$P(\hat{c}_k = j) = \frac{f_{out}^j}{\sum_{j'=1}^C f_{out}^{j'}} \quad (8)$$

where \hat{c}_k denotes the predicted class of the k -th sample. We will use f_{sum} to denote $\sum_{j'=1}^C f_{out}^{j'}$ for convenience. Furthermore, we use the cross-entropy to define the loss function of k -th sample as:

$$L_k = -\log(P(\hat{c}_k = c_k)) \quad (9)$$

where c_k denotes the actual class of the k -th sample.

We then minimize the cost function with gradient descent optimization and iteratively update the synaptic weight w_i by:

$$\Delta w_i = -\lambda \frac{\partial L_k}{\partial V_{peak}^j} \frac{\partial V_{peak}^j}{\partial w_i} \quad (10)$$

where λ is the learning rate which is set as 0.1, and $\frac{\partial L_k}{\partial V_{peak}^j}$ and $\frac{\partial V_{peak}^j}{\partial w_i}$ are evaluated as below:

$$\frac{\partial L_k}{\partial V_{peak}^j} = \begin{cases} -(f_{out}^j)^j \frac{f_{sum} - f_{out}^j}{f_{sum} f_{out}^j} & j = c_k \\ (f_{out}^j)^j \frac{1}{f_{sum}} & j \neq c_k \end{cases} \quad (11)$$

$$\frac{\partial V_{peak}^j}{\partial w_i} = \sum_{t_S \leq t_i < t_{peak}^j} K(t_{peak}^j - t_i) \quad (12)$$

where t_S is the starting time of an event stream.

Peak Detection There may contain multiple voltage peaks of a neuron in an event stream covering a period of time. The voltage peak is triggered by a burst of incoming feature spikes, indicating that the neuron has received a large amount of information at the time of the voltage peak. To further utilize these informative time points, we propose a *peak detection* mechanism (PD) to locate the peaks of the membrane voltage segment by segment. The principle of PD is as follows: within a search range with length t_R starting at t_S , t_{peak} of the neuron representing class j is defined as:

$$t_{peak}^j = \arg \max_t \{V^j(t) | t \in (t_S, t_S + t_R]\} \quad (13)$$

where $V^j(\cdot)$ is the membrane voltage of the neuron representing class j . If multiple time points in the current segment meet the criterion, the earliest one is chosen. After locating the voltage peak in the current segment for each neuron, t_S will be updated as:

$$t_S = \max_{1 \leq j \leq C} t_{peak}^j \quad (14)$$

The full procedure for SPA learning algorithm with PD is summarized in Algorithm 1.

Network Design We use the extracted feature spikes from $C1$ layer as the input of the classification process. Encoding neurons are fully connected to the decision neurons. Since the activity of one single neuron can be easily affected, the population coding is adopted to improve the reliability of information coding (Natarajan et al. 2008). In this paper, each input class is set to associate with a population of 10 decision neurons.

During training, the synaptic weights are firstly initialized with random values and updated using SPA learning algorithm. When training is done, we keep the synaptic weights fixed, and set the threshold of decision neurons V_{thr}^{dec} as 1. During testing, when the decision neuron’s membrane potential is higher than its threshold V_{thr}^{dec} , the neuron will fire

Algorithm 1 SPA learning algorithm for classification

Require: *FeatureSpikes* from encoding layer

Ensure: Synaptic weight w

```

1: function SPA(FeatureSpikes)
2:   Initialize the neuron membrane voltage  $V$  and synaptic weights  $w$ . Set the iteration time  $n_{iter}$ , the time length  $L$  and the learning rate  $\lambda$ ;
3:   while  $n_{iter}$  not reached do
4:     Calculate the membrane voltage  $V$  by (5);
5:     Initialize  $t_S = 0$ ;
6:     while  $t_S < L$  do
7:       Find the  $t_j^{peak}$  of each decision neuron  $j$  in the search range of  $(t_S, t_S + t_R]$  according to (13);
8:       Update  $w_i$  by  $\Delta w_i$  for each afferent  $i$  by (10);
9:       Update  $t_S$  according to (14);
10:    end while
11:  end while
12:  Return  $w$ ;
13: end function

```

a spike and its membrane potential will be reset to V_{reset} . The predicted class for the input is determined by averaging the firing rates of neurons per class and then choosing the class with the highest average firing rate.

Experimental Results

In this section, we evaluate the performance of our proposed approach on several public AER datasets and compare it with other AER classification methods.

Datasets

Four publicly available datasets are used to analyze the performance.

1) **MNIST-DVS** dataset (Lichtsteiner, Posch, and Delbruck 2008): it is obtained with a DVS by recording 10,000 original handwritten images in MNIST moving with slow motion.

2) **Neuromorphic-MNIST (NMNIST)** dataset (Orchard et al. 2015a): it is obtained by moving an ATIS camera in front of the original MNIST images. It consists of 60,000 training and 10,000 testing samples.

3) **Cards** dataset (Serrano-Gotarredona and Linares-Barranco 2015): it is captured by browsing a card deck in front of the sensitive DVS camera and recording the information in an event stream. The event stream consists of 10 cards for each of the 4 card types (spades, hearts, diamonds, and clubs).

4) **CIFAR10-DVS** dataset (Li et al. 2017): it consists of 10,000 samples, which are obtained by displaying the moving CIFAR10 images on a monitor and recorded with a fixed DVS camera.

Figure 3 shows some samples of these four datasets.

Experimental Settings

We randomly partition the used datasets into two parts for training and testing. The result is obtained over multiple runs with different training and testing data partitions. For fair comparison, the results of competing methods and ours are obtained under the same experimental settings. The results of competing methods are from the original papers (Orchard et al. 2015b; Lagorce et al. 2017; Zhao et al. 2015), or (if not in the papers) from the experiments using the code with our optimization.

Performance on Different AER Datasets

On MNIST-DVS Dataset The time length of reconstructing a digit silhouette in this dataset is no more than $80ms$, thus the search range t_R and the parameter τ_m are both set as $80ms$. This dataset has 10,000 samples, 90% of which are randomly selected for training and the remaining ones are used for testing. The performance is averaged over 10 runs.

Our model gets the classification accuracy of 96.7%, or 3.3% error rate on average. Table 1 shows that our model gets a higher performance and in addition achieves an more than 3 times smaller error rate than Zhao’s method (11.9%), HFirst (21.9%), and HOTS (19.7%) (Zhao et al. 2015; Orchard et al. 2015b; Lagorce et al. 2017).

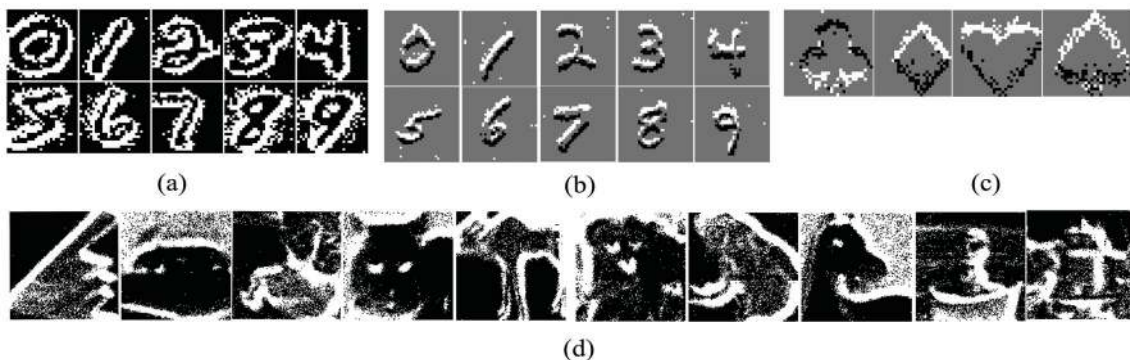


Figure 3: Some reconstructed images from the used datasets: (a) MNIST-DVS dataset; (b) NMNIST dataset; (c) Cards dataset; (d) CIFAR10-DVS dataset.

Table 1: Comparison of classification accuracy on four datasets.

	MNIST-DVS	NMNIST	Cards	CIFAR10-DVS
Zhao’s (Zhao et al. 2015)	88.1%	85.6%	86.5%	21.9%
HFirst (Orchard et al. 2015b)	78.1%	71.2%	97.5%	7.7%
HOTS (Lagorce et al. 2017)	80.3%	80.8%	100.0%	27.1%
This Work	96.7%	96.3%	100.0%	32.2%

On NMNIST Dataset This dataset records the event streams produced by 3 saccadic movements of the DVS camera. The time length of each saccadic movement is about $120ms$, thus the search range t_R and the parameter τ_m are set as $120ms$. This dataset is inherited from MNIST, and has been partitioned into 60,000 training samples and 10,000 testing samples by default.

Our model gets the classification accuracy of 96.3% on average. Table 1 shows our model outperforms Zhao’s method (Zhao et al. 2015), HFirst (Orchard et al. 2015b), and HOTS (Lagorce et al. 2017) by a margin of 10.7%, 25.1% and 15.5% respectively. Notice that HFirst has relatively poor performance on NMNIST, compared with the other methods. However, this drop in accuracy is expected because HFirst is designed to detect simple objects, while great variation of object appearance exists in the NMNIST dataset. In addition, SKIM network on NMNIST dataset (Cohen et al. 2016) achieves an accuracy of 92.9%, which is also 3.8% less than our model.

On Cards Dataset The time length of the recordings in this dataset is about $20ms$. Since $8ms$ is enough to reconstruct a card silhouette, we set the search range t_R and τ_m as $8ms$. For each category of this dataset, 50% are randomly selected for training and the others are used for testing. The performance is averaged over 10 runs to get the result.

Our model achieves the classification accuracy of 100% for the testing set. HFirst and HOTS also reach relatively high accuracy, while Zhao’s model only gets an accuracy of 86.5%. This is because the number of training samples in this dataset is very limited (only 5 samples per class), and at most one spike in Zhao’s method is emitted for each encoding neuron to represent features. Therefore, there is not enough effective information for tempotron classifier to train

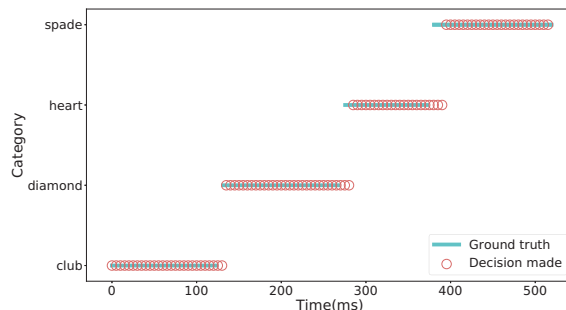


Figure 4: Performance of the proposed model on a continuous event stream from Cards dataset. All testing symbols are connected one by one into a continuous event stream and then fed to the model for evaluation. The cyan lines represent the ground truth of classification, and the red circles denote the decisions made by our model every $5ms$.

the SNN.

We further run our model on a continuous event stream which combines all the testing samples, since this dataset is originally in the form of a stream of events. The result is shown in Figure 4. Every $5ms$ we give a decision (one red circle in the figure). We can see that, at the beginning of the appearance of a new type, the decisions we made are slightly delayed. This is because, when a new type appears, the neurons representing the new class have not accumulated enough responses to outperform the neurons representing the former class. Nevertheless, after about $15ms$, the decisions can match very well with the ground truth.

On CIFAR10-DVS Dataset In this dataset, $80ms$ is enough to reconstruct a object silhouette, thus the search range t_R and the parameter τ_m are both set as $80ms$. We randomly select 90% of samples for training and the others for testing. The experiments are repeated 10 times to obtain the average performance.

The classification task of this dataset is more challenging because of the complicated object appearance and the large intra-class variance, therefore the classification results on this dataset of all the compared methods are relatively poor. Nevertheless, our model achieves the classification accuracy of 32.2%, which is still higher than other methods.

Effects of the SPA

In this section, we carry out more experiments to demonstrate the effects of our model using SPA in detail. The experiments are conducted on MNIST-DVS dataset and the parameter settings are the same as the previous section.

Sample Efficiency Sample efficiency measures the quantity of samples or information required for a model to reach a certain level of accuracy. In the AER object classification task, the length of the event stream determines the amount of information. We examine the impact of the time length on the algorithm. The experiments are conducted on recordings with the first $100ms$, $200ms$, $500ms$ and full length (about $2s$) of the original samples, respectively. Since Zhao’s method achieves a competitive classification result on the full length of MNIST-DVS as shown in Table 1, we list the results of both Zhao’s method and ours in Table 2.

It can be noticed that: 1) the accuracy of both two methods keeps increasing when longer recordings are used, which is because longer recordings provide more information; 2) our model consistently outperforms Zhao’s method on recordings with every time length in Table 2. In fact, even on the recordings with time length $100ms$, our model still yields a relatively better result than Zhao’s method on the recordings of full length. This result demonstrates that with the same or even less information, our model could reach a better classification accuracy, which proves its sample efficiency.

Table 2: Performance on recordings with different time length of MNIST-DVS dataset.

Time Length	Zhao’s	This Work
100 ms	76.9%	89.4%
200 ms	82.6%	92.7%
500 ms	85.9%	94.9%
Full (about 2s)	88.1%	96.7%

Inference with Incomplete Information Inference with incomplete information requires the model to be capable of responding with a certain level of accuracy, when information of the object is incomplete during testing. We use the recordings of $500ms$ for training and observe the performance within the first $300ms$ recordings of three methods, including the proposed SPA learning algorithm, the tempotron learning algorithm used in Zhao’s method, and the

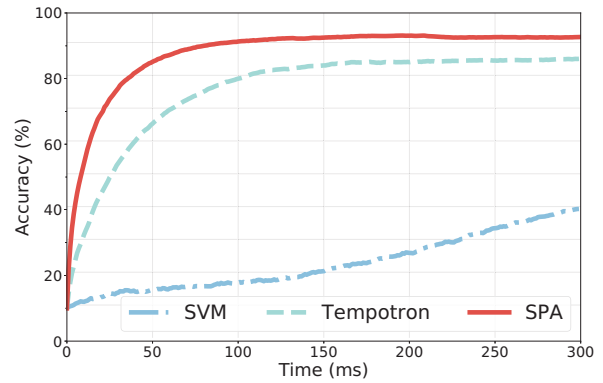


Figure 5: Performance of the inference with incomplete information on MNIST-DVS dataset.

nontemporal classifier SVM (with the same feature extraction procedure). The results are averaged over 10 runs and shown in Figure 5.

As the event stream flows in, the classification accuracy of models with the three algorithms keeps increasing. The model with SPA has the highest performance among all the methods, especially within the first $100ms$ when the input information is extremely incomplete. This is because in our SPA learning algorithm, each sample is trained several times based on the informative time points in every segment, which increases the diversity of the training information. Therefore, the model has a better generalization ability and can be promoted rapidly at the early stage, even though the information is incomplete.

Conclusion

In this paper, we propose an effective AER object classification model using a novel SPA learning algorithm of SNN. The SPA learning algorithm iteratively updates the weight by maximizing the probability of the actual class to which the sample belongs. A PD mechanism is introduced in SPA to locate informative time points segment by segment on the temporal axis, based on which the information can be fully utilized by the learning. Experimental results show that our approach yields better performance on four public AER datasets, compared with other benchmark methods specifically designed for AER tasks. Moreover, experimental results also demonstrate the advantage of sample-efficiency and the ability of inference with incomplete information of our model.

Acknowledgments

This work is partly supported by National Key Research and Development Program of China (2017YFB1002503), Zhejiang Lab (No. 2019KC0AD02), Ten Thousand Talent Program of Zhejiang Province (No.2018R52039), and National Science Fund for Distinguished Young Scholars (No. 61925603).

References

- Bohte, S. M.; Kok, J. N.; and La Poutre, H. 2002. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing* 48(1-4):17–37.
- Brandli, C.; Berner, R.; Yang, M.; Liu, S.-C.; and Delbruck, T. 2014. A 240×180 130 db $3 \mu\text{s}$ latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits* 49(10):2333–2341.
- Cohen, G. K.; Orchard, G.; Leng, S.-H.; Tapson, J.; Benosman, R. B.; and Van Schaik, A. 2016. Skimming digits: neuromorphic classification of spike-encoded images. *Frontiers in neuroscience* 10:184.
- Diehl, P. U., and Cook, M. 2015. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience* 9:99.
- Gütig, R., and Sompolinsky, H. 2006. The tempotron: a neuron that learns spike timing-based decisions. *Nature neuroscience* 9(3):420.
- Lagorce, X.; Orchard, G.; Galluppi, F.; Shi, B. E.; and Benosman, R. B. 2017. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(7):1346–1359.
- Leñero-Bardallo, J. A.; Serrano-Gotarredona, T.; and Linares-Barranco, B. 2011. A $3.6 \mu\text{s}$ latency asynchronous frame-free event-driven dynamic-vision-sensor. *IEEE Journal of Solid-State Circuits* 46(6):1443–1455.
- Li, H.; Liu, H.; Ji, X.; Li, G.; and Shi, L. 2017. Cifar10-dvs: An event-stream dataset for object classification. *Frontiers in Neuroscience* 11:309.
- Lichtsteiner, P.; Posch, C.; and Delbruck, T. 2008. A 128×128 120 db $15 \mu\text{s}$ latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits* 43(2):566–576.
- Liu, Q., and Furber, S. 2016. Noisy softplus: a biology inspired activation function. In *International Conference on Neural Information Processing*, 405–412. Springer.
- Liu, Q.; Pan, G.; Ruan, H.; Xing, D.; Xu, Q.; and Tang, H. 2019. Unsupervised aer object recognition based on multiscale spatio-temporal features and spiking neurons. *arXiv preprint arXiv:1911.08261*.
- Mohammed, A.; Schliebs, S.; Matsuda, S.; and Kasabov, N. 2012. Span: Spike pattern association neuron for learning spatio-temporal spike patterns. *International journal of neural systems* 22(04):1250012.
- Natarajan, R.; Huys, Q. J.; Dayan, P.; and Zemel, R. S. 2008. Encoding and decoding spikes for dynamic stimuli. *Neural computation* 20(9):2325–2360.
- Neil, D., and Liu, S.-C. 2016. Effective sensor fusion with event-based sensors and deep network architectures. In *2016 IEEE International Symposium on Circuits and Systems (IS-CAS)*, 2282–2285. IEEE.
- Orchard, G.; Jayawant, A.; Cohen, G. K.; and Thakor, N. 2015a. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience* 9:437.
- Orchard, G.; Meyer, C.; Etienne-Cummings, R.; Posch, C.; Thakor, N.; and Benosman, R. 2015b. Hfirst: A temporal approach to object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(10):2028–2040.
- Pérez-Carrasco, J. A.; Zhao, B.; Serrano, C.; Acha, B.; Serrano-Gotarredona, T.; Chen, S.; and Linares-Barranco, B. 2013. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing—application to feedforward convnets. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(11):2706–2719.
- Ponulak, F., and Kasiński, A. 2010. Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting. *Neural Computation* 22(2):467–510.
- Posch, C.; Matolin, D.; and Wohlgenannt, R. 2011. A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits* 46(1):259–275.
- Qi, Y.; Shen, J.; Wang, Y.; Tang, H.; Yu, H.; Wu, Z.; and Pan, G. 2018. Jointly learning network connections and link weights in spiking neural networks. In *IJCAI*, 1597–1603.
- Riesenhuber, M., and Poggio, T. 1999. Hierarchical models of object recognition in cortex. *Nature Neuroscience* 2(11):1019.
- Serrano-Gotarredona, T., and Linares-Barranco, B. 2015. Poker-dvs and mnist-dvs. their history, how they were made, and other details. *Frontiers in Neuroscience* 9:481.
- Sironi, A.; Brambilla, M.; Bourdis, N.; Lagorce, X.; and Benosman, R. 2018. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1731–1740.
- Wu, Z.; Pan, G.; Principe, J. C.; and Cichocki, A. 2014. Cyborg intelligence: Towards bio-machine intelligent systems. *IEEE Intelligent Systems* 29(6):2–4.
- Wu, Z.; Pan, G.; and Zheng, N. 2013. Cyborg intelligence. *IEEE Intelligent Systems* 28(5):31–33.
- Yu, Q.; Tang, H.; Tan, K. C.; and Li, H. 2013. Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns. *Plos one* 8(11):e78318.
- Zhang, M.; Wu, J.; Chua, Y.; Luo, X.; Pan, Z.; Liu, D.; and Li, H. 2019. Mpd-al: an efficient membrane potential driven aggregate-label learning algorithm for spiking neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1327–1334.
- Zhao, B.; Ding, R.; Chen, S.; Linares-Barranco, B.; and Tang, H. 2015. Feedforward categorization on aer motion events using cortex-like features in a spiking neural network. *IEEE Transactions on Neural Networks and Learning Systems* 26(9):1963–1978.