OPTIMISING EVENT-DRIVEN SPIKING NEURAL NETWORK WITH REGULARISATION AND CUTOFF

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking neural networks (SNNs), next generation of artificial neural networks (ANNs) with the benefit of energy efficiency, have achieved the accuracy close to its ANN counterparts, on benchmark datasets such as CIFAR10/100 and ImageNet. However, comparing with frame-based input (e.g., images), event-based inputs from e.g., Dynamic Vision Sensor (DVS) can make a better use of SNNs thanks to the SNNs' asynchronous working mechanism. In this paper, we strengthen the marriage between SNNs and event-based inputs with a proposal to consider anytime optimal inference SNNs, or AOI-SNNs, which can terminate anytime during the inference to achieve optimal inference result. Two novel optimisation techniques are presented to achieve AOI-SNNs: a regularisation and a cutoff. The regularisation enables the training and construction of SNNs with optimised performance, and the cutoff technique optimises the inference of SNNs on event-driven inputs. We conduct an extensive set of experiments on multiple benchmark event-based datasets, including CIFAR10-DVS, N-Caltech101 and DVS128 Gesture. The experimental results demonstrate that our techniques are superior to the state-of-the-art with respect to the accuracy and latency.

1 INTRODUCTION

SNNs have recently attracted significant research and industrial interests thanks to its energy efficiency and low latency Pfeiffer & Pfeil (2018), and there are neuromorphic chips such as Loihi Davies et al. (2018) and TrueNorth Akopyan et al. (2015) on which SNNs can be deployed. Mechanistically, SNNs mimic biological neurons, and the neurons process and forward spikes independently. With such an asynchronous working mechanism, only a (small) subset of neurons will be activated during inference. That is, energy efficiency is inherent to SNNs.

The asynchronous mechanism also suggests that event-based input may make a better use of SNNs. Actually, neuromorphic sensors such as Dynamic Vision Sensor Lichtsteiner et al. (2008); Delbrück et al. (2010); Gallego et al. (2020) and Dynamic Audio Sensor (DAS) Anumula et al. (2018) have been developed to generate binary "events", which are ideal inputs to SNN. For example, unlike conventional frame-based cameras which measure the "absolute" brightness at a constant rate, DVS cameras are bio-inspired sensors that *asynchronously* measure per-pixel brightness changes (called "events"), and output a stream of events that encode the time, location and sign of the brightness changes Gallego et al. (2022). DVS reveals the sparsity and asychronicity in recognition systems for computational efficiency Amir et al. (2017); Messikommer et al. (2020); Kim et al. (2021). To deal with event-based input, we propose to consider anytime optimal inference SNNs, or AOI-SNNs, which allow the termination at any time during the inference on a spike train (i.e., an input) and return the best possible inference result. Such SNNs enable the cutoff during the inference without (significantly) compromising the performance, and thus can achieve the best in terms of accuracy and latency.

Regarding the training of SNNs, a mainstream approach is through ANN-to-SNN conversion, which adopts the mature training regime of ANNs to first train a high-accuracy ANN, and then convert it into SNN. Such conversions via ANNs have resulted in research to focus on achieving the near-zero conversion loss. However, existing conversion methods Deng & Gu (2021); Bu et al. (2022); Han et al. (2020) mostly conduct empirical experiments on frame-based benchmark datasets such as ImageNet Deng et al. (2009) and CIFAR10/100 Krizhevsky & Hinton (2009). In this paper, we

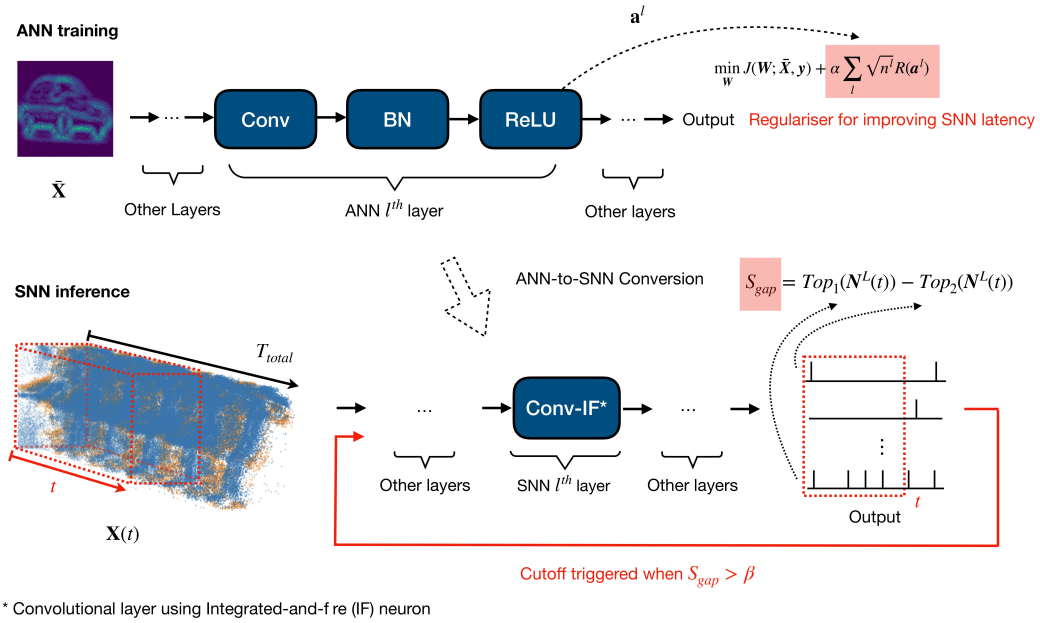


Figure 1: An illustrative diagram showing the regularisation for improving SNN latency and the cutoff mechanism for reducing latency on Cifar10-DVS dataset. Cutoff is triggered when S_{gap} is greater than β, a value dynamically determined by a confidence rate as introduced in Section 4.3.

will focus on event-based input, and therefore the AOI-SNNs, and explore effective training and inference methods to improve accuracy and latency together.

When considering ANN-to-SNN conversions to deal with DVS inputs, there are two possible ways. The first one aggregates the sparse events in the DVS stream into a frame-based input, on which the SNN processes as a whole. This resembles the ANN processing a static input (such as an image). As explained in Section 3.1, the frame-based input will base on the average spike rate, neglecting the spike timing information. The second is to directly work with the event-based input, by considering e.g., AOI-SNNs. An obvious benefit is that SNNs can exploit sparse events in the DVS input, enabling energy-efficient operation and reduced latency. In addition, unlike frame-based input, the event-based input does not need an encoder at or before the first layer, which allows SNNs to operate asynchronously and achieve extra low-latency (further explained in Section 3.1).

This paper makes two key technical contributions. Firstly, we propose a regularisation technique to influence the activation distribution during ANN training, which results in an SNN that can classify with less input information. As will be discussed in Related Work (Section 2), with our proposed regulariser, we can train an ANN without clipping and do not need to apply any quantisation-aware technique. Experiments in Section 5.1 show that we can achieve better accuracy than the state-of-the-art methods on both direct training and ANN-to-SNN conversion. Clipping (and quantisation-aware) techniques have been the status quo in this area due to the recent progress Li et al. (2021); Bu et al. (2022); Deng & Gu (2021); Wu et al. (2022) and our result suggests that there is an alternative, and probably better, way to get an improved SNN. Instead of simulating non-differentiable SNN activations during ANN training, our regulariser enables the attainment of a better distribution of SNN current by actively regularising the activations of the possible misclassifications. The regulariser is based on a new theoretical result (Section 4.1) that a smaller ratio of threshold voltages to average accumulated current can result in an SNN that can achieve optimised performance at any time during the inference.

The second contribution is that, instead of setting the inference length to always be T_{total}, we can explore an early cutoff mechanism that enables the SNN model to automatically achieve optimal latency and energy efficiency. As shown in Figure 1, the SNN model will run a monitoring mechanism to determine when it is sufficiently confident to make a decision. Once such a decision is made

at time $t < T_{total}$, a cutoff action is triggered so that the SNN will not take future inputs until the time T_{total}. Therefore, not only will this lead to lower latency (because decision is made at time t rather than T_{total}), but it will be also more energy-efficient (because no spike will be generated after time t).

2 RELATED WORK

Table 1: Technical ingredients of different conversion methods. OE and QA denote Outlier Elimination and Quantisation-aware technique respectively. COE: Clipping for Outlier Elimination; ROE: Regularisation for Outlier Elimination.

	Training		Inference		
	OE through	Apply QA	Soft-reset	Additive Noise	Cutoff
Rueckauer et al. (2017)	-	-	$\surd$	-	-
Deng & Gu (2021)	clipping (COE)	-	$\surd$	-	-
Wu et al. (2022)	clipping (COE)	-	$\surd$	$\surd$	-
Li et al. (2021)	clipping (COE)	$\surd$	$\surd$	-	-
Bu et al. (2022)	clipping (COE)	$\surd$	$\surd$	$\surd$	-
Ours	regularisation (ROE)	-	$\surd$	$\surd$	$\surd$

The application of SNNs to a data source can be separated into two phases: training and inference. Broadly speaking, the training algorithms for SNNs can be categorised into direct training (DT) and ANN-to-SNN conversion. Recently, Spike-based Error Backpropagation Wu et al. (2018; 2019); Fang et al. (2021b); Deng et al. (2022); Yao et al. (2021) direct train a neural network to process the temporal information of input spikes. However, either direct training or conversion algorithm Kugele et al. (2020); Wu et al. (2022) needs to collapse the input spikes into frames for the training. More specifically, the first layer in former SNN needs to wait for the full spike train within one frame to generate one spike, while the latter can respond very fast as long as the SNN receives spikes. Normally, the number of frames in direct training is kept small to reduce training complexity and determines the latency of SNN in inference. In contrast, ANN-to-SNN conversion can incorporate the maximum number of spikes during training to consider the SNN with optimal latency.

For the ANN-to-SNN conversion, early studies Rueckauer et al. (2017); Diehl et al. (2015) use the maximum value of activation to normalise the weights from ANN, and Sengupta et al. (2019) proves that the normalisation can also be achieved by greedily searching for the optimal threshold using the input spike train. A unified conversion framework is studied in Wu et al. (2022). Besides, there are hybrid methods Rathi et al. (2020); Rathi & Roy (2021) that combine conversion and direct training. Tandem Learning Wu et al. (2021) leverages the gradient from ANN to update SNN during training. The first two columns of Table 1 present the technical ingredients of different conversion methods for the training phase. Recent work Deng & Gu (2021); Wu et al. (2022) shows that, outlier elimination (OE) in ANN activations can be implemented by applying *clipping* operation after the Rectified Linear Unit (ReLU). Based on this, Li et al. (2021); Bu et al. (2022) further minimise the quantisation error by Quantisation-aware (QA) training. Different from the above methods, we develop a new regulariser to achieve the better performance *without* clipping, and moreover, noticeably, we are *free from* applying QA training.

For the inference phase, as indicated in the last two columns of Table 1, the soft-reset mechanism Rueckauer et al. (2017) and the additive white noise to membrane potential Deng & Gu (2021); Wu et al. (2022); Bu et al. (2022) can significantly increase the conversion efficiency. To the best of our knowledge, there is no existing work on cutoff in the inference phase, and our confidence-based method is the first of its kind.

3 PRELIMINARY

In this section, we discuss the event-based input in spiking neuron and introduce the ANN-to-SNN conversion. To facilitate the analysis, we use **bold symbol** to represent vector, l to denote the layer index, and i to denote the index of elements. For example, $\boldsymbol{a}^l$ is a vector and a_i^l is the i-th element

in $\boldsymbol{a}^l$. Inference time t represents the time length of input. T_{total} denotes the maximum time length of input and it can be various depending on dataset. $\boldsymbol{W}^l$ is weight matrix at the l-th layer.

3.1 INTEGRATED-AND-FIRE MODEL

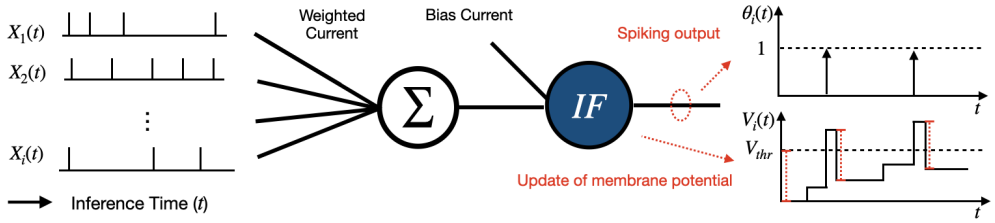


Figure 2: Inference in integrate-and-fire (IF) neuron with *reset by subtraction* mechanism.

Conversion-based SNN uses integrate-and-fire (IF) neuron as the basic computing unit to approximate ReLU in ANN Wu et al. (2022). Figure 2 illustrates the inference process in IF neurons. The input spike train $X_i(t)$ charges the membrane potential $V_i(t)$ with weighted current. The weighted current and bias current are translated from the weight $\boldsymbol{W}^l$ and bias $\boldsymbol{b}^l$ in ANN. When $V_i(t)$ reaches the threshold V_{thr}, the neuron will generate a spike and then reset the $V_i(t)$ by subtracting V_{thr}. The *reset by subtraction* mechanism was firstly suggested in Rueckauer et al. (2017) to reduce information loss during inference. The dynamics of IF neuron can be described as

$$\boldsymbol{V}^l(t) = \begin{cases} \boldsymbol{V}^l(t-1) + \boldsymbol{Z}^l(t) - \boldsymbol{\theta}^l(t)V^l_{thr} & l > 1 \\ \boldsymbol{Z}^1(t) & l = 1 \end{cases} \tag{1}$$

where $\boldsymbol{\theta}^l(t)$ is a step function i.e., $\theta^l_i(t) = 1$ if $V^l_i(t-1) + Z^l_i(t) \geq V^l_{thr}$ and $\theta^l_i(t) = 0$ otherwise. $\boldsymbol{Z^l}(t)$ is the input current such that

$$\boldsymbol{Z}^l(t) = \boldsymbol{W}^l\boldsymbol{\theta}^{l-1}(t) + \boldsymbol{b}^l \qquad \text{when } l > 1. \tag{2}$$

For the event-based inputs (e.g., from a DVS sensor) , $\boldsymbol{Z}^l(t)$ at the first layer, i.e., $\boldsymbol{Z}^1(t)$, can be initialised as

$$\boldsymbol{Z}^1(t) = \boldsymbol{W}^1\boldsymbol{X}(t) + \boldsymbol{b}^1 \tag{3}$$

where $\boldsymbol{X}(t)$ is the time-dependent spike train, i.e., the input may change the charging current with time during the inference. To consider the temporal information, we split the spike train into F frames and duration of each frame is equal to $T = T_{total}/F$, where $F \in \mathbb{Z}^+$. We write $\bar{\boldsymbol{X}}_{\boldsymbol{f}}$ to represent the average spiking rate of f-th frame, i.e., $\bar{\boldsymbol{X}}_f = 1/N_{\max}\sum_{t=T\cdot(f-1)}^{T\cdot f}\boldsymbol{X}(t)$ such that $N_{\max}$ is the maximum spikes of all training frames in dataset D. The spiking resolution of $X(t)$ can be roughly computed as $S_r = N_{\max}/T$. For event-based input, SNN can manifest faster inference due to immediate response after receiving the first spike, and it completes the inference whenever the spike train ends, i.e., at T_{total}. The event-based benchmarks are further introduced in AppendixA.4. This characteristic makes it possible that the inference time is dynamic for different inputs. In this paper, with the cutoff technique as in Section 4.3, we will show that the average latency of the inference in SNN can be further reduced (to some $t \leq T_{total}$).

3.2 TEMPORAL TRAINING

Regarding the direct training of SNN in Fang et al. (2021b); Deng et al. (2022), the resulted SNN can make decision by averaging output spikes of consecutive frames. Assuming that the inference of each frames is independent, such process can be approximated in ANN training by letting the loss function be

$$L_{TT} = \frac{1}{F}\sum_{f=1}^{F} L_{CE}(\boldsymbol{Y}_f, \hat{\boldsymbol{Y}}) \tag{4}$$

where $\boldsymbol{Y}_f$ is output of $\bar{\boldsymbol{X}}_f$ after softmax, $\hat{\boldsymbol{Y}}$ is the ground truth and L_{CE} is cross-entropy loss. Temporal training loss (L_{TT}) was suggested in Deng et al. (2022) that achieves better generalisation. To simplify the theoretical analysis, we let $F = 1$ in section 3.3 & 4. The further explanation of temporal training is given in appendix A.5, including the impact of F on ANN training and extension of theories to $F > 1$. To ensure the independence between frames, the membrane potential of hidden layer is reset after each frame, while that of output layer is reset after the last frame, which is feasible in hardware implementation Frenkel & Indiveri (2022); Khodamoradi et al. (2021).

3.3 ANN-TO-SNN CONVERSION

The conversion method is mainly based on integrated-and-fire (IF) neuron, which generates spikes depending on positive accumulated current, corresponding to ReLU activation in ANN. An existing conversion method Wu et al. (2022) uses current normalisation methods by letting

$$\frac{1}{T \cdot S_r} \sum_{t=0}^{T} \boldsymbol{Z}^1(t) = \boldsymbol{a}^1 \tag{5}$$

where $\boldsymbol{a}^1$ is the output of ReLU activation at the first layer of ANN. The spiking rate of each SNN neuron at layer l is defined as $\boldsymbol{r}^l(t) = \boldsymbol{N}^l(t)/t$, where $\boldsymbol{N}^l(t)$ is the number of spikes received up to time t by neuron at layer l. The relationship between spiking rate in SNN and activation in ANN has been theoretically proved in Wu et al. (2022), which gives

$$\boldsymbol{r}^l(t) = \tfrac{1}{V_{thr}^l}\left(\boldsymbol{W}^l \boldsymbol{r}^{l-1}(t) + \boldsymbol{b}^l\right) - \boldsymbol{\Delta}^l(t) \tag{6}$$

where $\boldsymbol{\Delta}^l(t) \triangleq \boldsymbol{V}^l(t)/(tS_r V_{thr}^l)$ represents the residual spiking rate. The spiking rate at the first layer can be initialised as $\boldsymbol{r}^1(t) = \boldsymbol{a}^1/V_{thr}^1 - \boldsymbol{\Delta}^1(t)$. Note that, we use tS_r to represent the timestep in Wu et al. (2022). Then, the current normalisation can be achieved by

$$\tilde{\boldsymbol{W}}^l \leftarrow \boldsymbol{W}^l,\ \tilde{\boldsymbol{b}}^l \leftarrow \frac{1}{\lambda^{l-1}}\boldsymbol{b}^l,\ V_{thr}^l \leftarrow \frac{\lambda^l}{\lambda^{l-1}} \tag{7}$$

where λ^l be the maximum value of the activation at layer l. For temporal training, the temporal input frames share the same λ^l.

4 METHODS

We introduce two novel techniques: one is for the training and the other for the inference. Section 4.1 presents the theoretical underpinning of the regulariser, which in turn is introduced in Section 4.2. This is followed by the introduction of cutoff mechanism in Section 4.3 for the inference.

4.1 ANYTIME OPTIMAL INFERENCE SNNS

The regulariser is based on an investigation into the design of AOI-SNNs. An AOI-SNN is able to perform optimally under different settings on the inference time for processing an input. When t is large enough to make $\Delta^l(t)$ negligible, we define the desired spiking rate as follows:

$$\boldsymbol{r}_d^l = \boldsymbol{a}^l/V_{thr}^1 \tag{8}$$

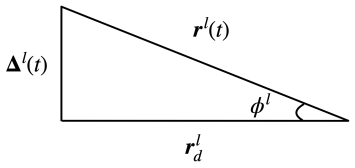


Figure 3: Graphic illustration of the desired spiking rate $\boldsymbol{r}_d^l$ and spiking rate $\boldsymbol{r}^l(t)$

We start from establishing a theoretical underpinning between spiking rate $\boldsymbol{r}^l(t)$ and its desired value $\boldsymbol{r}_d^l$. Let ϕ^l denote the angle between $\boldsymbol{r}_d^l$ and $\boldsymbol{r}^l(t)$. Then, we follow Banner et al. (2018) to use cosine similarity between $\boldsymbol{r}_d^l$ and $\boldsymbol{r}^l(t)$, i.e., $\cos(\phi^l)$, for the measurement of the performance of SNN by t. Actually, Banner et al. (2018) shows that the cosine similarity between full precision and quantised neural network has a high correlation with the final accuracy of the quantised neural network. Similarly, we expect that higher cosine similarity between $r^l(t)$ and r_d^l can result in less accuracy drop by t.

The following theorem states that, for any t, the performance of the SNN is of negative correlation with threshold V_{thr}^l, and positive correlation with L_2 norm over $\boldsymbol{a}^l$, as stated in the below theorem.

Theorem 4.1 *For any inference time, assuming that the residual spiking rate $\boldsymbol{\Delta}^l(t)$ is independent from $\boldsymbol{r}_d^l$, the cosine similarity between $\boldsymbol{r}_d^l$ and $\boldsymbol{r}^l(t)$ is inversely proportional to the ratio of threshold to average accumulated current,*

$$\cos(\phi^l) \propto \left(\sqrt{n^l}\frac{V_{thr}^l}{\|\boldsymbol{a}^l\|_2}\right)^{-1}$$

where n^l is the dimension of $\boldsymbol{a}^l$ and $\|\boldsymbol{a}^l\|_2/\sqrt{n^l}$ denotes the average accumulated current.

We give a proof sketch of the theorem. Because $\Delta^l(t)$ is independent from $\boldsymbol{r}_d^l$, the angle between these two vectors tends to be $\pi/2$ at high dimension. Then, by $\boldsymbol{r}^l(t) = \boldsymbol{r}_d^l - \boldsymbol{\Delta}^l(t)$, we get a right angle triangle with $\boldsymbol{r}_d^l$ and $\Delta^l(t)$ as the legs, and $\boldsymbol{r}^l(t)$ as the hypotenuse, as illustrated in Figure 3. Moreover, we have

$$\cos(\phi^l) = \frac{\|\boldsymbol{r}_d^l\|_2}{\|\boldsymbol{r}^l(t)\|_2} \geq \frac{\|\boldsymbol{r}_d^l\|_2}{\|\boldsymbol{r}_d^l\|_2 + \|\boldsymbol{\Delta}^l(t)\|_2} \tag{9}$$

We are interested in increasing the lower bound of Equation 9, so that we have greater $\cos(\phi^l)$ for different t. Combining with Equations (6) and (8), we have

$$\cos(\phi^l) \geq \frac{\|\boldsymbol{a}^l/V_{thr}^l\|_2}{\|\boldsymbol{a}^l/V_{thr}^l\|_2 + \|\boldsymbol{V}^l(t)/(tS_rV_{thr}^l)\|_2} = \frac{\|\boldsymbol{a}^l\|_2}{\|\boldsymbol{a}^l\|_2 + \|\boldsymbol{V}^l(t)/(tS_r)\|_2} \tag{10}$$

Assuming that elements in $\boldsymbol{V}^l(t)$ satisfy uniform distribution over the time t and they are in $[0, V_{thr}]$, we can derive $\mathbb{E}(\|V(t)/(tS_r)\|_2) \leq \sqrt{n^l}V_{thr}/(\sqrt{3}tS_r)$ (proof in Appendix A.2). Moreover, at high dimensions, the relative error made as considering $\mathbb{E}(\|V(t)/(tS_r)\|_2)$ instead of the random variable $\|V(t)/(tS_r)\|_2$ becomes asymptotically negligible Biau & Mason (2015); Banner et al. (2018). Therefore, Equation 10 can be computed with the following lower bound

$$\cos(\phi^l) \geq \frac{\|\boldsymbol{a}^l\|_2}{\|\boldsymbol{a}^l\|_2 + \sqrt{n^l}V_{thr}^l/(\sqrt{3}tS_r)} = \frac{\sqrt{3}tS_r}{\sqrt{3}tS_r + \sqrt{n^l}V_{thr}^l/\|\boldsymbol{a}^l\|_2} \tag{11}$$

which explicitly explains that (1) the increase of t to $t \gg \sqrt{n^l}V_{thr}^l/\|\boldsymbol{a}^l\|_2$ can increase the lower bound and (2) it is possible to minimise term $\sqrt{n^l}V_{thr}^l/\|\boldsymbol{a}^l\|_2$ for developing an SNN with optimised performance at any time during the inference. In other words, an AOI-SNN expects a good (small) ratio of threshold voltage V_{thr}^l to average accumulated current, i.e., $\|\boldsymbol{a}^l\|_2/\sqrt{n^l}$, while not degrading SNN classification performance. The point (2) corresponds with the theorem.

4.2 Regulariser for outlier elimination (ROE)

This section shows how to design a regulariser based on Theorem 4.1. Recall from Equation (7) that V_{thr}^l is determined by λ^l and λ^{l-1}, where λ^l is the maximum value of activation in the l-th layer. To simplify the complexity of optimisation, the impact of $1/\lambda^{l-1}$ is omitted and the ratio of threshold to expected current approximately becomes proportional to $\lambda^l/\|\boldsymbol{a}^l\|_2$. Therefore, we design a regulariser to minimise term $\lambda^l/\|\boldsymbol{a}^l\|_2$ to develop an AOI-SNN. Firstly, we use matrix $\boldsymbol{A}^l$ to represent a batch of $\boldsymbol{a}^l$ during training. Secondly, we simply use maximum value in $\boldsymbol{A}^l$ to approximate λ^l, i.e., $\lambda^l \approx \|\boldsymbol{A}^l\|_{\max}$. Then, we write $\|\boldsymbol{A}^l\|_{2,q} = (\sum_j(\sum_i A_{ij}^2)^{q/2})^{1/q}$ to denotes the $L_{2,p}$ over $\boldsymbol{A}^l$, where A_{ij}^l presents j-th $\boldsymbol{a}_{\boldsymbol{i}}^l$ in the batch and $q \in \mathbb{Z}$. Finally, we can let the penalty term be the ratio between $\|\boldsymbol{A}^l\|_{\max}$ and $\|\boldsymbol{A}^l\|_{2,q}$ with scale constant $\sqrt{n^l}$, i.e.,

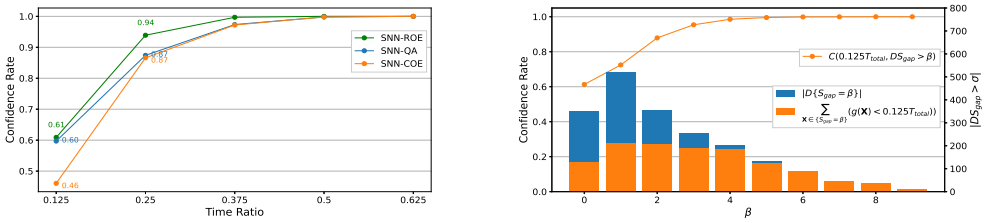
$$R(\boldsymbol{A}^l) = \sqrt{n^l}\frac{\|\boldsymbol{A}^l\|_{\max}}{\|(\boldsymbol{A}^l)\|_{2,q}} \tag{12}$$

We let q be $-\infty$ so that the penalty term can focus on the inputs with relatively small accumulated current in the batch. The final training objective is

$$L_{TT} + \alpha \sum_{l} \ln\left(R(\boldsymbol{A}^l)\right) \tag{13}$$

where α is a hyper-parameter to balance two loss terms. Logarithm is applied to reduce the impact from extremely large value. The regularisation-based training is to train an ANN based on $\bar{\boldsymbol{X}}_f$ resulting in an SNN, then SNN operates with the event-based input (Equation 3). A small $R(\boldsymbol{A})^l$ implies that it is less possible for λ^l to be an outlier and $\|\boldsymbol{a}^l\|_2$ is generally large.

4.3 CUTOFF MECHANISM TO REDUCE INFERENCE TIME



(a) Confidence w.r.t time ratio.

(b) Confidence w.r.t β at 0.125 T_{total}, for SNN-ROE

Figure 4: Evaluation of confidence on Cifar10-DVS

Thanks to the asynchronous working mechanism, event-driven SNNs can predict when only part of the spike train is processed. Nevertheless, a naive cutoff on the length of spike train (or the sampling time of event sensor) can easily result in accuracy loss. In this section, we suggest a principled method to determine the inference time. Technically, a new metric, called confidence rate and denoted as $C(\hat{t}, D\{S_{gap} > \beta\})$, is defined based on the statistical characteristics of processing a set D of inputs with respect to the discrete inference time $\hat{t}$ and S_{gap}. $S_{gap} > \beta$ operates as a condition to identify the samples in D that are suitable for cutoff. Actually, we are able to plot a curve of confidence rate $C(\hat{t}, D\{S_{gap} > \beta\})$ with respect to the time $\hat{t}$ and β, respectively. During the processing of an individual input $\boldsymbol{X}$, we will monitor another variable S_{gap}, and once it is able to ensure the confidence rate can reach certain degree, an early cutoff signal can be sent (see Figure 1). The following provides the details.

We write $\boldsymbol{X}[\hat{t}] = \sum_{t=0}^{\hat{t}} \boldsymbol{X}(t)$ to denote the accumulation of $\boldsymbol{X}(t)$ from 0 up to $\hat{t}$. Then, we let $f(\boldsymbol{X}[\hat{t}])$ return the prediction of f based on the partial input $\boldsymbol{X}[\hat{t}]$. Based on this, we define a function

$$g(\boldsymbol{X}) = \arg\min_{\hat{t}} \{\forall \hat{t}_1 > \hat{t} : \mathbf{1}(f(\boldsymbol{X}[\hat{t}_1]) = \boldsymbol{y})\} \tag{14}$$

to express the earliest time from which the model f is able to confidently and correctly classify according to the partial input. $\mathbf{1}(\cdot)$ is the indicator function, i.e., $\mathbf{1}(x_1 = x_2) = 1$ and $\mathbf{1}(x_1 \neq x_2) = 0$. $\mathbf{1}(f(\boldsymbol{X}[\hat{t}_1]) = \boldsymbol{y})$ suggests that $f(\boldsymbol{X}[\hat{t}])$ is the same as the ground truth $\boldsymbol{y}$. Then, recall that $\boldsymbol{N}^L(t)$ is the number of spikes received by t by the output layer L. We write $Top_k(\boldsymbol{N}^L(t))$ as the top k spikes that occur in some neuron of layer L. Then, we let

$$S_{gap} = Top_1(\boldsymbol{N}^L(t)) - Top_2(\boldsymbol{N}^L(t)) \tag{15}$$

be a variable denoting the gap of top-1 and top-2 number of spikes. A large S_{gap} implies little possibility of switching the prediction results during inference. Then, we let $D\{\cdot\}$ denote the inputs in subset of D that satisfy a certain condition. Now, we can define the confidence rate as follows:

$$\textit{Confidence rate: } C(\hat{t}, D\{S_{gap} > \beta\}) = \frac{1}{|D\{S_{gap} > \beta\}|} \sum_{\boldsymbol{X} \in D\{S_{gap} > \beta\}} (g(\boldsymbol{X}) \leq \hat{t}) \tag{16}$$

which intuitively computes the percentage of inputs in D that can achieve the prediction success on or before a prespecified time $\hat{t}$, i.e., $g(\boldsymbol{X}) \leq \hat{t}$. $|D\{S_{gap} > \beta\}|$ denotes the number of samples in D satisfying the condition. It is not hard to see that, when $\hat{t} = 0$, $C(\hat{t}, D\{S_{gap} > \beta)$ is also 0, and

with the increase of time $\hat{t}$, $C(\hat{t}, D\{S_{gap} > \beta)$ will also increase until reaching 1. Our algorithm searches for a minimum $\beta \in \mathbb{Z}^+$ at a specific $\hat{t}$, as expressed in the following optimisation objective:

$$\arg\min_{\beta} C(\hat{t}, D\{S_{gap} > \beta\}) \geq 1 - \epsilon \tag{17}$$

where ϵ is a pre-specified constant such that $1 - \epsilon$ represents an acceptable level of confidence for activating cut-off, and a set of β is extracted under different $\hat{t}$ using training samples.

Equation 16 is visualised in Figure 4 that shows the impact of inference time and β on confidence. Time ratio denotes the normalised inference time. We characterise the confidence metric with training samples and eventually use testing samples for evaluation. Note that, on Cifar10-DVS, all model achieves 100% for training accuracy, however, they perform differently on confidence. With regularisation, SNN-ROE can further improve the confidence than SNN-QA, e.g., it is 0.01 higher at $0.125T_{total}$ and 0.07 higher at $0.25T_{total}$. Therefore, SNN-ROE can have a better performance at any time during the inference, as there are more inputs join the early cutoff. Figure 4b presents that the input with large S_{gap} has more consistent prediction over time, which supports the use of $S_{gap} > \beta$ as the cutoff condition.

5 EXPERIMENT

We implement the ROE and conduct an extensive set of experiments to validate it. We consider its comparison with the state-of-the-art CNN-to-SNN conversion methods. In this section, 'SNN-QA' denotes the method in Bu et al. (2022), which includes both COE and QA during training, and outperforms the other methods on image input. In contrast, 'SNN-COE' denotes the SNN with only COE. Our proposed method is denoted by 'SNN-ROE'. To reduce the accuracy loss during inference, we followed Wu et al. (2022); Bu et al. (2022) to add extra current $V_{thr}^l/2$ to each neuron.

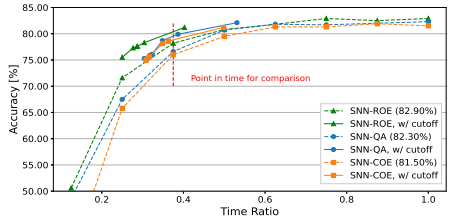
Our method is validated against three event-based datasets, e.g., Cifar10-DVS Li et al. (2017), N-Caltech101 Orchard et al. (2015) and DVS128 Gesture Amir et al. (2017). We train the neural network using Tensorflow with Keras API and convert it into SNN by SpKeras Wu et al. (2022). As Bu et al. (2022) did not cover the event-based input, we replicate their work as our baseline for comparison and set the quantisation length of SNN-QA to 16 for all datasets, which yields optimal performance. Note that, we use original input from DVS camera without any pre-processing for the inference so that SNN can remain asynchronous to the input events. The details of training setting are described in Appendix A.3.

5.1 EXPERIMENTAL RESULTS

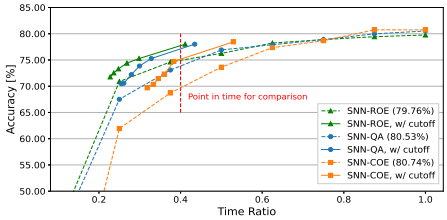
This section presents a comparison between SNN-ROE, SNN-QA and SNN-COE on accuracy w.r.t time ratio and performance improvement after cutoff. The inference time $t =$ Time Ratio $\times\, T_{total}$, where T_{total} is equal to $1.3s$, $0.3s$ and $1.2s$ for Cifar10-DVS, N-Caltech101 and DVS128 Gesture respectively.

It is not hard to see, with cutoff, the performance of all models is improved in Figure 5(a, b, c). For example, the accuracy curve moves above its original curve, which means that same accuracy can have less inference time for same model. Thanks to the increase of confidence (recall the results in Figure 4a), SNN-ROE can have general higher accuracy before the time point (red dash line) and it shows consistent results in different datasets. The confidence evaluation for N-Caltech101 and DVS128 Gesture is provided in Figure A-3. It has been argued in Yao et al. (2021) that the temporal information in Cifar10-DVS is not the dominant information, which is similar in N-Caltech101. Unlike N-Caltech101 and Cifar10-DVS, the correlation between the temporal events in DVS128 Gesture is high. This phenomenon can also be observed from the Figure A-1. Therefore, we set $F = 1$ on N-Caltech101 and Cifar10-DVS for efficient training and $F = 4$ on DVS128 Gesture to incorporate temporal information. Moreover, the result in Figure 5c shows that cutoff also can improve SNN trained with $F > 1$. Figure 5d presents that the time ratio becomes adaptive after applying cutoff and regulariser can generally increase the cutoff performance with more accurate predictions at early inference time.

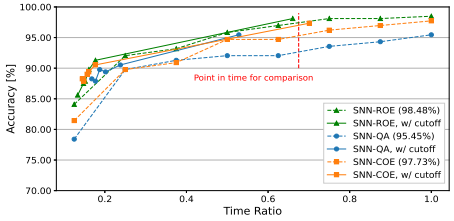
We collect the results around the time point in Table 2, which has similar inference time, to show the performance of each model on cutoff. SNN-ROE achieves the superior performance on accuracy and latency. Spiking resolution S_r is calculated to estimate maximum average spikes per second.



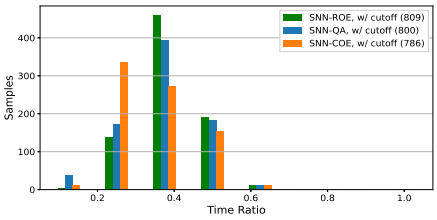
(a) Accuracy w.r.t time ratio on Cifar10-DVS



(b) Accuracy w.r.t time ratio on N-Caltech101



(c) Accuracy w.r.t time ratio on DVS128 Gesture



(d) Adaptive time ratio on Cifar10-DVS

Figure 5: Comparison of SNN using ROE, QA and COE on different datasets. (a,b,c) Accuracy for full-length input is shown in bracket and cutoff is based on $\epsilon = \{0.05, 0.04.0.03, 0.02, 0.01, 0.00\}$. The results around red dash line are summarised in Table 2. (d) The statistic data is extracted from testing samples with total number of right predictions in bracket.

Table 2: Comparison between the SNN-COE, SNN-QA and SNN-ROE with cutoff for similar inference t. The comparison with the state-of-the-art methods on event-based datasets. DT and CV represents direct training and ANN-to-SNN conversion.

Dataset	Models	Methods	Acc.	Ave. t
Cifar10-DVS (S_r=85.38, F=1)	Fang et al. (2021a)	DT	74.40%	1.3s
	Fang et al. (2021b)	DT	74.80%	1.3s
	Kugele et al. (2020)	CV	66.40%	-
	SNN-COE (w/ cutoff)	CV	78.60%	0.5s
	SNN-QA (w/ cutoff)	CV	79.90%	0.5s
	SNN-ROE (w/ cutoff)	CV	**81.20%**	**0.5**s
N-Caltech101 (S_r=5230, F=1)	She et al. (2022)	DT	71.20%	-
	Messikommer et al. (2020)	DT	74.50%	0.3s
	SNN-COE (w/ cutoff)	CV	74.72%	0.1s
	SNN-QA (w/ cutoff)	CV	78.00%	0.1s
	SNN-ROE (w/ cutoff)	CV	**78.00%**	**0.1**s
DVS128 Gesture (S_r=506.67, F=4)	Fang et al. (2021a)	DT	97.92%	6.0s
	Fang et al. (2021b)	DT	97.57%	6.0s
	Yao et al. (2021)	DT	97.57%	1.2s
	Kugele et al. (2020)	CV	95.56%	-
	SNN-COE (w/ cutoff)	CV	97.34%	0.8s
	SNN-QA	CV	93.56%	0.9s
	SNN-ROE (w/ cutoff)	CV	**98.10%**	**0.8**s

6 CONCLUSIONS

This paper promotes anytime optimal inference SNNs (AOI-SNNs), which maintain the optimal performance throughout the inference stage, and therefore are suitable for event-driven inputs such as those from dynamic vision sensor or dynamic audio sensor. Two technical novelties are proposed to optimise the attainment of AOI-SNNs, one for the training stage and the other for the inference stage. Our experiments demonstrate the superior performance with respect to the accuracy and latency, comparing to the state-of-the-art.

REFERENCES

F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, Nam Datta, P., G.J., and B. Taba. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *EEE transactions on computer-aided design of integrated circuits and systems*, 34(10):1537–1557, 2015.

Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7243–7252, 2017.

Jithendar Anumula, Daniel Neil, Tobi Delbruck, and Shih-Chii Liu. Feature representations for neuromorphic audio spike streams. *Frontiers in neuroscience*, 12:23, 2018.

Ron Banner, Itay Hubara, Elad Hoffer, and Daniel Soudry. Scalable methods for 8-bit training of neural networks. *Advances in neural information processing systems*, 31, 2018.

Gérard Biau and David M Mason. High-dimensional p p-norms. In *Mathematical statistics and limit theorems*, pp. 21–40. Springer, 2015.

Tong Bu, Wei Fang, Jianhao Ding, PENGLIN DAI, Zhaofei Yu, and Tiejun Huang. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7B3IJMM1k_M.

M. Davies, N. Srinivasa, T.H. Lin, G. Chinya, Y. Cao, S.H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, and Y. Liao. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.

Tobi Delbrück, Bernabe Linares-Barranco, Eugenio Culurciello, and Christoph Posch. Activity-driven, event-based vision sensors. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 2426–2429. IEEE, 2010.

J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 2(248-255): 1430, 6 2009.

Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=FZ1oTwcXchK.

Shikuang Deng, Yuhang Li, Shanghang Zhang, and Shi Gu. Temporal efficient training of spiking neural network via gradient re-weighting. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_XNtisL32jv.

P.U. Diehl, D. Neil, J. Binas, M. Cook, S.C. Liu, and M. Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. *International Joint Conference on Neural Networks*, pp. 1–8, 7 2015.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021a.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2661–2671, 2021b.

Charlotte Frenkel and Giacomo Indiveri. Reckon: A 28nm sub-mm2 task-agnostic spiking recurrent neural network processor enabling on-chip learning over second-long timescales. In *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 65, pp. 1–3. IEEE, 2022.

G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44(01):154–180, jan 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2020.3008413.

Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.

B. Han, G. Srinivasan, and K. Roy. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13558–13567, 2020.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR.

Alireza Khodamoradi, Kristof Denolf, and Ryan Kastner. S2n2: A fpga accelerator for streaming spiking neural networks. In *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 194–205, 2021.

Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim. N-imagenet: Towards robust, fine-grained object recognition with event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2146–2156, 2021.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *2009 IEEE conference on computer vision and pattern recognition*, 2009.

Alexander Kugele, Thomas Pfeil, Michael Pfeiffer, and Elisabetta Chicca. Efficient processing of spatio-temporal data streams with spiking neural networks. *Frontiers in Neuroscience*, 14:439, 2020.

Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11:309, 2017.

Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ann: Towards efficient, accurate spiking neural networks calibration. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6316–6325. PMLR, 18–24 Jul 2021.

Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128$\times$128 120 db 15μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *European Conference on Computer Vision*, pp. 415–431. Springer, 2020.

Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015.

M. Pfeiffer and T. Pfeil. Deep learning with spiking neurons: opportunities and challenges. *Frontiers in neuroscience*, 12:774, 2018.

Nitin Rathi and Kaushik Roy. Diet-snn: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Nitin Rathi, Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=B1xSperKvH`.

B. Rueckauer, I.A. Lungu, M. Hu, Y.and Pfeiffer, and S.C. Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11:682, 2017.

Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in neuroscience*, 13:95, 2019.

Xueyuan She, Saurabh Dash, and Saibal Mukhopadhyay. Sequence approximation using feedforward spiking neural network for spatiotemporal learning: Theory and optimization methods. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=bp-LJ4y_XC`.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

Dengyu Wu, Xinping Yi, and Xiaowei Huang. A little energy goes a long way: Build an energy-efficient, accurate spiking neural network from convolutional neural network. *Frontiers in neuroscience*, 16, 2022.

Jibin Wu, Chenglin Xu, Xiao Han, Daquan Zhou, Malu Zhang, Haizhou Li, and Kay Chen Tan. Progressive tandem learning for pattern recognition with deep spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12(331), 5 2018.

Y. Wu, L. Deng, G. Li, J. Zhu, Y. Xie, and L. Shi. Direct training for spiking neural networks: Faster, larger, better. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 33:1311–1318, 7 2019.

Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10221–10230, 2021.

A Appendix

A.1 Notation table

Symbol	Definition	Symbol	Definition
i	Element index	l	Layer index
T_{total}	Total inference time	F	Number of input frame
T	Duration of one frame	t	Inference time
N_{max}	Maximum spikes in training dataset	S_r	Spiking resolution
$\boldsymbol{V}^l(t)$	Membrane potential	$\boldsymbol{Z}^l(t)$	Weighted current
$\boldsymbol{\theta}^l(t)$	Step function	V_{thr}	Threshold voltage
$\boldsymbol{W}^l$	Weight	$\boldsymbol{b}^l$	Bias
$\boldsymbol{X}(t)$	Input spike train	$\hat{\boldsymbol{X}}_f$	Spiking rate of f-th frame
$\boldsymbol{Y_f}$	Output of $\hat{\boldsymbol{X}}_f$	$\boldsymbol{Y}$	Ground truth
$\boldsymbol{N}^l(t)$	Number of spikes received	$\boldsymbol{r}^l(t)$	Spiking rate
$\boldsymbol{\Delta}^l(t)$	Residual spiking rate	$\boldsymbol{\lambda}_d^l$	Maximum value of activation
$\boldsymbol{r}_d^l$	Desired spiking rate	ϕ^l	Angular between $\boldsymbol{r}_d^l$ and $\boldsymbol{r}^l(t)$
$\boldsymbol{a}^l$	Activation	n	Dimension of $\boldsymbol{a}^l$
$\boldsymbol{A}^l$	A batch of activation	L_{TT}	Temporal training loss
$\hat{t}$	Discrete inference time	$C(\hat{t}, \cdot)$	Confidence rate
$g(\boldsymbol{X})$	Gap of top-1 and top-2 number of spikes	β	Constant value for cutoff
D	Training dataset	$R(\cdot)$	Regulariser term

A.2 INEQUATION PROOF

We follow Banner et al. (2018) to derive the bound of expected norm of a random variable vector. By Jensen's inequality, it gives

$$\mathbb{E}(\|\boldsymbol{V}(t)\|_2) = \mathbb{E}(\sqrt{\sum_i V_i(t)^2}) \leq \sqrt{\mathbb{E}(\sum_i V_i(t)^2)} = \sqrt{\sum_i \mathbb{E}(V_i(t)^2)} \tag{1}$$

As the $V_i(t)$ is a uniform random variable in range $[0, V_{thr}]$, the expected value of $V_i^2(t)$ can be computed as follows

$$\mathbb{E}(V(t)_i^2) = \int_0^{V_{thr}} x^2 \frac{1}{V_{thr}} dx = \frac{V_{thr}^2}{3} \tag{2}$$

which yields

$$\mathbb{E}(\|\boldsymbol{V}(t)\|_2) \leq \frac{\sqrt{n} V_{thr}}{\sqrt{3}} \tag{3}$$

Since t and S_r are constant values, the following inequality holds

$$\mathbb{E}(\|\boldsymbol{V}(t)/(tS_r)\|_2) \leq \frac{\sqrt{n} V_{thr}}{\sqrt{3} t S_r} \tag{4}$$

A.3 EXPERIMENT SETUP

The network architectures for difference datasets are given in Table A-1, which are modified from VGG-11 Simonyan & Zisserman (2014) for Cifar10-DVS & N-Caltech101 and VGG-like structure Fang et al. (2021b) for DVS128 Gesture.

Table A-1: Network architectures for difference datasets. C64k8s4 represents the convolutional layer with $filters = 64$, $kernel\ size = 4$ and $strides = 4$. The default values of Kernel size and strides are 3 and 1 respectively. AP2 is the average pooling layer, MP2 is the max pooling layer with $kernel\ size = 2$ and FC is the fully-connected layer.

Dataset	Network Architecture
Cifar10-DVS N-Caltech101	C64k8s4-C64-C128-C256s2-C256-C512s2 -C512-C512s2-C512-AP2-FC512-Output(10 or 101)
DVS128 Gesture	C128k8s4-{C128-MP2}*5 -FC512-FC128-Output(11)

Batch Normalisation Ioffe & Szegedy (2015) is applied after each convolutional and fully-connected layer to accelerate the convergence of ANN training. For all experiment, the learning rate is set to 0.1 and decays to zero after 300 epochs based on cosine decay schedule Loshchilov & Hutter (2016). Weight decay is set to 0.0005. We set α to 0.003 for the regulariser proposed in Section 4.2 and use pixel shifting as the data augmentation for all models, i.e., both width and height are randomly shifted by the range [-20%,20%]. Dropout is applied after fully-connected layer for DVS128 Gesture to improve the training and the dropout rate is 0.2. We set the batch size to 128 for $F = 1$ and 32 for $F > 1$ to reduce memory consumption.

For SNN training, we inherit the conversion methods and most of the notation from Wu et al. (2022); Bu et al. (2022). Particularly, the relationship between ReLU and IF is from Wu et al. (2022) and the relationship between quantised ReLU and IF is from Bu et al. (2022). Moreover, ROE minimisation is based on Wu et al. (2022) and operates as a penalty term during ANN training.

A.4 EVENT-BASED DATASETS

The samples in the event-based datasets record the event addresses with on/off events over a period of time. For Cifar10-DVS, it consists of 10,000 samples extracted from Cifar10Krizhevsky & Hinton (2009). Each sample has 128×128 spatial resolution. The length of each spike train is less equal to $1.3s$. For N-Caltech101, it has 8709 samples categorised into 101 classes. The number of samples in each class ranges from 31 to 800. The length of each spike train is about $0.3s$. The width in x-direction does not exceed 240 pixels and in y-direction does not exceed 180 pixels. For this two datasets, we use 90% samples in each class for training and 10% for testing. DVS128 Gesture consists of 1341 samples with 11 categories. Each sample is repetitive over $6.0s$.

A.5 APPLYING TEMPORAL TRAINING IN ANN

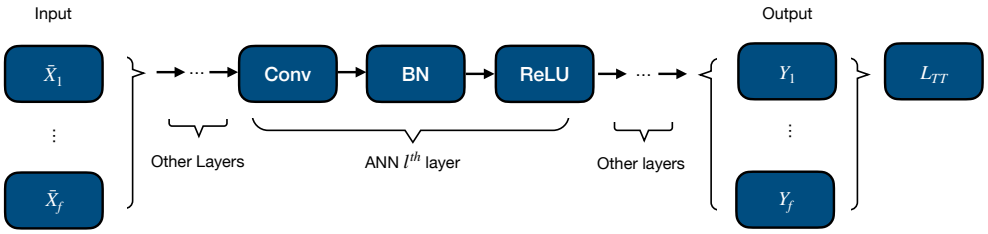


Figure A-1: Forward propagation in temporal training.

Similar to the direct training in Fang et al. (2021b), the temporal training in ANN, shown in Figure A-1, reshapes the temporal frames before forwarding them into the neural network and computes average loss after multiple outputs for optimisation. However, temporal training uses ReLU as the activation function and has no iterative operation during forward propagation. Although it ignores the correlation between the neighbouring frames in hidden layers, our experiment shows that SNN still can achieve good performance. Normally, iterative operation can be expensive when the number of iteration is large, i.g., large memory required Fang et al. (2021b). Figure A-2 presents that the increase of F can improve the accuracy on DVS128 Gesture, while it has little effect on Cifar10-DVS. We did not examine F in N-Caltch101 due to its large size.

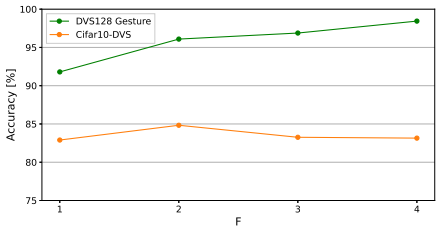


Figure A-2: ANN accuracy w.r.t F on Cifar10-DVS and DVS128 Gesture.

Moreover, since temporal training treats consecutive frames as individual frames and generates most spike for the prediction, regulariser and cutoff can be directly applied when $F > 1$,

A.6 ADDITIONAL EXPERIMENTS

Figure A-3 shows the confidence comparison on N-Caltech101 and DVS128 Gesture. The results are consistent, SNN-ROE can have better confidence at early inference time.

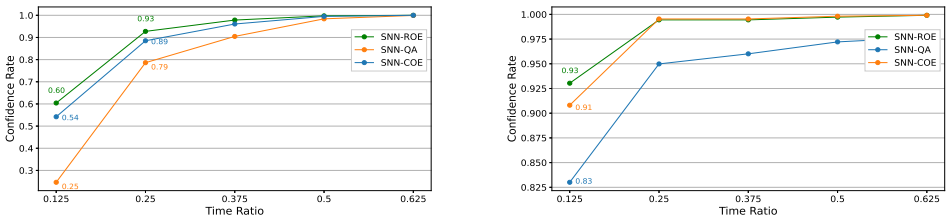


Figure A-3: Confidence w.r.t time ratio. for N-Caltech101 (left) and DVS128 Gesture (Right).

Figure A-4 shows that the proposed regulariser improves $cos(\phi)$, i.e., the cosine similarity between spiking rate $r(t)$ and desired spiking rate r_d, over different layer and reduces the conversion error at early inference time. After increasing α above 0.003, the improvement in $cos(\phi)$ becomes limited.

Therefore, we remain $\alpha = 0.003$ in other experiments. Figure A-5 shows the difference of $cos(\phi)$ on SNN using ROE, QA and COE. SNN-ROE has general higher $cos(\phi)$ than the other methods. Meanwhile, Figure A-4 and A-5 also show that the increase of inference time benefits $cos(\phi)$.

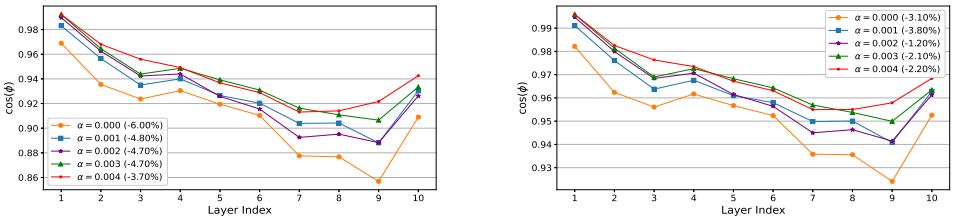


Figure A-4: Comparison of $cos(\phi)$ for SNN-ROE w.r.t different setting of α, at $0.375T_{total}$ (left) and $0.500T_{total}$ (right), on Cifar10-DVS. The conversion error is shown in the bracket.

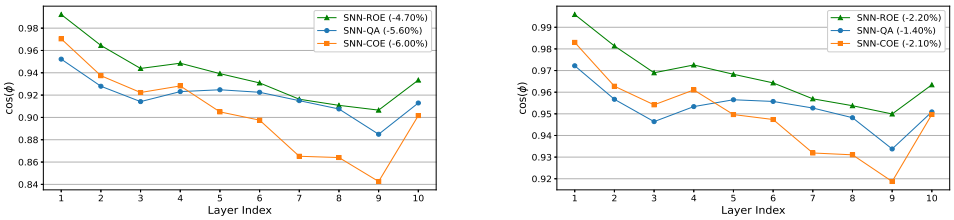


Figure A-5: Comparison of $cos(\phi)$ for SNN-ROE ($\alpha = 0.003$), SNN-QA and SNN-COE, at $0.375T_{total}$ (left) and $0.500T_{total}$ (right), on Cifar10-DVS. The conversion error is shown in the bracket.

Figure A-6 indicates that the change of α and batch size can influence the performance of resulted SNN. It can be easily found that ROE-SNN can achieve better accuracy at $0.125T_{total}$ with larger batch size. However, too small or large batch size can significantly degrade the normal ANN training. Thus, we set the batch size wisely that can result in an optimal SNN.

Figure A-7 presents the activation distributions for ROE, COE and QA over different layers. Comparing with QA, ROE helps ANN retain a Gaussian-like distribution. On the other hand, ROE brings the $99.99th$ percentile value closer to the mean of the activation than COE.

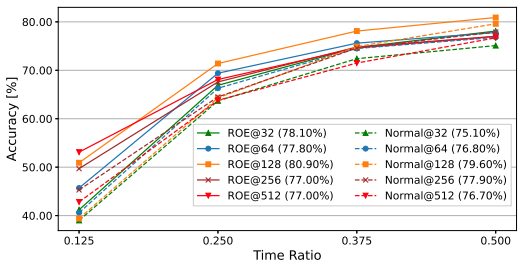


Figure A-6: Impact of α and batch size on resulted SNN at early inference time, for Cifar10-DVS. We use $method@batch\ size$ to present training method with the setting of batch size. 'ROE' represents $\alpha = 0.003$ and 'Normal' means training without ROE. The accuracy at $0.500T_{total}$ is shown in the bracket.

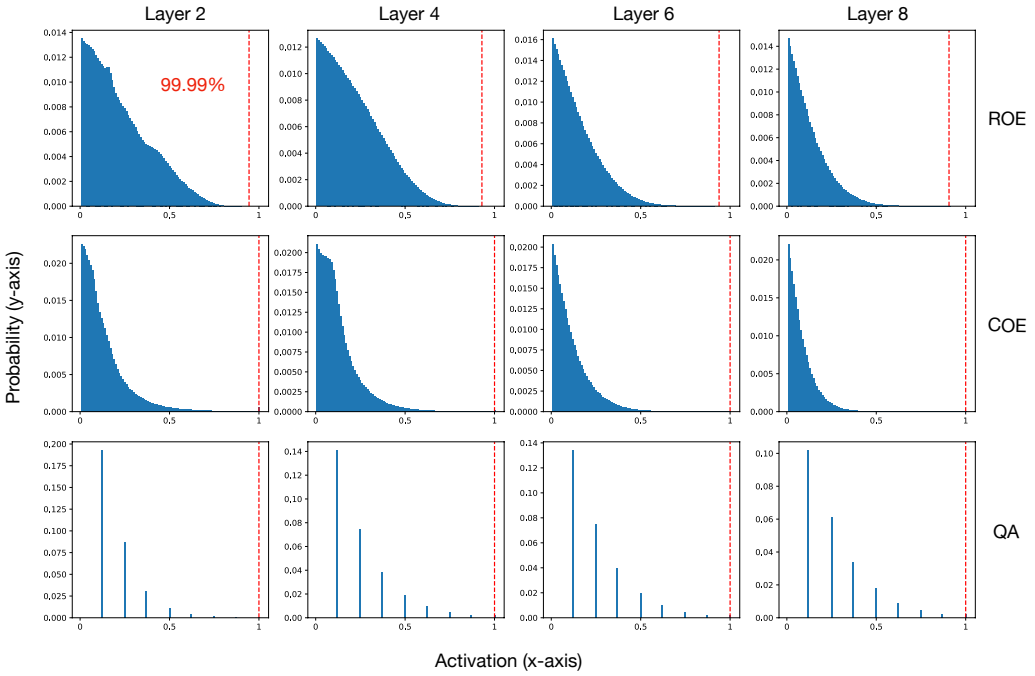


Figure A-7: Comparison of Normalised activation distribution of ANN using different methods, ROE, COE and QA, for Cifar10-DVS. The dashed line indicates the $99.99th$ percentile of activation.