Recognizing Scoring in Basketball Game from AER Sequence by Spiking Neural Networks

Jiangrong Shen[1, 2, 3], Yu Zhao[1, 2], Jian K. Liu[3], and Yueming Wang*[2, 4, 5]

[1]The College of Computer Science and Technology, Zhejiang University.
[2]The Qiushi Academy for Advanced Studies, Zhejiang University.
[3]Centre for Systems Neuroscience, Department of Neuroscience, Psychology and Behaviour, University of Leicester.
[4]State Key Lab of CAD&CG, Zhejiang University.
[5]Zhejiang Lab.

***Abstract*—The automatic score detection and recognition in basketball game has important application potentials, for examples, basketball technique analysis and 24 second control in the game. Although existing studies have been conducted on broadcast videos, most of them usually learned a machine learning algorithm on long videos recorded by traditional cameras. Address Event Representation (AER) sensor provides a possibility to deal with the problem by a human sensing manner. It represents the visual information as a series of spike-based events and records event sequences. Compared to traditional videos, AER events can fully utilize their addresses and timestamp information, forming precise spatio-temporal features with significantly less storage cost. More importantly, it issues spikes which can be naturally processed by human-style spiking neural networks (SNNs). In this paper, we propose to recognize scoring in basketball game from AER sequences. A new model is designed to extract dynamic features and discriminate different event streams using SNN. To handle the imbalance problem between positive and negative samples, we use an imbalanced Tempotron algorithm in our SNN model. Meanwhile, an AER sequence dataset of basketball games is collected. The experimental results demonstrate that our method achieves better performance compared with existing models.**

***Index Terms*—Basketball scoring recognition, Spiking neural networks, AER, Encoding, Supervised rules**

I. Introduction

Automatic scoring recognition in basketball games is highly helpful in both professional basketball technique analysis and amateur basketball games [1] [2] [3]. Most researches in this area were achieved by recording games in videos and building a mathematical model by machine learning methods. Recently, Address Event Representation (AER) sensors provide a new way to deal with the problem. It is a type of neuromorphic vision system and records changes in the scene, i.e., a series of spike-based events representing visual information, resulting in an event sequence. In this sequence, spikes are generated when events such as object moving happen. Since spikes transferring is a common way in human brain sensing, it can be naturally connected to a human-style model to recognize scoring, e.g., spiking neural network (SNN). This paper aims to detect basketball scoring based on AER sequences.

*Corresponding author. Email: ymingwang@zju.edu.cn.

Some successful studies about AER vision sensors have been proposed, such as the Asynchronous Time-based Image Sensor (ATIS) [4], event-driven Dynamic Vision Sensor (DVS) [5] and the DAVIS sensor [6]. The sensors are driven by relative changes of light intensity, if the change exceeds the threshold, an event containing a tuple of the timestamp and address will be emitted in the corresponding pixel. If there is no change in intensity of the pixel, then no spike appears. By this event-triggering manner, AER based cameras can output high temporal resolution (in the range of microseconds) with low bit-rate compared to traditional cameras, which makes it very suitable for use in resource-constrained scenarios. There have been various applications based on AER dataset. Hu et al. summarized the DVS-based benchmark datasets for object tracking, action recognition and object recognition [7]. These AER datasets were collected by displaying existing benchmark videos on a monitor and recoding the screen by a DVS sensor. The benchmark datasets are conductive to the development of encoding methods and learning algorithms to process and recognize event-based spatio-temporal patterns. In sport videos, the attractive scenes often have limited time and high speed. The DVS sensor is more suitable for sports video application because of its higher time resolution and lower storage redundancy.

How to utilize the rich spatio-temporal information contained in AER representations for dynamic event recognitions is still a problem. Dynamic event recognition is believed to originate from the representations of dynamic visual features. Humans can easily discriminate different objects within a short time [8]. Spiking neural networks (SNNs) have rich biological plausibility and they communicate via discrete spikes instead of numerical values [9]. In SNNs, a neuron is activated only when it receives an input spike, hence inactive neurons without any input spikes can be put into low power mode to save power. Although effort has been made to build biological plausible systems using spike [10] or mimic the visual formation in human retina via a more biological way [11], most of them focus on the static image classification tasks not dynamic one. Therefore, robust object recognition in spiking neural systems remains a challenging in neuromorphic

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

computing area as it needs to solve both the effective encoding of sensory information and its integration with downstream learning neurons.

There have been some studies to develop different SNN models to utilize the rich spatio-temporal information contained in AER representations for dynamic object recognitions. Serre et al. have proposed a hierarchical visual system which can extract AER based features within the pattern complexities and position invariance [12]. Chen et al. proposed a novel method to extract size and position information from moving objects, which can perform well especially in human postures detection in real-time video captured by AER based sensors [13]. Zhao et al. used a convolution-based method to extract features from AER events by introducing an event-driven convolution mixed network [14]. Peng et al. have developed a feature extraction named Bag of Event (BOE) to capture the features from AER sensors within joint probability model [15]. The above studies explored how to build and process AER based representations from the sensors or CNN-based models. However, there is still a problem in SNN for the imbalanced data. The imbalance problem means that one of the two classes having more sample than the other class [16] [17]. Obviously, it occurs in the basketball scoring recognition problem for the relatively few positive samples (scoring) compared to negative samples (Failed to score). In this case, the negative class tends to be overwhelmed during training process with the common SNN classifiers. In data mining research field, there are several approaches to solve this problem [17]. At the data level, different forms of re-sampling can change the dataset distribution. At the algorithm level, the loss of different class can be adjusted to counter the data imbalance. Lin et al. proposed focal loss that adds a factor to standard cross entropy criterion [18]. This loss can down-weights the loss for well-classified examples and focus on hard and misclassified samples. It is proved efficient to prevent the vast number of easy negatives from overwhelming the detector during training. However, these algorithms cannot be applied to SNNs models directly.

Motivated by those previous works, this paper proposes a robust spike-based network for scoring event recognition in basketball game. This spiking neural system consists of sparse temporal encoding and Tempotron classifier. The sparse temporal coding part consists of feature extraction, peak detection and spike train generation [14]. The HMAX model with S1 layer and C1 layer extracts the crucial spatio-temporal features from the input AER sequences. The peak detection part controls the switch of spike train generation. Moreover, we adopt balance factor into the primal Tempotron algorithm [19] in order to relieve the imbalance between positive samples and negative samples. Some details of the contribution can be summarized as follows.

- A new event streams dataset for basketball scoring recognition is collected by DVS sensor. The event streams are segmented automatically and preprocessed to dislodge noises. These event streams are split into positive class (scored) and negative class (Failed to score). There are 6267 samples totally, where contains only 512 positive ones. This dataset could build a bridge between a real-world task of basketball scoring recognition and the dynamic human-style visual formation.
- This work aims to solve the dynamic event detection in basketball game. It is more difficult because of not only its complexities of dynamic input spatio-temporal patterns, but also the difficulties lying in the relation between the related series frames. Through the sparse temporal coding, the complex spatio-temporal patterns in AER data can be encoded efficiently as spike trains. Meanwhile, the proposed imbalanced Tempotron method overcomes the data imbalance in the basketball scoring dataset and improves the recognition of key events effectively.

This paper evaluates the proposed model on the newly released basketball scoring dataset. Experimental results show the proposed framework is not only capable of extracting rich spatio-temporal features, but also recognizing dynamic traces with a good performance.

II. Method

The framework used in our paper is shown as Figure. 1. There are three parts consisting of feature extraction with Hmax model, peak detection and spike train feature generation and pattern classification with imbalanced Tempotron algorithm. To start with, the incoming events are gathered into peak detection part, which are fed into the detection LIF neuron in that part. Once the potential of that LIF neuron reaches a relatively high value, a peak is emitted and events that caused this peak are segmented as the input data of feature extraction part for this peak. After the detection of that peak, the gate between feature extraction and imbalanced Tempotron classification is open, the recognition process is triggered and the features extracted by feature extraction part are employed as input spike train for the classifier and transmitted into final recognition process. After all the events in one event stream are segmented into small segmentations and fed into feature extraction part and recognition part, the process for this event stream is finished. Then the recognized category of this event streams is achieved by the imbalanced Tempotron classifier.

A. Peak Detection and Spike Pattern Generation

Note that each complete event stream usually contains thousands of events and the time interval between two events can be very small, which causes one event could not carry enough information to recognizing its corresponding event stream belongs to which category. Hence, we need a mechanism to decide when to carry out classification. Here we adopt the time domain clustering algorithm with motion symbol detector module proposed in [14].

The motion symbol detector module contains one leaky integration neuron and peak detection unit. As illustrated in Figure. 2 (b), the potential of that neuron is updated by:

$$V(t) = \sum_{t_i} K(t - t_i) + V_{rest}, \tag{1}$$

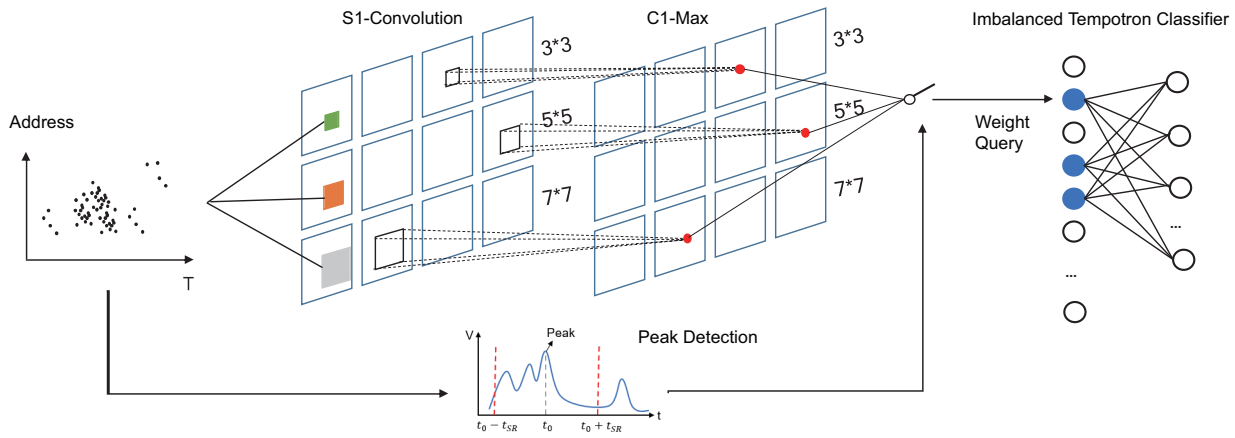

Fig. 1. The framework for basketball scoring recognition, which consists of feature extraction, peak detection and imbalanced Tempotron classifier. The feature extraction contains S1 layer and C1 layer for event stream convolution and max operation respectively. Peak detection part is implemented by a detection LIF neuron. To start with, the incoming events are gathered into peak detection part, which are fed into the detection LIF neuron. Once the potential of that LIF neuron reaches a relatively high value, a peak is emitted. After a peak is detected by peak detection part, the gate between feature extraction and imbalanced Tempotron classification is open, the recognition process is triggered and the features extracted by feature extraction part are transmitted into final recognition process. The input AER data are convoluted by S1-convolution with different filters, which attains feature maps. These filters contain different scales and orientations. After that, the neurons in feature maps are completed by max operation. Only the neuron with max value survives. These remaining spike events containing in these survival neurons are transmitted into the final recognition process as the extracted feature. To reduce unnecessary memory accessing in imbalanced Tempotron classifier in recognition process, the weight query mechanism is introduced to search weights matching these survival neurons, which are marked by blue circles in this figure.

Where t_i is the time when event comes in. V_{rest} is the rest potential of the leaky integrate neuron and typically set as 0. A normalized PSP kernel K vanishing for $t_i > t$ is as follows:

$$K(t-t_i) = V_0(exp(-\frac{t-t_i}{\tau_m}) - exp(-\frac{t-t_i}{\tau_s})), \quad (2)$$

where V_0 is used to normalize the maximum of kernel to be 1.0. The parameters τ_m and τ_s denote the decay time constants of membrane integration and synaptic currents, respectively.

The peak detection unit is applied to detect local temporal peaks according to neuron's potential. In detail, if the potential at time t_0 is bigger than that in the time range $[t_0 - t_{SR}/2, t_0 + t_{SR}/2]$, then t_0 is considered as a peak. Once a peak is detected, the switch of classification processing is opened. In addition, we design a refractory time to make the motion symbol detector remain silenced to avoid small peaks caused by background noise events. With motion symbol detection unit, C1 feature maps are converted to spike trains and fed into LIF neurons when peaks detected. These neurons work simultaneously according to the weight query table to avoid huge memory consumption. Finally, we adopt Tempotron algorithm for these LIF neurons to classify different spike patterns.

B. Feature Extraction with HMAX Model

To extract features from event streams, we adopt hierarchical HMAX model with S1 layer and C1 layer. Different from static image processing, only when one input address event comes in, the convolution and max operations are triggered. For S1 layer, it convolves the input event streams with multiple Gabor filters. Each filter has different receptive field size to respond best for basic feature of certain orientation, which means it can select the corresponding feature. The sizes of these Gabor filters contain four scales $\sigma = [3, 5, 7, 9]$ and four orientations $\theta = [0, 45, 90, 135]$. Hence these are totally 16 different filters. The filter function is as follows:

$$G(x,y) = exp(-\frac{X^2 + \gamma^2 Y^2}{2\sigma^2}) * cos(\frac{2\pi}{\lambda} * X), \quad (3)$$

which satisfies

$$\begin{aligned} X &= xcos\theta + ysin\theta, \\ Y &= -xsin\theta + ycos\theta, \end{aligned} \quad (4)$$

where λ and θ denote the wavelength and effective width, their values are set to be $[1.5, 2.5, 3.5, 4.6]$ and $[1.2, 2.0, 2.8, 3.6]$ respectively. γ represents the wavelength, which is set to be 0.3 as tuned in [20] [21].

During event streams convolution, forgetting mechanism is introduced to implement continuously event-driven processing. As shown in Figure 2 (a), when an event comes in, the convolution operation on the position specified by the address of that event is integrated to update the response map. Besides, the forgetting mechanism makes the values of response map decrease toward the resting potential as time goes by. In this way, the effects of much earlier events are eliminated and the effects of closer events are improved for its stronger correlation. For simplicity, a constant linear leakage is adopted.

After convolution operation, C1 layer performs the max operation over the corresponding receptive field in S1 response maps. Through this competition, only the neuron with max value survives. Then each survival neuron in C1 maps represents a certain feature.

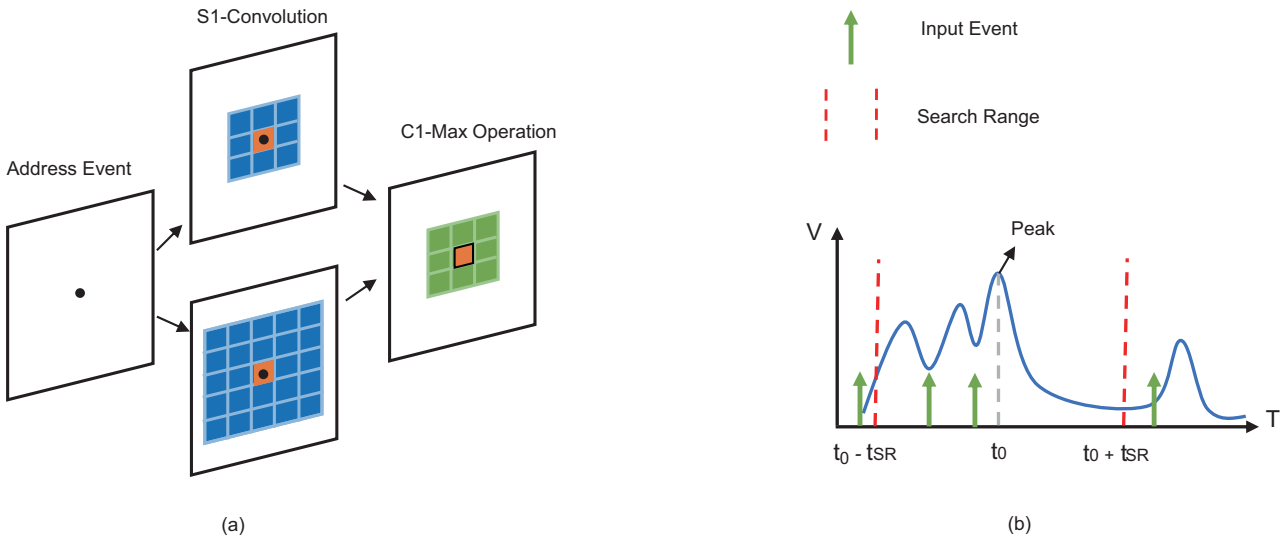

Fig. 2. The feature extraction and peak detection parts in the framework. (a) describes the processes of convolution and max operation for the coming events (black dot). The address of that event specifies the position where convolutional kernel is overlaid into the response map. After the convolution operation, the feature map holding up the information extracted from filters with different scales and orientations is computed (blue squares). Those feature maps are fed into C1 layer and all the neurons in one receptive field of those feature maps are competed against each other. The neuron with max value would survive as the final feature of C1. (b) is to find whether there is a peak in the search range. Once the peak is detected, the Tempotron classifier would collect the entire feature after C1 which used as the input spike train of classifier.

C. Tempotron for Imbalanced Data

After the feature extraction process, the imbalanced Tempotron algorithm is employed as the classifier to recognize the category of the input basketball scoring event stream. Based on Leaky Integrate-and-Fire (LIF) neuron model, a two-layer tempotron network is built. The neurons in the output layer are fully connected to the input neurons. Each neuron is permitted to fire only once. Moreover, the built network is trained by the improved Tempotron algorithm, which can relieve the data imbalanced problem.

1) LIF Neuron Model: Given an LIF neuron j, suppose there are N presynaptic afferents contributing to it. Neuron j is driven by exponential decaying synaptic currents generated by its N presynaptic neurons. Then the subthreshold membrane voltage of neuron j is a weighted sum of postsynaptic potentials (PSPs) contributed by all incoming spikes:

$$V_j(t) = \sum_{i=1}^{N} W_{ij} \sum_{t_i<t} K(t - t_i) + V_{rest}, \tag{5}$$

where W_{ij} is the synaptic efficacy between postsynaptic neuron j and presynaptic afferent i, t_i and V_{rest} denote the firing time of presynaptic afferent i, and the rest potential of postsynaptic neuron j, respectively. A normalized PSP kernel K vanishing for $t_i > t$ is the same as Equation. 2. The postsynaptic neuron j fires a spike once its voltage V_j crosses the firing threshold V_{thr}. That is, neuron j generates an output spike at that time. Since we only consider the situation that postsynaptic neuron is fired only once in this paper, the voltages of that fired neuron smoothly decline to V_{rest} by shutting down all the following incoming spikes.

2) Tempotron Learning for Imbalanced Data: Based on LIF, we propose an improved Tempotron learning method, called imbalanced Tempotron Learning algorithm. In classification processing, the input patterns to the neurons belong to one of two types of $\oplus$ and $\ominus$. When a $\oplus$ is presented to the neuron, it fires a spike, and when a $\ominus$ appears, the neuron does not fire. Tempotron rule learns the synaptic weights of W_{ij} with gradient descent to minimize the error signals:

$$E_j = \begin{cases} \alpha(V_{thr} - V_j(t_{max})) & if \quad y = 1 \quad and \quad \oplus error, \\ \alpha(V_j(t_{max}) - V_{thr}) & if \quad y = 1 \quad and \quad \ominus error, \\ \beta(V_{thr} - V_j(t_{max})) & if \quad y = 0 \quad and \quad \oplus error, \\ \beta(V_j(t_{max}) - V_{thr}) & if \quad y = 0 \quad and \quad \ominus error, \end{cases} \tag{6}$$

where t_{max} is the time point that the neuron reaches its maximum voltage, and V_{thr} is the threshold for neurons to fire a spike. $y = 1$ and $y = 0$ denote the positive class and negative class respectively. $\oplus$ error means the error that the neuron should emit a spike but it does not, and $\ominus$ error is the error that the neuron should not emit a spike but it does. Different from the primal Tempotron method, we add α and β parameters to balance the training process. When the number of negative samples is much larger than positive ones, we set $\alpha > \beta$ to improve the effects of positive samples and weaken the effects of negative samples and vice versa. Actually, this loss function is a more general form, which is equal to primal function when λ and β are set to be 1.0. Based on that loss function, the gradients of parameter W_{ij} are computed follows:

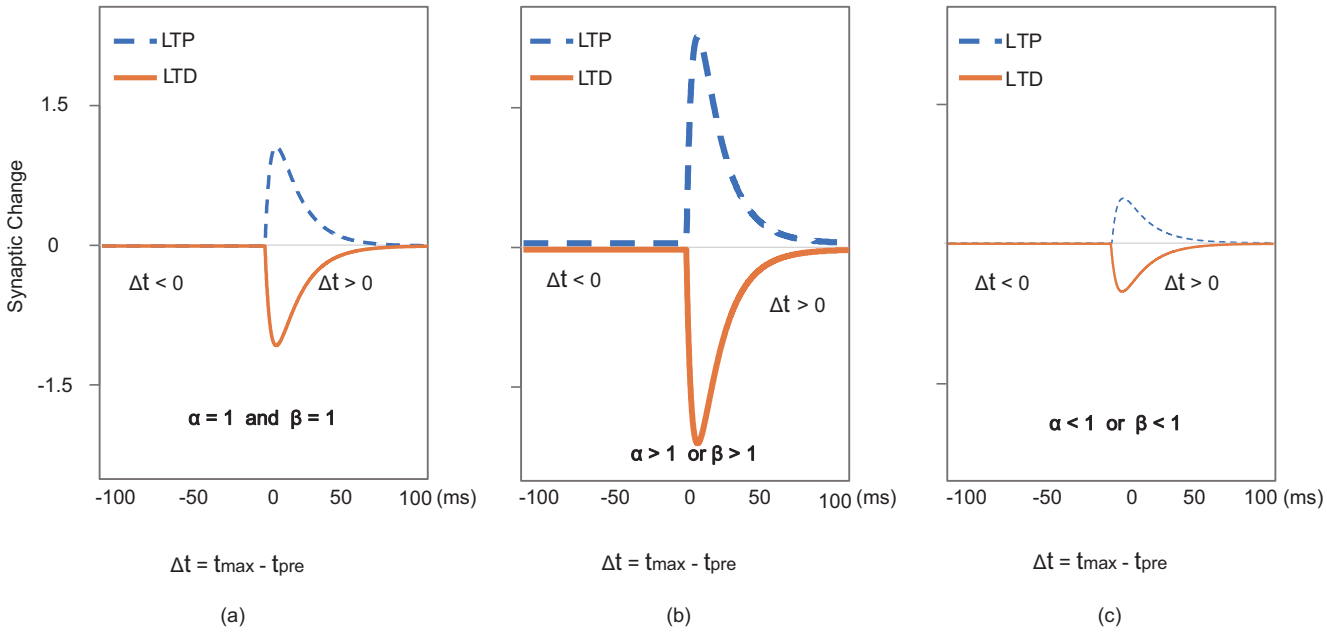

Fig. 3. The learning windows for imbalanced Tempotron algorithm with different factors. (a) shows the learning window when $\alpha = \beta = 1$, which is the same as original Tempotron. (b) is the learning window when factors satisfied $\alpha > 1$ or $\beta > 1$. The weight changes are improved because of the big factor. In contrast, once the values of factors become smaller than one, the corresponding learning window would be downscaled according to these factors.

$$\begin{cases} \Delta W_{ij}^{+} = \lambda_w \alpha \sum_{t_i<t_{max}} G_{ij}\, K(t_{max}-t_i) \quad if \quad y=1 \\ \qquad\qquad\qquad\qquad\qquad\qquad and \quad \oplus\, error, \\ \Delta W_{ij}^{-} = -\lambda_w \alpha \sum_{t_i<t_{max}} G_{ij}\, K(t_{max}-t_i) \quad if \quad y=1 \\ \qquad\qquad\qquad\qquad\qquad\qquad and \quad \ominus\, error, \\ \Delta W_{ij}^{+} = \lambda_w \beta \sum_{t_i<t_{max}} G_{ij}\, K(t_{max}-t_i) \quad if \quad y=0 \\ \qquad\qquad\qquad\qquad\qquad\qquad and \quad \oplus\, error, \\ \Delta W_{ij}^{-} = -\lambda_w \beta \sum_{t_i<t_{max}} G_{ij}\, K(t_{max}-t_i) \quad if \quad y=0 \\ \qquad\qquad\qquad\qquad\qquad\qquad and \quad \ominus\, error, \end{cases} \tag{7}$$

where λ_w is the weight learning rate for imbalance Tempotron classifier. As illustrated in Figure. 3, the scale of learning window is adjusted according to the factor values. When those two factors are both equal to 1.0, the synaptic change is the same as original Tempotron. Once these factors are bigger than 1.0, weight change scales are improved according to their values. On the contrary, the learning window is reduced for the smaller factors than 1.0. Then, the weights of network are updated by gradient descent rule according to the synaptic changes described above.

III. Results

In this section, we firstly describe the data collection process, which contains the automatic segmentation of event streams and data preprocessing. Then the potential of imbalance Tempotron under different parameters is investigated. Finally, the framework is applied into the basketball scoring recognition, its performance is compared with other methods such as unsupervised SNN, SVM and original Tempotron.

A. Data Collection

The original data of basketball playing is collected by DVS128 sensor. The size of screen is set as $128 * 128$, other parameters are the default ones. Then the collected data is recorded as AER data. The '.dat' data file could be transferred into '.mat' data file, which contains two variables of 'allAddr' and 'allTs'. To label samples and identify whether the basketball scored, we visualize the AER data of fixed time interval as one frame image to tag its label. With the tagged labels, we could employ the clustering mechanism to segment original data into positive event streams and negative ones.

1) Automatic Segmentation of Event Streams: After data collection, numerous positive event streams and negative ones are mixed together, we need to segment the original data to get single event stream. Then each event stream could describe one complete scoring process. Event clustering method is employed to do data segmentation. Through tagged labels for all events, we cluster the adjacent events with the same label as one event stream. In addition, the events with far time intervals are considered as different event streams. The reference time interval is set as 100ms. In this way, the continuous events are segmented as different kinds of event streams. To avoid noises during segmentation, we abandon the meaningless event streams by limiting the shortest and longest numbers of them.

2) Data Preprocessing: For the problem about basketball scoring trace recognition, there are some key issues. In the real-world scenes, the process of data collection exists some noises including objects occlusion by basketball players or audiences. To avoid these noises, we employ essential regions extraction based on events clustering. The clustering centre

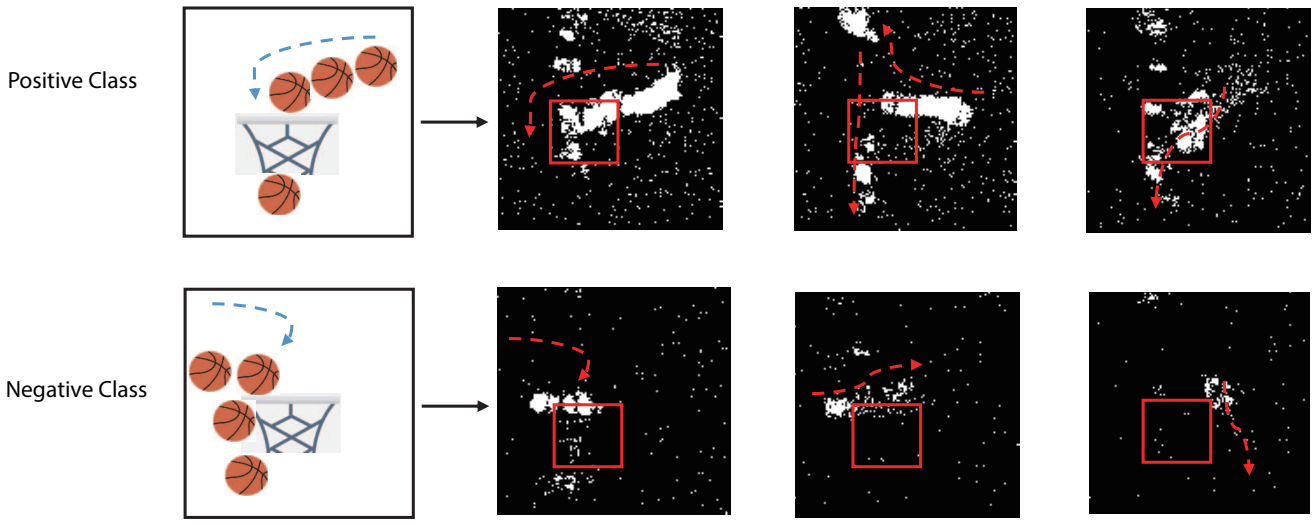

Fig. 4. The visualization of positive samples (scoring or scored) and negative samples (Failed to score). The above row shows the diagrammatic drawing of scoring process with three AER data examples. The red lines draw the ball moving trajectories when scoring to the baskets. The baskets are marked by a square with red color. The under row illustrates the situations that fails to score. There exist many different conditions for the negative class.

is used as the centre of the whole essential region. In this way, most of these occlusion noises are excluded efficiently. Besides, due to the human factors in the process of data segmentation and labelling, the discontinuity of the event stream would cause some events to flow very long and some events to flow very short, which is extremely unfavourable for the training of the model. To solve this problem, we defined the reference event length to throw away the unqualified samples.

After data preprocessing, we collected 6267 samples in total. There are 512 positive samples, which indicate the basketball scored processes. As illustrated in Figure. 4, it shows the reconstructed AER positive samples and negative samples with real-word simulation process. This reconstruction is the events accumulation through all the time within one event streams. Positive class denotes the condition that ball successes to score. Different positive samples has different trajectories but they all has the same process that basketball goes through the basketry. On the contrast, the negative samples are the situations failing to score. These samples have various trajectories when moving nearby the basketry. Therefore, each event stream describes the corresponding process in the real world. The dataset can be downloaded from https://www.dropbox.com/s/xeufx1b5io864v3/basketball_data_streams.mat?dl=0.

B. Experiment Settings

There exists typical imbalance between positive samples and negative ones for this basketball scoring dataset. The positive samples which play a more important role than negative ones only account for 8.17% percent. In terms of this issue, the imbalanced Tempotron is utilized to solve the basketball scoring recognition problem. The training set and test set are split to make them contain 80% and 20% samples for each category.

For the network used as classifier, there are $128 * 128$ input neurons and 64 output neurons in the input layer and output layer respectively. The learning rate λ_w is set to be 0.01. For each neuron model, the rest potential V_{rest} and firing threshold are 0 and 1.0 respectively. The time constants satisfy $\tau_m = 4 * \tau_s = 15ms$.

C. The Influence of Parameters

To investigate the essential factor on the learning efficiency of our method, the simulations are conducted under different parameter settings. Four groups of factors are employed to show the influences of different proportions between α and β. The classification accuracy and true positive rate are recorded to observe the recognition performance.

The classification performance under different parameters α and β is illustrated in Table.I. 'Train TP' and 'Test TP' denote the true positive rate on training set and test set respectively. Overall, the best classification accuracies are obtained when $\alpha = 2.0$ and $\beta = 1.0$. The test accuracies are 87.95%, 88.63%, 88.88%, 89.36%, and 88.67% for the primal Tempotron and imbalanced Tempotron method with different parameters, respectively. Firstly, since higher parameters help promote the learning ability for both positive samples and negative samples, when we set $\alpha = 2.0$ and $\beta = 2.0$, the performance is promoted compared with primal Tempotron algorithm. Besides, we find that when α is set to 2.0 and β is tuned from 2.0 to 1.0, the performance increases as smaller β emphasizes the effect of positive samples and weakens the influence of negative ones, which makes neurons are more focused on the informative samples. Therefore, the network becomes more selective to positive samples. However, when we continue decreasing the value of β to 0.5, the performance of the imbalanced Tempotron model is depressed. This is

TABLE I
COMPARISON OF PARAMETERS α AND β IN IMBALANCED TEMPOTRON LEARNING.

Algorithm	α	β	**Train TP**	**Train Accuracy**	**Test TP**	**Test Accuracy**
Tempotron	1.0	1.0	0.9636	0.995	0.7955	0.8795
Imbalanced Tempotron	2.0	2.0	0.9755	0.9972	0.8404	0.8867
Imbalanced Tempotron	2.0	1.0	0.9726	0.9979	**0.8411**	**0.8936**
Imbalanced Tempotron	2.0	0.8	0.9710	0.9980	0.8406	0.8888
Imbalanced Tempotron	2.0	0.5	0.9679	0.9980	0.8266	0.8863

TABLE II
COMPARISON OF PERFORMANCE BETWEEN IMBALANCED TEMPOTRON LEARNING AND OTHER METHODS.

Algorithm	**Train Accuracy**	**Test Accuracy**
Unsupervised SNN	0.88	0.62
SVM	0.79	0.72
Tempotron	0.97 ± 2.2	0.88 ± 1.2
Imbalanced Tempotron	0.98 ± 1.8	0.91 ± 1.7

because, the more useful information on essential samples can be lost, thus lower performance is obtained.

D. Compared with Other Methods on AER dataset

In this section, experiments are conducted to compare our approach with existing methods. Firstly, we compare our model with unsupervised SNN learning method proposed in [22], to evaluate the effectiveness of the imbalanced Tempotron algorithm for AER data recognition. This unsupervised STDP model consists of input layer and inhibition layer with lateral inhibition mechanism. Since the input size of input AER data is too big for this model, we convert the address of each event into spike trains and ignore the time of each event. That is, we regard the reconstructed static image as input instead of dynamic event streams. Then we assess the learning ability of our method in comparison with Support Vector Machine (SVM) [23] and original Tempotron. The training accuracy and test accuracy are recorded to show the comparison among those methods.

As shown in Table. II, our imbalanced Tempotron method achieves highest performance about 91% in test dataset. The training accuracies of both the original Tempotron and imbalanced Tempotron are around 98%, but the test accuracies decrease to 88% and 91%. It is caused by the quite few positive samples used for test are hard to recognized after training process. There are different kinds of event streams for positive sample in training set and test set. The imbalanced Tempotron achieves better performance because the learning ability of network for positive samples is improved during training process. Hence, the network becomes more balanced than the original Tempotron. In addition, the unsupervised STDP achieves 62% accuracy on test set. The reason is that the encoding process ignores the event time, which loses a lot of pivotal information. Moreover, all these SNN-based methods perform better than SVM method in training dataset, which indicates the AER data is more suitable for SNN-based model for its dynamic event-driven property.

IV. CONCLUSION

In this paper, we collect the basketball scoring dataset by DVS camera. The AER dataset consumes lower memory redundancy for it only records the dynamic varied events. After segmenting this dataset into event streams with positive samples and negative samples, we explore the classification performance of this problem. For this typically imbalanced dataset, we propose the improved Tempotron algorithm with balanced factors. The effect of different ratios between factors is explored. The big ratio improves the learning efficiency for positive samples and further increases the total classification accuracy. With this imbalanced Tempotron method, the basketball scoring recognition can achieve higher performance than other methods. In future work, the proposed framework could be applied to more time series datasets such as EEG, human pose trajectory, etc. Furthermore, more pruning methods would be combined with the proposed network to significantly reduce the network redundancy. Those applications and models could be further extended into neuromorphic chips for lower cost consumption.

ACKNOWLEDGMENT

We thank Qi Xu for his invaluable advice and feedback on this paper. This work was partly supported by the grants from National Key $R\&D$ Program of China (2018YFA0701400), National Natural Science Foundation of China (No. 61673340), Zhejiang Provincial Natural Science Foundation of China (LZ17F030001), Zhejiang Lab (2019KE0AD01), and the Royal Society Newton Advanced Fellowship (No. NAF-R1-191082).

REFERENCES

[1] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 7132–7141.

[2] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, "HDLTex: Hierarchical Deep Learning for Text Classification," *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 364–371, Dec. 2017, arXiv: 1709.08267.

[3] T. Cazenave, "Residual Networks for Computer Go," *IEEE Transactions on Games*, vol. 10, no. 1, pp. 107–110, 2018.

[4] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, Jan. 2011.

[5] T. Delbruck, "Fun with Asynchronous Vision Sensors and Processing," in *Proceedings of the 12th international conference on Computer Vision - Volume Part I*, ser. ECCV'12. Florence, Italy: Springer-Verlag, Oct. 2012, pp. 506–515.

[6] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240x180 130dB 3μs Latency Global Shutter Spatiotemporal Vision Sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[7] Y. Hu, H. Liu, M. Pfeiffer, and T. Delbruck, "DVS Benchmark Datasets for Object Tracking, Action Recognition, and Object Recognition," *Frontiers in Neuroscience*, vol. 10, 2016.

[8] Z. Yu, J. K. Liu, S. Jia, Y. Zhang, Y. Zheng, Y. Tian, and T. Huang, "Towards the Next Generation of Retinal Neuroprosthesis: Visual Computation with Spikes," *arXiv:2001.04064 [q-bio]*, Jan. 2020, arXiv: 2001.04064.

[9] S. Ghosh-Dastidar and H. Adeli, "Spiking Neural Networks," *International Journal of Neural Systems*, vol. 19, no. 04, pp. 295–308, Aug. 2009.

[10] J. Hu, H. Tang, K. Tan, and H. Li, "How the Brain Formulates Memory: A Spatio-Temporal Model Research Frontier," *IEEE Computational Intelligence Magazine*, vol. 11, no. 2, pp. 56–68, May 2016.

[11] Q. Xu, Y. Qi, H. Yu, J. Shen, H. Tang, and G. Pan, "CSNN: An Augmented Spiking based Framework with Perceptron-Inception," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. Stockholm, Sweden: International Joint Conferences on Artificial Intelligence Organization, Jul. 2018, pp. 1646–1652.

[12] T. Serre, A. Oliva, and T. Poggio, "A Feedforward Architecture Accounts for Rapid Categorization," *Proceedings of the National Academy of Sciences*, vol. 104, no. 15, pp. 6424–6429, Apr. 2007.

[13] S. Chen, P. Akselrod, B. Zhao, J. A. Perez Carrasco, B. Linares-Barranco, and E. Culurciello, "Efficient Feedforward Categorization of Objects and Human Postures with Address-Event Image Sensors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 302–314, Feb. 2012.

[14] B. Zhao, R. Ding, S. Chen, B. Linares-Barranco, and H. Tang, "Feedforward Categorization on AER Motion Events Using Cortex-Like Features in a Spiking Neural Network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 1963–1978, Sep. 2015.

[15] X. Peng, B. Zhao, R. Yan, H. Tang, and Z. Yi, "Bag of Events: An Efficient Probability-Based Feature Extraction Method for AER Image Sensors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 4, pp. 791–803, Apr. 2017.

[16] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special Issue on Learning from Imbalanced Data Sets," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, Jun. 2004.

[17] N. V. Chawla, "Data Mining for Imbalanced Datasets: An Overview," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA: Springer US, 2010, pp. 875–886.

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *arXiv:1708.02002 [cs]*, Feb. 2018, arXiv: 1708.02002.

[19] R. Gütig and H. Sompolinsky, "The Tempotron: A Neuron that Learns Spike Timing–Based Decisions," *Nature Neuroscience*, vol. 9, no. 3, pp. 420–428, Mar. 2006.

[20] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust Object Recognition with Cortex-Like Mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411–426, Mar. 2007.

[21] T. Serre, "Learning a Dictionary of Shape-Components in Visual Cortex: Comparison with Neurons, Humans and Machines," p. 211, 2006.

[22] P. U. Diehl and M. Cook, "Unsupervised Learning of Digit Recognition using Spike-Timing-Dependent Plasticity," *Frontiers in Computational Neuroscience*, vol. 9, p. 99, 2015.

[23] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support Vector Machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, Jul. 1998.